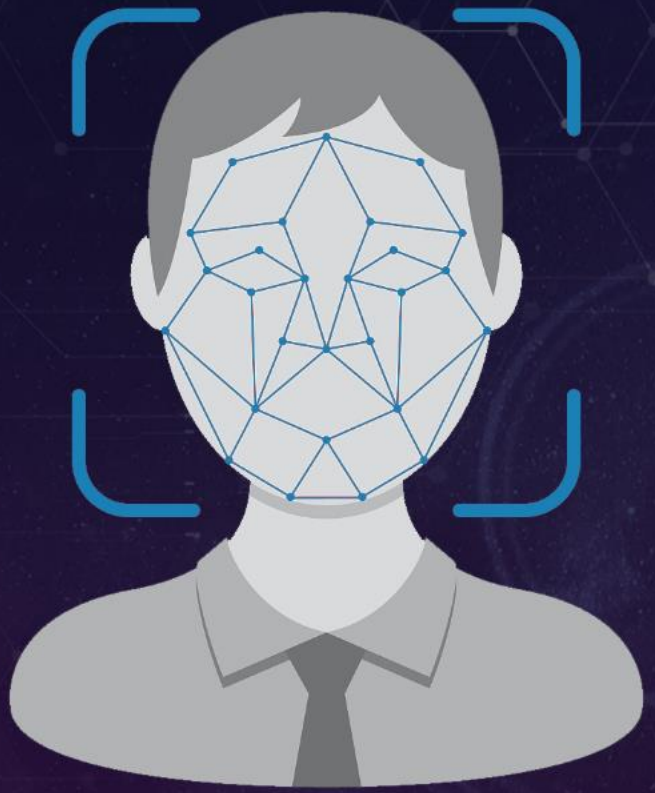
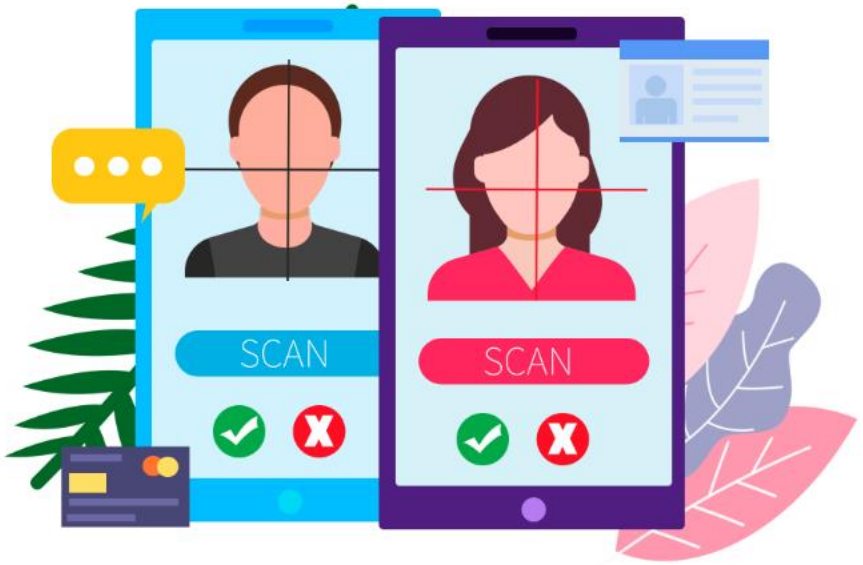
Sudo권



W H O A R E YOU ?

CONTENTS



개요

데이터

모델

검증

구현

개선점

프로젝트 개요

기획 의도

팀원 소개

SWOT 분석

개발 환경

공중화장실 성범죄 '연간 1000건' 발생… 불법 촬영 범죄 51%

2020-10-27 쿠키뉴스

공중 화장실·목욕탕 몰래 침입, 5년간 3.7배 증가

2019-09-27 서울경제

기획의도

특정 성별이 허용되지 않는 장소에 출입 시 이를 제지할 수 있는 예방적 수단의 필요성

이미지 분석을 사용하여 다중이용시설에 출입하는 사람들의 성별을 구분하고 인식.

팀원 소개

박관익

- 모델 구축
- 모델 검증

최은정

- 데이터 전처리
- 웹 구현

정인혁

- 데이터 수집
- 데이터 전처리

안심해치

'해치'란?

'시비와 선악을 판단하여 안다고 하는 상상의 동물'

'부정한 사람은 뿔로 받아서 몰아내는 동물'

남녀를 구별하여 부적합한 사람을 선별해내는 프로젝트의 성격에 맞는 캐릭터인 해치

SWOT 분석

S
- 조기에 범죄를 예방할 수 있음.
- 인력을 사용하는 것보다 경제적.
- 코로나로 인한 비접촉이 권장되는 상황에 적합.

W
- 학습 데이터 부족으로 인한 인식오류 발생 (ex. 중성적인 사람, 의도적인 여장/남장)

O
- 기존에 존재하지 않았던 서비스이기 때문에 고객 유치에 유리.

