
ONECLICK

이미지 분류

Image classification

기계학습







Training / 학습



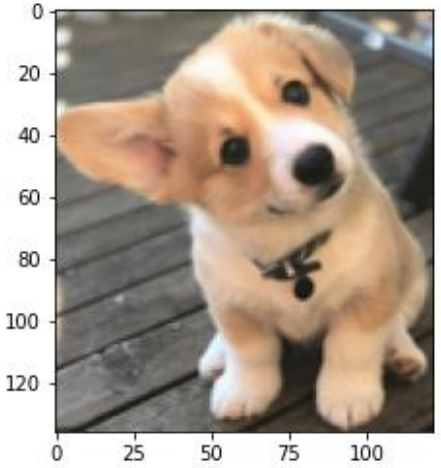

Model

강아지

인공지능

Image Classification

이미지의 뜻



이미지 데이터는 몇 차원일까요?

A: 2차원

B: 3차원

C: 4차원

Image Classification

이미지의 뜻

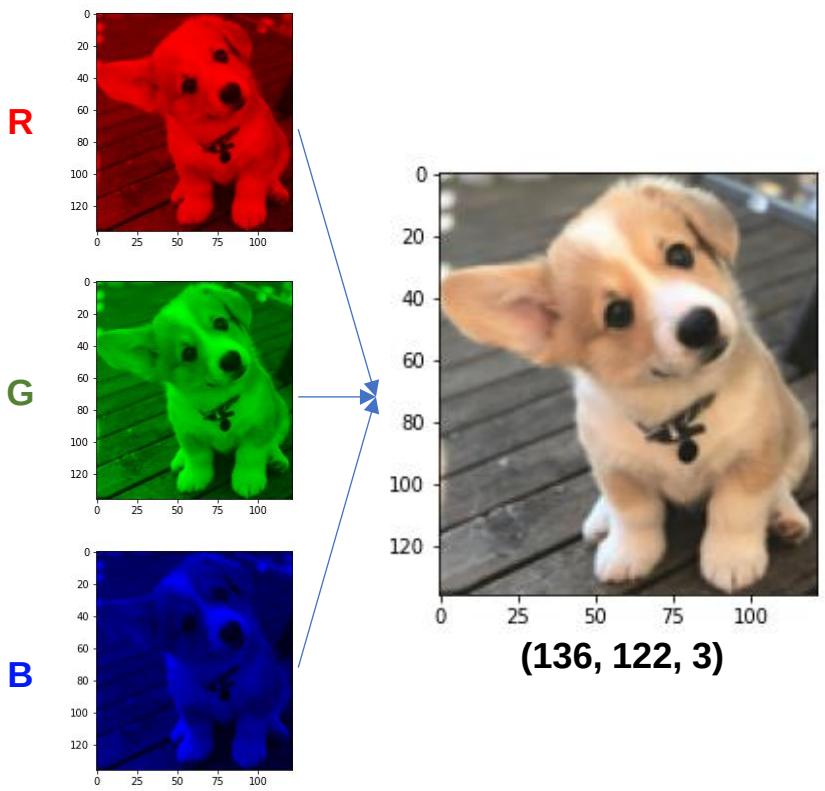


Image Classification

이미지의 뜻

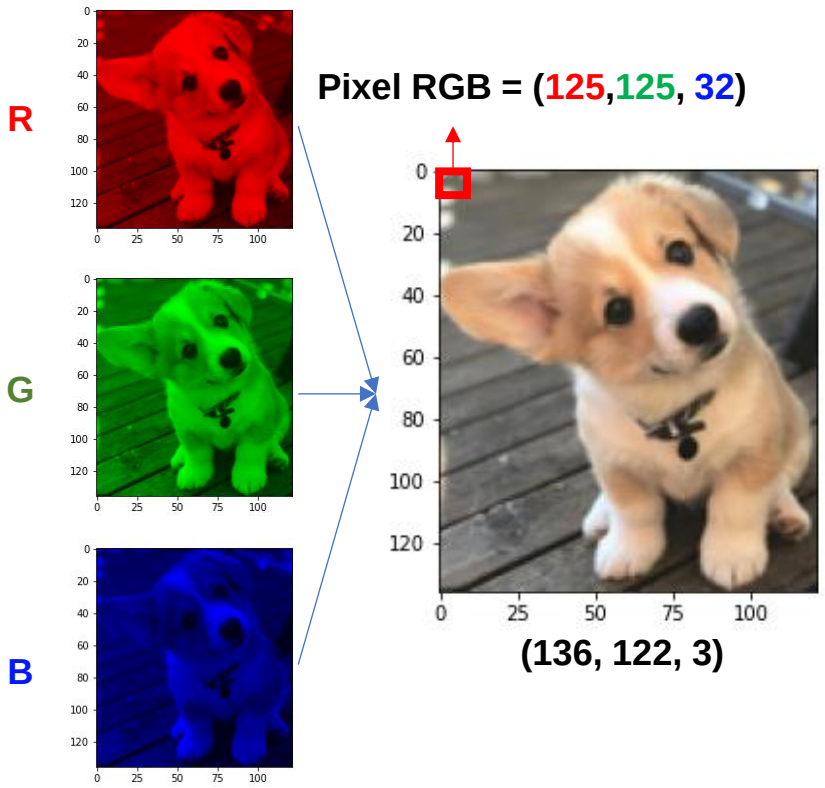


