수리 x2
car수리

차량이미지 딥러닝 인식기반 수리비용 예측서비스

넷이서 5인분

넷이서 5인분

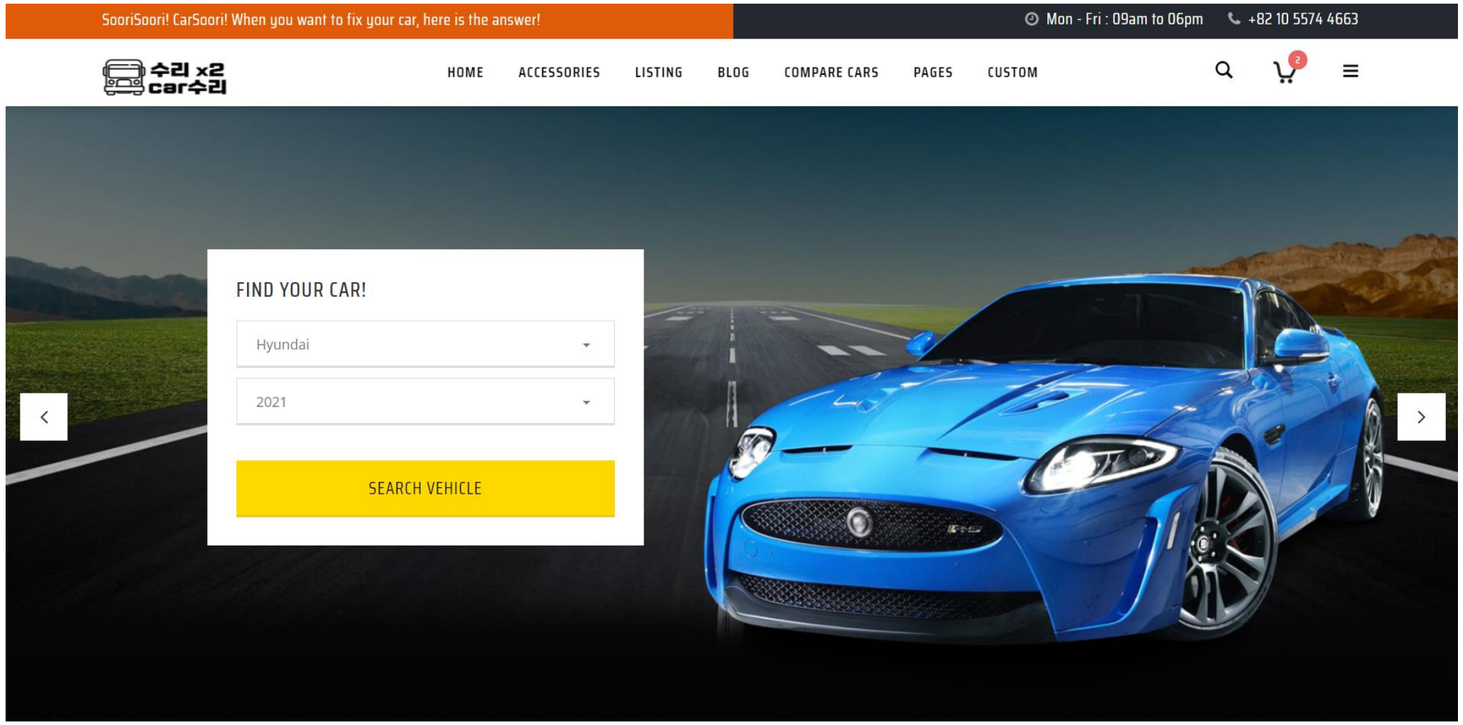
김소연

김한비

이지은

임상오

목 차



CHAPTER.1

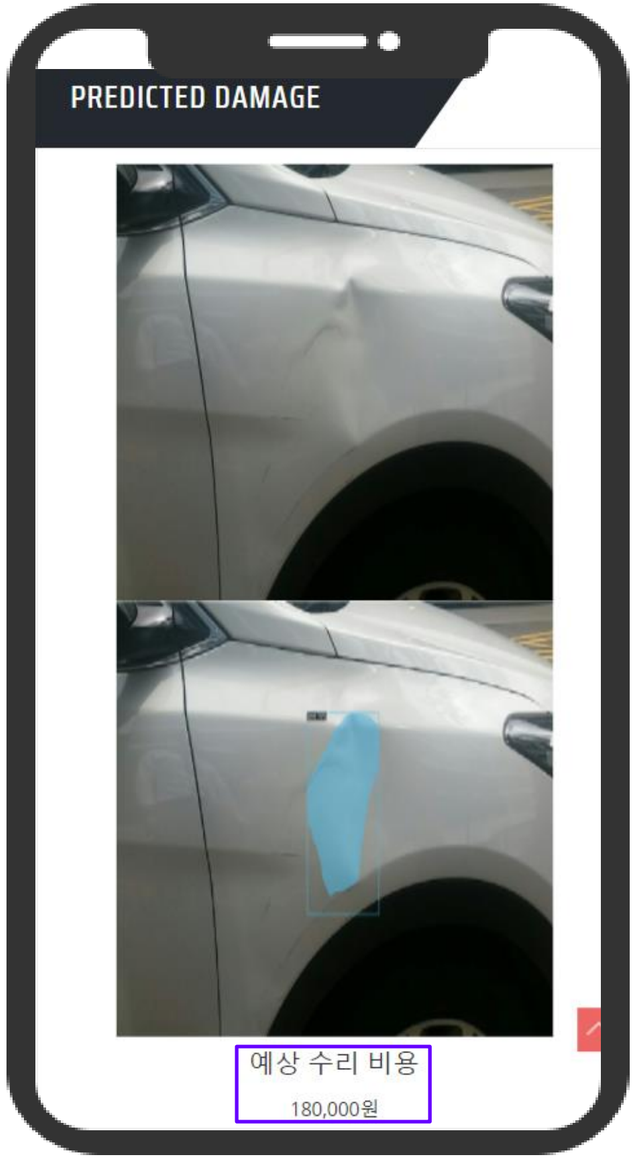
서비스 소개

서비스 소개

내차에 스크래치가 ?
수리업체마다 천차만별인 수리비..

찰칵! 한번으로 내차의 손상도 파악은 기본
자동차 수리비와 내 주변 수리업체까지
한번에 확인할 수 있는 서비스

수리 x2
car수리



서비스소개

자동차 손상도 체크

사진 한장으로 손상을 파악?

손상된 위치는 물론
손상의 종류 및 손상도를 짧은 시간안에
확인하는 서비스

#스크래치 #찌그러짐 #문콕

수리비 견적

수리 예상 금액은 얼마 ?

전국 정비업체의 견적 데이터를 분석
내 차의 수리비를
미리 예측하는 서비스

#비교견적 #저렴한 #잘고침

업체 추천

내 주변 수리업체는 어디?

예측된 비용으로 수리할 수 있는
나와 가장 가까운 수리업체를
추천해주는 서비스

#내주변 #리뷰좋음 #신속

서비스 비교분석

carooo VS 수리 x2 car수리

서비스	자동차 수리비 견적을 내 주변 수리업체에서 빠르게 확인할 수 있는 서비스	차량 손상 수리비용을 미리 예측하고 내 주변 수리업체를 추천하는 서비스
강점	자동차 보험서비스, 유지보수, 용품구입까지 필요한 다양한 서비스를 한 곳에서 확인하고 처리할수 있음	축적된 수리비 데이터를 분석 적정 금액을 예측하여 과도한 금액청구 방지, 보험사기 방지
약점	각 업체의 견적을 하나씩 확인하여 스스로 비교견적을 해야함	수리업체와의 제휴

CHAPTER.2

분석 프로세스 & 모델

분석과정

데이터 수집 및 처리

차손상 이미지 구글 크롤링
수리업체 비용 (별도 연락)
전처리 (마스킹, 리사이징)

Beautiful Soup Selenium OpenCV

모델링

U-net (eddynet)
Mask R-CNN
(Backbone - RESNET50 & RESNET101)