T
- 얼굴 인식을 위한 카메라 설치가 필수적이며 초기 설치 비용 발생

개발 환경


TensorFlow
K
Keras


python


Flask


Google Drive


Google
colab


PC


B
Bootstrap

데이터 전처리

데이터 수집

데이터 선별

데이터 분류

데이터 라벨링

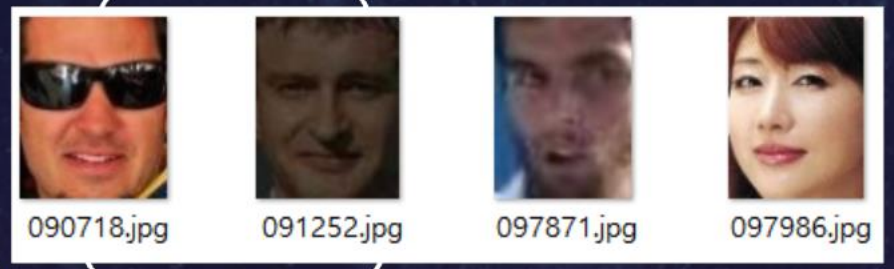
kaggle

남녀 사진

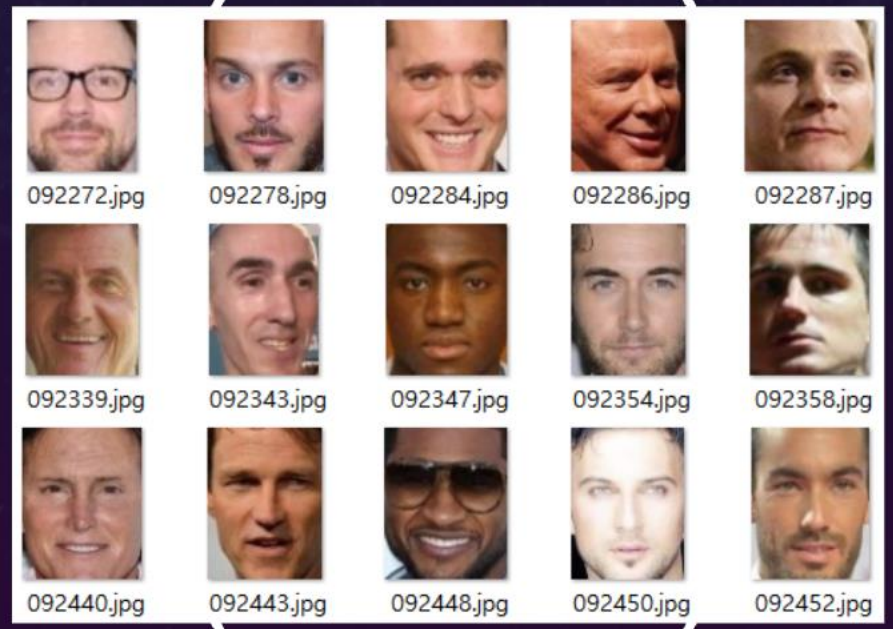
마스크 사진

캐글 데이터셋 사용

데이터 선별



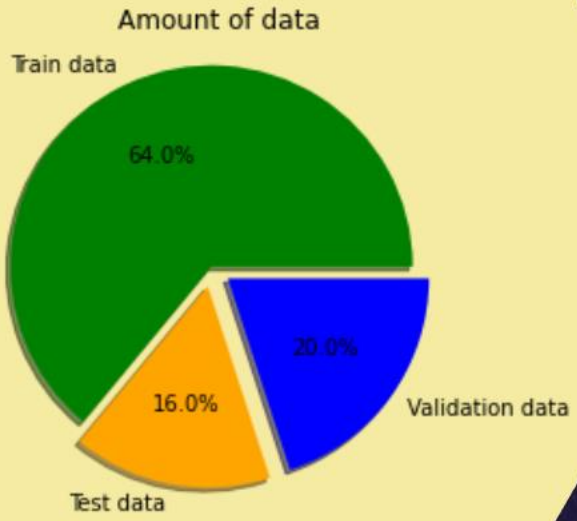
선글라스, 어두운, 저화질, 잘못된 성별사진 등 부적합한 데이터 제거

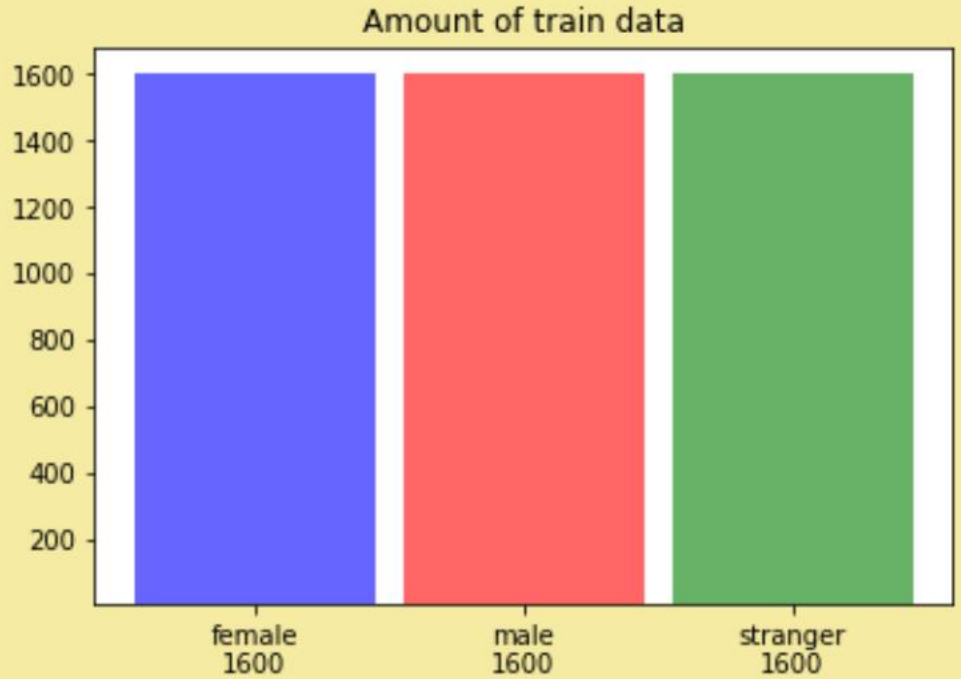


데이터 분류

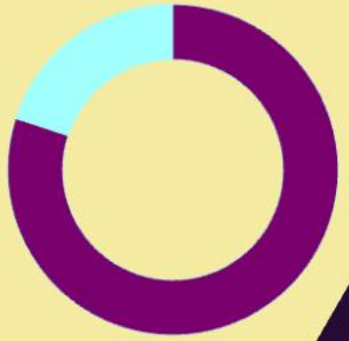
카테고리별 2000장씩 선별하여 학습

Train		Val		Test
1600	➤	0.2	➤	400

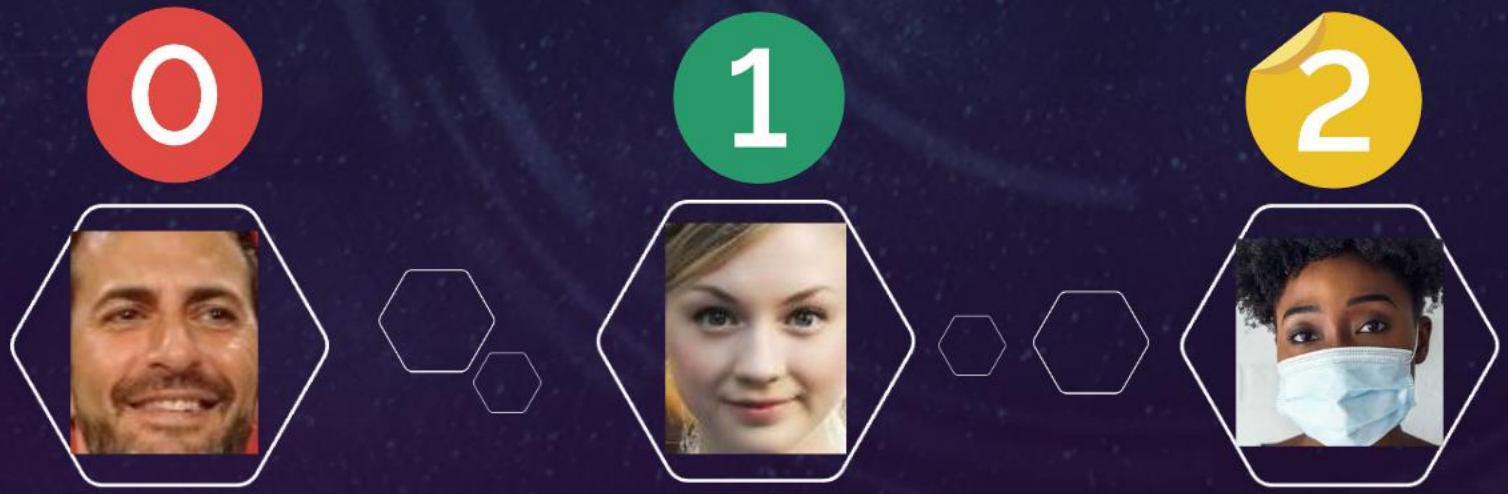




데이터의 80%를
Training에 사용



One-Hot Encoding



세 가지 종류로 라벨링

Model Algorithm

CNN

Ensemble

VGG16

First Model

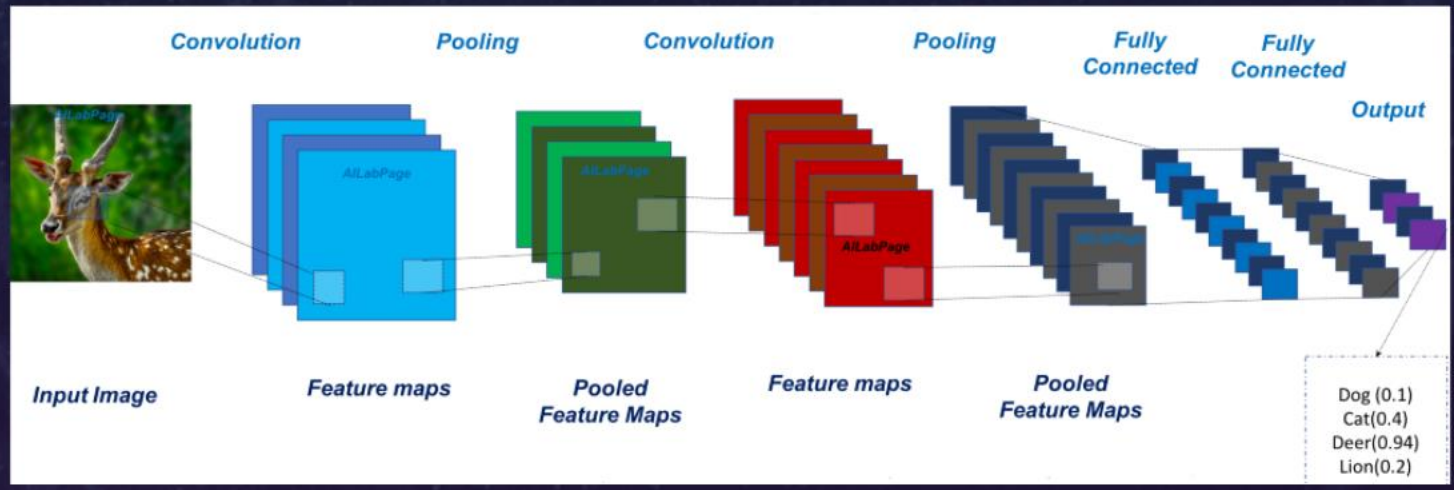
CNN

Explanation

Structure

Hyper Parameter

Convolutional Neural Network



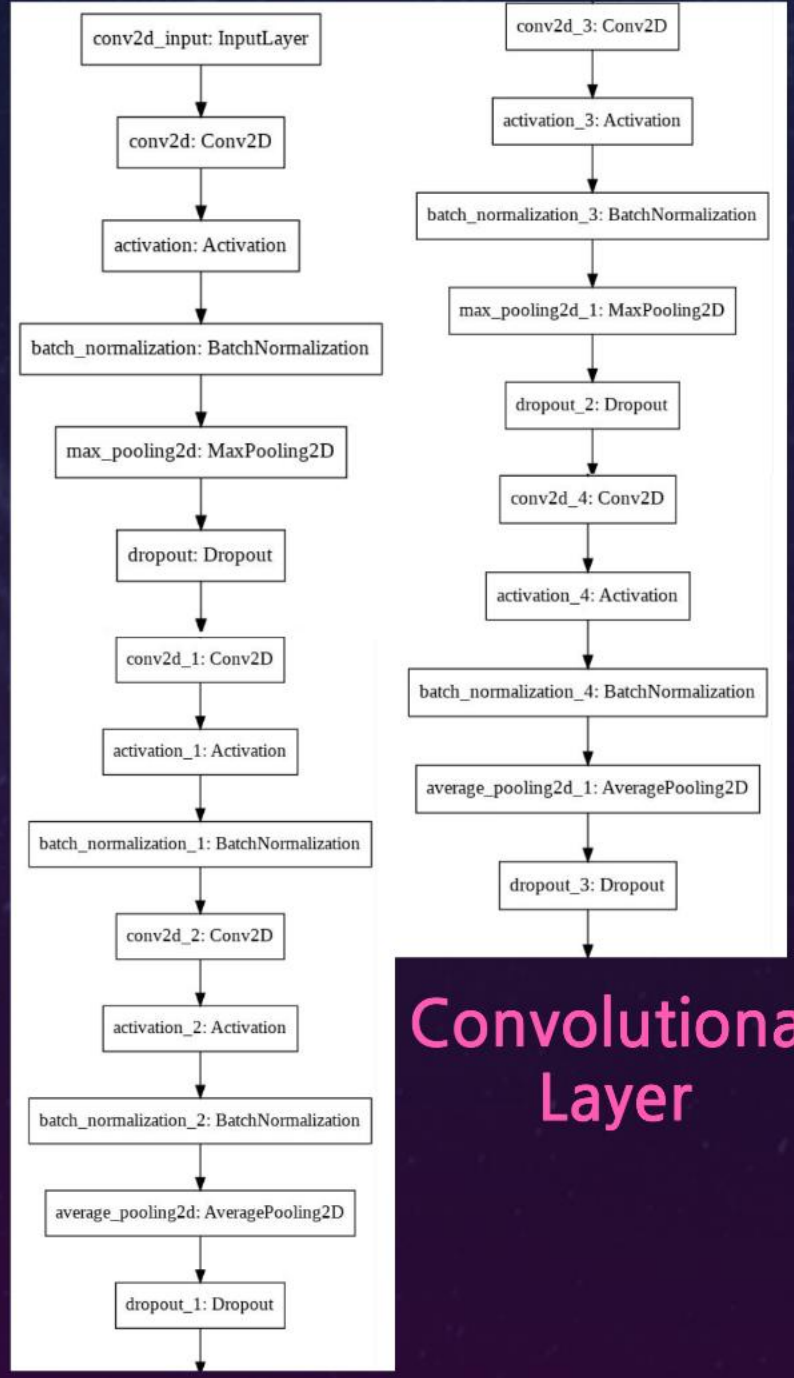
- 합성곱 신경망
- 기존의 구조에서 'convolutional layer'와 'pooling layer'를 추가한 네트워크
- Conv - ReLU - Pooling 으로 구성
- 입력 데이터에서 "filter"를 통해 특징을 추출하여 "feature map"을 생성
- layer가 깊어질수록 고급 정보가 추출됨