Image Classification

이미지의 뜻

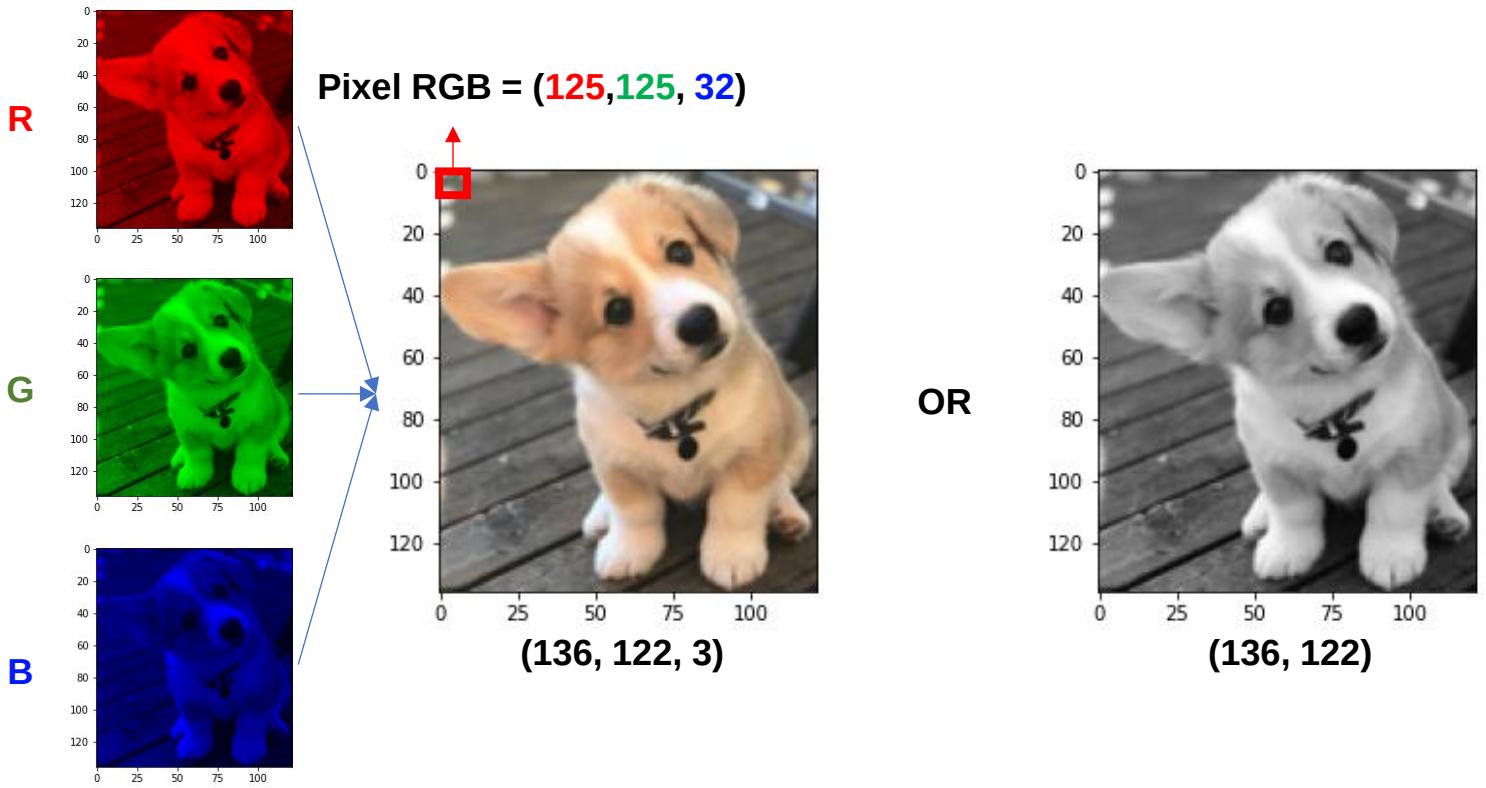
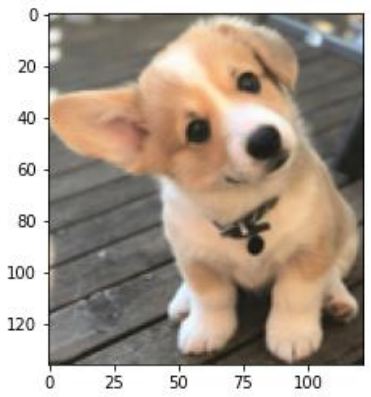


Image Classification

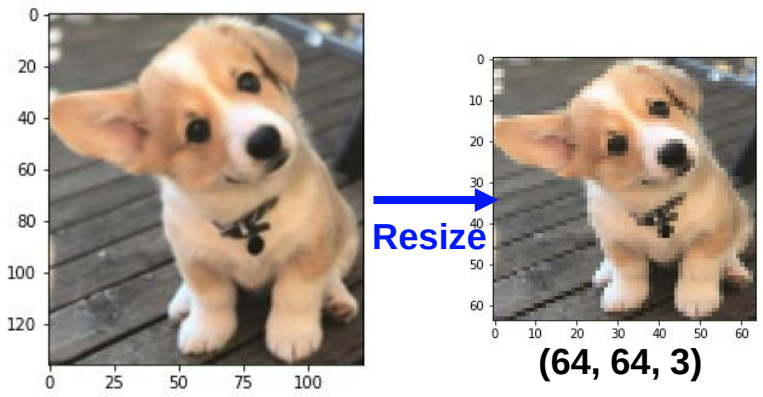
가장 간단한 이미지 분류 모델!



(136, 122, 3)

Image Classification

가장 간단한 이미지 분류 모델!



(136, 122, 3)

(64, 64, 3)

Image Classification

가장 간단한 이미지 분류 모델!

