CNN을 활용하여 표정 인식하기

For Those who have Trouble Recognizing Facial Expressions

CNN팀 - 20고진영, 22길태호, 22양지윤

Table of contents

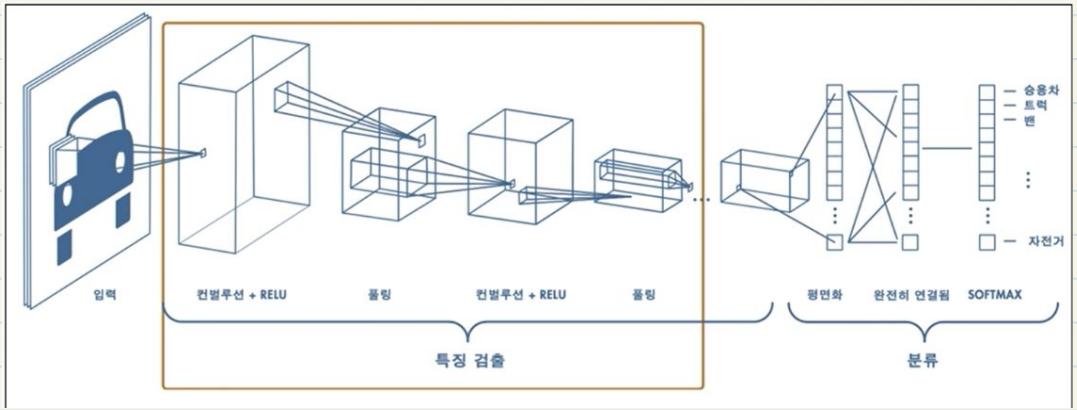
01

Concepts of CNN

CNN 기초 개념 설명

CNN(Convolution Neural Network)은 무엇인가?

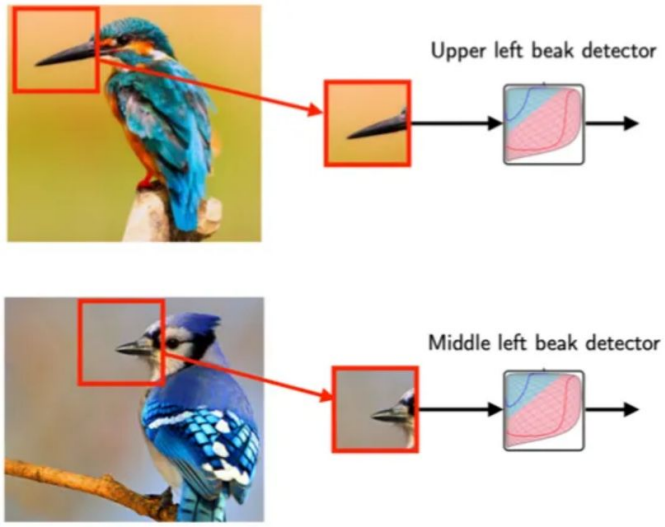
합성곱 신경망이란 입력된 이미지의 특징을 추출하기 위해 도입된 기법.



<CNN 관련 용어 정리>

- 합성곱
- 커널
- Conv2D()
- MaxPooling()
- Flatten()

CNN은 무엇인가?



CNN의 중요 Point!

- 새의 주요 특징은 '부리'
- 전체 이미지를 보는 것 보다 새의 부리 부분을 잘라 보는게 더 효율적!
- 이미지 전체보다는 부분을!
- 이미지의 한 픽셀과 주변 픽셀들의 연관성을!

