

[멋사 13기][AI_유다인]

1. 에폭(epoch) 변경 방법 시도

에폭이란? 전체 훈련 데이터를 한번 완전히 학습시키는 과정

- **Epoch:** 전체 데이터셋을 한 번 학습하는 과정
- **Batch:** 데이터셋을 작은 그룹으로 나누어 학습하는 단위
- **Iteration:** 한 번의 Batch를 학습하는 과정

#1차 시도

```
history = model.fit(
    train_generator,
    epochs=15,
    validation_data=validation_generator
)
```

... /usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning:
self._warn_if_super_not_called()
Epoch 1/15
552/1281 ----- 31s 43ms/step - accuracy: 0.5271 - loss: 0.6977

10회였던 횟수를 15회로 변경하여 실행

```

Epoch 1/15
1281/1281 ----- 68s 51ms/step - accuracy: 0.5309 - loss: 0.6954 - val_accuracy: 0.5359 - val_loss: 0.7106
Epoch 2/15
1281/1281 ----- 61s 48ms/step - accuracy: 0.5587 - loss: 0.6813 - val_accuracy: 0.6227 - val_loss: 0.6548
Epoch 3/15
1281/1281 ----- 87s 52ms/step - accuracy: 0.6339 - loss: 0.6455 - val_accuracy: 0.6408 - val_loss: 0.6239
Epoch 4/15
1281/1281 ----- 61s 48ms/step - accuracy: 0.6438 - loss: 0.6305 - val_accuracy: 0.6433 - val_loss: 0.6154
Epoch 5/15
1281/1281 ----- 82s 48ms/step - accuracy: 0.6630 - loss: 0.6148 - val_accuracy: 0.6583 - val_loss: 0.6160
Epoch 6/15
1281/1281 ----- 63s 49ms/step - accuracy: 0.6605 - loss: 0.6113 - val_accuracy: 0.6908 - val_loss: 0.5766
Epoch 7/15
1281/1281 ----- 62s 48ms/step - accuracy: 0.6826 - loss: 0.5962 - val_accuracy: 0.6746 - val_loss: 0.5965
Epoch 8/15
1281/1281 ----- 64s 50ms/step - accuracy: 0.6890 - loss: 0.5911 - val_accuracy: 0.6964 - val_loss: 0.5774
Epoch 9/15
1281/1281 ----- 80s 48ms/step - accuracy: 0.6822 - loss: 0.5813 - val_accuracy: 0.6627 - val_loss: 0.6105
Epoch 10/15
1281/1281 ----- 63s 49ms/step - accuracy: 0.6943 - loss: 0.5768 - val_accuracy: 0.7052 - val_loss: 0.5660
Epoch 11/15
1281/1281 ----- 63s 49ms/step - accuracy: 0.7062 - loss: 0.5680 - val_accuracy: 0.6858 - val_loss: 0.5944
Epoch 12/15
1281/1281 ----- 62s 48ms/step - accuracy: 0.7217 - loss: 0.5598 - val_accuracy: 0.6627 - val_loss: 0.5901
Epoch 13/15
1281/1281 ----- 65s 51ms/step - accuracy: 0.7146 - loss: 0.5585 - val_accuracy: 0.7171 - val_loss: 0.5734
Epoch 14/15
1281/1281 ----- 65s 51ms/step - accuracy: 0.7042 - loss: 0.5649 - val_accuracy: 0.7152 - val_loss: 0.5554
Epoch 15/15
1281/1281 ----- 65s 51ms/step - accuracy: 0.7121 - loss: 0.5617 - val_accuracy: 0.6977 - val_loss: 0.5609