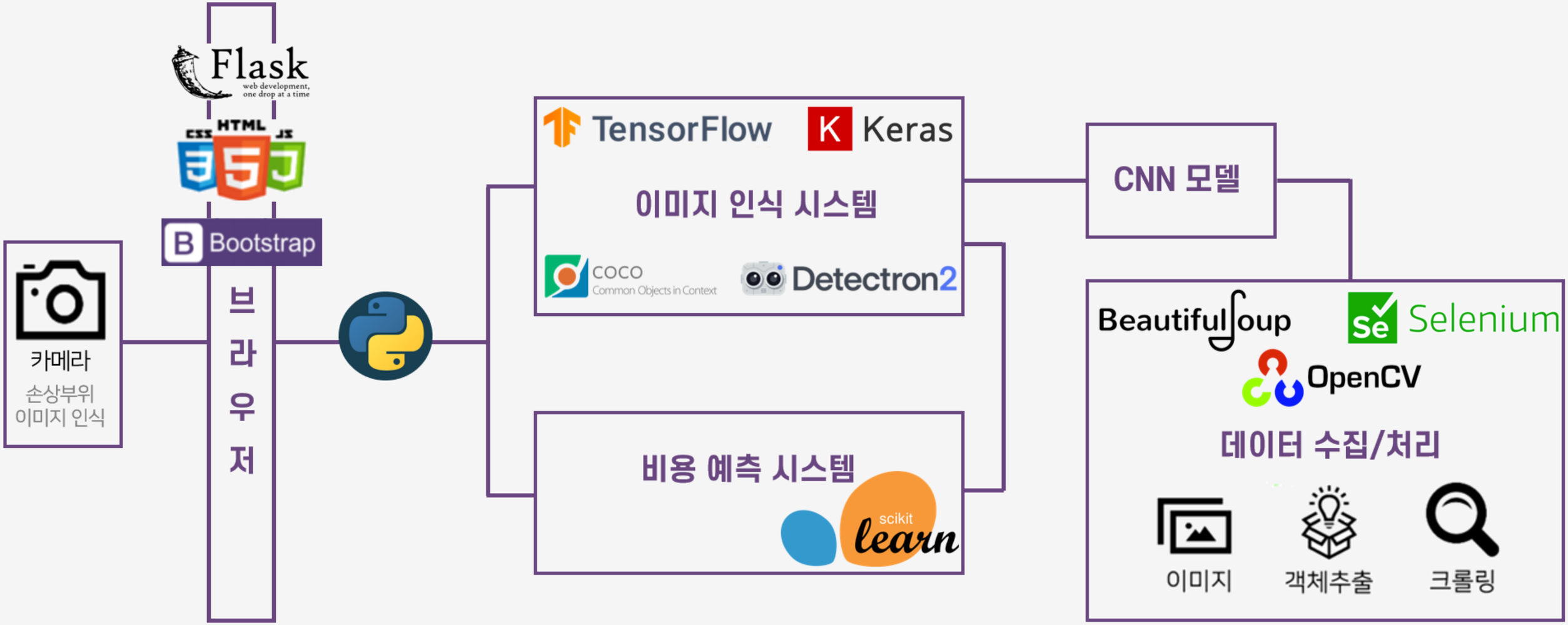
TensorFlow Keras COCO Common Objects in Context Detectron2

웹 브라우저 구현

Flask를 활용한 파이썬 연동
Bootstrap 사용
html, css, js를 활용한 홈페이지 수정

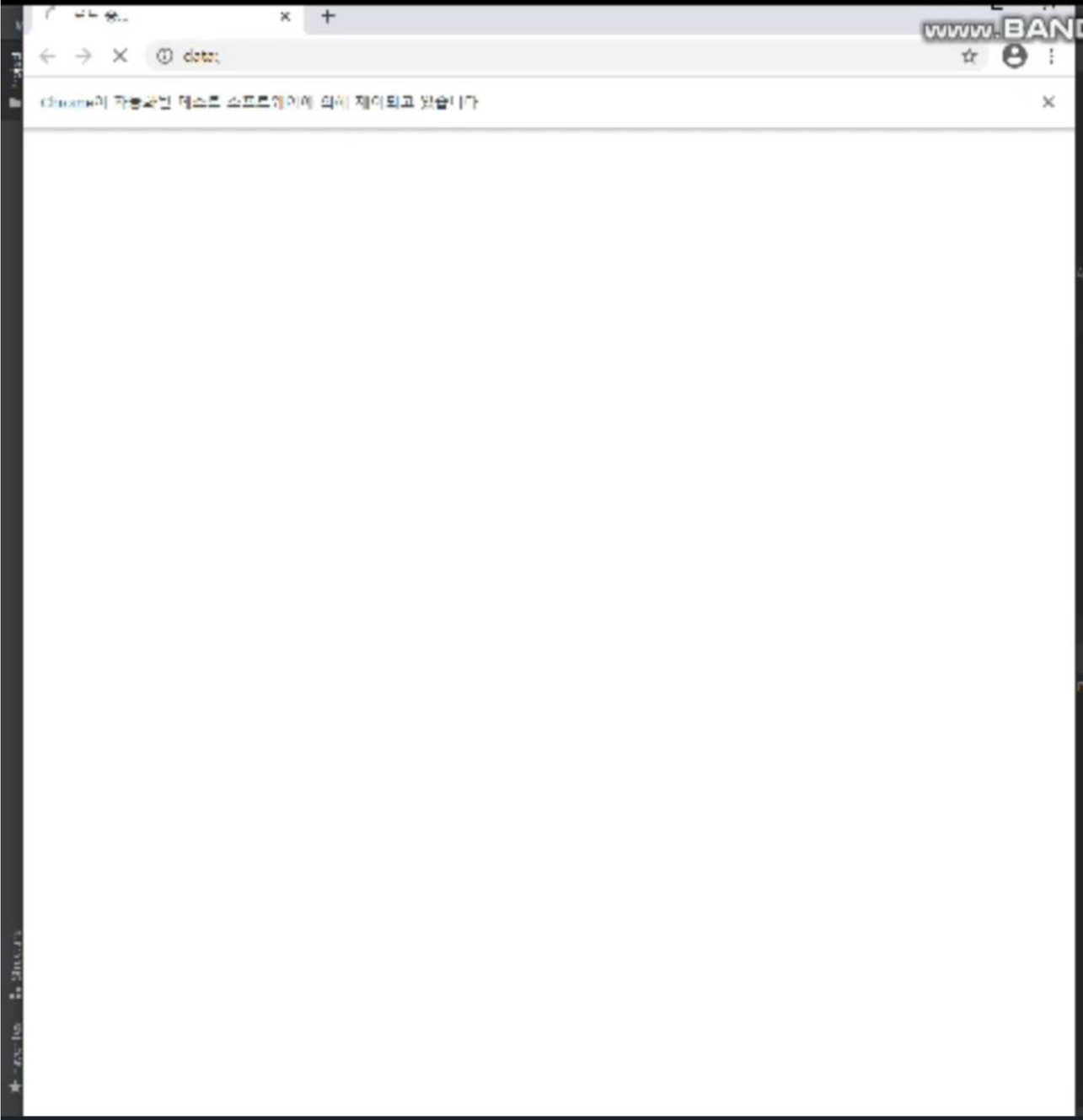
Flask web development, one drop at a time HTML CSS JS Bootstrap

아키텍처 정의서


Flask
web development, one drop at a time
HTML
CSS
JS
Bootstrap
카메라
손상부위
이미지 인식
브라우저
TensorFlow
Keras
이미지 인식 시스템
COCO
Common Objects in Context
Detectron2
비용 예측 시스템
scikit
learn
CNN 모델
BeautifulSoup
Selenium
OpenCV
데이터 수집/처리
이미지
객체추출
크롤링