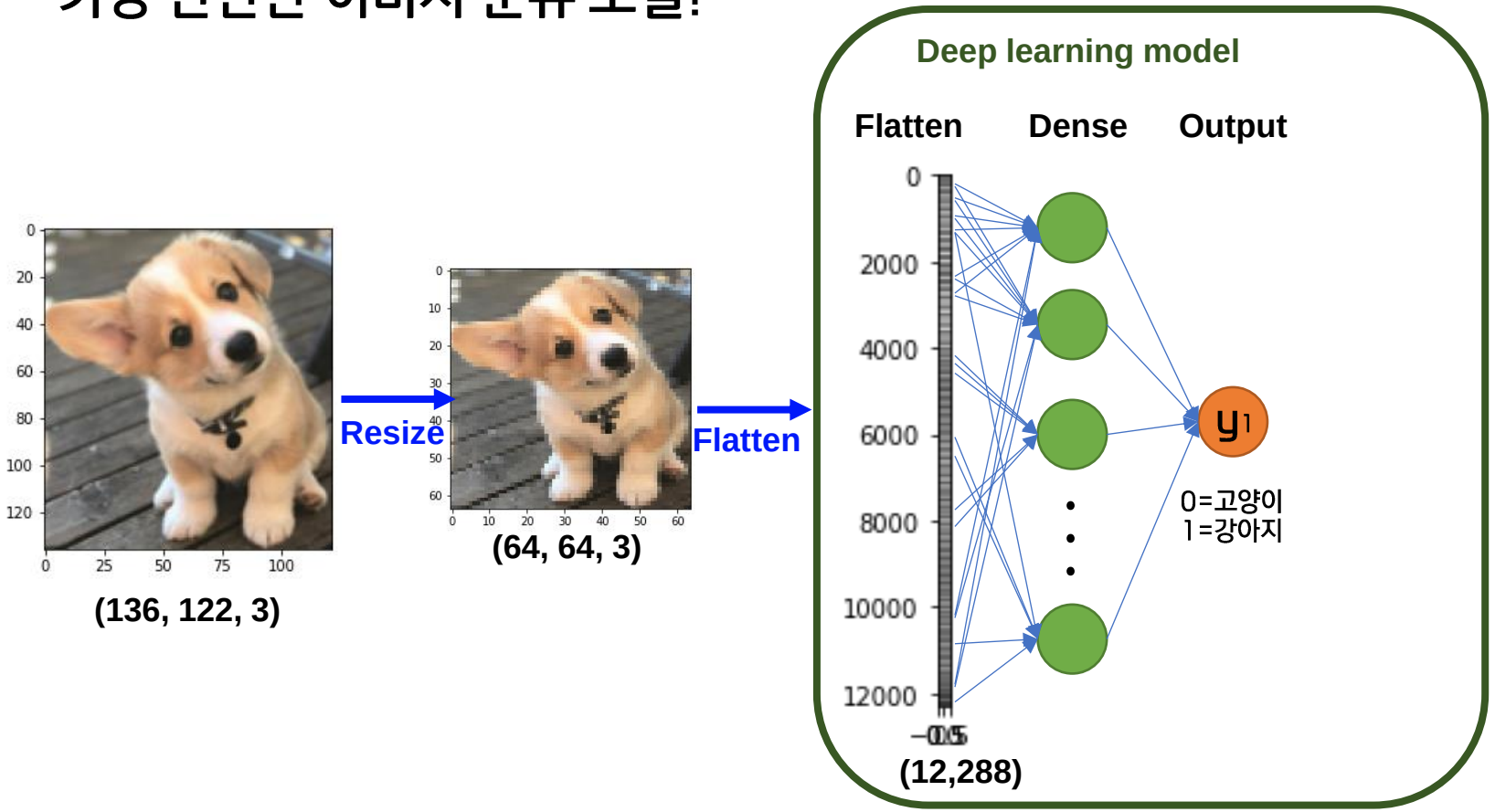
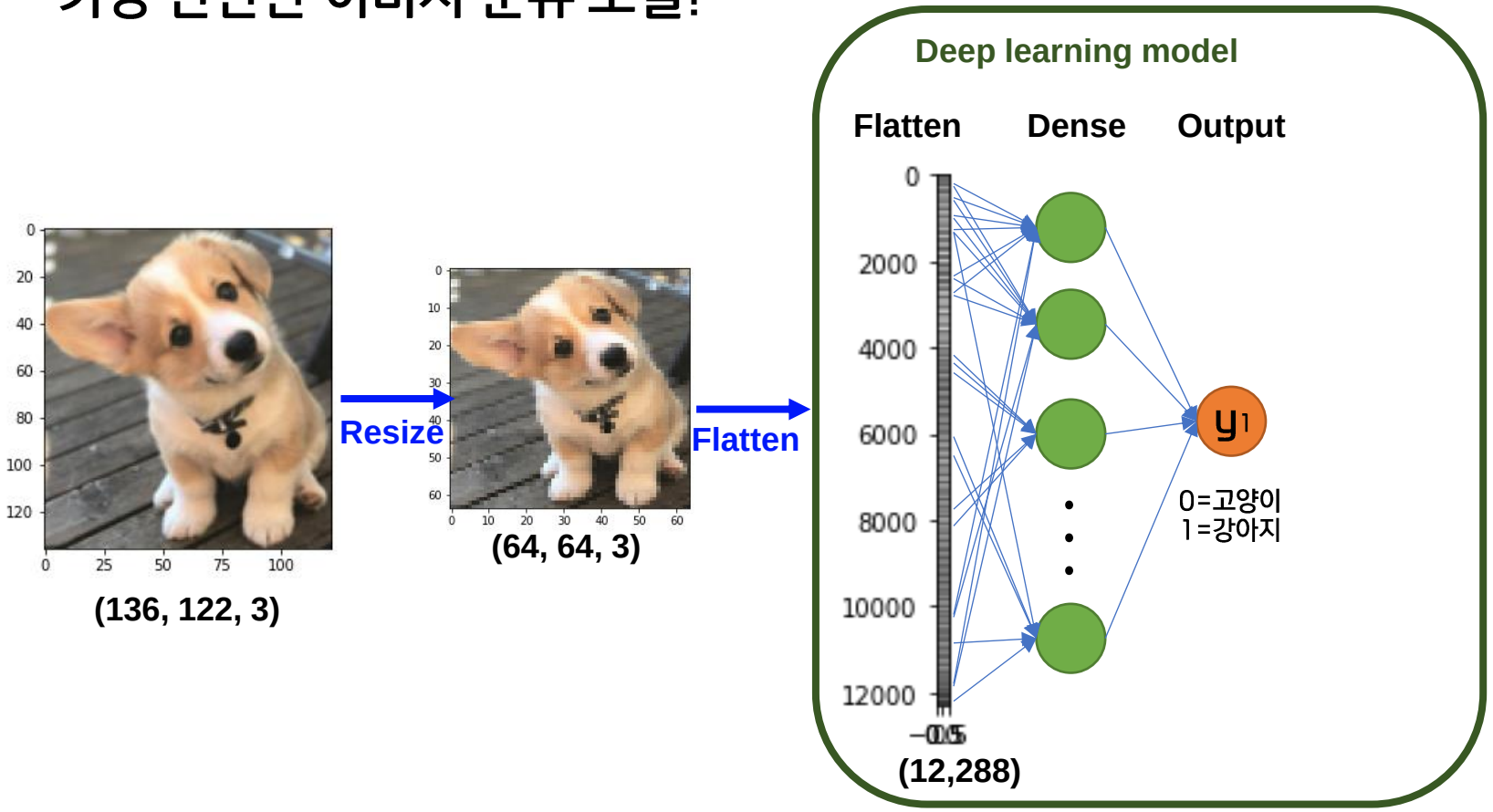