```

1회의 경우

val_accuracy: 0.5359

va_loss: 0.7106

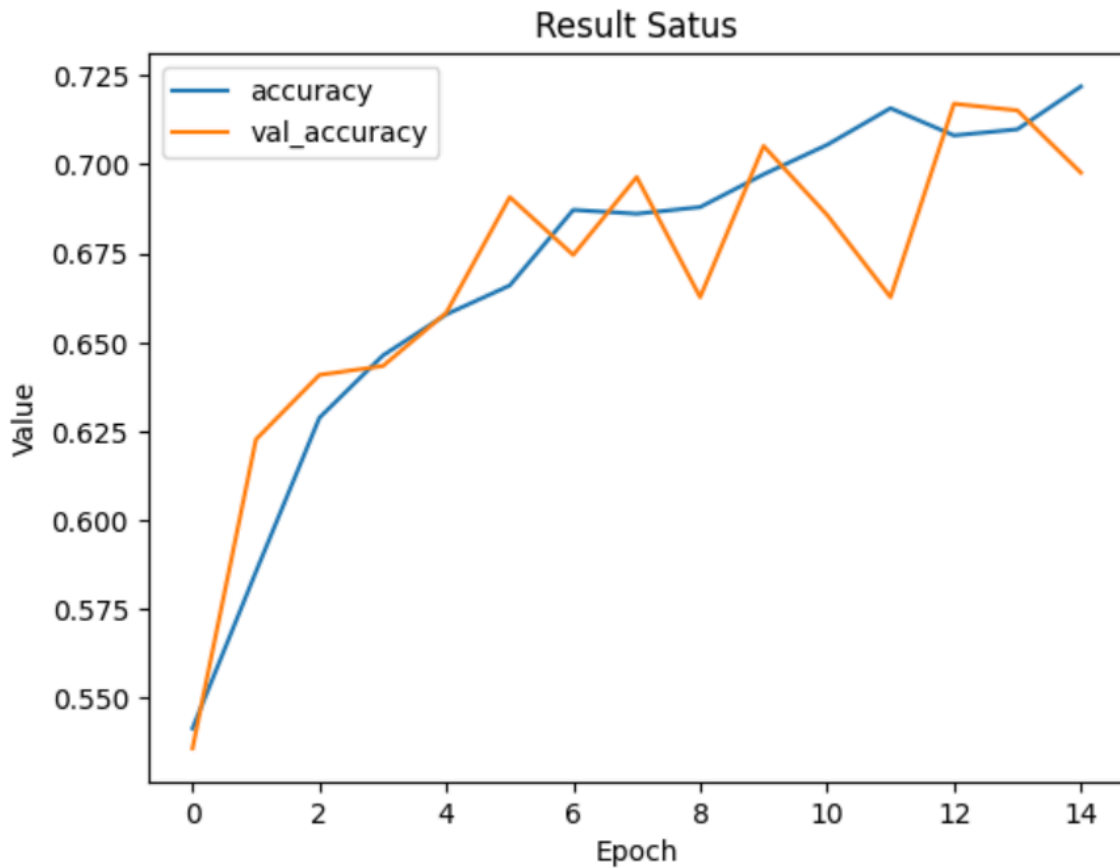
15회의 경우

val_accuracy:0.6977

va_loss:0.5609

정확도는 0.1618, 약 30.2% 상승

손실도는 0.1497, 약 21.1% 하락



예측할 이미지를 업로드하세요.

파일 선택 dog.88.jpg

• **dog.88.jpg**(image/jpeg) - 33505 bytes, last modified: 2025. 6. 2. - 100% done

Saving dog.88.jpg to dog.88.jpg

1/1 ----- 0s 48ms/step

dog.88.jpg: 강아지입니다.