Projects related to CNN

02

Project Introduction

CNN 프로젝트 소개

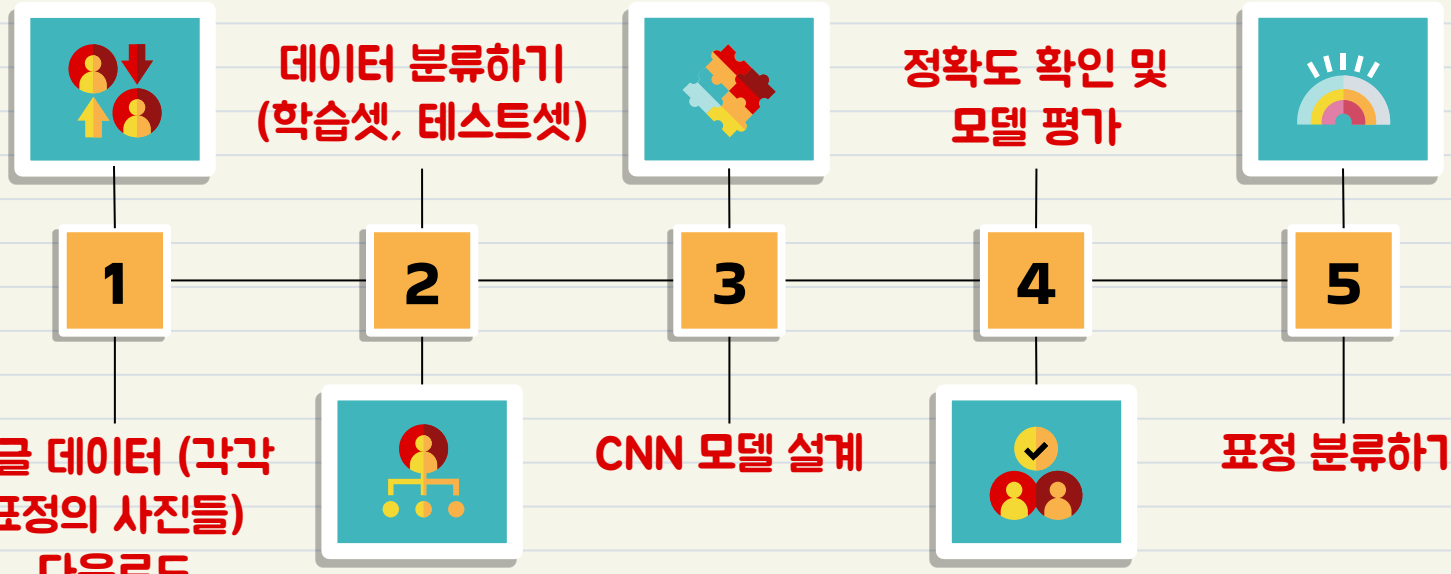
Project Motivation

〈이상한 변호사 우영우〉

*자폐스펙트럼장애를 가진 우영우는 사람의 감정을 읽는 것을 어려워하여 사진을 보며 표정에 따른 감정을 학습함

*감정을 인식하는 데 어려움을 갖는 사람들을 위해 CNN을 활용한 감정 분류 모델을 만들어봄

Project Outline

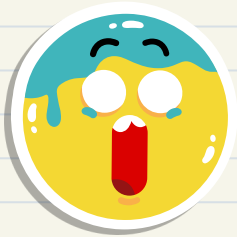


7 Emotions in Our Project

Happy

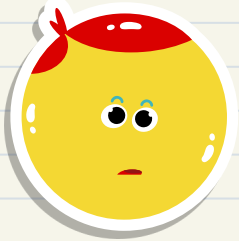
Sad

Angry



Fear

Surprise



Neutral

Disgust

03

Project Codes

프로젝트 코드 설명 및 개념 보충

Project Codes

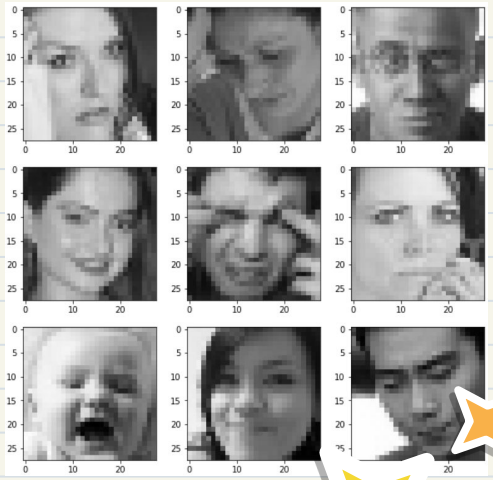
〈캐글에서 사진 데이터 가져오기〉

#캐글 파일 다운로드

```
import os
os.environ['KAGGLE_USERNAME'] = "jiyunishere"
os.environ['KAGGLE_KEY'] =
"7e690b22a3d4e2b8059cc2e961b180f3"
!kaggle datasets download -d msambare/fer2013
```

#ZIP파일 압축 풀기

```
from zipfile import ZipFile
file_name = "/content/fer2013.zip"
with ZipFile(file_name, 'r') as zip:
  zip.extractall()
  print('done')
```



