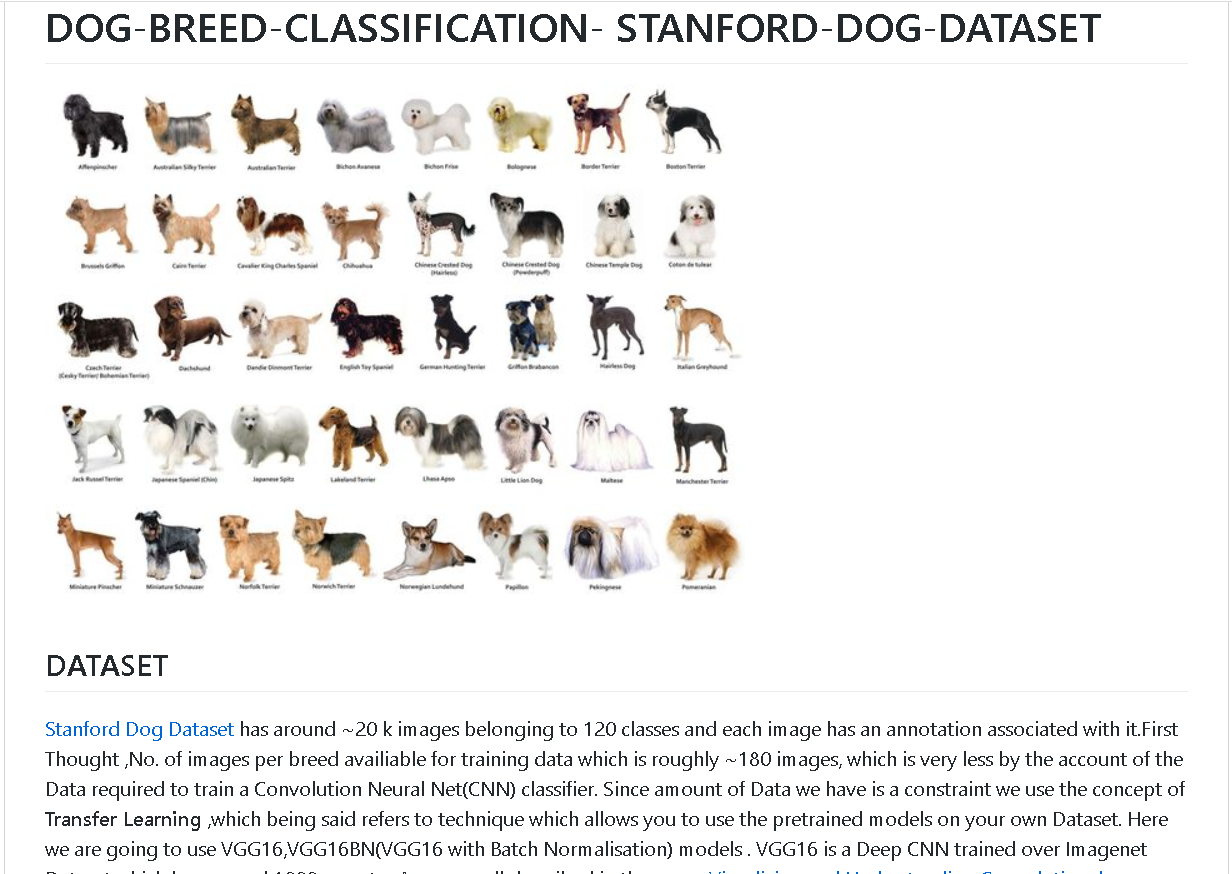
오픈소스 전문 프로젝트

Report #05

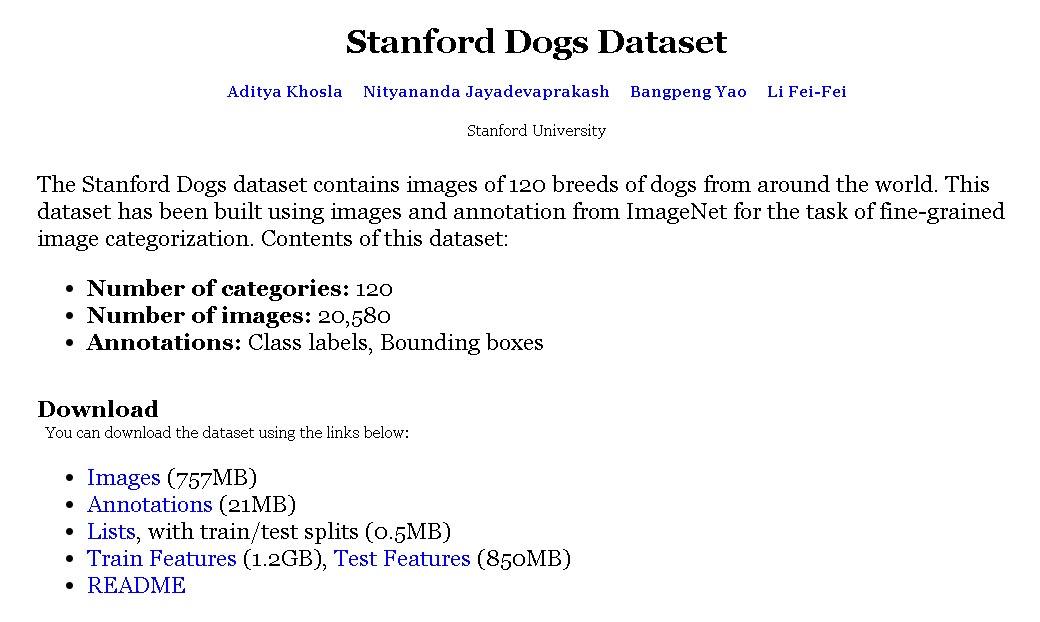
1차 데이터 및 분석 계획서

