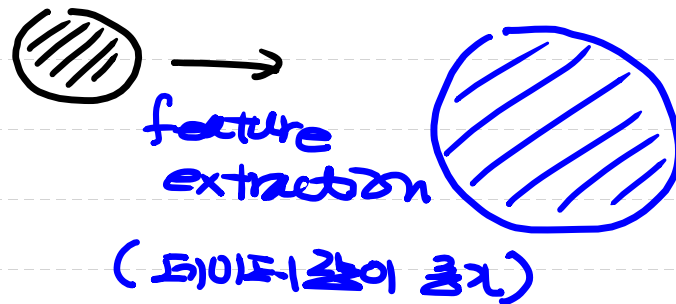


04/21

① MNIST 데이터를 이용한 DNN (Deep Neural Network) 

② 복잡한 (Size가 큰) image를 학습하려면 어떻게 해야 하나요?

→ image의 특성을 추출하기 그 특성을 학습 ⇒ CNN
feature extraction



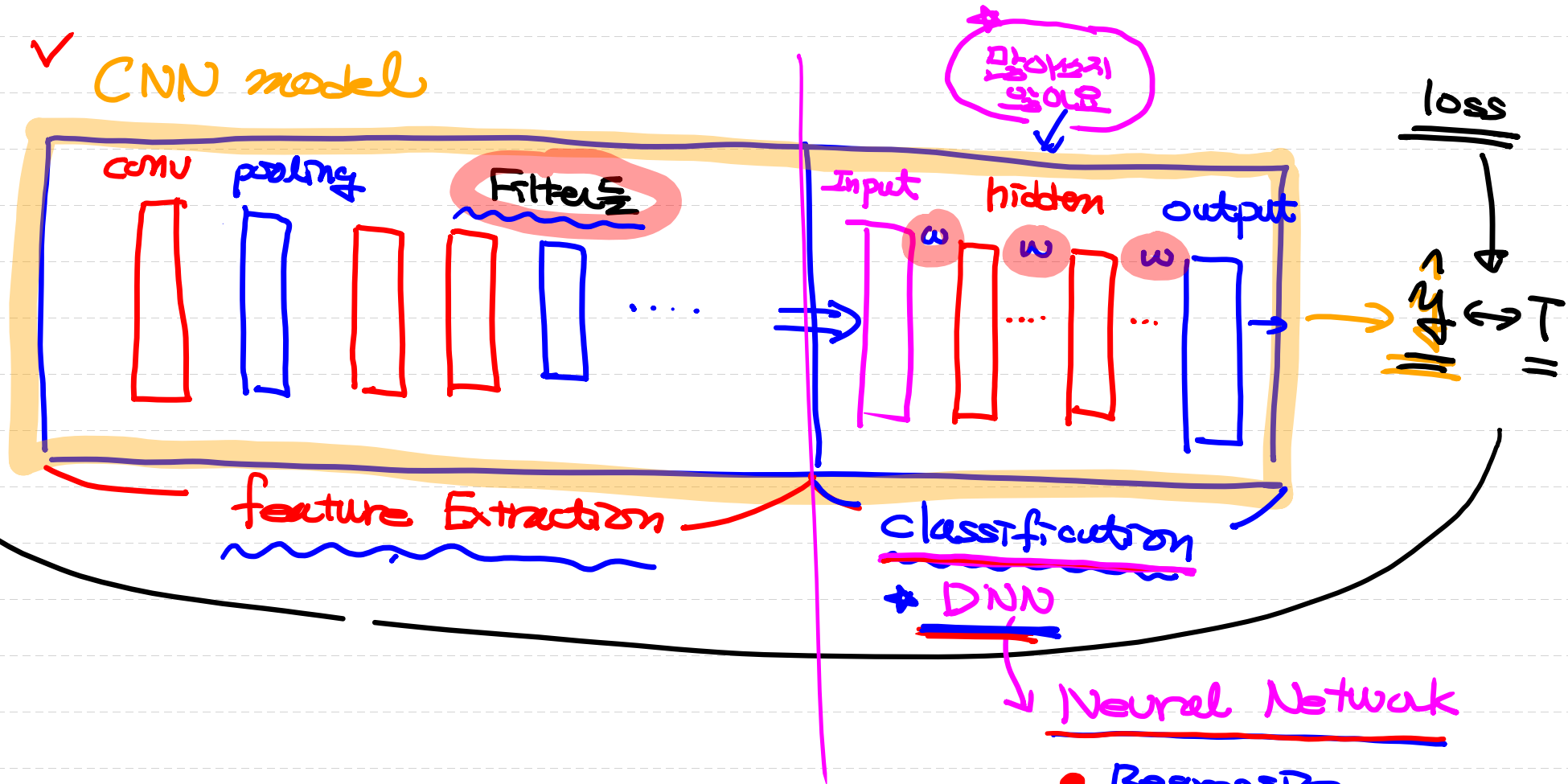
③ 충분한 양의 데이터가 있는 경우 CNN 학습을 하면 → Accuracy 95%
(25,000 장)

④ 작은 양의 데이터를 이용해서 학습 (CNN 학습) → Overfitting
(4,000 장)

⑤ 학습이미지의 개수를 증가시키는 효과를 내기 위해 → Image Augmentation (증가)

• 층수를 작으면 overfitting이 줄어든 모델 성능이 좋아져요!!