Project Codes

〈필요한 라이브러리 불러오기〉

```
import os, re, glob
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
from keras.models import Sequential
from keras.layers import Dropout, Activation, Dense
from keras.layers import Flatten, Convolution2D,
MaxPooling2D
from keras.models import load_model
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
import numpy as np
from tensorflow import keras
from tensorflow.keras.preprocessing.image import load_img,
img_to_array
from tensorflow.keras.applications.resnet50 import
preprocess_input
import requests
from PIL import Image
from io import BytesIO
from IPython.display import Image
from urllib.request import urlopen
from PIL import Image
```

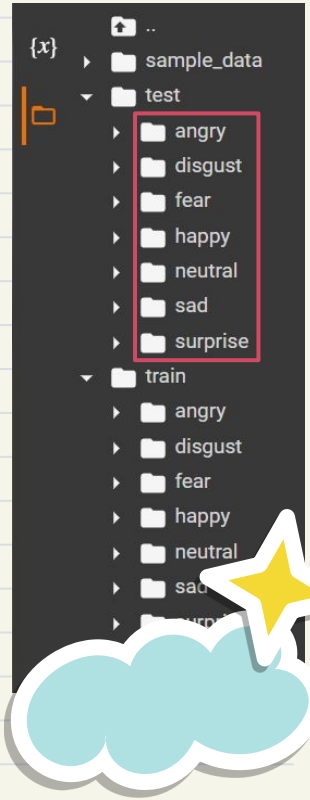
Project Codes

〈train, test, validation set 분류하기〉

```
#폴더에서 train데이터 불러오기 및 카테고리 분류
groups_folder_path = '/content/train/'
categories = ["angry", "disgust", "fear", "happy", "neutral", "sad", "surprise"]

image_w = 28
image_h = 28

X = []
Y = []
```



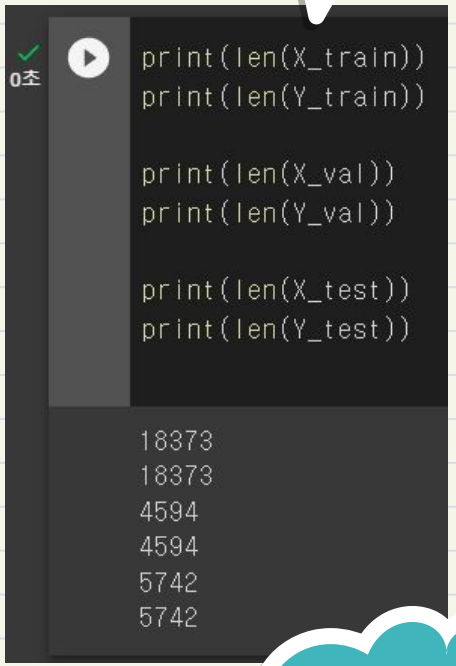
Project Codes

```
#train, test, validation set 분류하기
for idex, categorie in enumerate(categories):
    label = [0 for i in range(num_classes)]
    label[idex] = 1
    image_dir = groups_folder_path + categorie + '/'

    for top, dir, f in os.walk(image_dir):
      for filename in f:
          img = cv2.imread(image_dir+filename)
          img = cv2.resize(img, None, fx=image_w/img.shape[1],
fy=image_h/img.shape[0])
          X.append(img/256)
          Y.append(label)

X = np.array(X)
Y = np.array(Y)

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2)
X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, test_size=0.2)
```



Project Codes

〈CNN 모델 설계하기〉

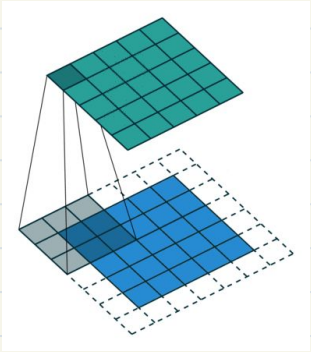
```
# CNN 모델 설계
model = keras.Sequential([
    layers.Conv2D(32, kernel_size=3, padding='valid',
activation='relu', input_shape=(28, 28, 3)),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Dropout(0.2),
    layers.Conv2D(64, kernel_size=3, activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Dropout(0.2),
 layers.Conv2D(128, kernel_size=3, activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.2),
    layers.Dense(7, activation='softmax')])
```