이미지 크롤링 영상

www.BANDICAM.com
www.BANDICAM.com



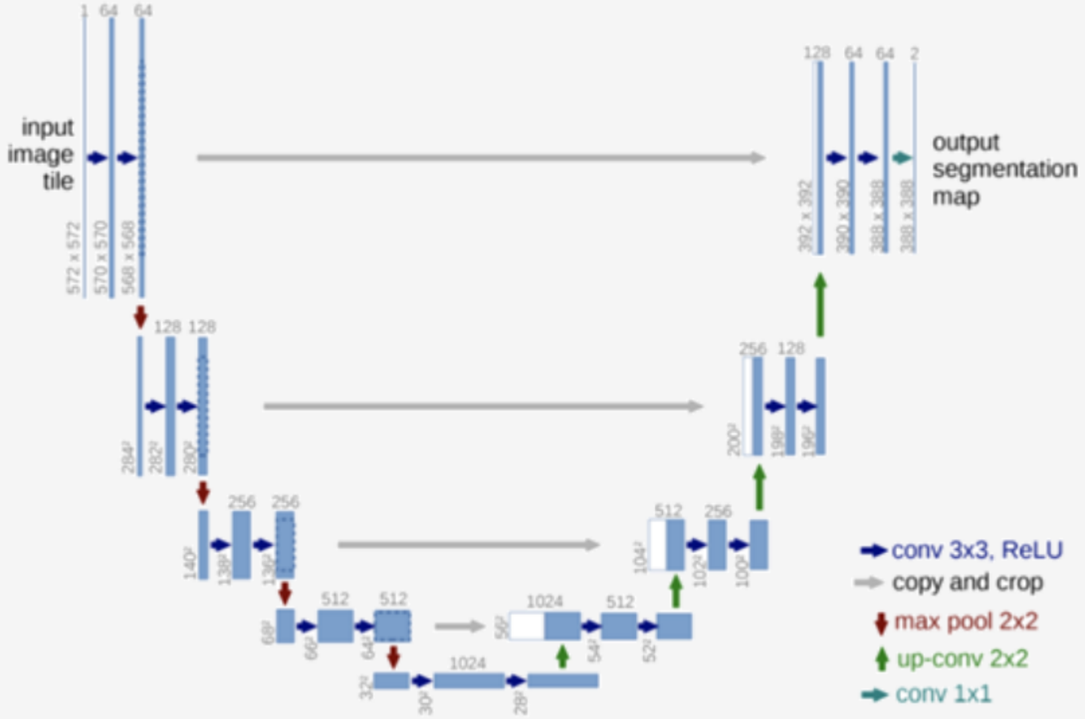
EDDYNET MODELING

Classification

Semantic Segmentaion

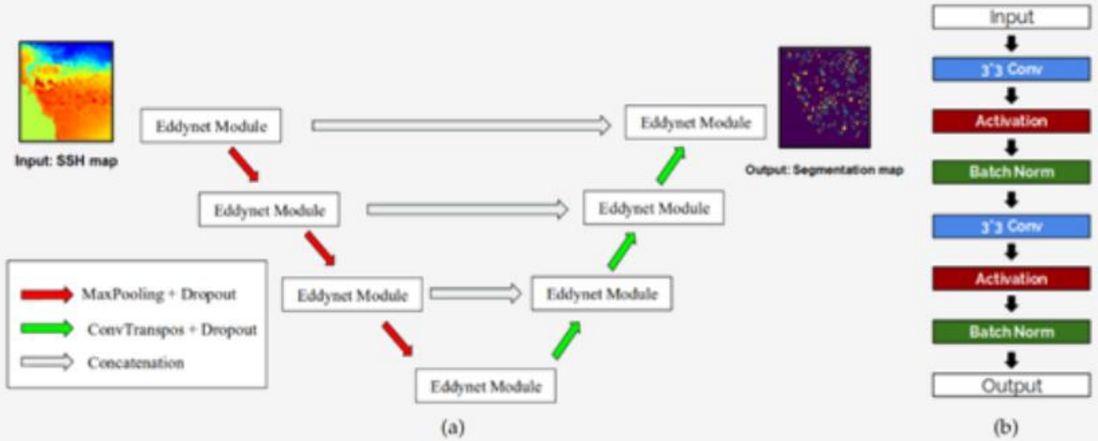
EDDYNET MODELING

U-net



U-Net Architecture

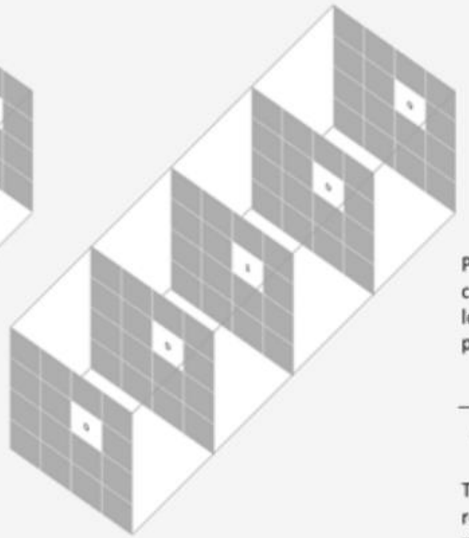
Eddy net



EDDYNET MODELING

pixel-wise cross entropy loss

Prediction for a selected pixel



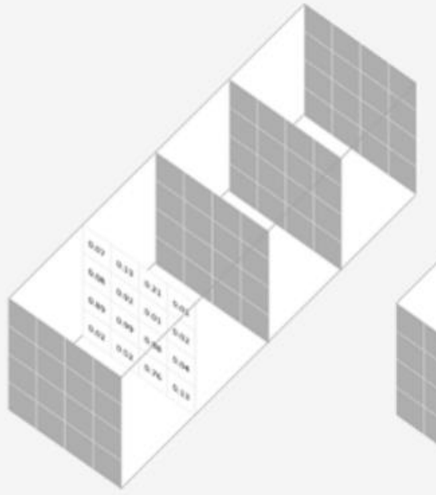
Target for the corresponding pixel

Pixel-wise loss is calculated as the log loss, summed over all possible classes

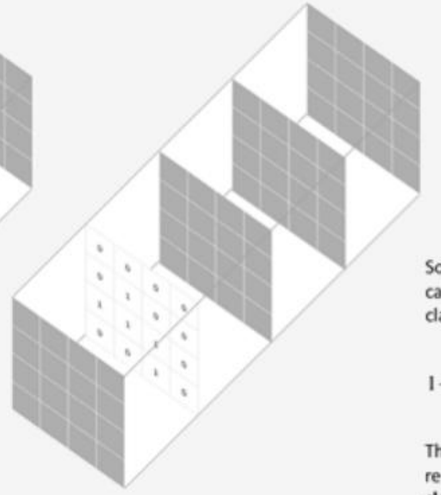
$$-\sum_{classes} y_{true} \log\left(y_{pred}\right)$$

This scoring is repeated over all **pixels** and averaged

Soft Dice loss



Prediction for a selected class



Target for the corresponding class

Soft Dice coefficient is calculated for each class mask

$$1-\frac{2\sum_{pixels} y_{true}, y_{pred}}{\sum_{pixels} y_{true}^2+\sum_{pixels} y_{pred}^2}$$