Image Classification

가장 간단한 이미지 분류 모델!



Problem: 입력 값이 너무 많아요!

Image Classification

가장 간단한 이미지 분류 모델!
혹은 Grayscale 사용

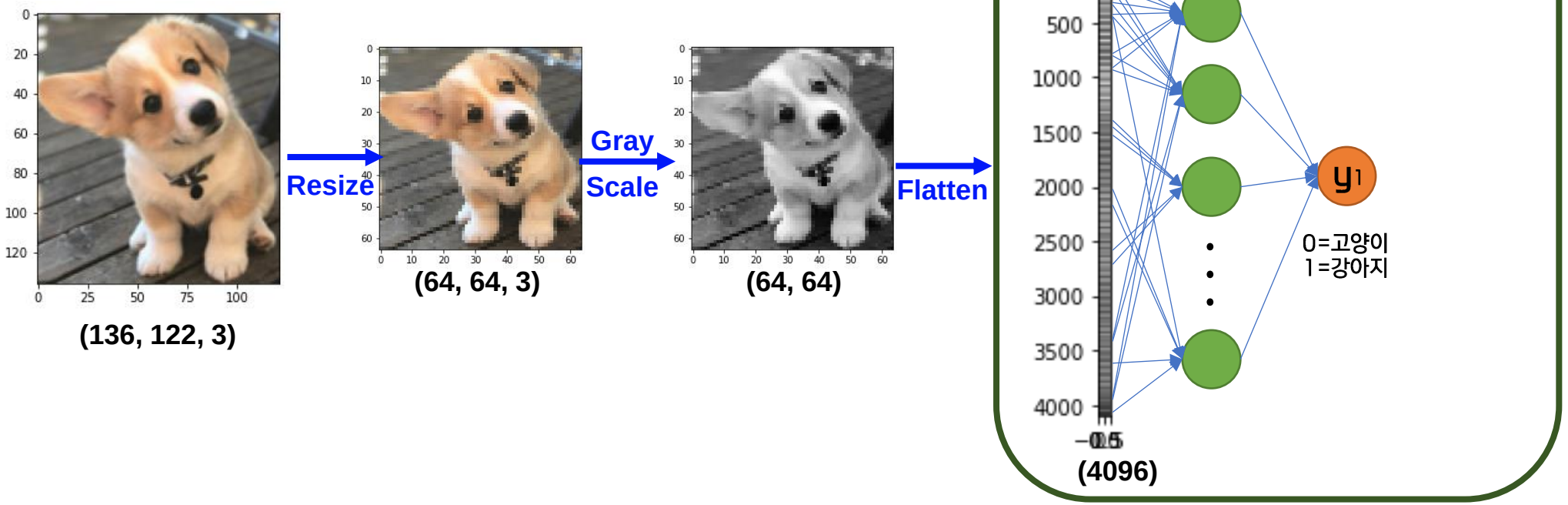


Image Classification

조금 더 복잡한 모델!

Image Classification

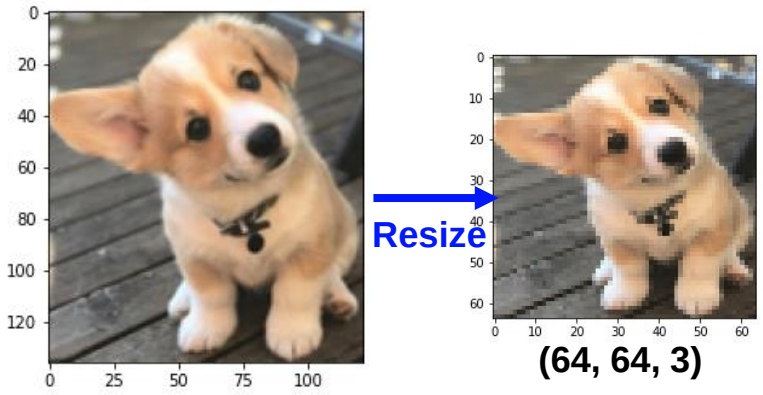
조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류

Image Classification

조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류
…그러니까 Convolutional? 무슨뜻이죠?

Image Classification

조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류
…그러니까 Convolutional? 무슨뜻이죠?

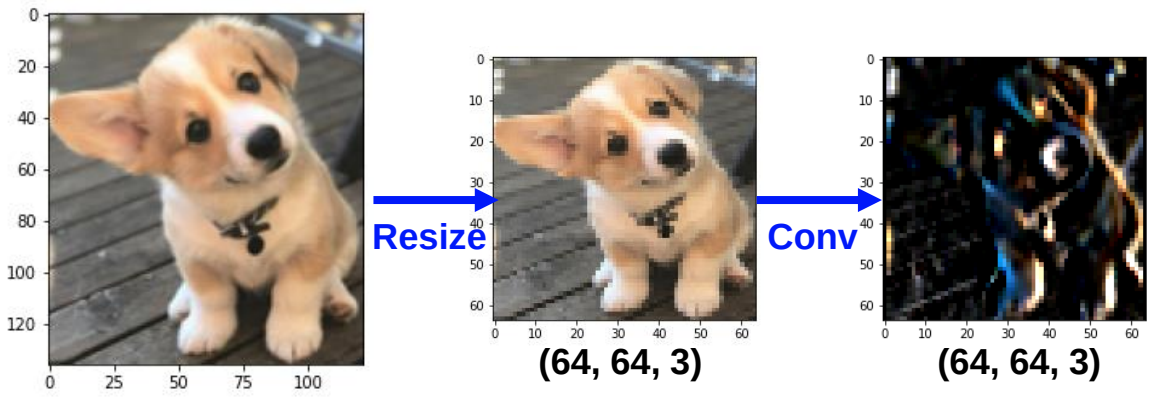


1	0	-1

필터

Image Classification

조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류
…그러니까 Convolutional? 무슨뜻이죠?