✓ CNN model



• Regression

* SUM, Decision Tree
KNN, Naive Bayes
인공물 기법

⑥

어떻게 하면

✓

시스템도 잘 작동하게

더 좋은 성능을 낼 수 있는가??

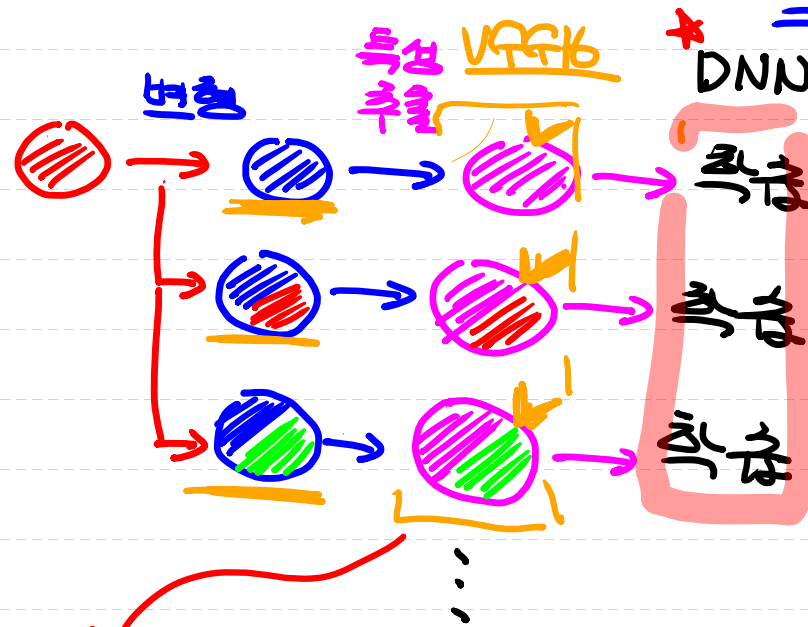
Pretrained Network

Feature Extraction 결과
+ 위의 classifier

★ ⑦ 작은양의 데이터를 "VGG16"을 이용하여 feature extraction을 진행.
 DNN을 수행
 → Overfitting 발생.

↓ ndarray
 → 학습

★ ⑧ Image Augmentation 을 이용하여 작은양의 데이터를 증가



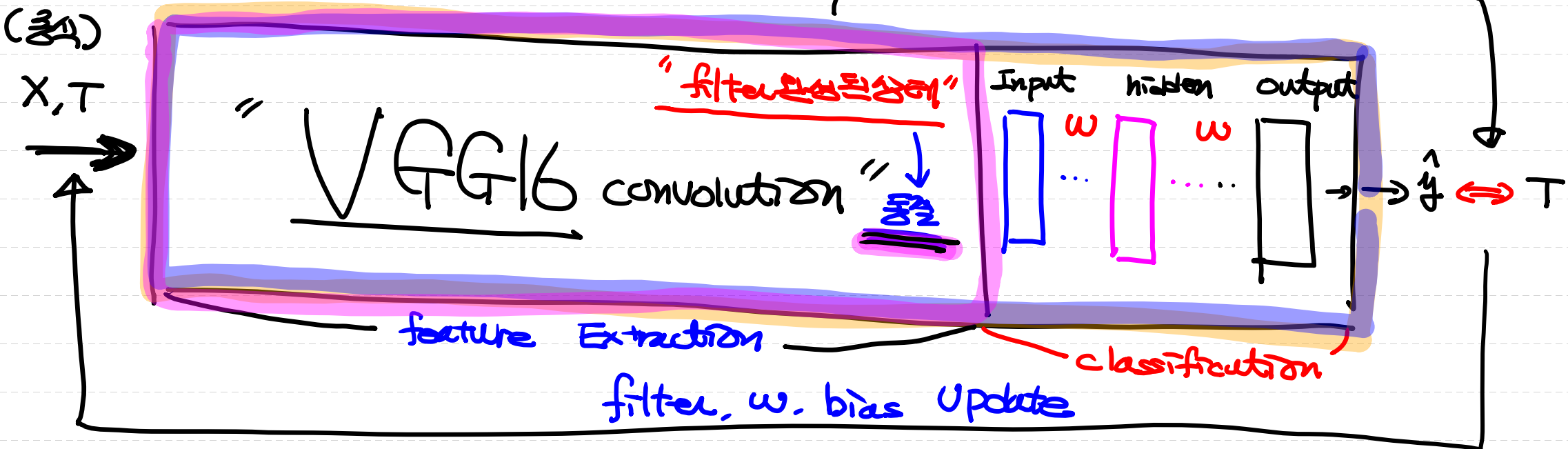
Overfitting
 이 감소 !!