This scoring is repeated over all **classes** and averaged

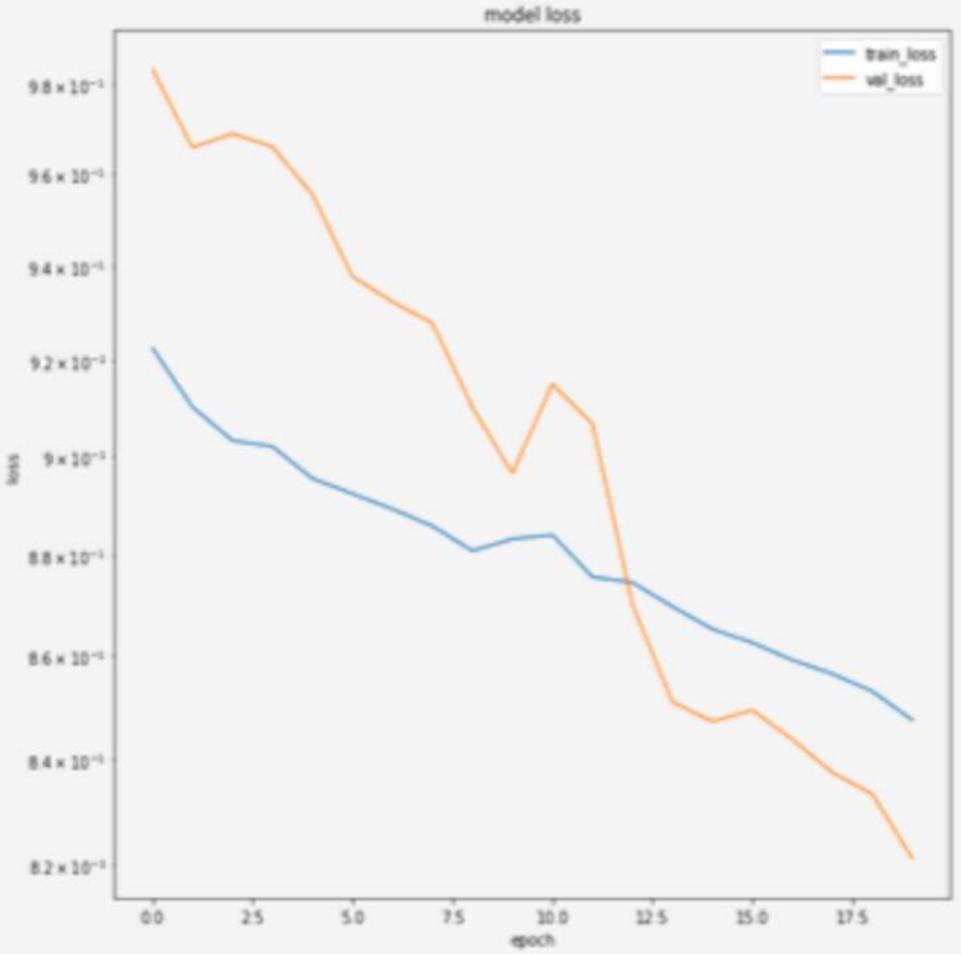
이미지 마스킹 영상

www.BANDICAM.com

```
drawing = False  # true if mouse is pressed
mode = True

# mouse callback function
def paint_draw(event, former_x, former_y, flags, param):
    global current_former_x, current_former_y, drawing, mode

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        current_former_x, current_former_y = former_x, former_y

    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing == True:
            if mode == True:
                cv2.line(bg, (current_former_x, current_former_y), (former_x, former_y), brush_color, brush_size)
                cv2.line(image, (current_former_x, current_former_y), (former_x, former_y), brush_color, brush_size)
                current_former_x = former_x
                current_former_y = former_y
    elif event == cv2.EVENT_LBUTTONUP:
        drawing = False
        if mode == True:
            cv2.line(bg, (current_former_x, current_former_y), (former_x, former_y), brush_color, brush_size)
```

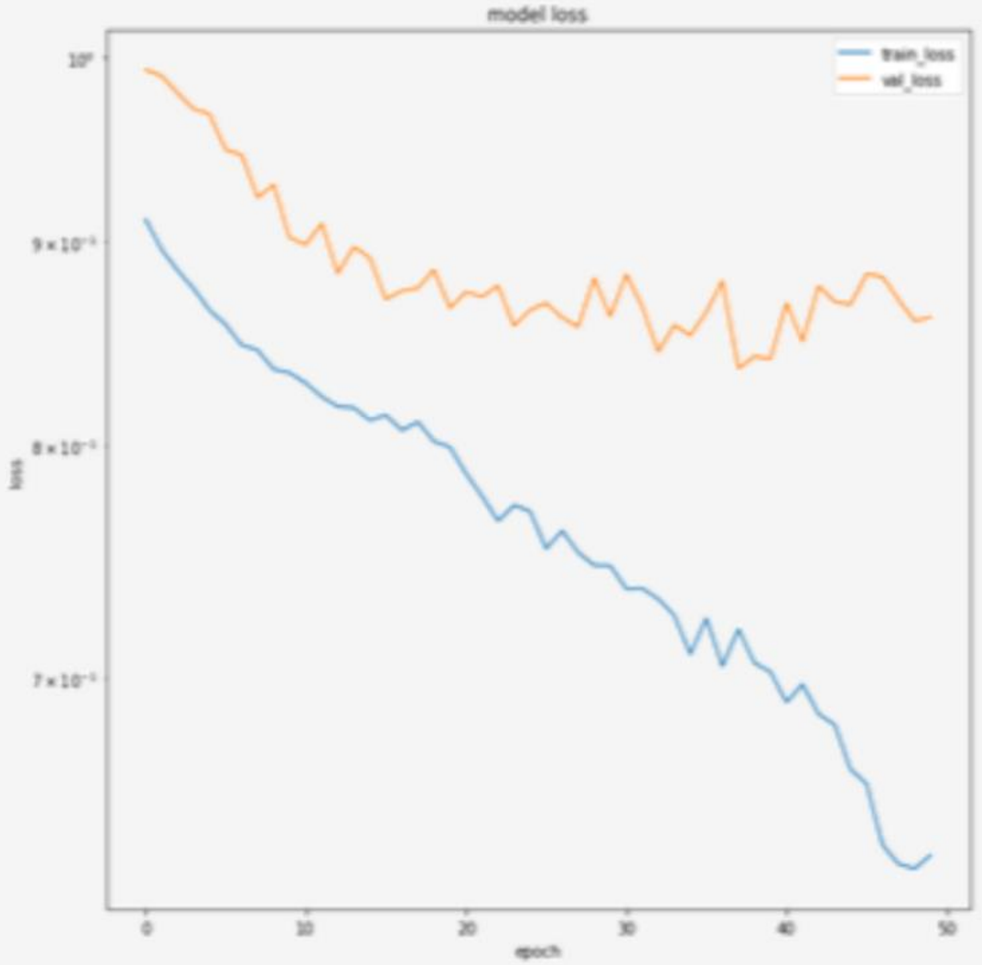
```
C:\IT\workspace_python\venv\Scripts\python.exe C:/IT/workspace_python/venv/pkg/final_project/easking_brush.py
number?:
```