1. CONV -> RELU -> MAXPOOL
2. (CONV -> RELU)*2 -> AVGPOOL
3. CONV -> RELU -> MAXPOOL
4. CONV -> RELU -> AVGPOOL

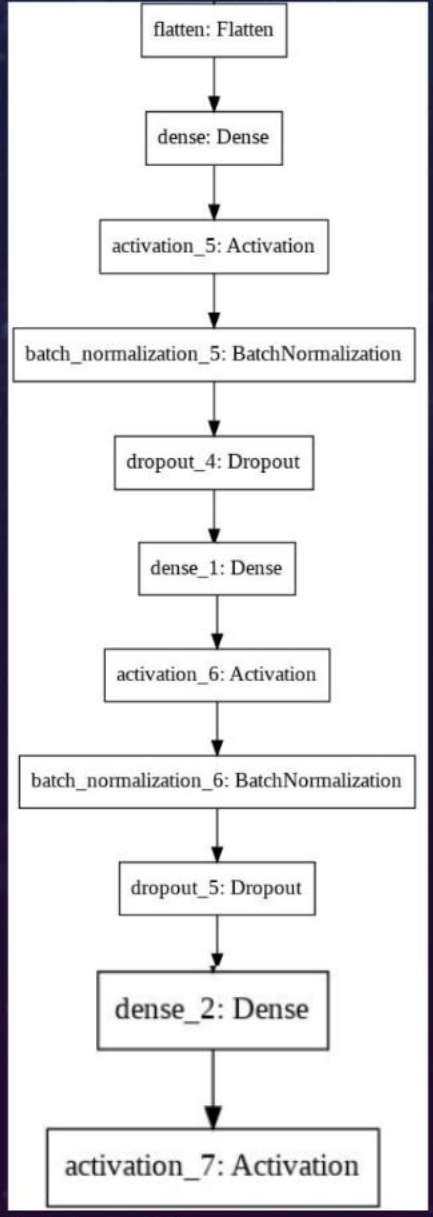
---

5. DENSE -> RELU
6. DENSE -> RELU
7. DENSE -> SOFTMAX

Structure



Convolutional Layer



Fully Connected Layer

Hyper Parameter

compile

- optimizer adam
- loss categorical_crossentropy
- metrics accuracy

EarlyStopping

- monitor val_loss
- mode auto
- patience 8

Fit

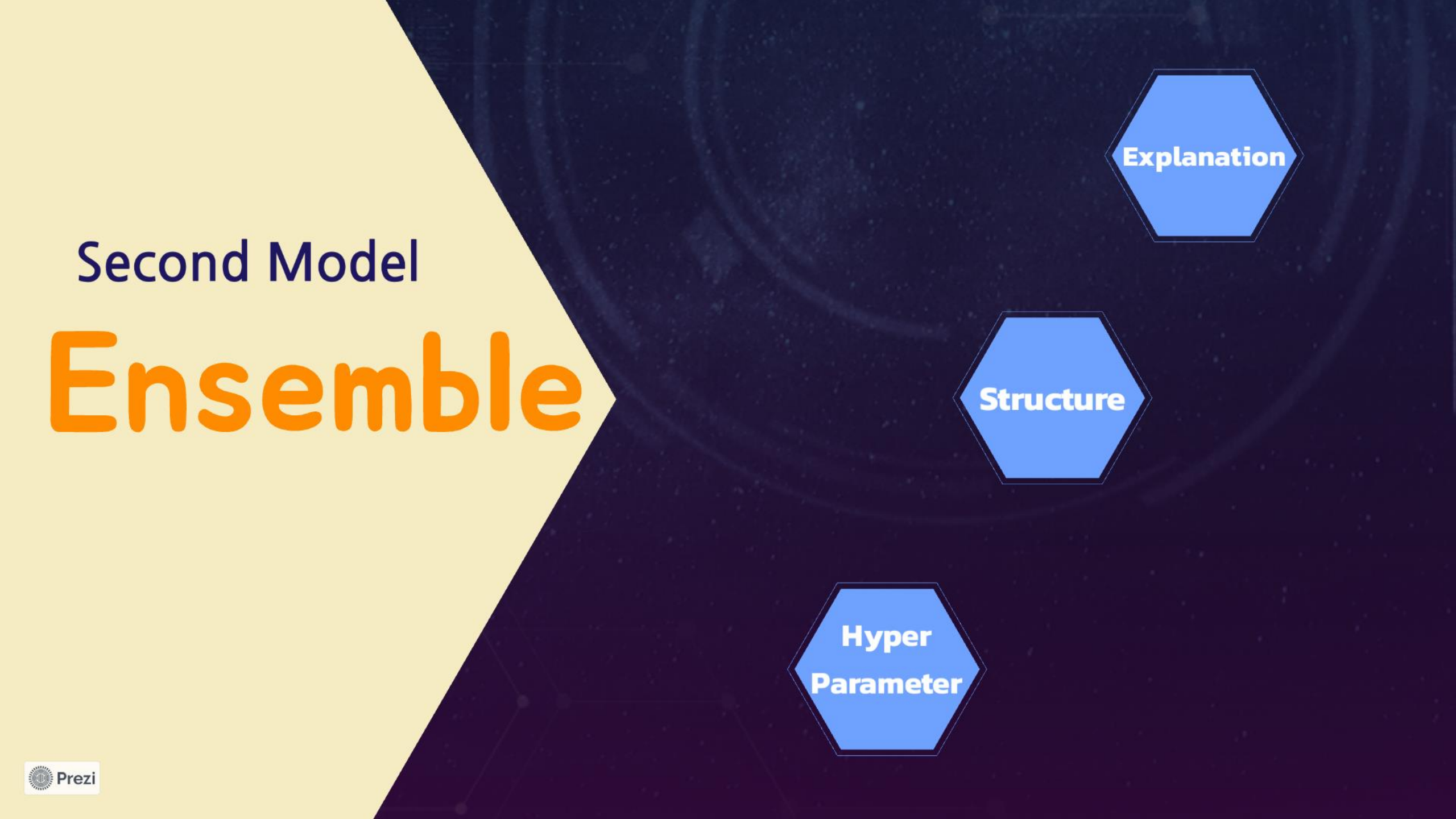
- batch_size 16
- epochs 10
- validation_split 0.2

ImageDataGenerator

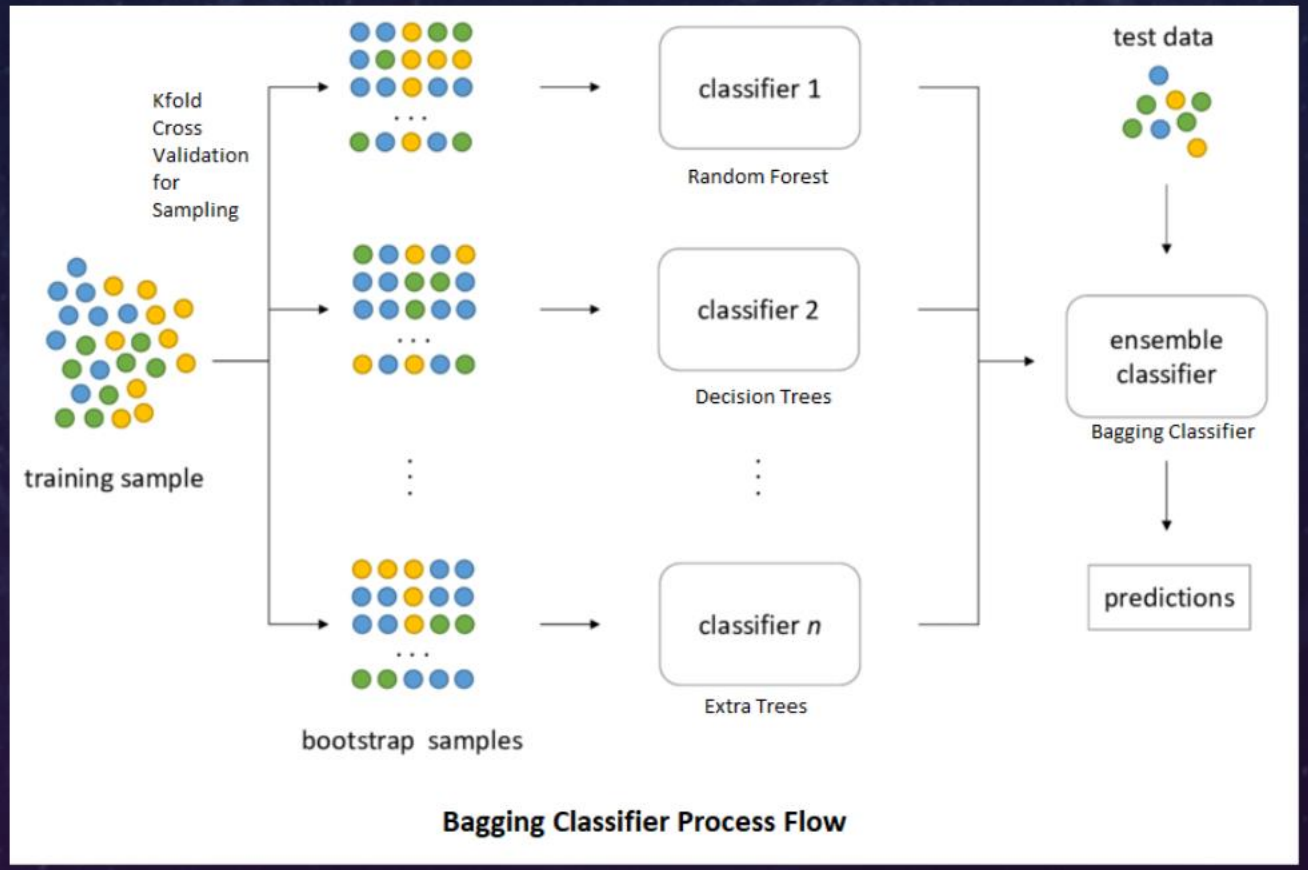
- rotation_range 40
- width_shift_range 0.2
- height_shift_range 0.2
- zoom_range 0.2
- horizontal_flip True
- fill_mode nearest

Second Model

Ensemble



Ensemble



Bagging Classifier Process Flow

- 여러개의 모델을 조화롭게 학습시켜 모델의 예측 결과들을 이용
- 여러개의 결정트리를 결합하여 좋은 성능을 내는 기법
- Bagging 기법과 Boosting 기법이 존재

Structure

Model

Model 1

```
model1 = Conv2D(16,(7,7), activation='relu')(input_model)
model1 = Conv2D(32,(6,6), activation='relu', padding='same')(model1)
model1 = BatchNormalization()(model1)
model1 = MaxPooling2D((2,2))(model1)
model1 = Conv2D(32,(6,6), activation='relu' ,padding='same')(model1)
model1 = Conv2D(64,(5,5), activation='relu' ,padding='same')(model1)
model1 = BatchNormalization()(model1)
model1 = AveragePooling2D((2, 2))(model1)
model1 = Conv2D(64,(5,5), activation='relu' ,padding='same')(model1)
model1 = Conv2D(128,(5,5), activation='relu' ,padding='same')(model1)
model1 = BatchNormalization()(model1)
model1 = AveragePooling2D((2, 2))(model1)
model1 = Conv2D(256,(4,4), activation='relu' ,padding='same')(model1)
model1 = Conv2D(256,(4,4), activation='relu' ,padding='same')(model1)
model1 = BatchNormalization()(model1)
model1 = MaxPooling2D((2, 2))(model1)
model1 = Conv2D(512,(3,3), activation='relu' ,padding='same')(model1)
model1 = Conv2D(512,(3,3), activation='relu' ,padding='valid')(model1)
model1 = BatchNormalization()(model1)
model1 = Flatten()(model1)
```