★ filter값이 변함
 epoch 만큼 수행.

→ 변경되면 어도모 ⇒ "동작"

★ CNN

Sequential



VGG16 convolution "필터 완성된 상태"

"filter 완성된 상태"

至今

Input hidden output

a

w

loss

$$y \Leftrightarrow T$$

feature Extraction

• classification

filter, w. bias Update

↓ 결과

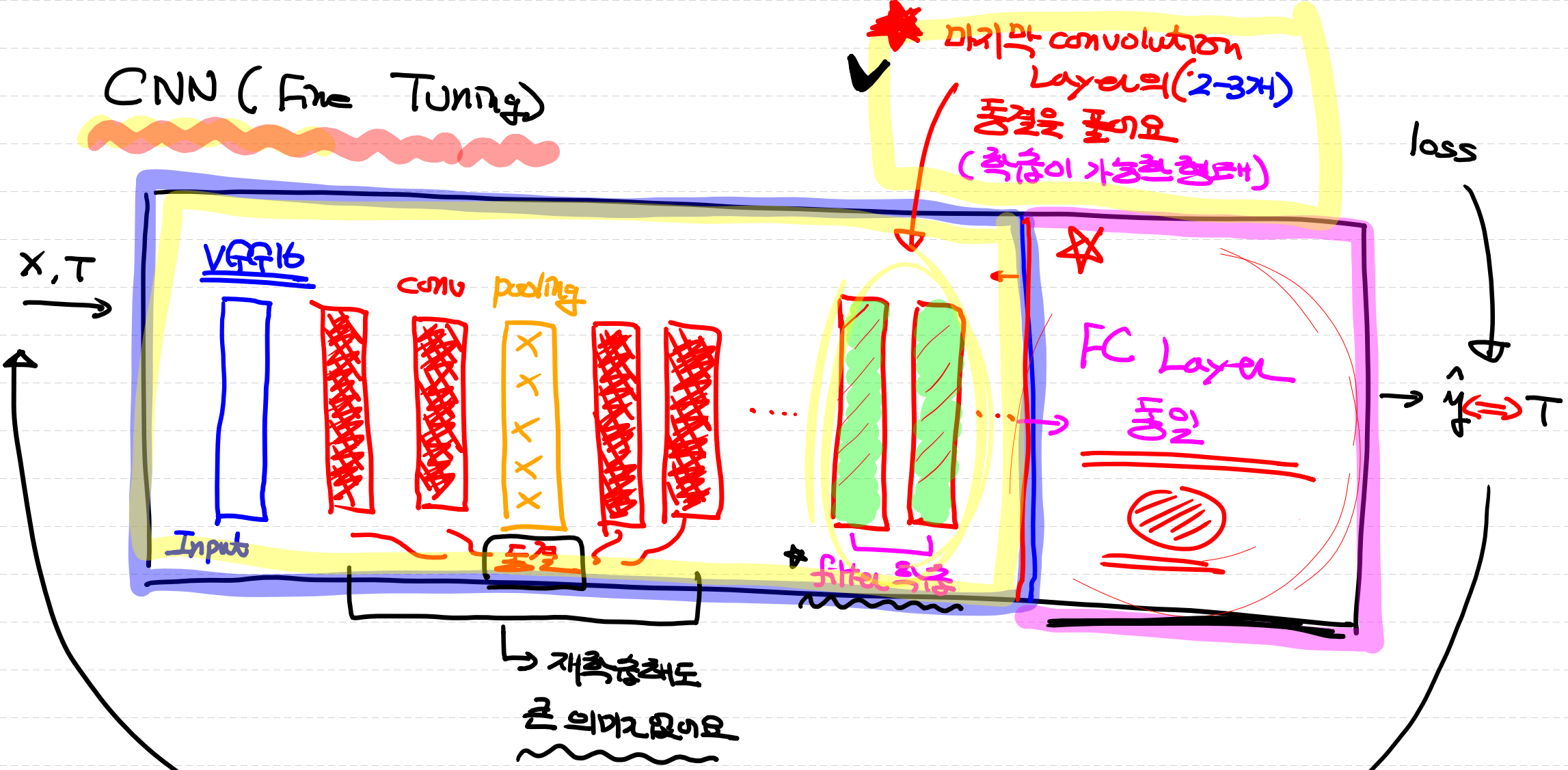
→ 약 90%.