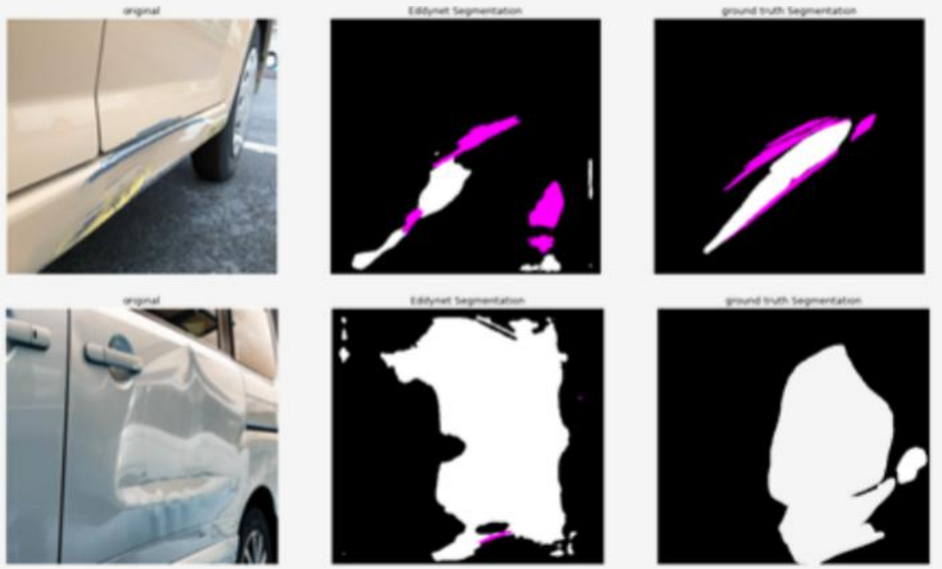
EDDYNET MODELING



Damage type	scratch, dent
Train data	200
Input image size	128X128
Epoch	20

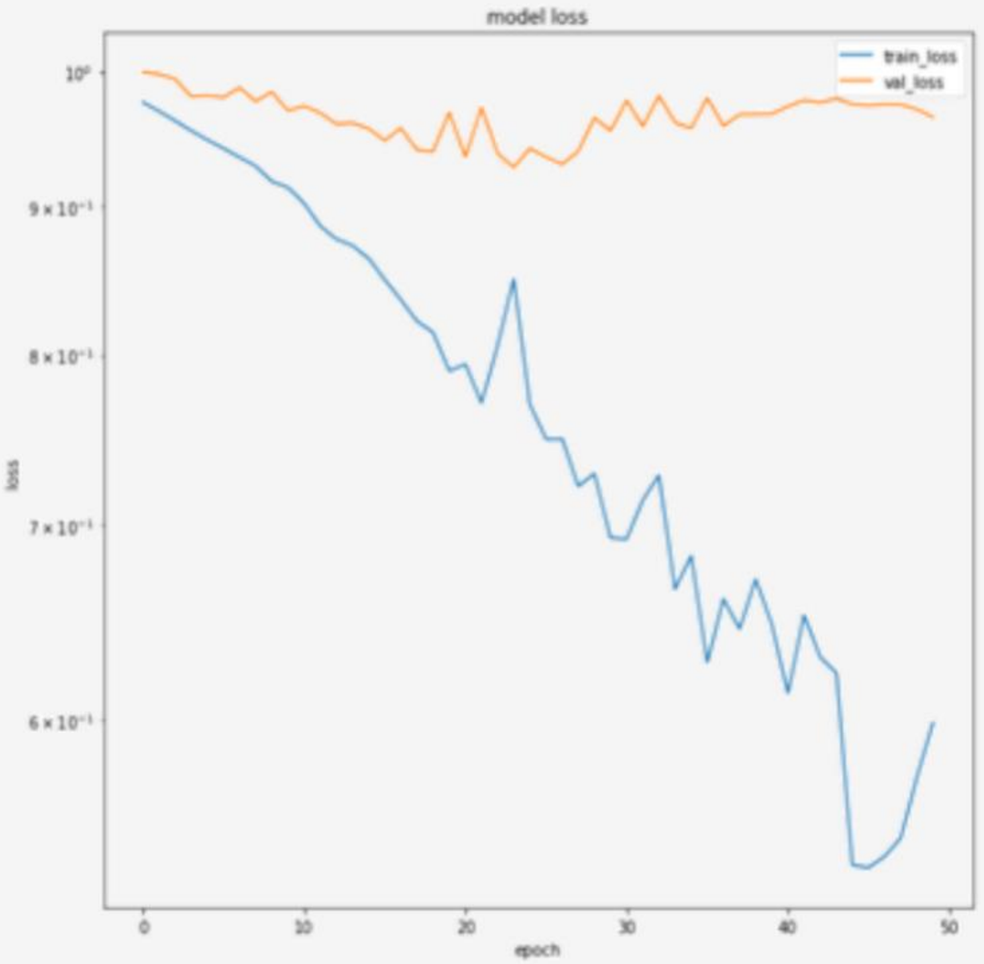
EDDYNET MODELING

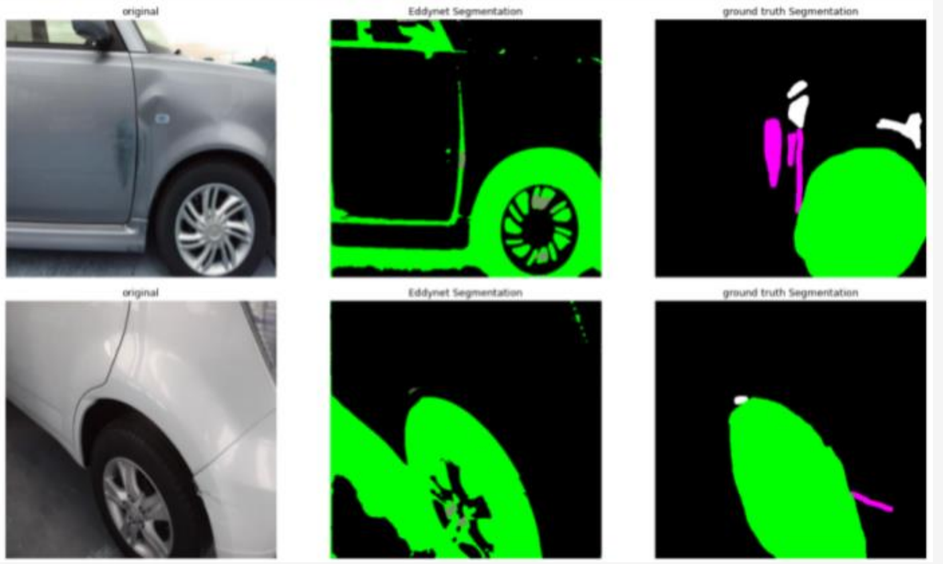




Damage type	scratch, dent
Train data	200
Input image size	512X512
Epoch	50

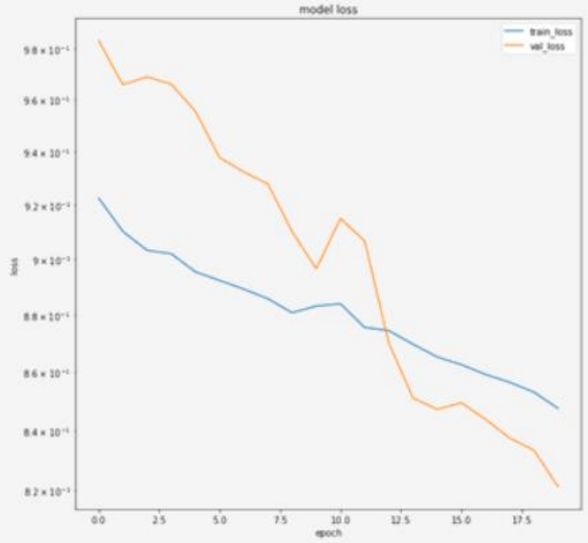
EDDYNET MODELING



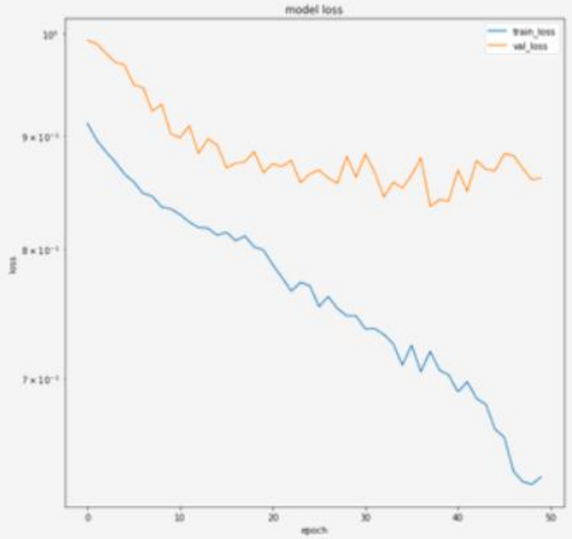


Damage type	scratch, dent
Train data	200
Input image size	512X512
Epoch	50

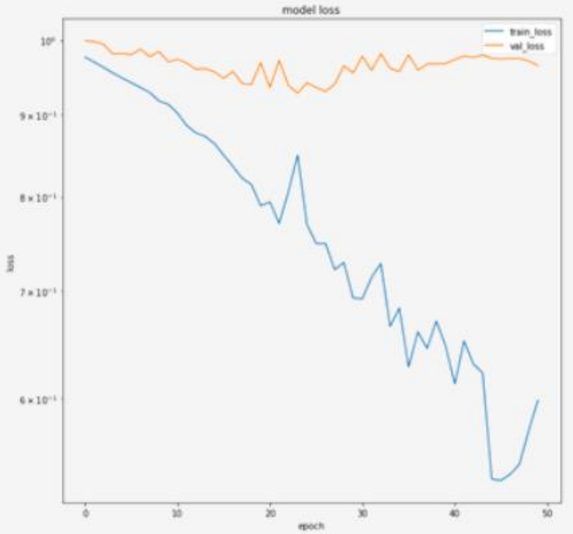
EDDYNET MODELING



Damage type	scratch, dent
Train data	200
Input image size	128X128
Epoch	20
Learming rate	0.001
Optinizer	Adam
Patience	20
Loss	dice_coef_loss
Acc	categorical accuracy



Damage type	scratch, dent
Train data	200
Input image size	512X512
Epoch	50
Learming rate	0.001
Optinizer	Adam
Patience	20
Loss	dice_coef_loss
Acc	categorical accuracy



Damage type	scratch, dent
Train data	200
Input image size	512X512
Epoch	50
Learming rate	0.001
Optinizer	Adam
Patience	20
Loss	dice_coef_loss
Acc	categorical accuracy

MASK R-CNN MODELING

News

- We are pleased to announce the COCO 2020 Detection, Keypoint, Panoptic, and DensePose Challenges.
- The new rules and awards for this year challenges encourage innovative methods.
- Results to be announced at the Joint COCO and LVIS Recognition ECCV workshop.