Model 2

```
model2 = Conv2D(16,(4,4), activation='relu')(input_model)
model2 = Conv2D(16,(4,4), activation='relu', padding='same')(model2)
model2 = BatchNormalization()(model2)
model2 = MaxPooling2D((3, 3))(model2)
model2 = Conv2D(32,(3,3), activation='relu', padding='same')(model2)
model2 = Conv2D(32,(3,3), activation='relu', padding='same')(model2)
model2 = BatchNormalization()(model2)
model2 = AveragePooling2D((2, 2))(model2)
model2 = Conv2D(32,(3,3), activation='relu', padding='same')(model2)
model2 = Conv2D(64,(2,2), activation='relu' ,padding='same')(model2)
model2 = BatchNormalization()(model2)
model2 = AveragePooling2D((2, 2))(model2)
model2 = Conv2D(64,(2,2), activation='relu' ,padding='same')(model2)
model2 = Conv2D(64,(2,2), activation='relu' ,padding='same')(model2)
model2 = BatchNormalization()(model2)
model2 = AveragePooling2D((2, 2))(model2)
model2 = Conv2D(128,(1,1), activation='relu' ,padding='same')(model2)
model2 = Conv2D(128,(1,1), activation='relu' ,padding='same')(model2)
model2 = BatchNormalization()(model2)
model2 = AveragePooling2D((2, 2))(model2)
model2 = Conv2D(256,(1,1), activation='relu' ,padding='same')(model2)
model2 = Conv2D(512,(1,1), activation='relu' ,padding='valid')(model2)
model2 = BatchNormalization()(model2)
model2 = Flatten()(model2)
```

Merge

```
merged = Concatenate()([model1, model2])
merged = Dense(units = 512, activation = 'relu')(merged)
merged = BatchNormalization()(merged)
merged = Dropout(rate = 0.2)(merged)
merged = Dense(units = 64, activation = 'relu')(merged)
merged = Dense(units = 32, activation = 'relu')(merged)
merged = Dense(units = 16, activation = 'relu')(merged)
merged = Dense(units = 8, activation = 'relu')(merged)
merged = Dense(units = 4, activation = 'relu')(merged)
output = Dense(activation = 'softmax', units = 3)(merged)

model = Model(inputs= [input_model], outputs=[output])
```

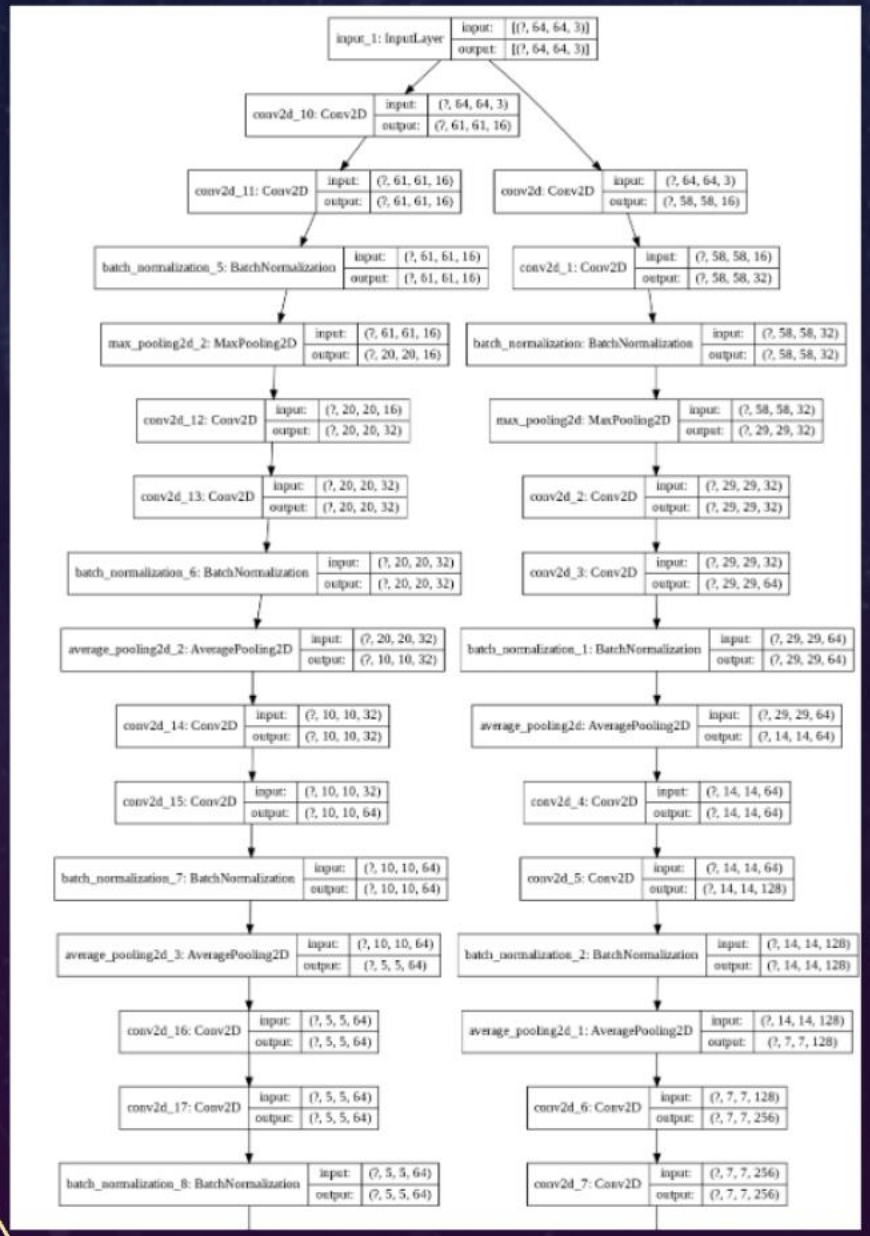
- **Model 1**

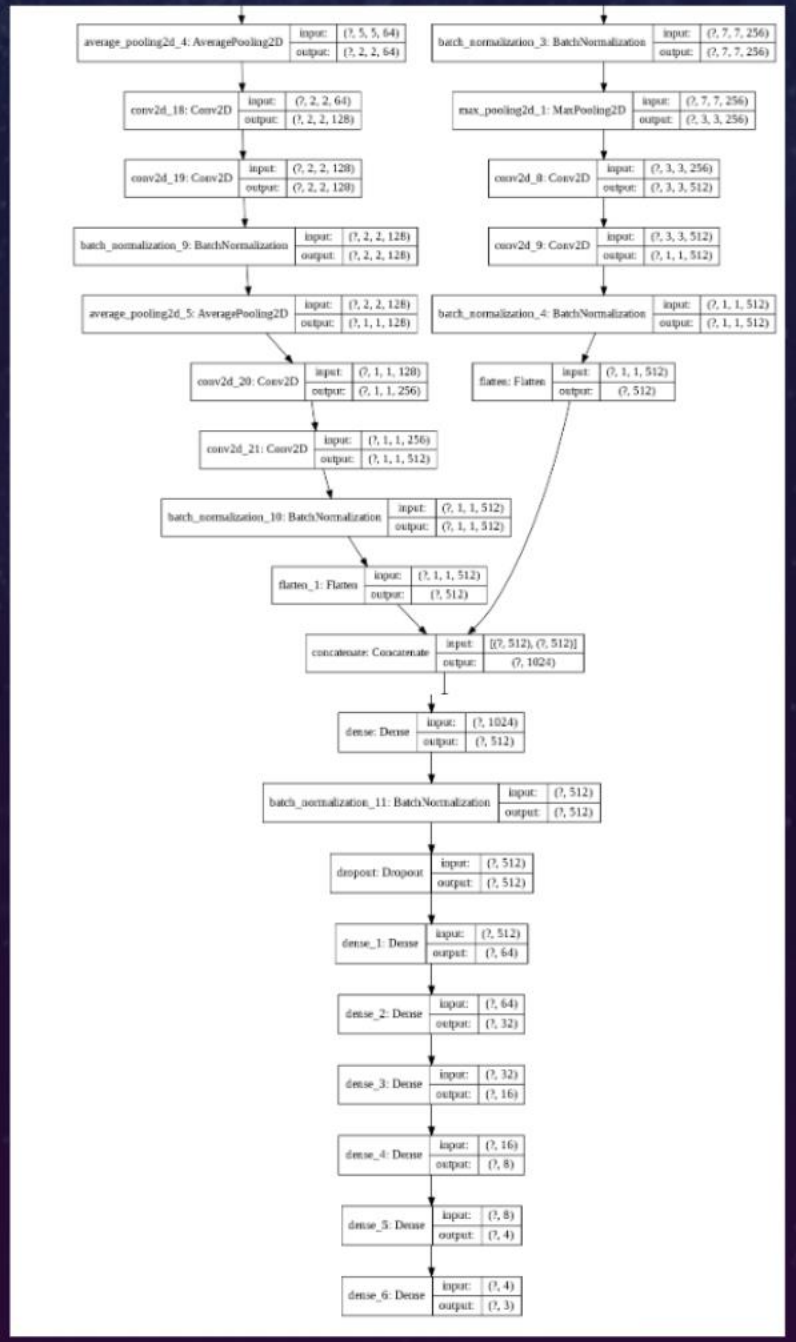
ConV	10
MaxPooling	2
AveragePooling	2
BatchNormalization	5

- **Model 2**

ConV	10
MaxPooling	1
AveragePooling	4
BatchNormalization	6

Plot





Hyper Parameter

compile

- optimizer adam
- loss categorical_crossentropy
- metrics accuracy

EarlyStopping

- monitor val_loss
- mode auto
- patience 8

Fit

- batch_size 16
- epochs 10
- validation_split 0.2

ImageDataGenerator

- rotation_range 40
- width_shift_range 0.2
- height_shift_range 0.2
- zoom_range 0.2
- horizontal_flip True
- fill_mode nearest