2조 – 유지원, 김혁, 박소연, 한현지

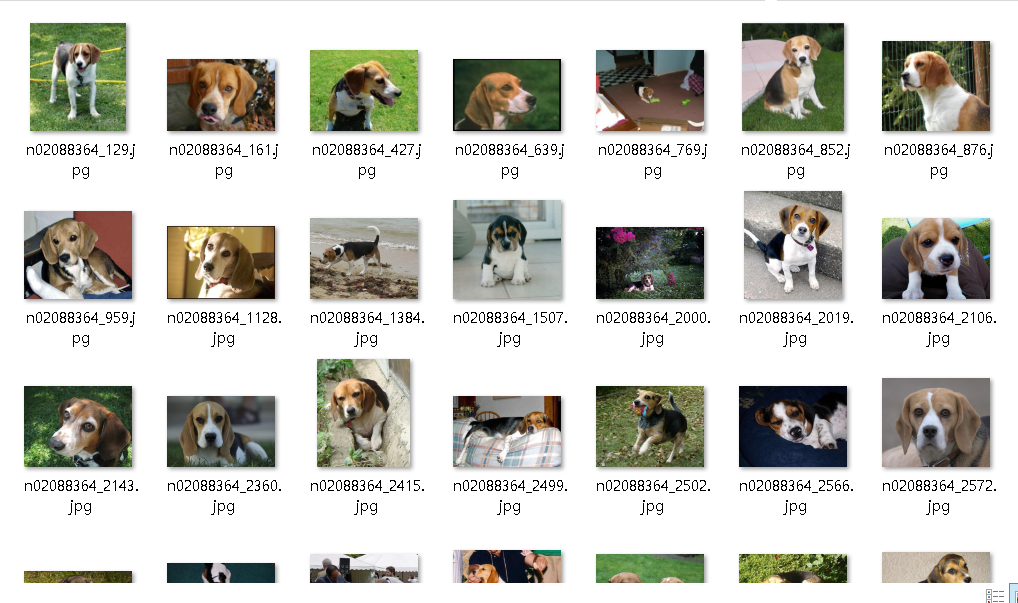
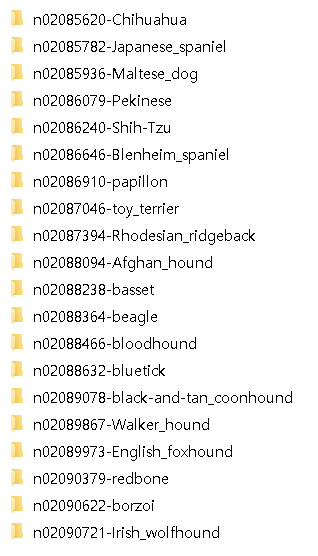
◈ 데이터 수집



