Image Classification

Using CNN

김 영 준

CONTENTS

01

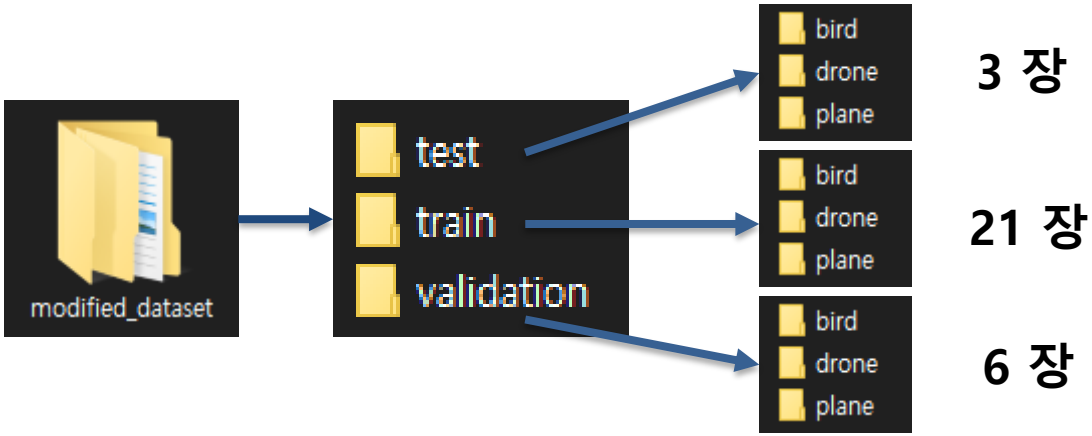
Bird

Drone

Plane

CNN을 사용한
Classification Algorithm을 개발.

- 개발환경 : Google Colab
- Dataset : Total 90 images.
  - Directory Structure



1. 이미지 수집

- 구글링을 통한 이미지 크롤링

2. 이미지 리사이징

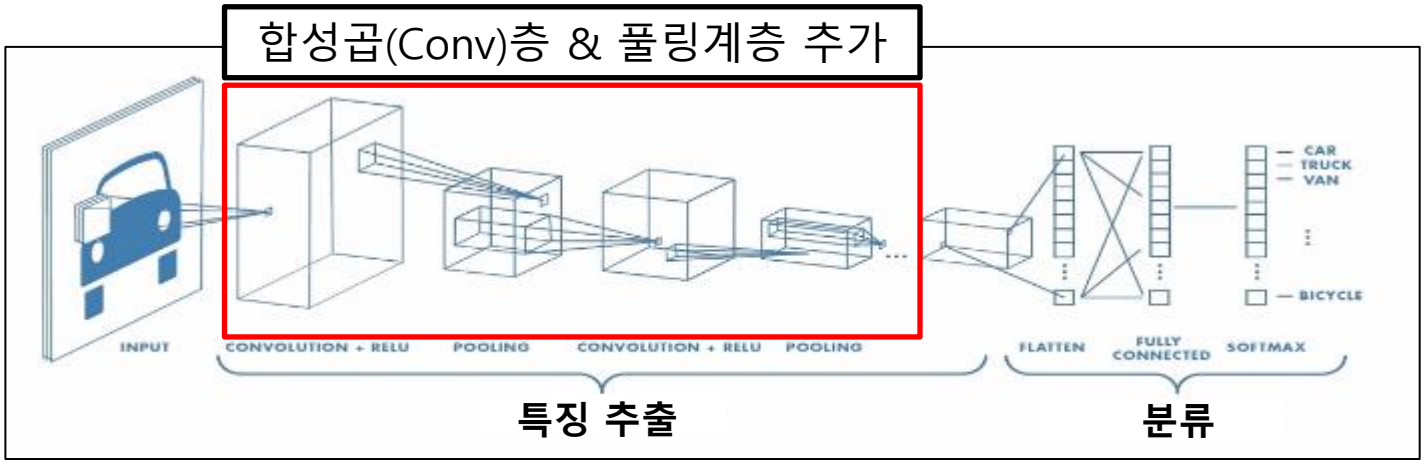
https://github.com/rladudwnss/CNN_Classification-flying-object/blob/main/resize_image.py

3. Colab에서 신경망 구축

https://github.com/rladudwnss/CNN_Classification-flying-object/blob/main/classification_usingCNN.ipynb