What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

Collaborators

Tsung-Yi Lin Google Brain
Genevieve Patterson MSR, Trash TV
Matteo R. Ronchi Caltech
Yin Cui Google
Michael Maire TTI-Chicago
Serge Belongie Cornell Tech
Lubomir Bourdev WaveOne, Inc.
Ross Girshick FAIR
James Hays Georgia Tech
Pietro Perona Caltech
Deva Ramanan CMU
Larry Zitnick FAIR
Piotr Dollár FAIR

Sponsors

CVDF
Microsoft
facebook
Mighty Ai

“문제점

모델 구축을 위한 양질의 데이터셋 부족

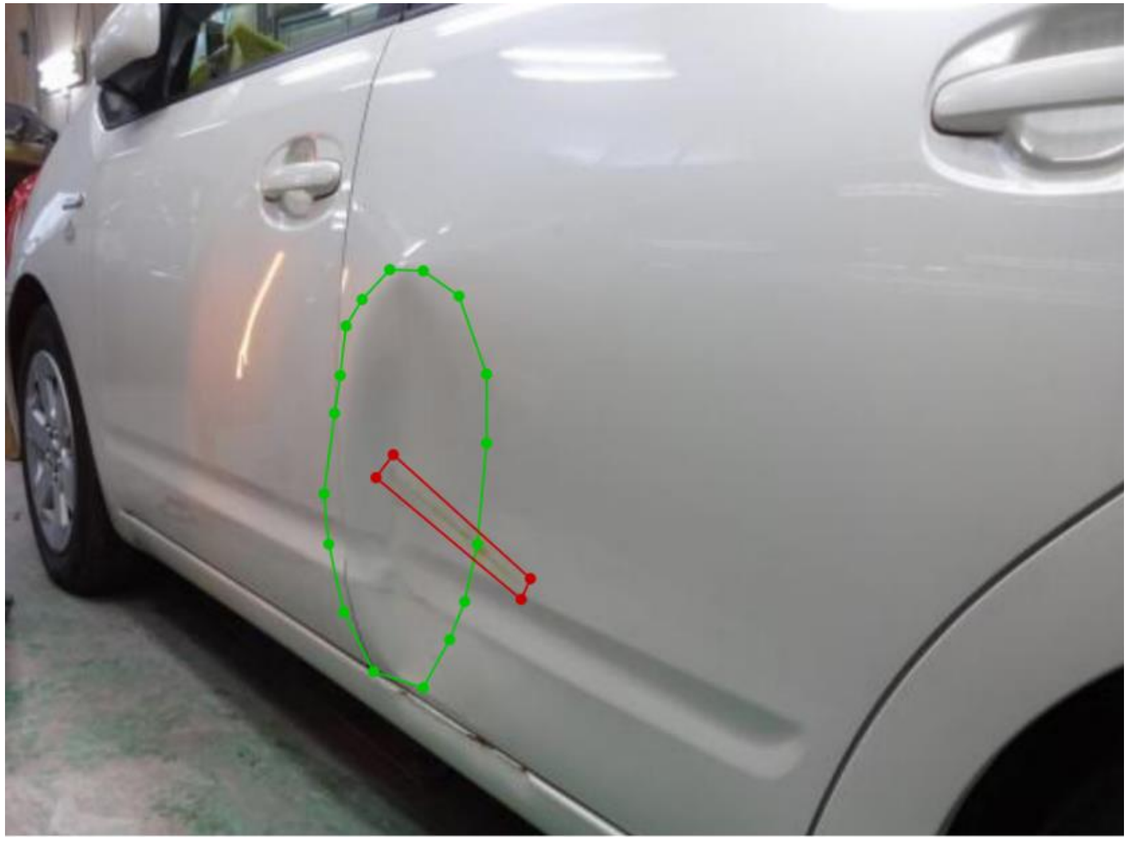
“해결방안

COCO data set 활용

- 33만건의 이미지와 20만장이상의 이미지 레이블 보유
- 80개의 각기 다른 사물 분류 가능

사이트 : https://cocodataset.org/#home

MASK R-CNN MODELING



“문제점

손상 부위의 레이블이 중복될 경우 한가지만 선택

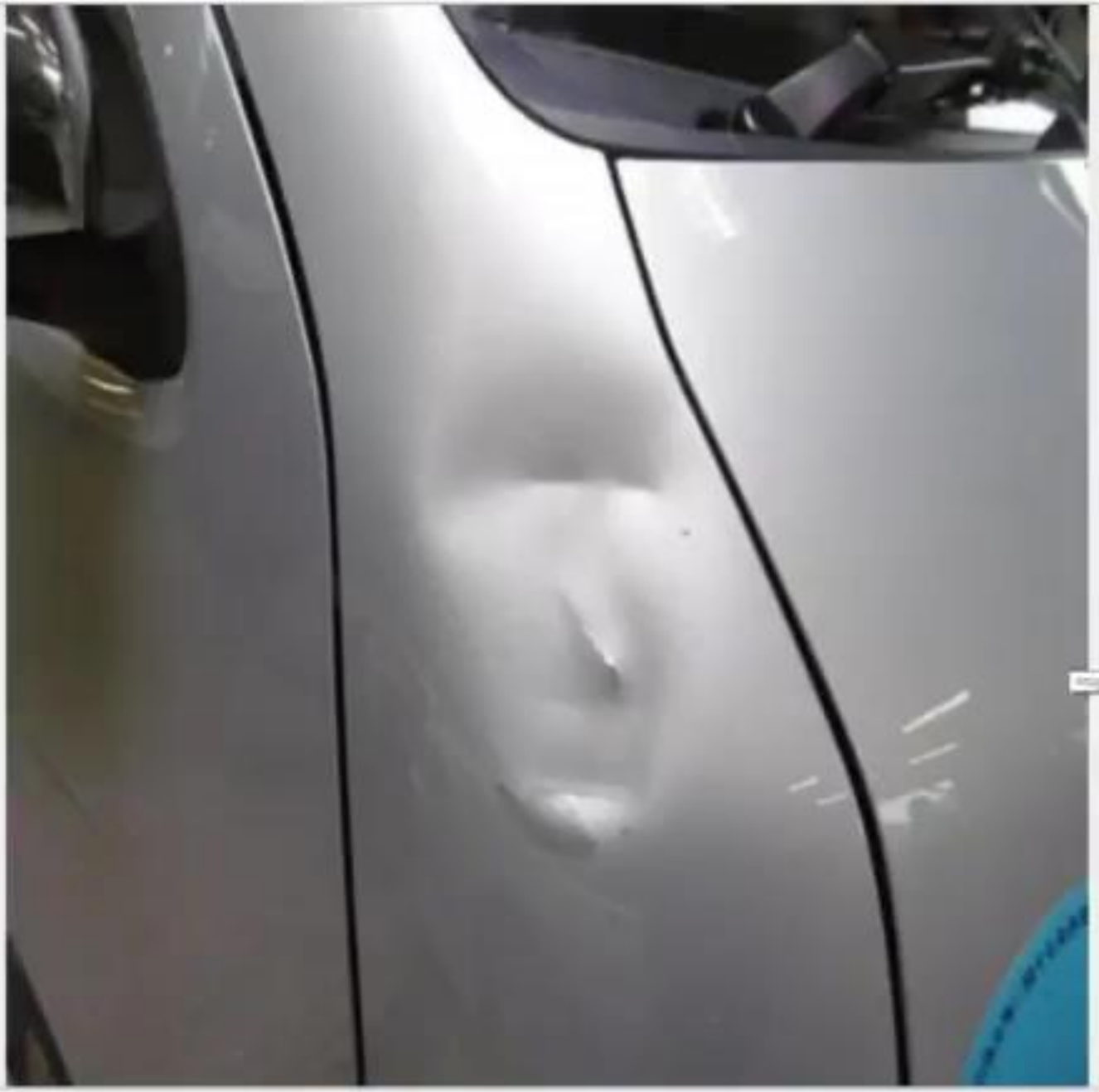
“해결방안

Labelme 활용

- 다각형을 그려 중복되는 레이블도 선택 가능

이미지 마스킹 영상

www.BANDICAM.com

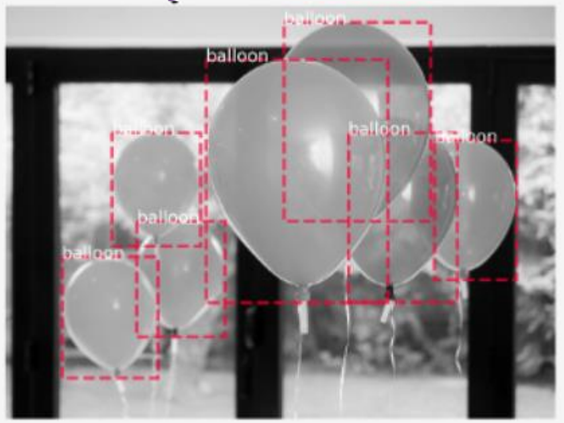


MASK R-CNN MODELING

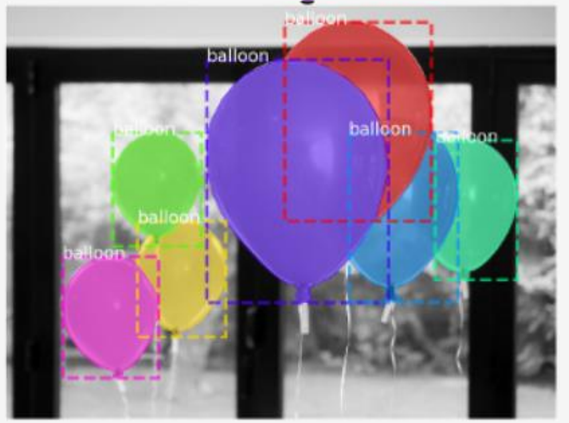
Classificatior

Semantic Segmentation

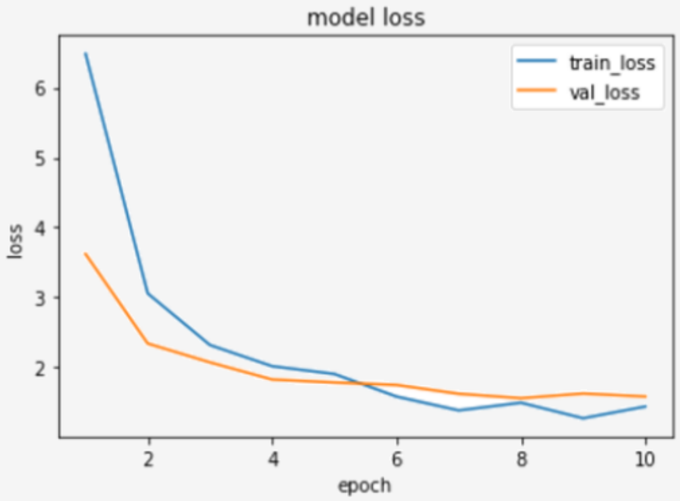
Object Detection



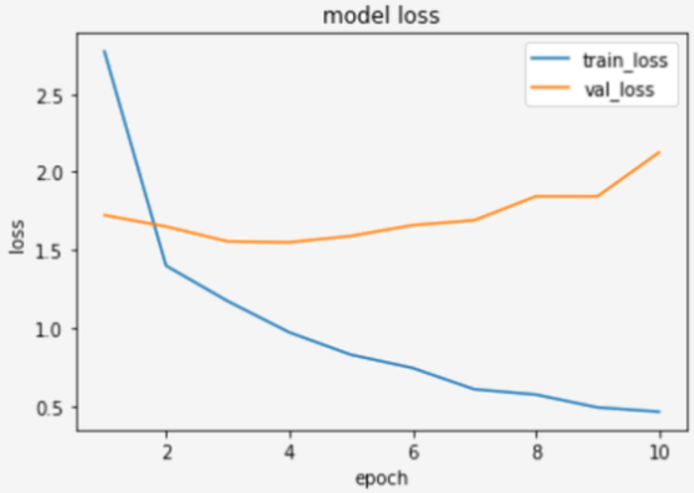
Inastance Segmentation



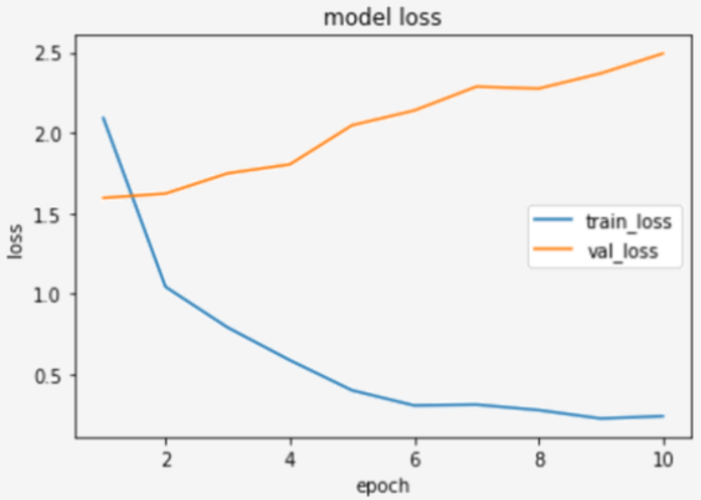
MASK R-CNN MODELING



Damage type	single class(damage)
Input image size	1024X1024
Backbone network	RESNET101
Epoch	10
Steps per Epoch	10
Learming rate	0.001
Optimizer	SGD
Loss	Crossentropy + Smooth
	l1 loss



Damage type	single class(damage)
Input image size	1024X1024
Backbone network	RESNET101
Epoch	10
Steps per Epoch	50
Learming rate	0.001