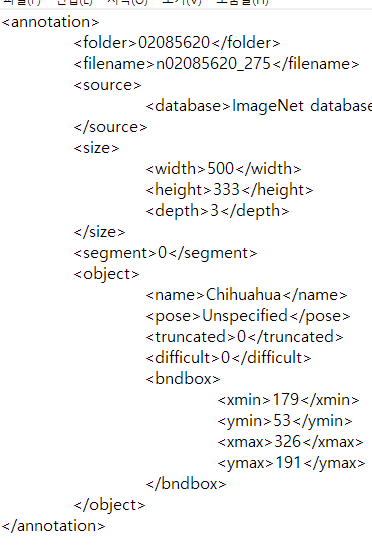
* 깃허브에 강아지 품종 분류 모형 검색(Dataset에 보면 stanford dog dataset을 기반으로 하였음을 알 수 있다)



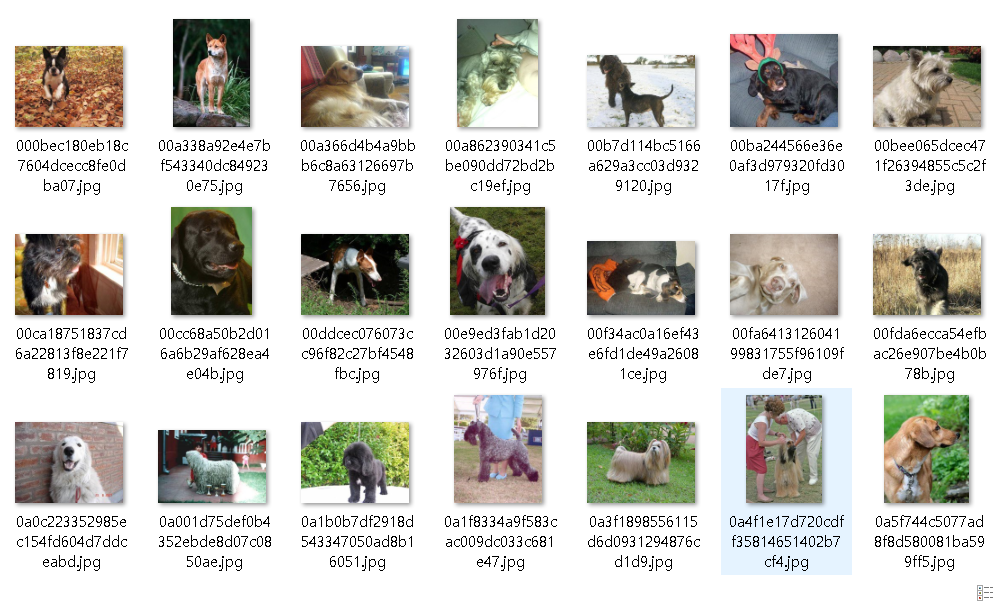
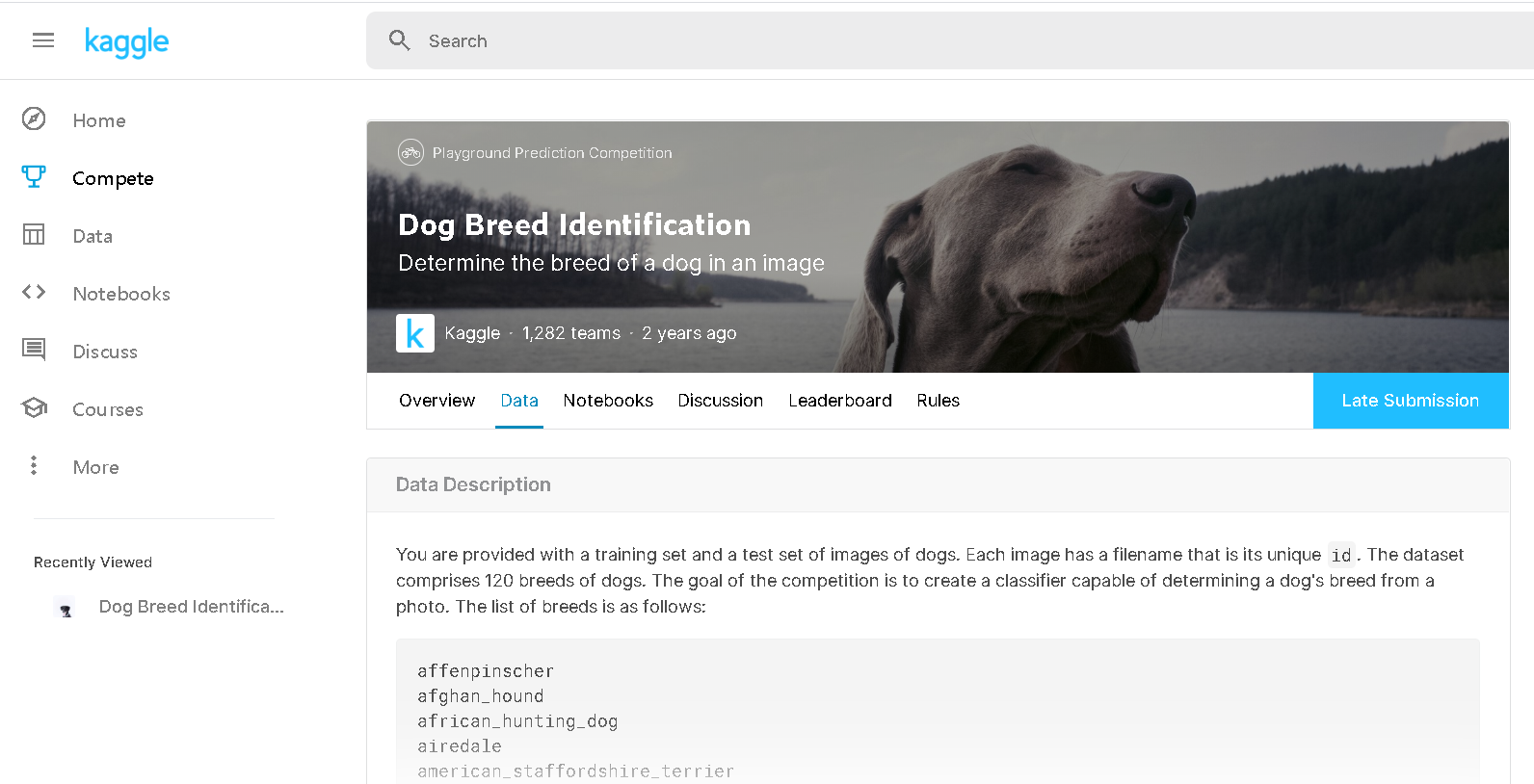
Github에 강아지 품종 분류 모형을 검색해보니 대체로 Stanford dogs dataset을 기반으로 모델을 설계했음을 알 수 있었다. 스탠포드 데이터셋을 보니 품종 별로 폴더에 정리되어 있어서 사용하기 간편했다. 같이 제공하는 annotations 파일에는 사진의 사이즈와 클래스 레이블과 같은 사진 파일 별 정보와 강아지 분석을 위한 사진 내의 강아지 얼굴 위치 데이터가 저장되어 있다.