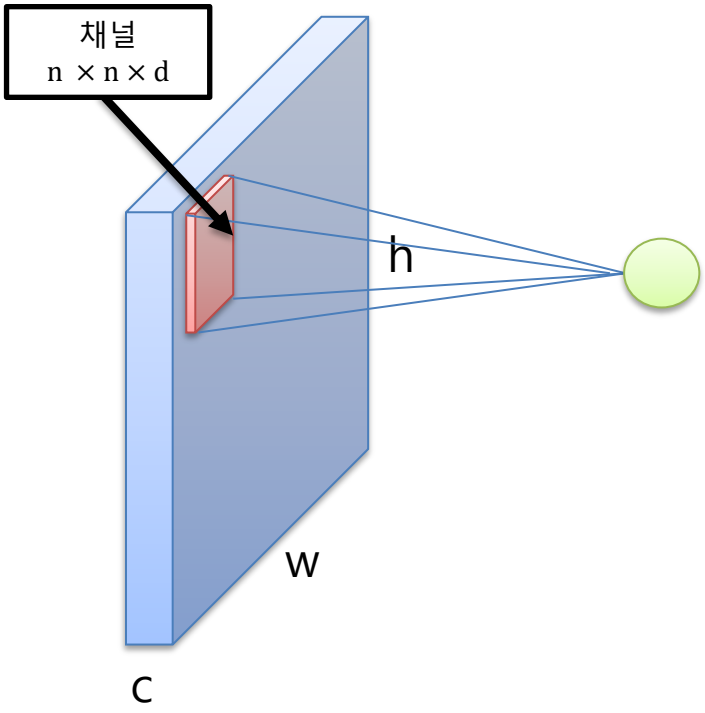
What is CNN?

Convolution Neural Network

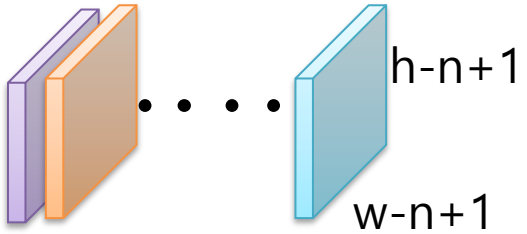


- 동물의 시신경 구조를 모방한 기술
- 특징맵을 생성하기 때문에 "비전" 분야에서 성능이 우수
- CNN은 3차원 데이터 (W, H, Depth)를 입력 받기 때문에 데이터를 제대로 이해할 가능성이 높다.

Convolution Layer : channel을 통한 activation maps 생성



Input Image

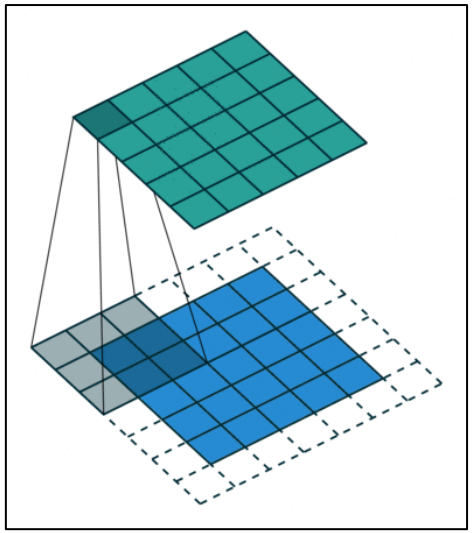


fitler의 수

Filter의 사이즈는 거의 홀수

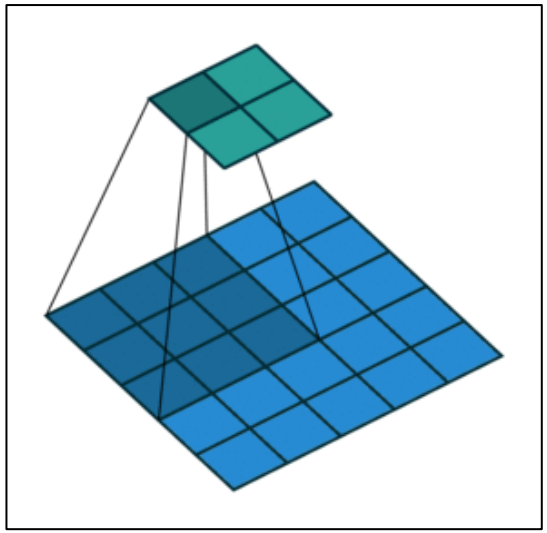
- 짝수면 비대칭이 됨
- 왼쪽, 오른쪽을 다르게 줘야 함
- 하나의 픽셀(중심 픽셀)이 존재해야 함

Convolution Layer : Padding & Stride



- stride : 1
- Input data 사이드를 특정 값으로 채우는 기법

-> padding을 통해 사이즈를 천천히 줄여 특징 데이터의 손실을 줄여준다!