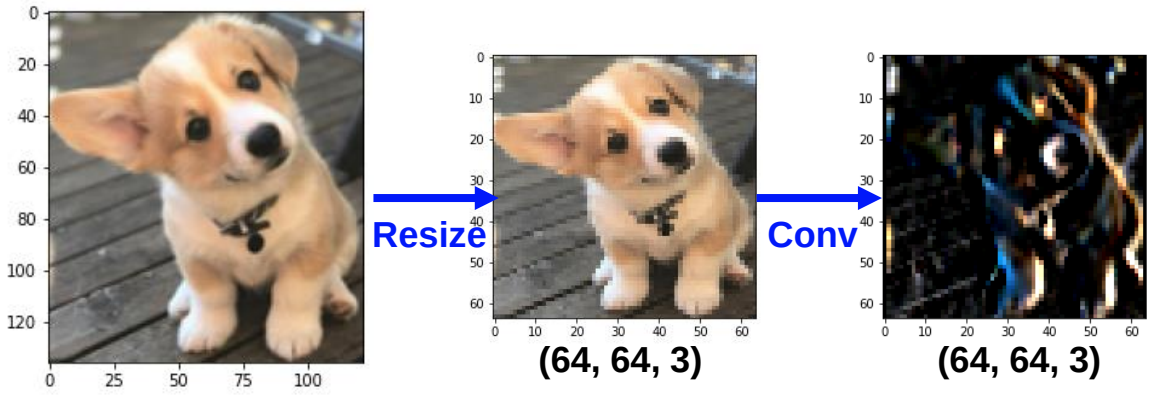


1	0	-1

필터

Image Classification

조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류
…그러니까 Convolutional? 무슨뜻이죠?



(136, 122, 3)

(64, 64, 3)

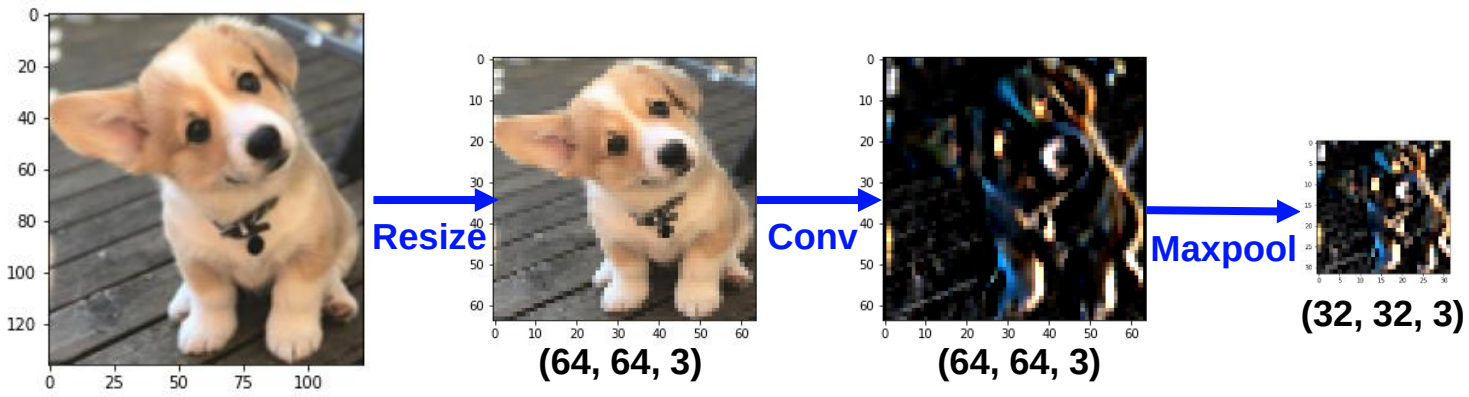
(64, 64, 3)

1	0	-1
1	0	-1
1	0	-1

필터

Image Classification

조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류
…그러니까 Convolutional? 무슨뜻이죠?



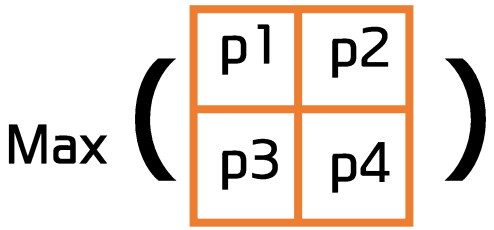
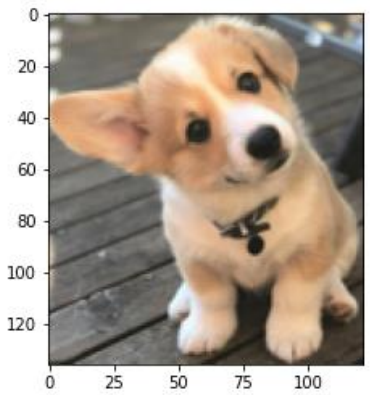


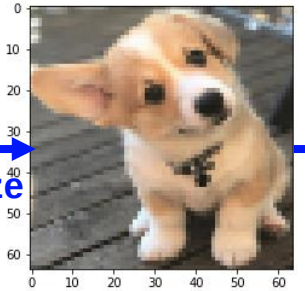
Image Classification

조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류
…그러니까 Convolutional? 무슨뜻이죠?