Optimizer	SGD
Loss	Crossentropy + Smooth
	l1 loss



Damage type	single class(damage)
Input image size	1024X1024
Backbone network	RESNET101
Epoch	10
Steps per Epoch	100
Learming rate	0.001
Optimizer	SGD
Loss	Crossentropy + Smooth
	l1 loss

MASK R-CNN MODELING

Detectron2

Detectron2 is Facebook AI Research's next generation software system that implements state-of-the-art object detection algorithms. It is a ground-up rewrite of the previous version, Detectron, and it originates from maskrcnn-benchmark.

What's New

- It is powered by the PyTorch deep learning framework.
- Includes more features such as panoptic segmentation, Densepose, Cascade R-CNN, rotated bounding boxes, PointRend, DeepLab, etc.
- Can be used as a library to support different projects on top of it. We'll open source more research projects in this way.
- It trains much faster.
- Models can be exported to TorchScript format or Caffe2 format for deployment.

See our blog post to see more demos and learn about detectron2.

Installation

See INSTALL.md.

Getting Started

Follow the installation instructions to install detectron2.

See Getting Started with Detectron2, and the Colab Notebook to learn about basic usage.

Learn more at our documentation. And see projects/ for some projects that are built on top of detectron2.

Model Zoo and Baselines

We provide a large set of baseline results and trained models available for download in the Detectron2 Model Zoo.