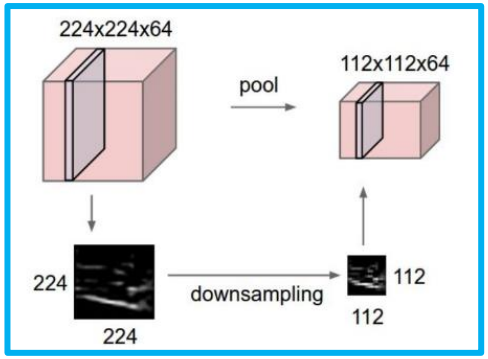


- stride : 2
- 필터와 다음 필터 사이의 간격
- Output Size

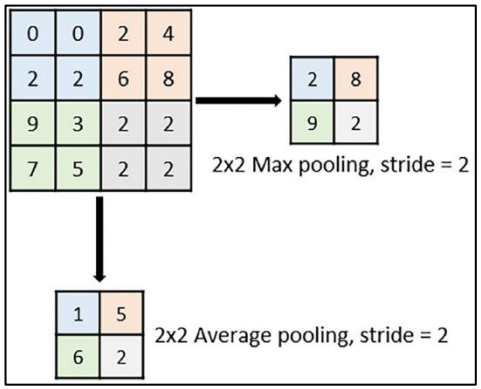
$$= \frac{Input(w,h) - Filter(w,h)}{stride} + 1$$

Convolution Layer : Pooling

Pooling 사용 이유 : padding으로 사이즈를 보존하지만 결국 줄여야 함.
즉 down sampling의 역할을 수행해 줌



Pooling layer에서 down sampling을 진행
사이즈 감소 + depth 유지
처리방법
1) Max Pooling(제일 많이 쓰임)
2) Average Pooling
3) Min Pooling



따라서

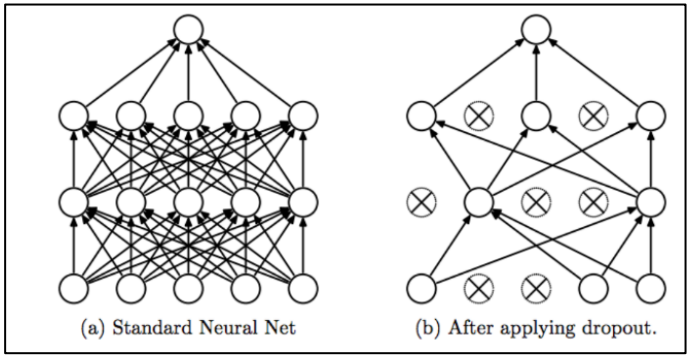
Output data Height : $\frac{H+2P-FH}{S}+1$

Output data Width : $\frac{H+2P-FW}{S}+1$

Fully Connected Layer : Flatten Layer & Softmax Layer

```
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(1, activation='softmax'))
```

- **Flatten**()을 통해 학습을 위해 2차원 데이터를 1차원 데이터로 바꾸어 준다.
- 활성화 함수는 **relu**, 마지막층(classification)을 위해 **softmax**를 사용
- **Dropout**을 통해 과적합을 막아준다.(학습 과정에서 무작위로 뉴런의 집합을 제거