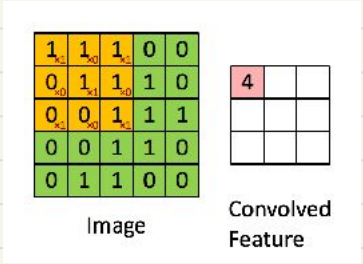
```
# 모델 컴파일 및 fit시키기
model.compile(optimizer='adam',
loss='categorical_crossentropy',
metrics=['accuracy'])

history=model.fit(X_train, Y_train,
validation_data=(X_val, Y_val), epochs=7)

model.save('Gersang.h5')
```

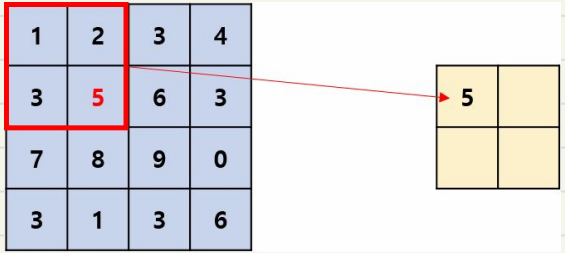
What are These Functions?





Conv2D()

* 필터 적용: 합성곱 연산을 통해 출력 이미지 생성
* 스트라이드: 필터의 이동량으로, 출력 이미지 변화
* 패딩: 입력 이미지와 출력 이미지의 크기를 같도록 이미지 크기 키움



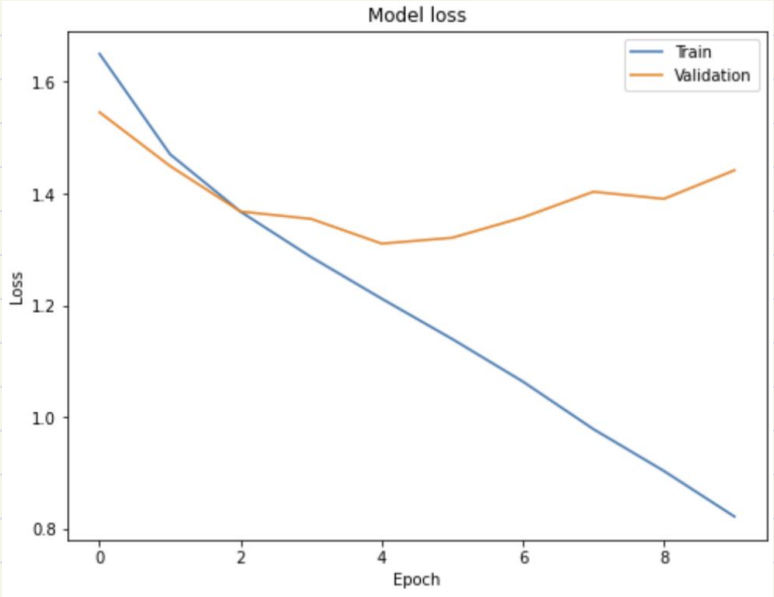
Max Pooling( )

크기를 줄이고 특정 feature를 강조하기 위하여 Pooling을 하며, Max Pooling은 가장 큰 숫자만 선택

Project Codes

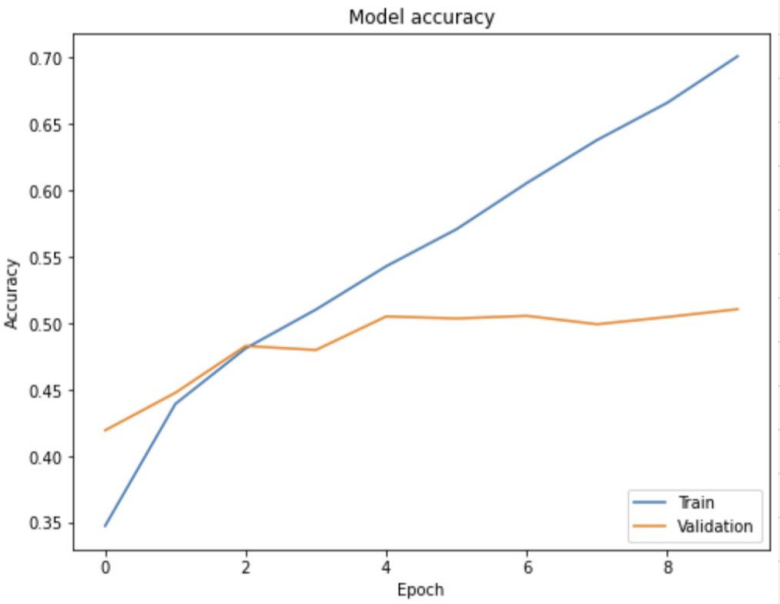
〈손실함수와 정확도 그래프로 확인하기〉

```
# 손실함수 그래프 그리기
fig, ax = plt.subplots(figsize=(8, 6))
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper
right')
plt.show()
```