“문제점

모델 구축에 긴시간 소요됨

“해결방안

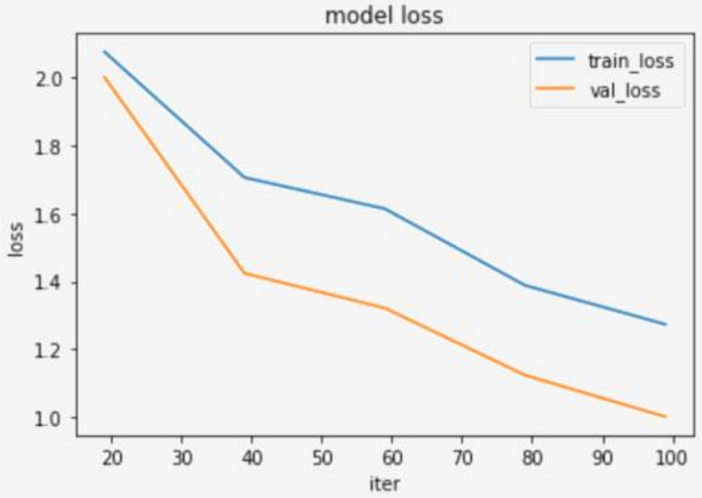
빌드인 모델 (Model zoo)을 사용

가중치를 계산하는 방식으로

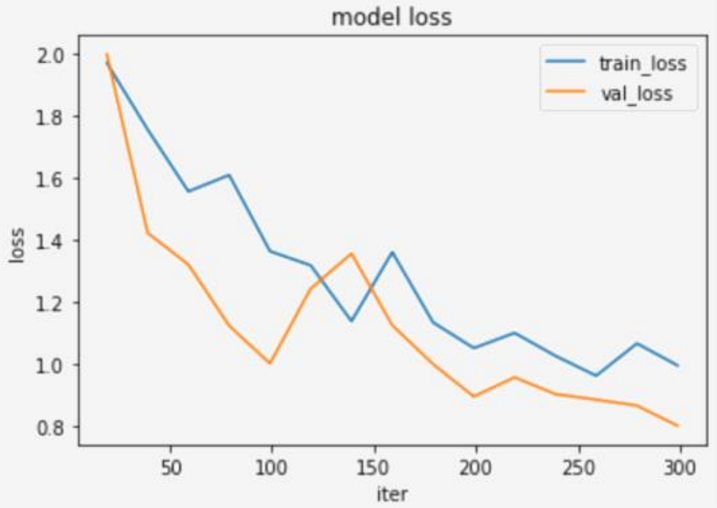
빠른 결과 도출 가능

사이트 : https://github.com/facebookresearch/detectron2

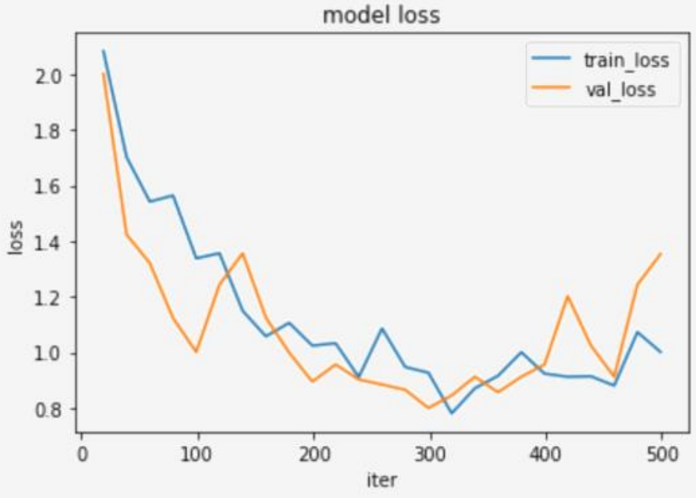
MASK R-CNN MODELING



Damage type	scratch, dent
Input image size	1000X800
Backbone network	RESNET50
Epoch	30
Iterations	100
Learming rate	0.02
Optimizer	SGD
Loss	Crossentropy + Smooth
	l1 loss



Damage type	scratch, dent
Input image size	1000X800
Backbone network	RESNET50
Epoch	30
Iterations	300
Learming rate	0.02
Optimizer	SGD
Loss	Crossentropy + Smooth
	l1 loss

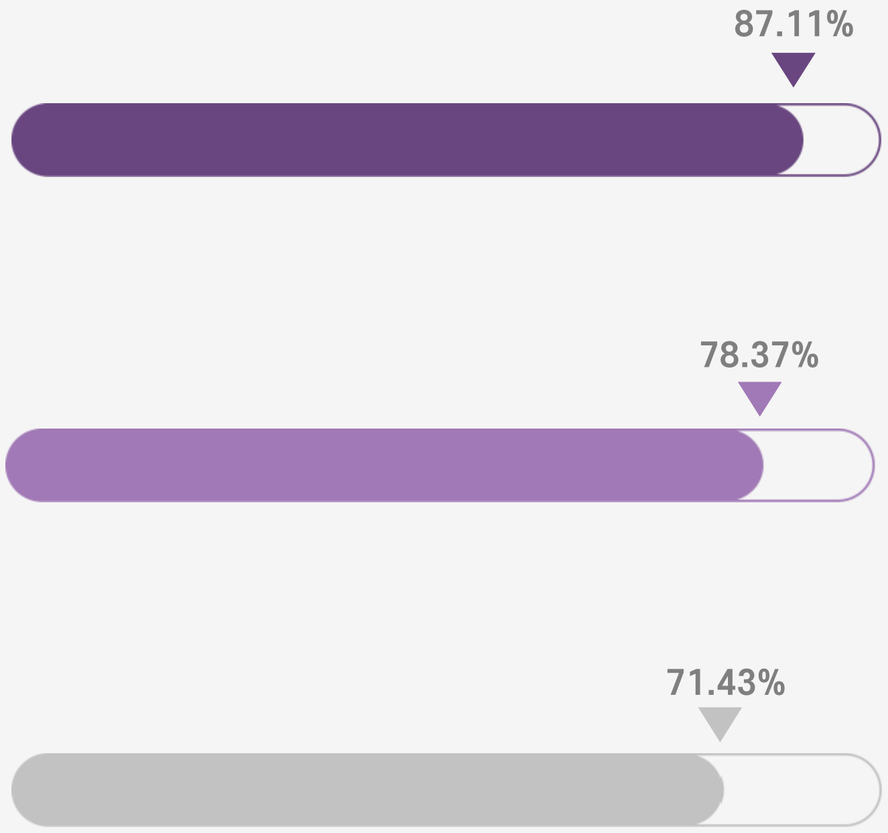


Damage type	scratch, dent
Input image size	1000X800
Backbone network	RESNET50
Epoch	30
Iterations	500
Learming rate	0.02
Optimizer	SGD
Loss	Crossentropy + Smooth
	l1 loss

CHAPTER.3

분석 결과 및 결론

분석 결과



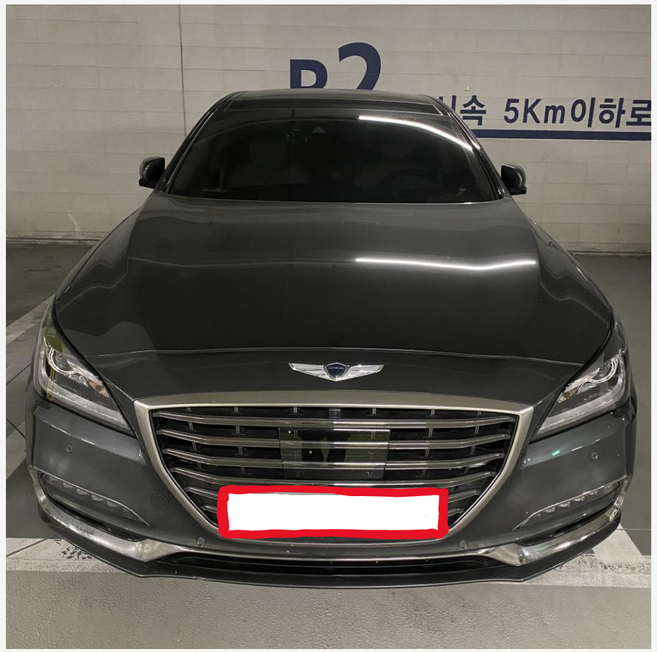
Mask R-CNN (Detectron2)	- Accracy : 87.11% - Loss : 0.8486 - Val_loss : 0.9182
Eddy net	- Accracy : 78.37% - Loss : 0.65 (dice-coef) - Val_loss : 0.86 (dice-coef)
Mask R-CNN (COCO)	- Accracy : 71.43% - Loss : 0.2399 - Val_loss : 2.4922

분석 결과

Scratch

Dent

향후과제



1. 사진의 각도 포맷을 정해 정확한 손상을 잡도록 해야함.

2. 비용데이터 확보가 어려워 정확한 금액 예측에 어려움이 있음.

발표 내용에 관해
궁금한 점이 있으시다면
자유롭게 질문해주세요!

Q&A