Overfitting은 꼭이되지 않을까 기대?

⑨ 여기서 더 권익도를 높이려면 어떻게 해야 할까요??

⇒ Fine Tuning (마세조형)

CNN (Fine Tuning)



fine Tuning을 하려면

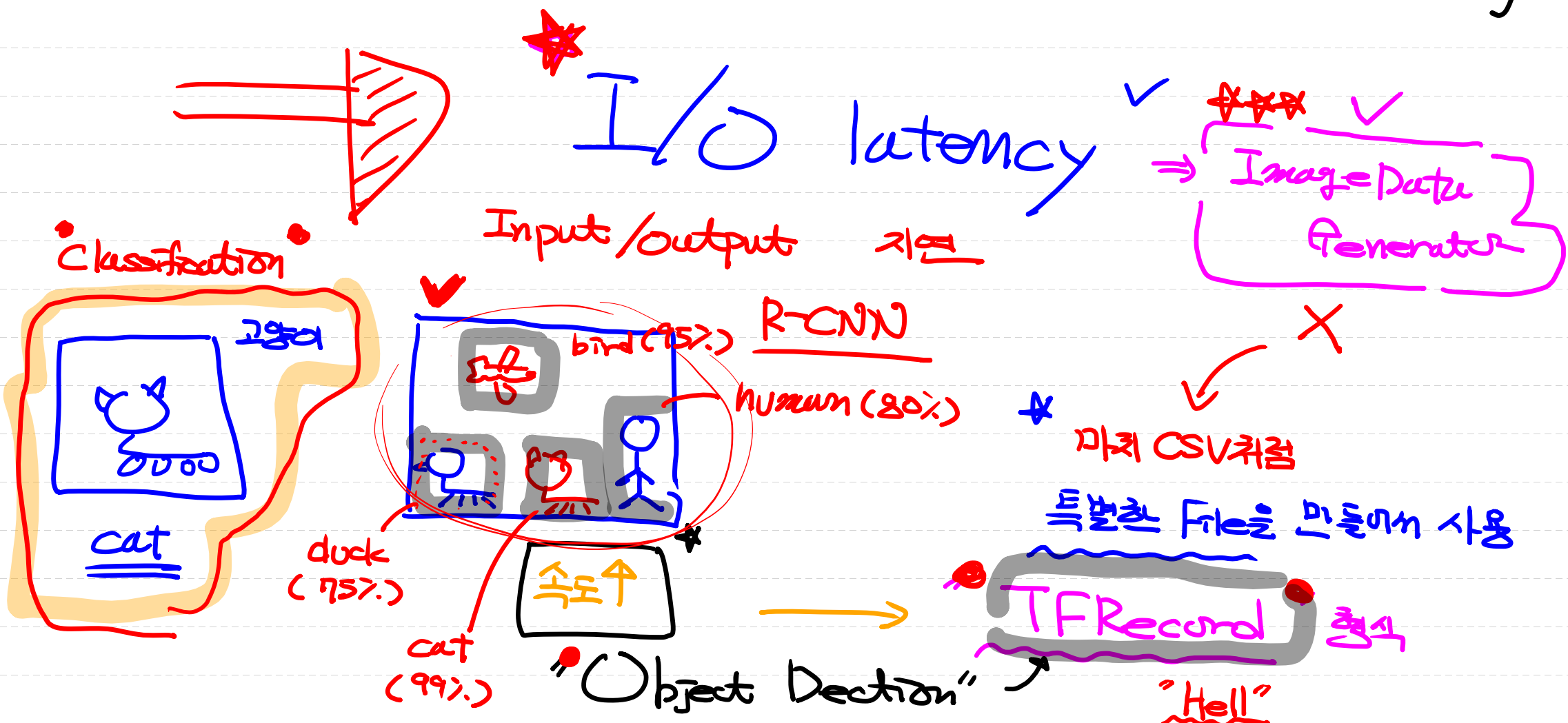


Classifier가 학습이 되어 있어야 해요.



code 3
→ Colab에서
실행해 보세요!!

여기까지 사용 중 performance 이 높을수록 영향을
가치는 요소가
어떻게 있을까요???



⑨ 92% (cat-dog-small) ^{*}
fine tuning

* 모델의 성능을 높이려면 ?? *

↳ ("VGG16") 대신에 ^{*} EfficientNet 이용 ↑

↳ 새로운 기법이 추가로 필요!!

↓
구현해보아요 (colab)