Third Model

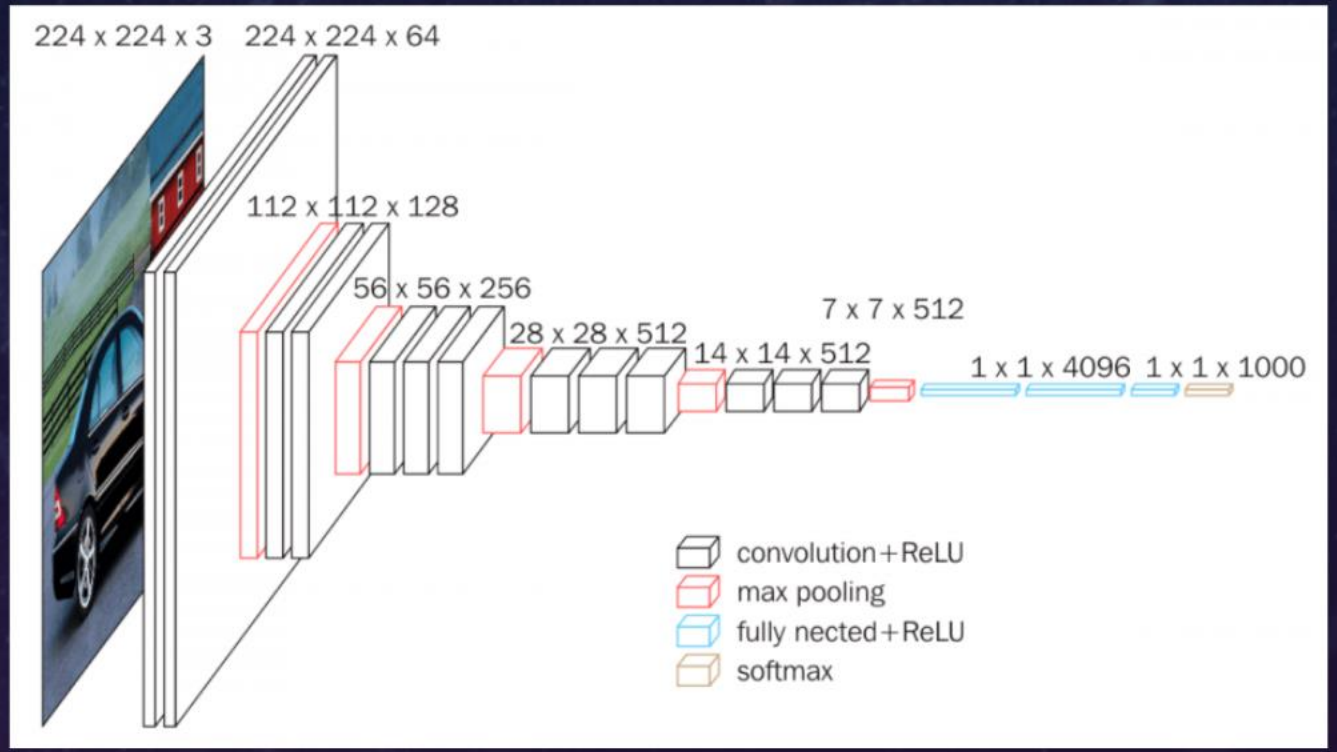
VGG16

Explanation

Structure

Hyper Parameter

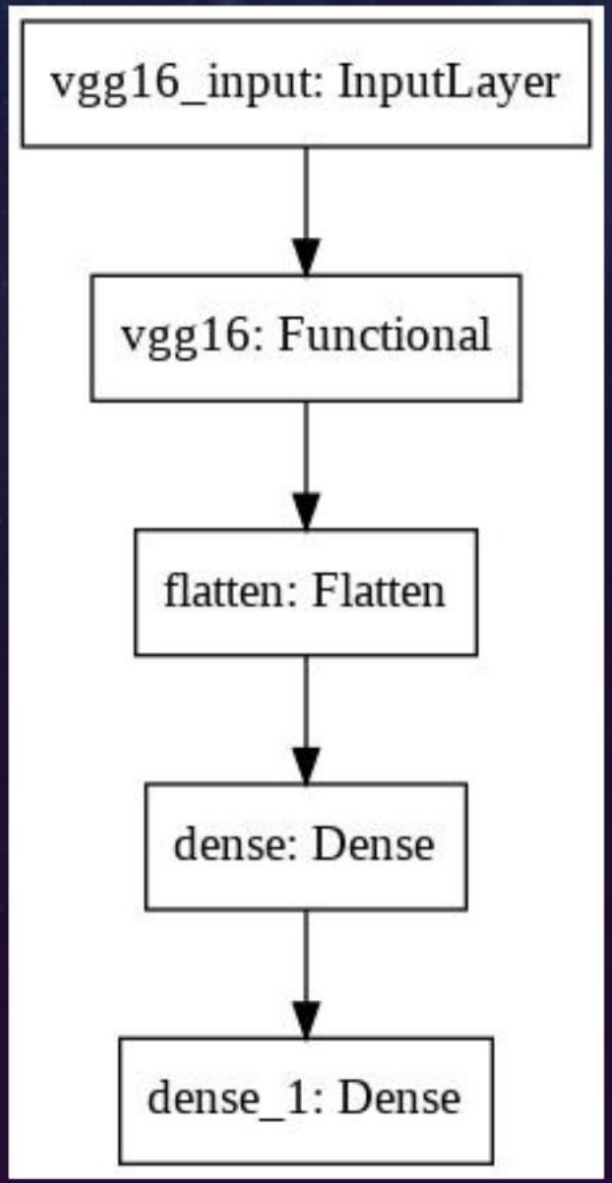
Visual Geometry Group 16



- CNN 이미지 분류 알고리즘의 하나로써 2014년 ILSVRC에서 준우승한 모델
- VGG16은 총 16개의 층으로 구성됨
- VGGNet은 네트워크의 깊이가 성능에 어떤 영향을 미치는지 잘 보여준 모델
- 네트워크의 영향만을 확인하고자 Conv Filter 사이즈를 가장 작은 3x3으로 고정함
- (224 x 224 x 3) 크기의 Input Image를 받을 수 있다

1. vgg16
2. Dense
3. Softmax

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Functional)           (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 256)               6422784
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 771
=================================================================
Total params: 21,138,243
Trainable params: 6,423,555
Non-trainable params: 14,714,688
```



Structure

Hyper Parameter

compile

- optimizer adam
- loss categorical_crossentropy
- metrics accuracy

EarlyStopping

- monitor val_loss
- mode auto
- patience 8

Fit

- batch_size 16
- epochs 10
- validation_split 0.2

ImageDataGenerator

- rotation_range 40
- width_shift_range 0.2
- height_shift_range 0.2
- zoom_range 0.2
- horizontal_flip True
- fill_mode nearest

Validation

Ensemble

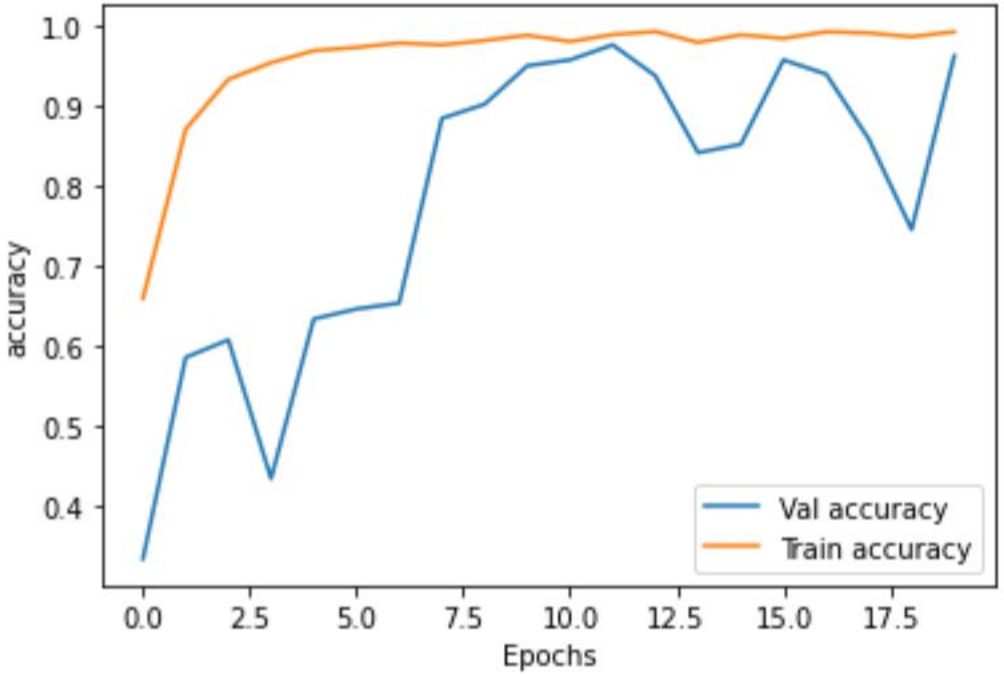
VGG16

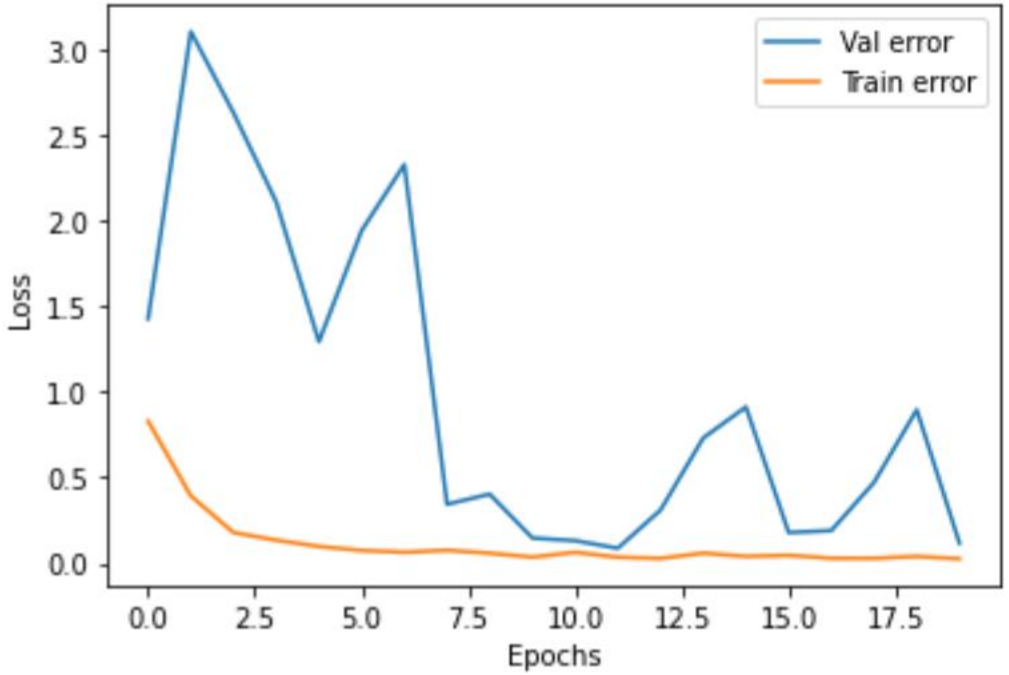
CNN

Ensemble Validation

Training

Test

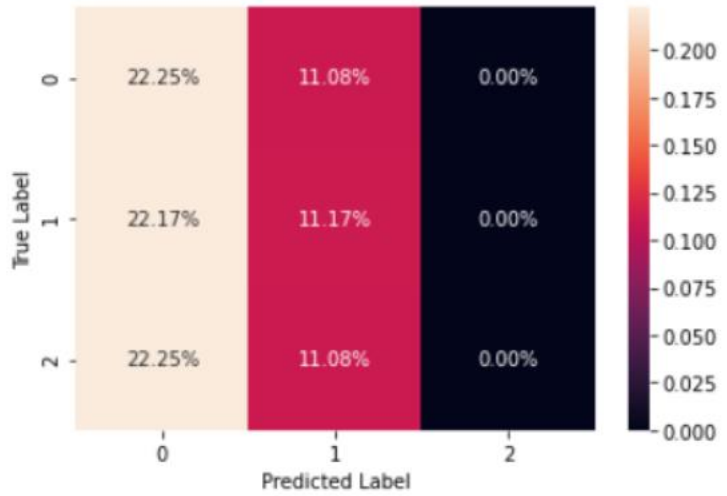


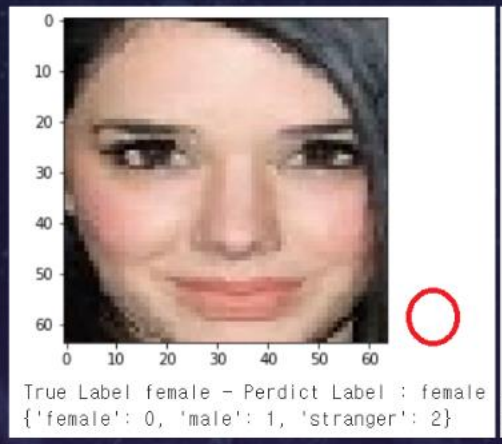


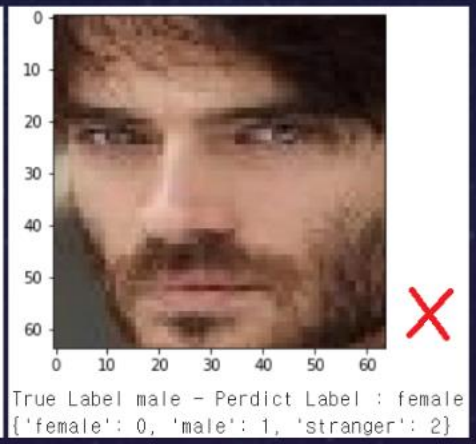
Accuracy & Loss

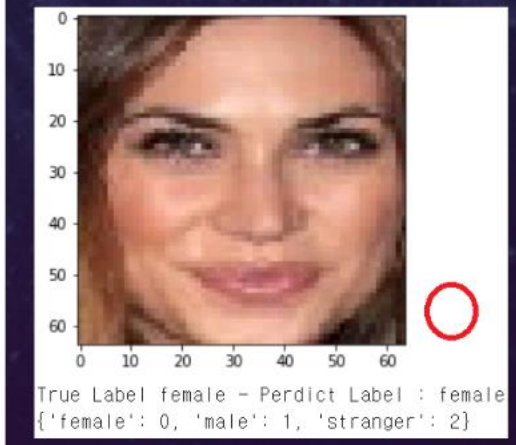
```
#evaluate the model
scores = model.evaluate_generator(evaluation_generator)
print("₩n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