Project Codes

```
# 정확도 그래프 그리기
fig, ax = plt.subplots(figsize=(8,6))
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='lower
right')
plt.show()
```



Project Codes

〈모델 평가하기〉

#모델 평가하기
model = load_model('Gersang.h5')

model.evaluate(X_test,Y_test)[1]

```
180/180 [==============================] - 1s 3ms/step - loss: 1.4748 - accuracy: 0.5063
0.5062695741653442
```

Project Codes

〈이미지를 입력하여 결과 확인하기〉

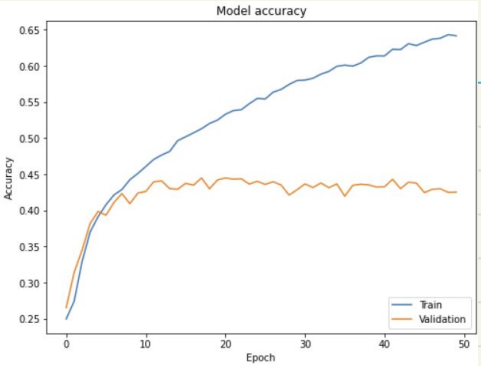
```
url='https://bit.ly/3IDqQdW'

response = requests.get(url)
i = Image.open(BytesIO(response.content))
image = i.resize((28, 28))
img_array = img_to_array(image)

preprocessed_img = preprocess_input(img_array)
preprocessed_img = np.expand_dims(preprocessed_img, axis=0)
model = load_model('Gersang.h5')
preprocessed_data = preprocess_input(preprocessed_img)

predicted_probabilities = model.predict(preprocessed_data)
predicted_category = np.argmax(predicted_probabilities)
categories = ["angry", "disgust", "fear", "happy", "neutral", "sad", "surprise"]
print(categories[predicted_category])
```

```
1/1 [==============================] - 0s 72ms/step
angry
```



What Problems?

훈련 데이터의 정확도 증가, 검증 데이터의 정확도 감소
=> 과적합 문제로 판단

What Attempts?

<함수의 인자 조정하기>

컨벌루션 레이어의 필터 수를 줄여 복잡도를 감소시키려 하였고 Dropout의 인자 등을 조정하였지만 큰 변화를 찾지 못함.

<K겹 교차 검증>

과적합을 피하기 위해 시도해봤지만 코드에 에러가 발생하였고 에러를 해결해도 정확도가 높아지지 않았음.

<다른 데이터 사용하기>

데이터의 양이 충분하지 않다고 판단하여 Kaggle에서 다른 데이터를 불러옴.

Conclusion?

다른 해결책을 떠올려도 더이상 할 수 있는 방안이 없었다.

캐글의 다른 분들이 만든 모델의 정확도도 높지 않았다.

그럽게 저희는 한계를 느끼고 프로젝트를 마무리 했다는,,,,,슬픈 이야기가,,,

〈우리의 느낀점〉

고진영

프로젝트를 진행하면서 좋은 모델 선정의 중요성에 대해서 알게 되었다. 1주차부터 책에서 공부한 내용들, 그리고 각 팀에서 준비한 발표 내용들을 통해 공부한 내용이 프로젝트를 준비하면서 많은 도움이 되었다. 마지막으로 팀플이 매우 잘 된 것 같아서 좋았다.

길태호

모델의 정확도를 높이는 동시에 과적합을 피하는 것이 어렵다는 사실을 체감할 수 있었다. 이번 프로젝트 덕분에 많이 배울 수 있었다.

양지윤

CNN에 대해 공부하면서 어려운 내용이 많아 이해하기 힘들었다. 하지만 계속 찾아보고 이해하려고 노력하니 조금이라도 CNN에 대해 알게 되어 뿌듯하다. 또한 같이 프로젝트를 하니 다양한 아이디어가 많이 나와 좋았다.

04

Project Practice

Colab으로 프로젝트 실습

링크: cnn 프로젝트 진행중 - Colaboratory (google.com)

원하는 표정의 사진을 다운받아 모델에 입력해봐요!

Thank You

감사합니다