강아지 사진을 첨부한 결과 예측은 성공했지만 훈련 데이터에서 71.21%, 검증 데이터에서 69.77% 결과 확인

15회에서 20회로 늘려 다시 시도

#2차 시도

```
[7] history = model.fit(  
    train_generator,  
    epochs=20,  
    validation_data=validation_generator  
)
```

20으로 설정 후,

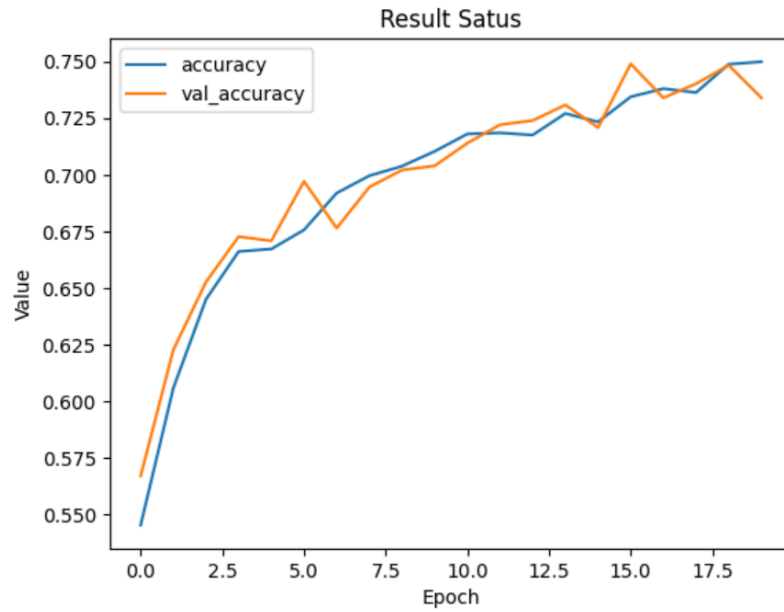
1회는 val_accuracy: 0.5671 - val_loss: 0.6701

20회는 val_accuracy: 0.7339 - val_loss: 0.5195

1차 시도에 비해 훈련 데이터에서 75.38%,

검증 데이터에서 73.39%로 나온 것을 볼 수 있음

Epoch 1/20	1281/1281	-----	68s	51ms/step	- accuracy: 0.5248 - loss: 0.6976 - val_accuracy: 0.5671 - val_loss: 0.6701
Epoch 2/20	1281/1281	-----	65s	50ms/step	- accuracy: 0.5892 - loss: 0.6688 - val_accuracy: 0.6227 - val_loss: 0.6461
Epoch 3/20	1281/1281	-----	64s	50ms/step	- accuracy: 0.6363 - loss: 0.6369 - val_accuracy: 0.6527 - val_loss: 0.6163
Epoch 4/20	1281/1281	-----	64s	50ms/step	- accuracy: 0.6609 - loss: 0.6170 - val_accuracy: 0.6727 - val_loss: 0.6129
Epoch 5/20	1281/1281	-----	68s	53ms/step	- accuracy: 0.6623 - loss: 0.6046 - val_accuracy: 0.6708 - val_loss: 0.5954
Epoch 6/20	1281/1281	-----	65s	51ms/step	- accuracy: 0.6787 - loss: 0.5931 - val_accuracy: 0.6971 - val_loss: 0.5761
Epoch 7/20	1281/1281	-----	65s	51ms/step	- accuracy: 0.6966 - loss: 0.5784 - val_accuracy: 0.6765 - val_loss: 0.6180
Epoch 8/20	1281/1281	-----	86s	54ms/step	- accuracy: 0.7025 - loss: 0.5710 - val_accuracy: 0.6946 - val_loss: 0.5933
Epoch 9/20	1281/1281	-----	77s	50ms/step	- accuracy: 0.7053 - loss: 0.5665 - val_accuracy: 0.7021 - val_loss: 0.5613
Epoch 10/20	1281/1281	-----	64s	50ms/step	- accuracy: 0.7079 - loss: 0.5665 - val_accuracy: 0.7039 - val_loss: 0.5639
Epoch 11/20	1281/1281	-----	82s	50ms/step	- accuracy: 0.7168 - loss: 0.5545 - val_accuracy: 0.7139 - val_loss: 0.5545
Epoch 12/20	1281/1281	-----	64s	50ms/step	- accuracy: 0.7271 - loss: 0.5508 - val_accuracy: 0.7220 - val_loss: 0.5461
Epoch 13/20	1281/1281	-----	64s	50ms/step	- accuracy: 0.7144 - loss: 0.5625 - val_accuracy: 0.7239 - val_loss: 0.5463
Epoch 14/20	1281/1281	-----	64s	50ms/step	- accuracy: 0.7268 - loss: 0.5414 - val_accuracy: 0.7308 - val_loss: 0.5378
Epoch 15/20	1281/1281	-----	82s	50ms/step	- accuracy: 0.7263 - loss: 0.5351 - val_accuracy: 0.7208 - val_loss: 0.5392
Epoch 16/20	1281/1281	-----	64s	50ms/step	- accuracy: 0.7407 - loss: 0.5148 - val_accuracy: 0.7489 - val_loss: 0.5023
Epoch 17/20	1281/1281	-----	67s	52ms/step	- accuracy: 0.7424 - loss: 0.5181 - val_accuracy: 0.7339 - val_loss: 0.5326
Epoch 18/20	1281/1281	-----	66s	51ms/step	- accuracy: 0.7340 - loss: 0.5265 - val_accuracy: 0.7402 - val_loss: 0.5179
Epoch 19/20	1281/1281	-----	65s	50ms/step	- accuracy: 0.7551 - loss: 0.5137 - val_accuracy: 0.7483 - val_loss: 0.5048
Epoch 20/20	1281/1281	-----	65s	50ms/step	- accuracy: 0.7538 - loss: 0.5088 - val_accuracy: 0.7339 - val_loss: 0.5195