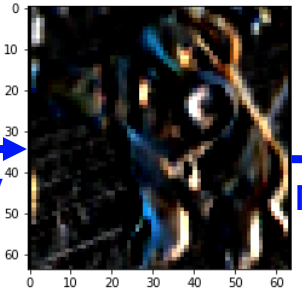
(136, 122, 3)

Resize



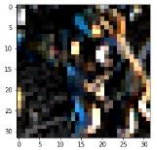
(64, 64, 3)

Conv



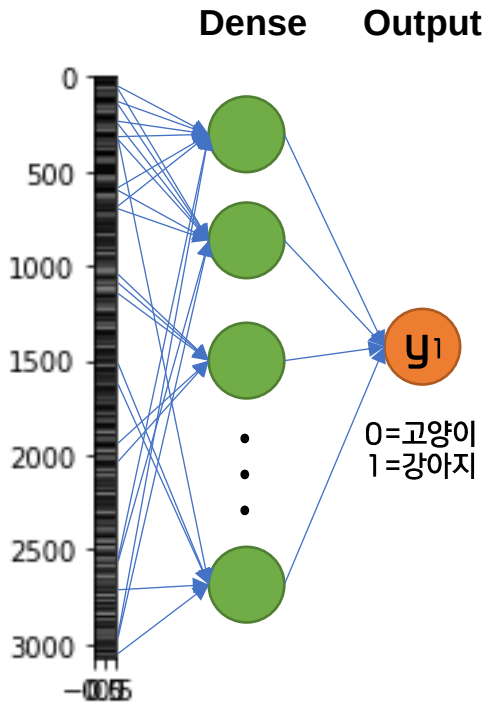
(64, 64, 3)

Maxpool



(32, 32, 3)

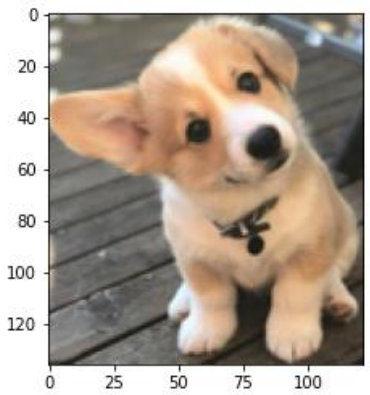
Flatten



(3072)

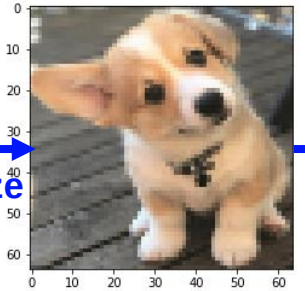
Image Classification

조금 더 복잡한 모델!
Convolutional Layer를 사용한 이미지 분류
…그러니까 Convolutional? 무슨뜻이죠?



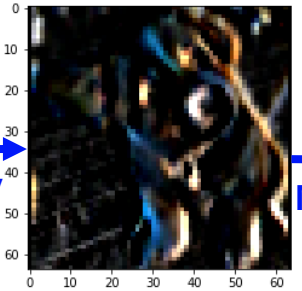
(136, 122, 3)

Resize



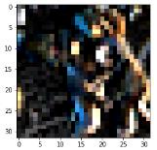
(64, 64, 3)

Conv



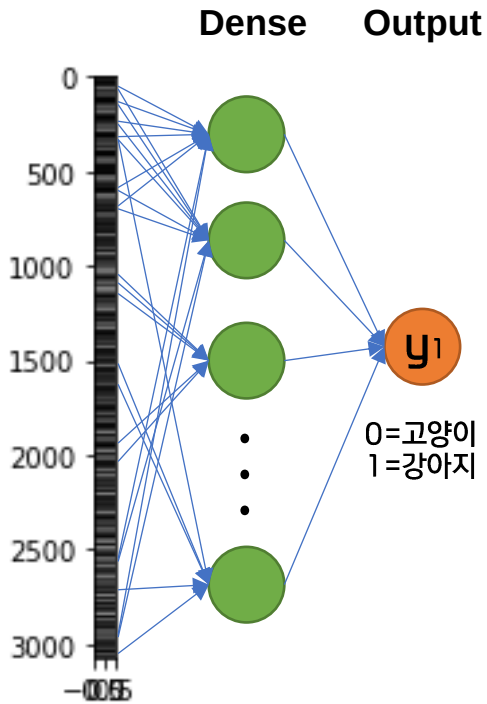
(64, 64, 3)

Maxpool



(32, 32, 3)

Flatten



(3072)

코드 작성해보기

강아지 고양이 분류해보기

코랩을 활용하여 이미지 학습 실행
코드는 지난번과 동일하게 아래 알고리즘으로 동작합니다.