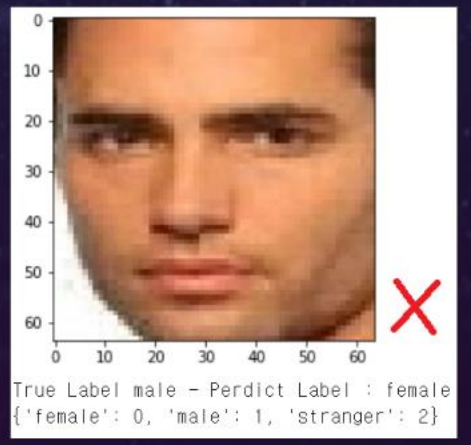
```
WARNING:tensorflow:From <ipython-input-17-acb3e08369cc>:2: Model
Instructions for updating:
Please use Model.evaluate, which supports generators.

accuracy: 92.67%
```

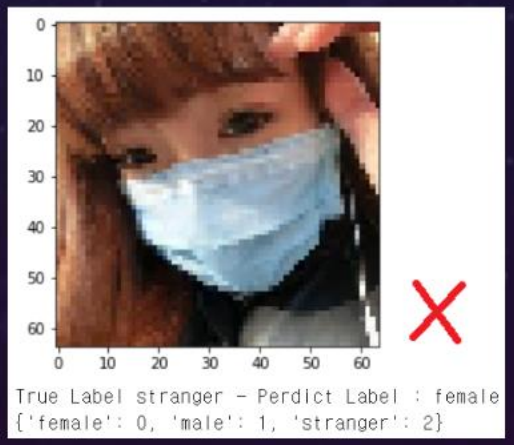












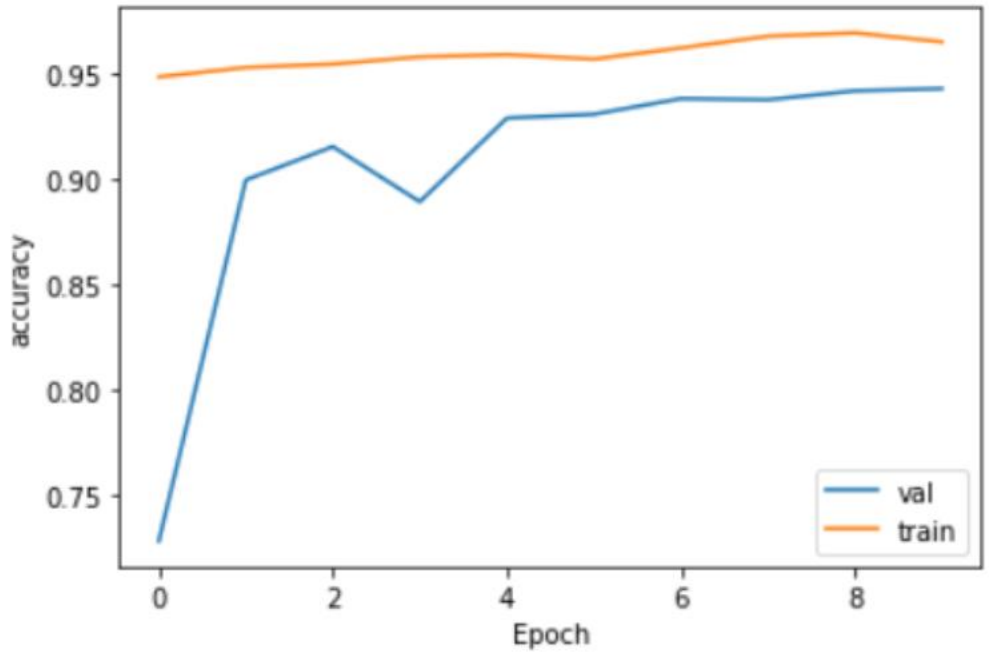
Evaluate & Visualization

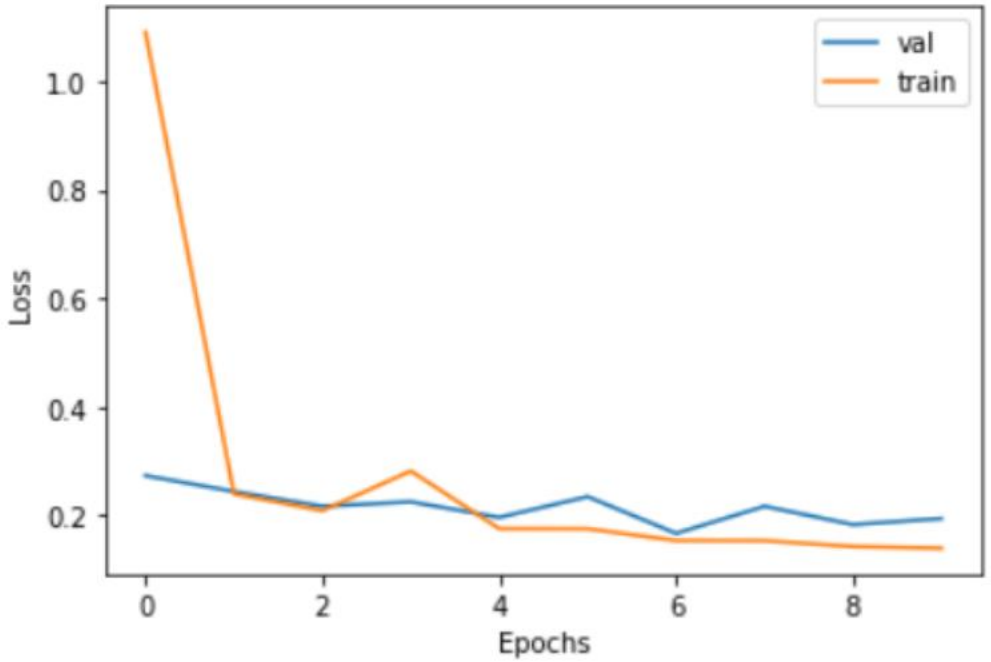
VGG 16

Validation

Training

Test

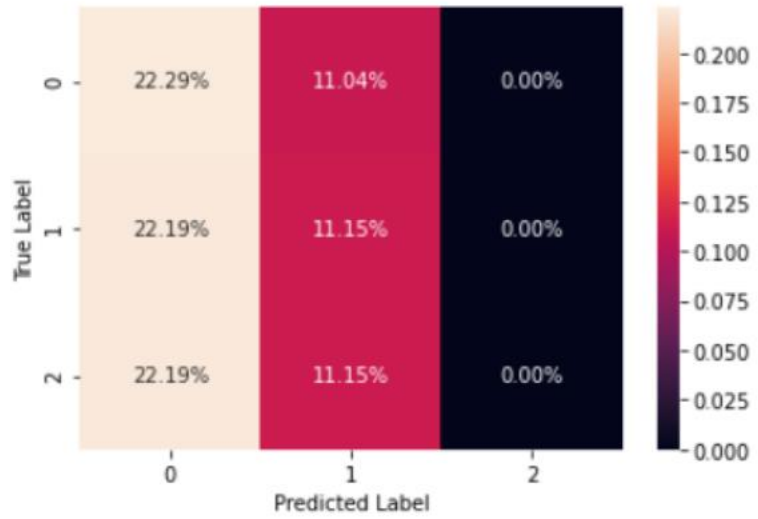


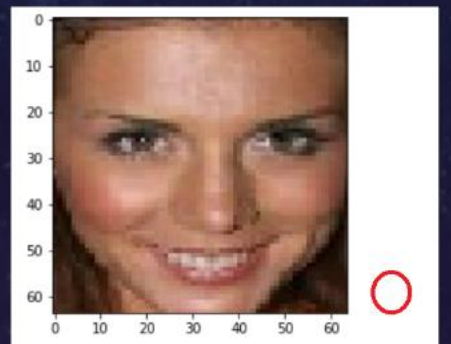


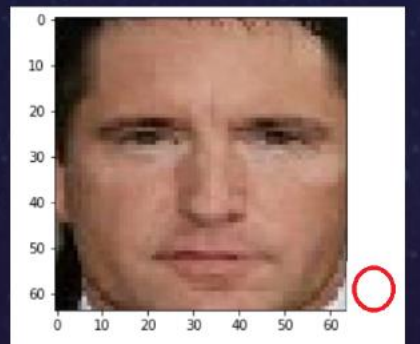
Accuracy & Loss

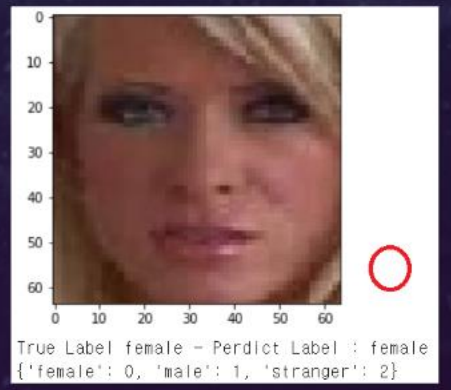
```
results_with_tuning = model.evaluate_generator(test_gen,
                                               len_test_data // batch_size,
                                               verbose=1)
```

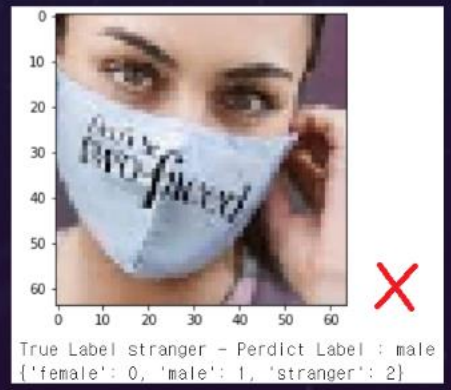
```
16/16 [==============================] - 3s 209ms/step - loss: 0.0692 - accuracy: 0.9771
```