- **cat.98.jpg**(image/jpeg) - 28927 bytes, last modified: 2025. 6. 2. - 100% done
 Saving cat.98.jpg to cat.98.jpg
 1/1 ----- 0s 136ms/step
 cat.98.jpg: 고양이입니다.

2. EarlyStopping 사용

EarlyStopping은 불필요한 학습을 방지하고 과적합을 줄이는 콜백 함수
 불필요한 Epich 반복을 줄이고 개선점이 보이지 않으면 학습을 멈춤

```

# EarlyStopping 설정
early_stopping = EarlyStopping(
    monitor='val_loss', # 검증 손실을 기준으로 모니터링
    patience=3, # 개선되지 않으면 3번 기다렸다가 종료
    restore_best_weights=True # 가장 성능 좋은 모델 가중치 복원
)

# 모델 학습 (EarlyStopping 적용)
model.fit(
    train_generator,
    validation_data=validation_generator,
    epochs=50,
    callbacks=[early_stopping] # EarlyStopping 추가
)

```

기존 코드에 최대 50회까지 학습 할 수 있도록 설정

```

1281/1281 ----- 65s 51ms/step - accuracy: 0.6741 - loss: 0.6070 - val_accuracy: 0.6883 - val_loss: 0.5956
Epoch 13/50
1281/1281 ----- 65s 51ms/step - accuracy: 0.6721 - loss: 0.6049 - val_accuracy: 0.6715 - val_loss: 0.5905
Epoch 14/50
1281/1281 ----- 66s 51ms/step - accuracy: 0.6809 - loss: 0.6034 - val_accuracy: 0.6908 - val_loss: 0.5838
Epoch 15/50
1281/1281 ----- 82s 51ms/step - accuracy: 0.6862 - loss: 0.5946 - val_accuracy: 0.7114 - val_loss: 0.5744
Epoch 16/50
1281/1281 ----- 66s 51ms/step - accuracy: 0.6871 - loss: 0.5926 - val_accuracy: 0.6702 - val_loss: 0.5927
Epoch 17/50
1281/1281 ----- 66s 51ms/step - accuracy: 0.6814 - loss: 0.5823 - val_accuracy: 0.7002 - val_loss: 0.5696
Epoch 18/50
1281/1281 ----- 65s 51ms/step - accuracy: 0.6994 - loss: 0.5846 - val_accuracy: 0.7014 - val_loss: 0.5776
Epoch 19/50
1281/1281 ----- 68s 53ms/step - accuracy: 0.7134 - loss: 0.5798 - val_accuracy: 0.7077 - val_loss: 0.5670
Epoch 20/50
1281/1281 ----- 64s 50ms/step - accuracy: 0.6860 - loss: 0.5883 - val_accuracy: 0.7196 - val_loss: 0.5703
Epoch 21/50
1281/1281 ----- 64s 50ms/step - accuracy: 0.7113 - loss: 0.5692 - val_accuracy: 0.6827 - val_loss: 0.5901
Epoch 22/50
1281/1281 ----- 64s 50ms/step - accuracy: 0.7087 - loss: 0.5707 - val_accuracy: 0.7345 - val_loss: 0.5280
Epoch 23/50
1281/1281 ----- 64s 50ms/step - accuracy: 0.7081 - loss: 0.5734 - val_accuracy: 0.7214 - val_loss: 0.5587
Epoch 24/50
1281/1281 ----- 64s 50ms/step - accuracy: 0.7138 - loss: 0.5721 - val_accuracy: 0.7370 - val_loss: 0.5518
Epoch 25/50
1281/1281 ----- 64s 50ms/step - accuracy: 0.7088 - loss: 0.5598 - val_accuracy: 0.7358 - val_loss: 0.5388
<keras.src.callbacks.history.History at 0x7d3d99398850>