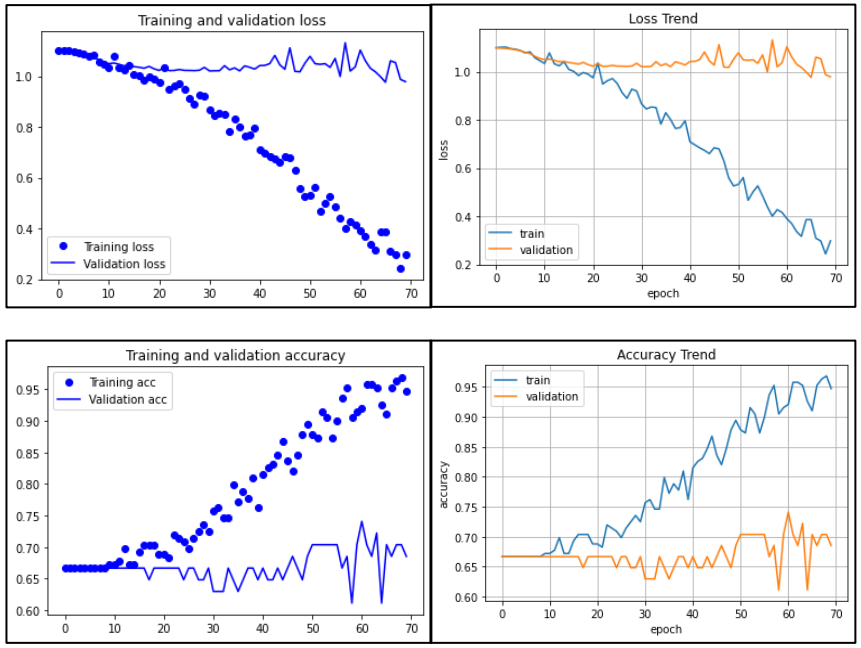


Colab 에서 데이터 확인 & 신경망 구축

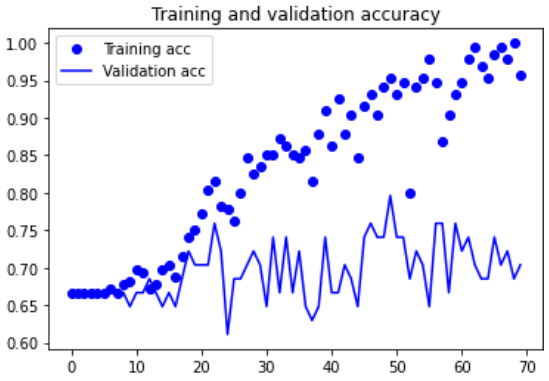
```
훈련용 새 이미지 전체 개수: 21
훈련용 드론 이미지 전체 개수: 21
훈련용 비행기 이미지 전체 개수: 21
검증용 새 이미지 전체 개수:  6
검증용 드론 이미지 전체 개수:  6
검증용 비행기 이미지 전체 개수:  6
테스트용 새 이미지 전체 개수:  3
테스트용 드론 이미지 전체 개수:  3
테스트용 비행기 이미지 전체 개수:  3
```

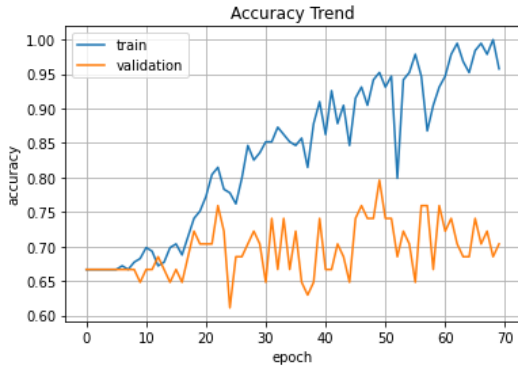
(256, 256, 3)의 데이터를 갖는다.

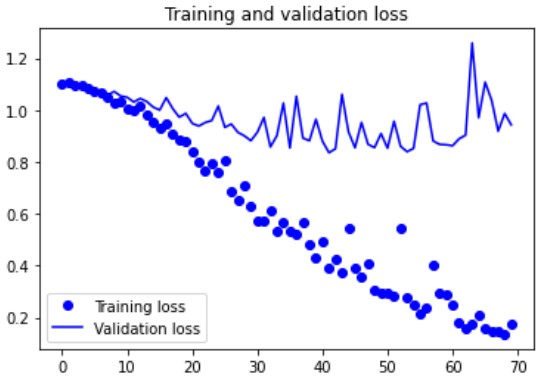
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 32)      896

 max_pooling2d (MaxPooling2D  (None, 127, 127, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 125, 125, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 62, 62, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 30, 30, 128)      0
 2D)

 flatten (Flatten)           (None, 115200)            0

 dense (Dense)               (None, 256)               29491456

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 29,584,961
Trainable params: 29,584,961
Non-trainable params: 0
_________________________________________________________________
```

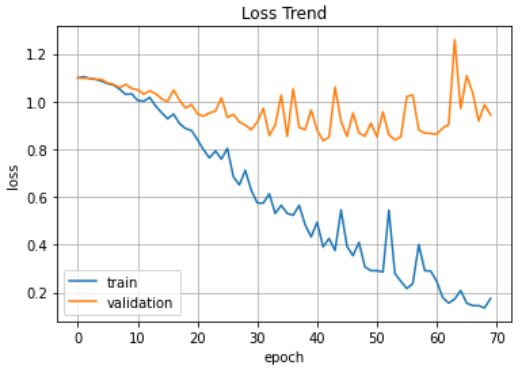


