KT AIVLE School

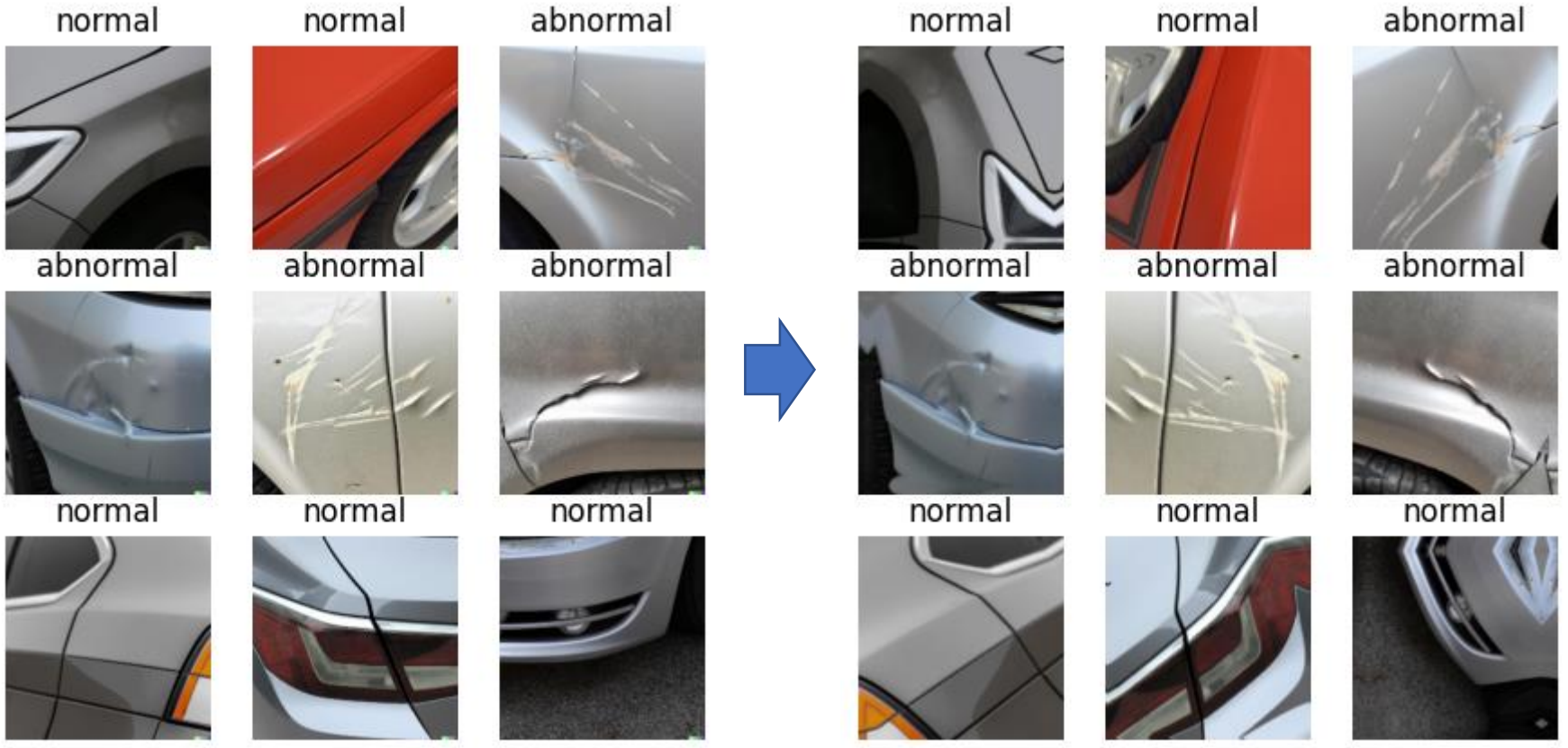
4차 미니프로젝트_조별 발표 템플릿

AI 03반 11조

Let’s make it possible

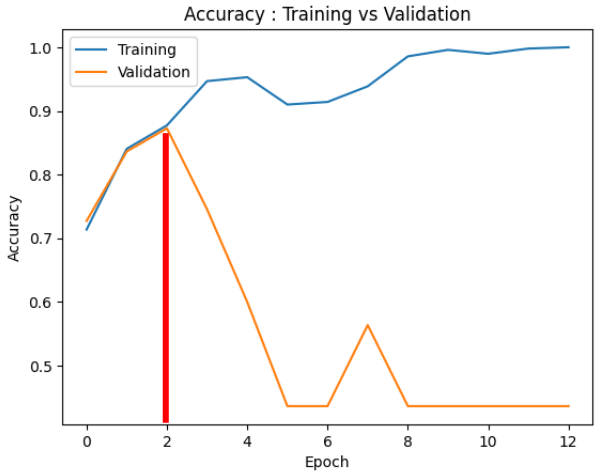
1. 데이터 전처리 (Transfer-Learning)