```

25회에서 학습 종료

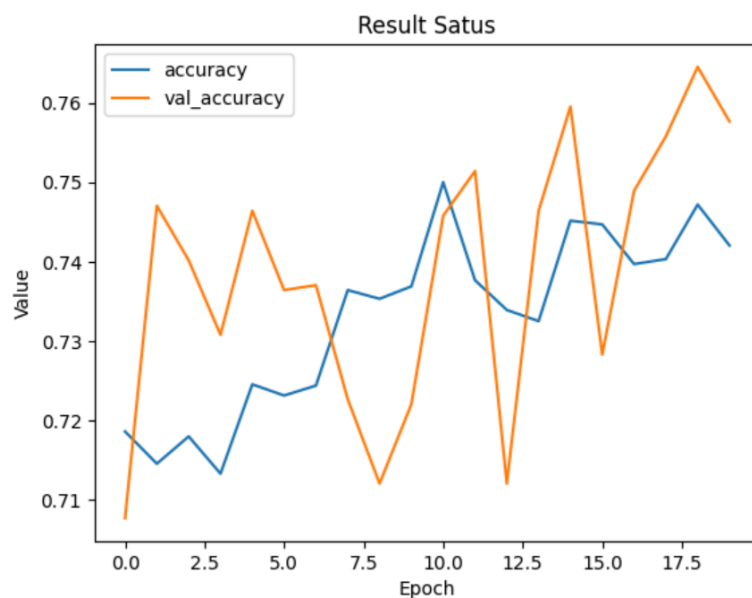
Epoch 1/20
 1281/1281 ----- 67s 52ms/step - accuracy: 0.7227 - loss: 0.5651 - val_accuracy: 0.7077 - val_loss: 0.5673
 Epoch 2/20
 1281/1281 ----- 66s 51ms/step - accuracy: 0.7137 - loss: 0.5565 - val_accuracy: 0.7470 - val_loss: 0.5321
 Epoch 3/20
 1281/1281 ----- 66s 51ms/step - accuracy: 0.7178 - loss: 0.5612 - val_accuracy: 0.7402 - val_loss: 0.5363
 Epoch 4/20
 1281/1281 ----- 65s 50ms/step - accuracy: 0.7201 - loss: 0.5462 - val_accuracy: 0.7308 - val_loss: 0.5343
 Epoch 5/20
 1281/1281 ----- 65s 51ms/step - accuracy: 0.7332 - loss: 0.5423 - val_accuracy: 0.7464 - val_loss: 0.5252
 Epoch 6/20
 1281/1281 ----- 66s 52ms/step - accuracy: 0.7178 - loss: 0.5550 - val_accuracy: 0.7364 - val_loss: 0.5355
 Epoch 7/20
 1281/1281 ----- 66s 51ms/step - accuracy: 0.7247 - loss: 0.5450 - val_accuracy: 0.7370 - val_loss: 0.5256
 Epoch 8/20
 1281/1281 ----- 65s 51ms/step - accuracy: 0.7474 - loss: 0.5324 - val_accuracy: 0.7227 - val_loss: 0.5371
 Epoch 9/20
 1281/1281 ----- 65s 51ms/step - accuracy: 0.7320 - loss: 0.5356 - val_accuracy: 0.7121 - val_loss: 0.5567
 Epoch 10/20
 1281/1281 ----- 65s 51ms/step - accuracy: 0.7272 - loss: 0.5469 - val_accuracy: 0.7220 - val_loss: 0.5415
 Epoch 11/20
 1281/1281 ----- 66s 51ms/step - accuracy: 0.7497 - loss: 0.5407 - val_accuracy: 0.7458 - val_loss: 0.5213
 Epoch 12/20
 1281/1281 ----- 66s 51ms/step - accuracy: 0.7344 - loss: 0.5305 - val_accuracy: 0.7514 - val_loss: 0.5180