```
from keras import layers
from keras import models

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
                        input_shape=(256, 256, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='softmax'))

model.summary()
```





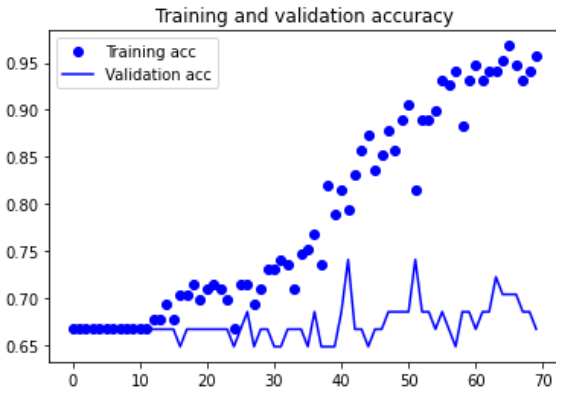


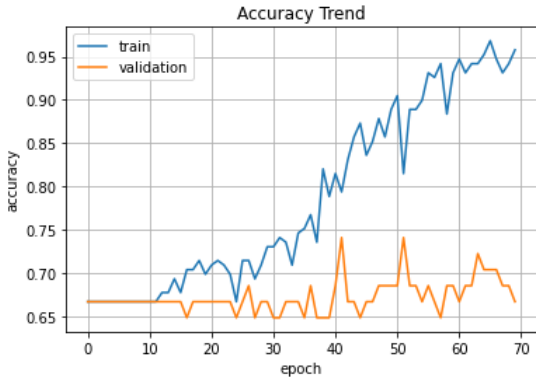


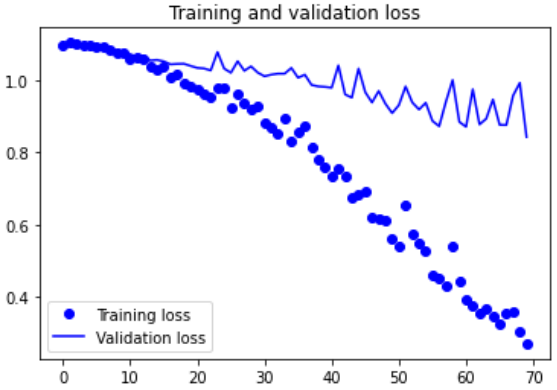

```
1/1 [==============================] - 1s 1s/step - loss: 0.9430 - acc: 0.7037
Test accuracy:  0.7037037014961243
```

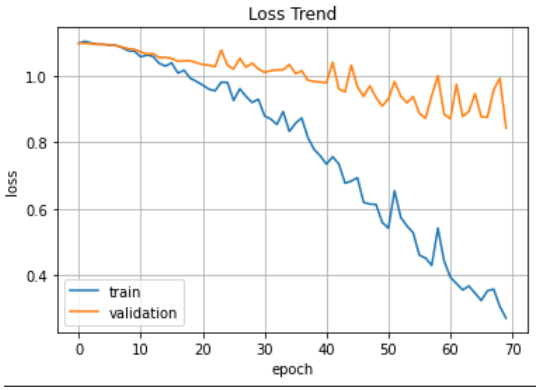
02

```
from keras import layers
from keras import models

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
                        input_shape=(256, 256, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
#model.add(layers.Dropout(0.5))
model.add(layers.Dense(1, activation='softmax'))

model.summary()
```










```
1/1 [==============================] - 1s 679ms/step - loss: 0.8432 - acc: 0.6667
Test accuracy:  0.6666666865348816
```

class / test	Bird	Drone	Plane
1	o	o	x
2	o	o	x
3	o	x	x

```
1/1 [==============================] - 1s 1s/step - loss: 0.9787 - acc: 0.6852
Test accuracy:  0.6851851940155029
```

Plane class에 대해서 굉장히 낮은 정확도를 보여준다.

03

- 실제 획득한 데이터에서는 Preprocessing 부분을 더 신경쓴다.
- 양질의 데이터셋을 수집
- CNN의 신경망 구축에 대한 추가적인 공부 필요
- 그래프, 정확도 등 분석하고 이해하는 공부 필요

THANK YOU

Youngjun's git
https://github.com/rladudwnss?tab=repositories
참고링크
https://bigdaheta.tistory.com/48
https://seongkyun.github.io/study/2019/11/21/cnn_problem/
https://foxtrotin.tistory.com/473