1. Normalization (Min-Max Scaler)
2. Data Augmentation
   - RandomFlip(“horizontal”)
   - RandomRotation(0.2) # 20도 회전
   - RandomZoom(0.2) # 20% 확대 또는 축소



2. 모델링

1. CNN 모델 설계
   - 4 Layer
   - Standardization (RGB 별 처리)
   - Data augmentation X
   - 파라미터 수 : 8.6MB




```python
## Functional API
# 1. 세션 클리어
clear_session()

# 2. 레이어 엮기
il = Input(shape=(224,224,3))

conv1 = Conv2D(128, (3,3), (1,1), 'same', activation='relu')(il)
conv1 = BatchNormalization()(conv1)
conv1 = MaxPool2D((2,2), (2,2))(conv1)

conv2 = Conv2D(256, (3,3), (1,1), 'same', activation='relu')(conv1)
conv2 = BatchNormalization()(conv2)
conv2 = MaxPool2D((2,2), (2,2))(conv2)

conv3 = Conv2D(128, (3,3), (1,1), 'same', activation='relu')(conv2)
conv3 = BatchNormalization()(conv3)
conv3 = MaxPool2D((2,2), (2,2))(conv3)

conv4 = Conv2D(64, (3,3), (1,1), 'same', activation='relu')(conv3)
conv4 = BatchNormalization()(conv4)
conv4 = MaxPool2D((2,2), (2,2))(conv4)

full = Flatten()(conv4)
full = Dense(128, activation='relu')(full)

ol = Dense(1, activation='sigmoid')(full)

# 3. 모델의 시작과 끝 지정
model = Model(il, ol)

# 4. 컴파일
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy']
              )

model.summary()
```

2. 모델링

2. 성능지표

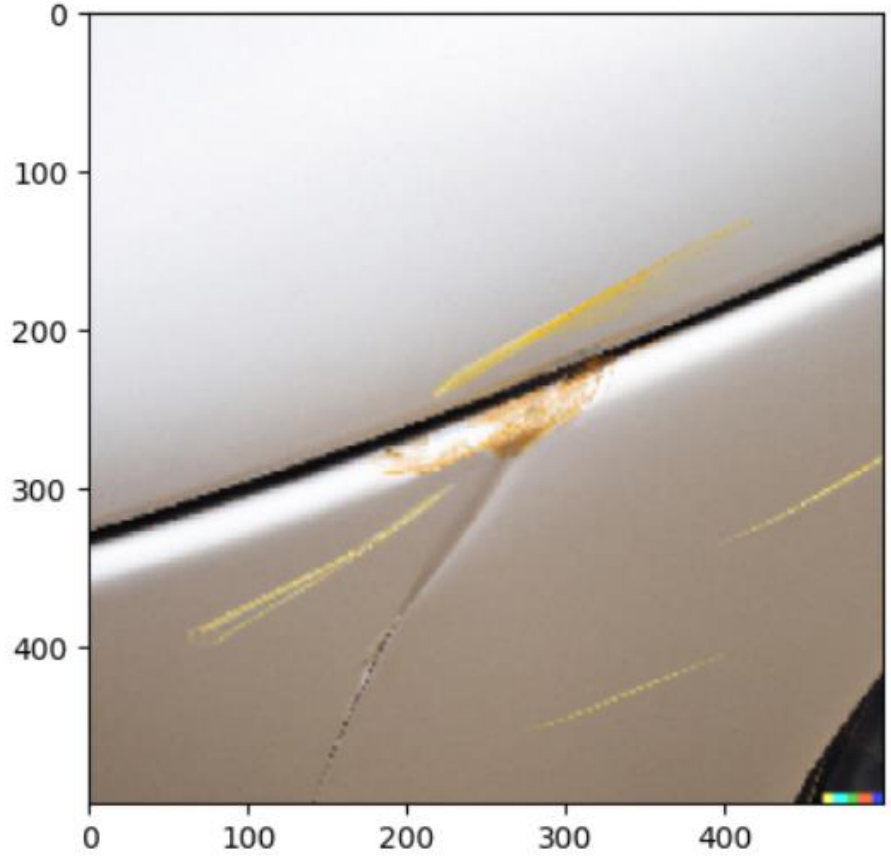
Model	Total Params	Accuracy	Recall
CNN	8.68 MB	0.89	0.83
InceptionV3	83.18 MB	1.00	1.00
ResNet-50	90.00 MB	0.88	0.94
Inception+ResNetV2		0.93	0.95
VGG16	56.14 MB	0.94	1.00
EfficientNetV2L	449.17 MB	0.70	0.48

- 파라미터 수가 많을 수록 성능이 좋아지지만, 너무 많은 파라미터를 학습하면 성능이 떨어진다.

2. 모델링

3.예측 결과

```
id = 1
다음 그림은 abnormal 입니다.
모델의 예측 : Normal
모델의 카테고리별 확률 :
{'abnormal': 46.0}
틀렸어요
```




```
id = 21
다음 그림은 abnormal 입니다.
모델의 예측 : abnormal
모델의 카테고리별 확률 :
{'Normal': 0.0, 'abnormal': 99.0}
정답입니다
```