 Epoch 13/20
 1281/1281 ----- 66s 51ms/step - accuracy: 0.7355 - loss: 0.5304 - val_accuracy: 0.7121 - val_loss: 0.5445
 Epoch 14/20
 1281/1281 ----- 65s 51ms/step - accuracy: 0.7376 - loss: 0.5320 - val_accuracy: 0.7464 - val_loss: 0.5205
 Epoch 15/20
 1281/1281 ----- 65s 51ms/step - accuracy: 0.7295 - loss: 0.5401 - val_accuracy: 0.7595 - val_loss: 0.5100
 Epoch 16/20
 1281/1281 ----- 65s 51ms/step - accuracy: 0.7481 - loss: 0.5255 - val_accuracy: 0.7283 - val_loss: 0.5254
 Epoch 17/20
 1281/1281 ----- 64s 50ms/step - accuracy: 0.7491 - loss: 0.5054 - val_accuracy: 0.7489 - val_loss: 0.5104
 Epoch 18/20
 1281/1281 ----- 63s 49ms/step - accuracy: 0.7418 - loss: 0.5303 - val_accuracy: 0.7558 - val_loss: 0.5052
 Epoch 19/20
 1281/1281 ----- 64s 50ms/step - accuracy: 0.7489 - loss: 0.5148 - val_accuracy: 0.7645 - val_loss: 0.4858
 Epoch 20/20
 1281/1281 ----- 83s 50ms/step - accuracy: 0.7419 - loss: 0.5177 - val_accuracy: 0.7577 - val_loss: 0.5025


20회로 설정한 결과

1회는 val_accuracy: 0.7077 - val_loss: 0.5673

20회는 val_accuracy: 0.7577 - val_loss: 0.5025

훈련 데이터에서 74.19%, 검증 데이터에서 75.77% 나온 걸 확인



• **cat.97.jpg**(image/jpeg) - 23845 bytes, last modified: 2025. 6. 2. - 100% done
Saving cat.97.jpg to cat.97 (1).jpg
1/1  0s 131ms/step
cat.97 (1).jpg: 고양이입니다.

- 훈련 정확도와 검증 정확도가 점진적으로 상승
- 에폭(epoch)이 증가할수록 검증 정확도가 좋아지는 것을 확인할 수 있음

느낀 점)

EarlyStopping을 사용했을 때 미리 예상했던 것보다 정확도가 적게 나온다는 점을 보고
왜 그런 결과가 나오게 된 것인지 더 탐구해봐야겠다는 생각이 들었으며
AI 학습 실습을 직접 경험해볼 수 있어서 학습에 도움이 되었습니다.