1. 데이터 불러오기
2. 모델 만들기
3. 모델 설정하기 (i.e. 학습속도)
4. 모델 학습하기
5. 결과 확인하기

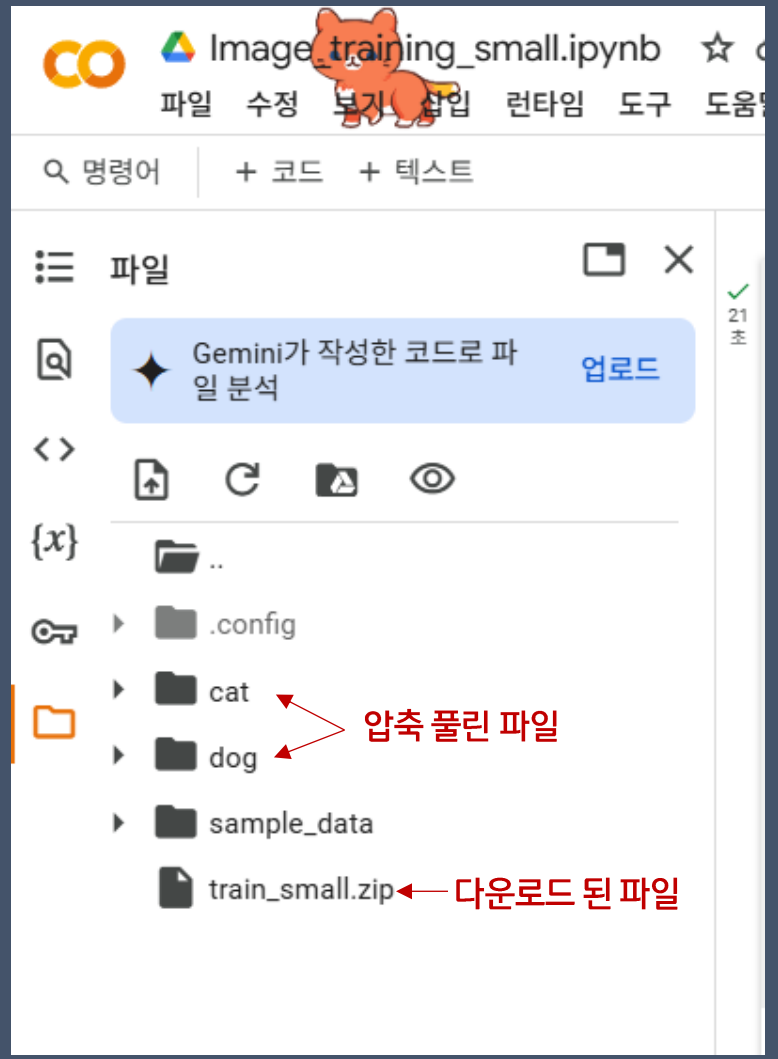


코랩 코드
https://colab.research.google.com/drive/10ILc901n66W4FYgQIdbtxg-ONTIZLimD?usp=sharing

데이터 파일(받을 필요 없음)
https://drive.google.com/file/d/1ckRJPia7Zo7Z7hGeVEvD1tbLHowJCkY2/view?usp=sharing

데이터 다운받기

```python
# 파일 다운로드 및 압축풀기 (코랩 전용 코드)
from google.colab.patches import cv2_imshow
import gdown
file_id = "1ckRJPia7Zo7Z7hGeVEvD1tbLHowJCkY2"
output = "train_small.zip"
gdown.download(id=file_id, output=output, quiet=False)
import zipfile
with zipfile.ZipFile('train_small.zip', 'r') as zip_ref:
    zip_ref.extractall()
```



실행 후 결과

패키지 불러오기

```
# 라이브러리 임포트
import tensorflow as tf
from tensorflow.keras.layers import Dense,Flatten,Conv2D,MaxPooling2D
from tensorflow.keras import Input
import matplotlib.pyplot as plt
import numpy as np
import cv2
import os
```

각각 폴더에서 첫번째 이미지 그려보기

```
#%% 데이터 불러오기
imgs_cat = os.listdir('.//cat')
imgs_dog = os.listdir('.//dog')

# 강아지 이미지 한 번 그려보기
img = cv2.imread('.//cat//' + imgs_cat[0])
cv2_imshow(img)

# 고양이 이미지 한 번 그려보기
img = cv2.imread('.//cat//' + imgs_cat[0])
cv2_imshow(img)
```

이미지 불러오고 라벨 붙이기

```
# 데이터 전처리
x = []
y = []

for i in range(1000):
    # load cat images
    img = cv2.imread('.//cat//' + imgs_cat[i])
    img = cv2.resize(img, (96, 96)) # 이미지 사이즈 변경 (96,96)
    x.append(img/127.5-1) # 이미지 정규화
    y.append(0) # 고양이 라벨 = 0

    # load dog images
    img = cv2.imread('.//dog//' + imgs_dog[i])
    img = cv2.resize(img, (96, 96))  # 이미지 사이즈 변경 (96,96)
    x.append(img/127.5-1) # 이미지 정규화
    y.append(1) # 강아지 라벨 = 1
```

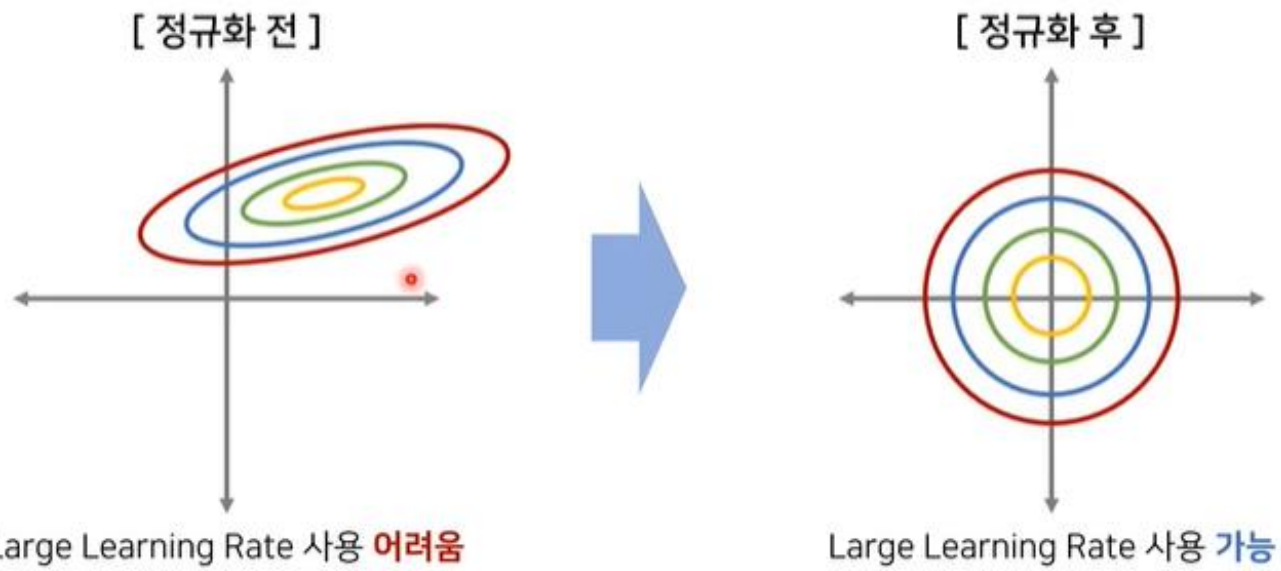
정규화란?