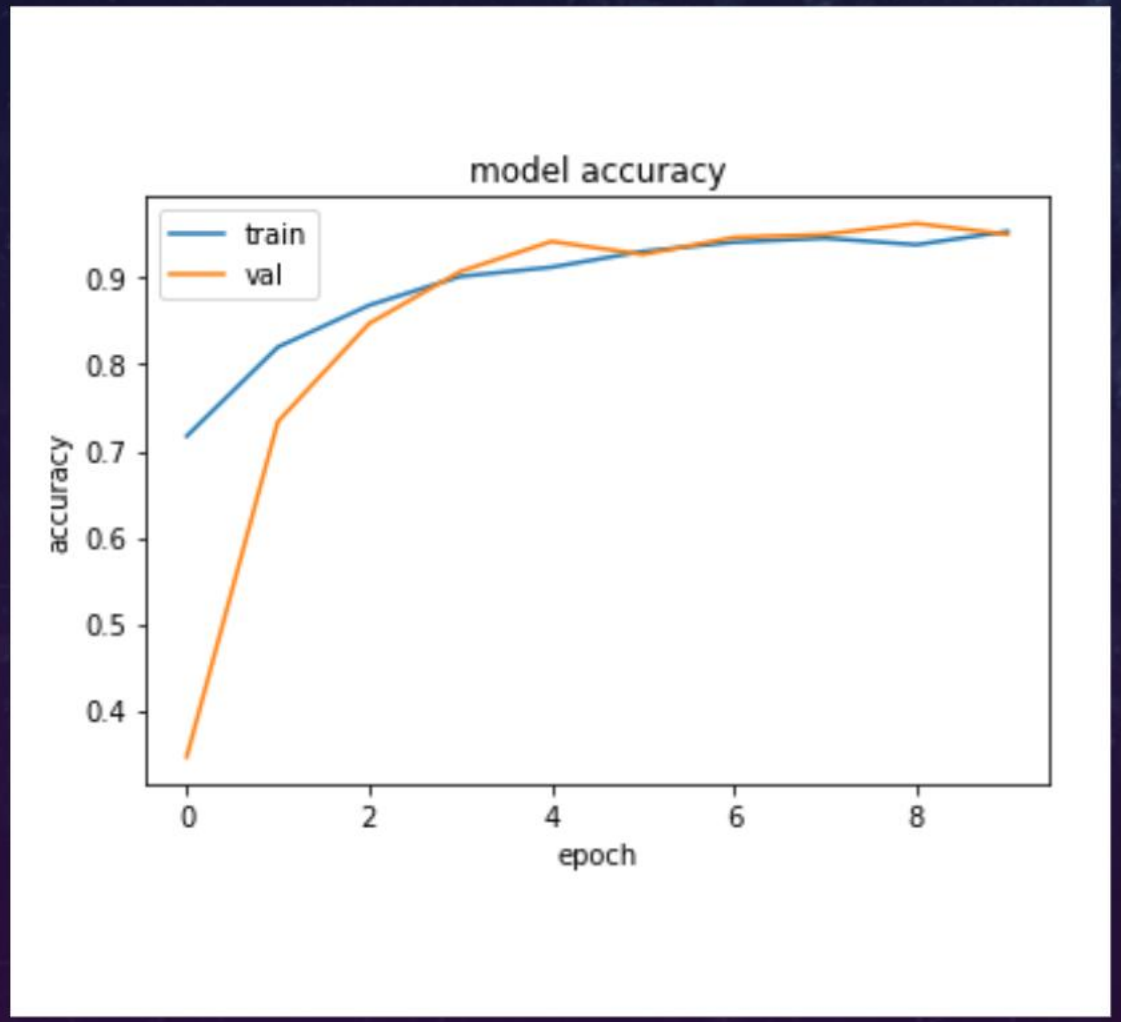
Evaluate & Visualization

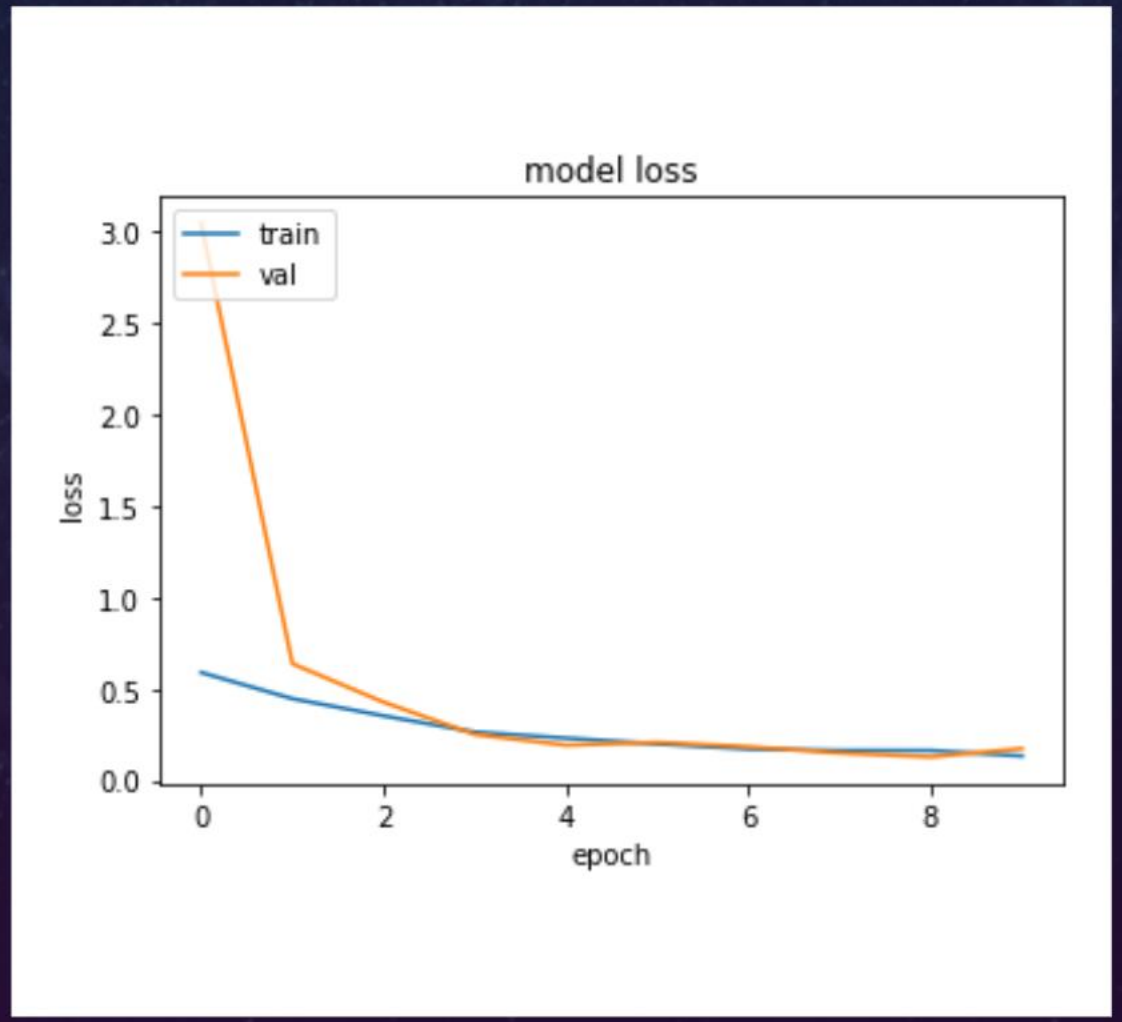
CNN

Validation

Training

Test





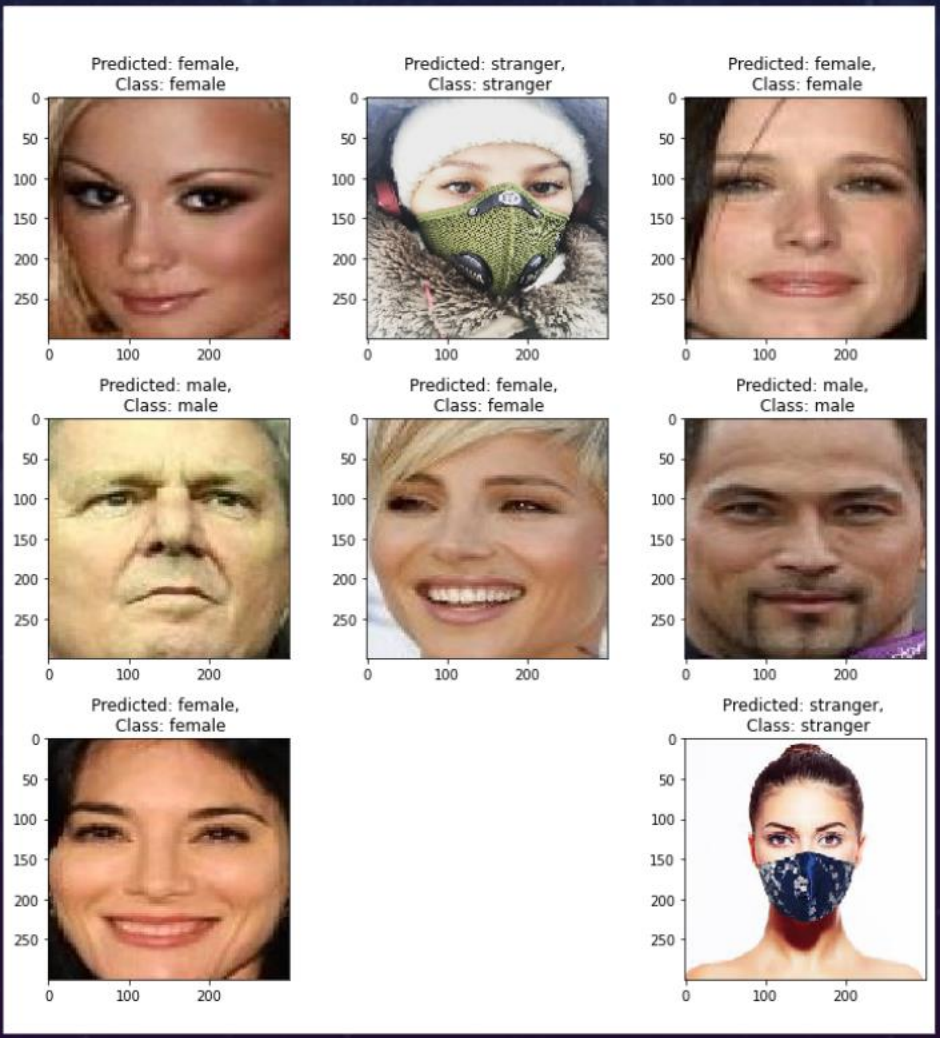
Accuracy & Loss

```
#X_test=np.ndarray(X_test)
loss_and_metrics = model.evaluate(X_test, y_test_labels, batch_size=48)
print('## evaluation loss and_metrics ##')
print(loss_and_metrics)
```

```
10/10 [==============================] - 0s 33ms/step - loss: 0.1407 - accuracy: 0.9604
## evaluation loss and_metrics ##
[0.14073750376701355, 0.9604166746139526]
```

```
print(classification_report(predictions, y_test))