AI 모델은 입력과 출력 모두 -1~1 사이 값일 때, 가장 학습이 잘됨.
따라서 데이터를 대략적으로 -1과 1사이로 바꿔주는 작업을 정규화라고 함.

예를들어 픽셀데이터는 0~255 사이 값을 가짐 때문에 이를 -1~1 값으로 변경

픽셀 값을 127.5로 나누고, 1을 빼줌.

이런 정규화 방식을 Scaling이라고 함
또 다른 정규화 방식으로 z-score를 사용하거나,
때에 따라 정규화를 사용하지 않는 경우도 있음



학습 데이터와 검증용 데이터로 나누기

```
# 리스트-> Array로 변경
x = np.stack(x)
y = np.array(y)

# 데이터를 training set과 validation set으로 나누기
x_train = x[:1500,:,:,:]
y_train = y[:1500]
x_val = x[1500:,:,:]
y_val = y[1500:]

# onehot으로 변경
# 강아지의 경우 (0,1), 고양이의 경우 (1,0)
y_train = tf.one_hot(y_train, 2)
y_val = tf.one_hot(y_val, 2)
```

CNN 모델 생성

```
# 모델 생성
model = tf.keras.Sequential()
model.add(Conv2D(16, (3, 3), input_shape=(96, 96, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(16, (3, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(64, activation = 'relu'))
model.add(Dense(2, activation = 'softmax'))
```

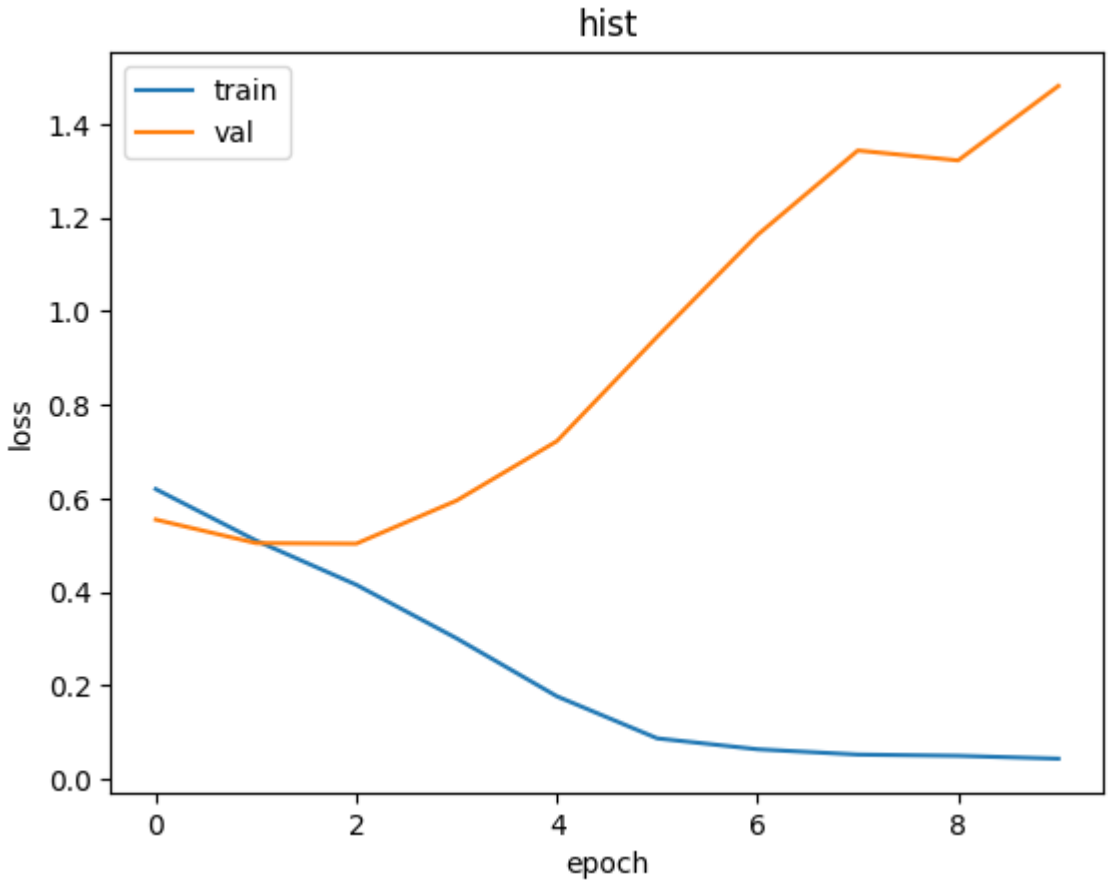
모델 설정 및 학습

```
#%%
# 모델 설정
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='categorical_crossentropy',
metrics=['Acc'])

# 모델 학습
history=model.fit(x_train,  y_train, validation_data=(x_val, y_val),
batch_size = 16, epochs = 10)
```

결과 확인

```
#%%
# 모델 결과 확인
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.title('hist')
plt.show()

#%% 테스트
print(model.predict(np.array([x_train[501]])))
cv2_imshow((x_train[501]+1)*127.5)
```

몇번째 epoch 모델이 제일 좋은 모델일까요?



강아지 고양이 분류해보기

코랩 코드 (2000장, 정확도: 64%)
https://colab.research.google.com/drive/10ILc901n66W4FYgQIdbtxg-ONTIZLimD?usp=sharing
데이터 2,000장:
https://drive.google.com/file/d/1ckRJPia7Zo7Z7hGeVEvD1tbLHowJCkY2/view?usp=sharing

코랩 코드 (16,000장, 정확도:75%)
https://colab.research.google.com/drive/1QSt53v3IUMsSGWxgdhUhDF4RzYnjxBv4?usp=sharing
데이터 25,000장:
https://drive.google.com/file/d/1w5GEU6H6OKz3eHF5rerix0L4IReMzXp5/view?usp=sharing

데이터 크기와 정확도의 관계