```

	precision	recall	f1-score	support
0	0.91	0.99	0.95	149
1	0.99	0.91	0.95	188
2	0.98	1.00	0.99	143
accuracy			0.96	480
macro avg	0.96	0.97	0.96	480
weighted avg	0.96	0.96	0.96	480



Evaluate & Visualization

Validation

Ensemble

VGG16

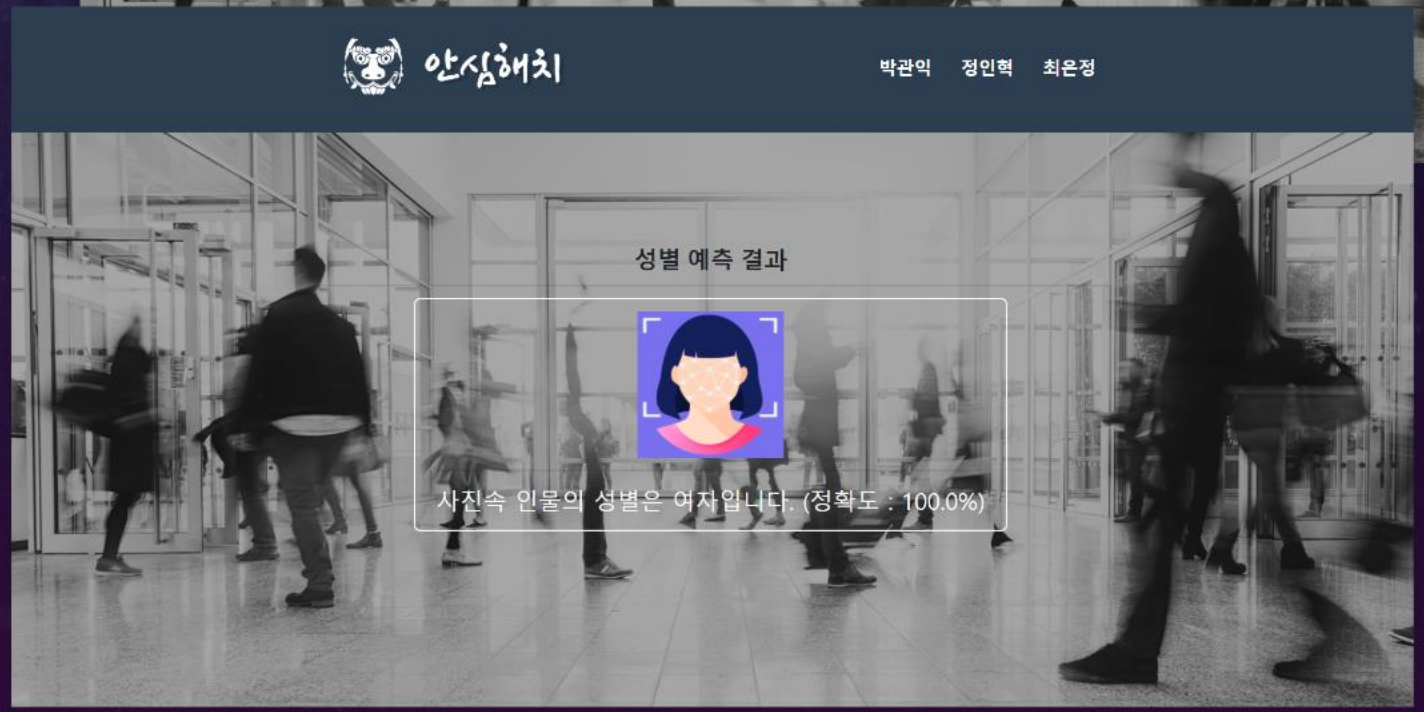
CNN

시연

웹을 사용한 구현 방식



이미지를 선택하여
입력하면
남자, 여자, 수상한 사람
으로 구별

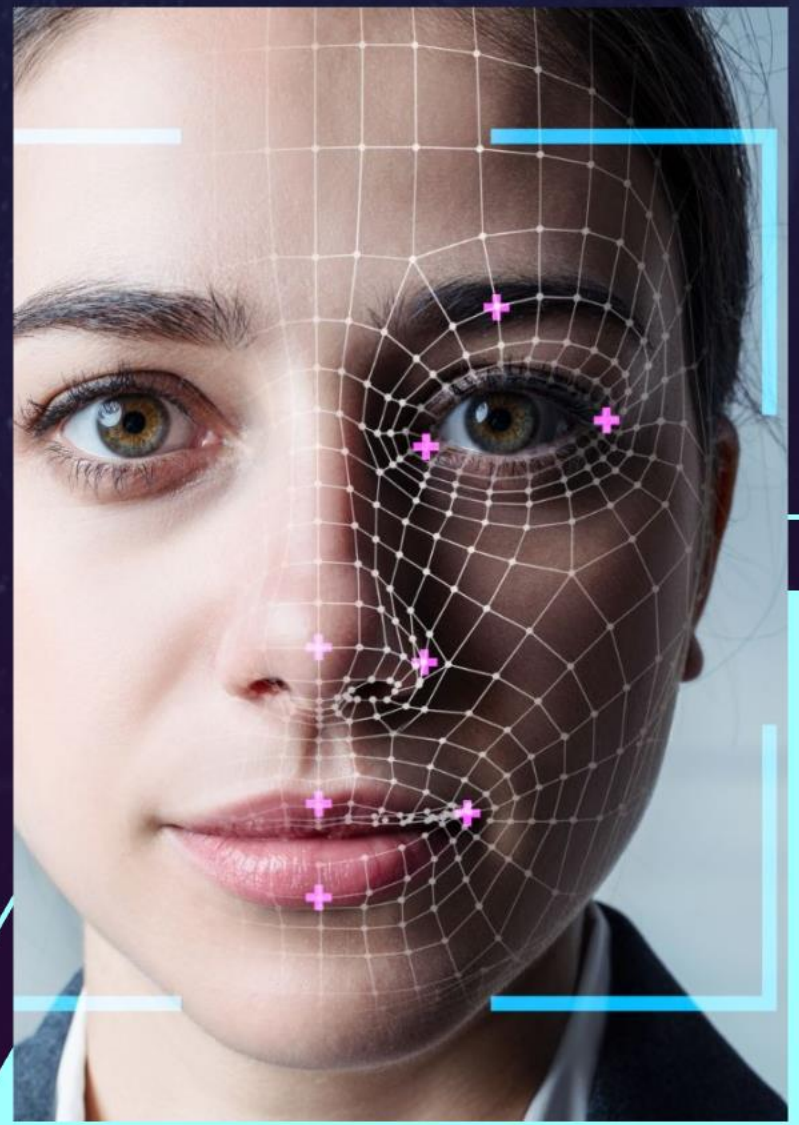


개선방향

IoT와 결합해 직접적으로 출입을 제한하고 유사시 소리 등으로 알려주는 시스템 구축

현재는 얼굴로만 구분하지만 신장이나 체형같은 정보를 추가로 인식할 수 있게 하여 정확도 향상

마스크를 쓴 상태에서 성별 구분이 가능하도록 하고 헬멧, 선글라스 등 다양한 변수를 추가로 인식할 수 있도록 학습시킴



Thank you!