* Stanford dog dataset 이미지 폴더(품종별로 폴더에 저장되어 있다.)



* annotation 파일안의 사진 파일 별 정보

* Kaggle competition : Dog Breed Identification 대회에서 제공하는 데이터

Stanford dogs dataset만으로는 부족할 수 있고, 확실한 많은 데이터가 구현에 더 도움이 될 것이라 생각하여서 Kaggle에서도 강아지 품종 분류 모형을 검색하여 데이터를 수집하였다. Stanford dogs dataset과 다르게 품종별로 폴더에 저장되어 있는 것이 아니라 약 10000장의 사진이 저장되어 있었는데 데이터와 함께 각 사진이 어떤 품종인지 알려주는 csv 파일(label.csv)이 있어 분류하는 것이 그렇게 어렵지 않을 것으로 보인다.

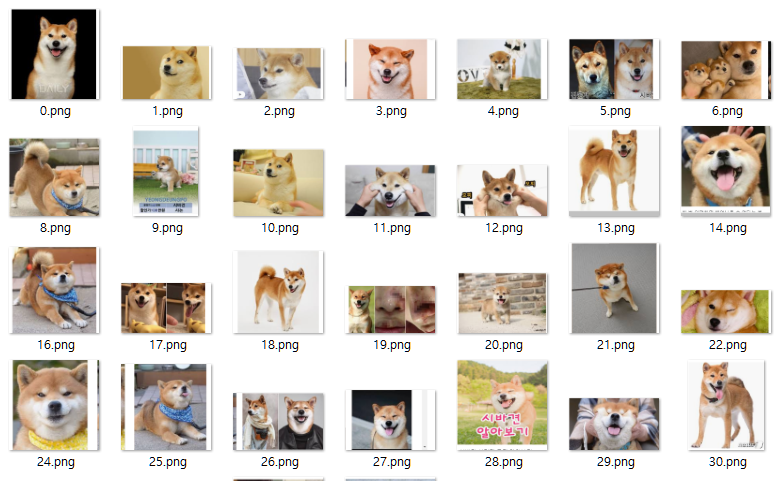
Stanford Stanford dogs dataset에서 품종별로 사진 100장 이상, 총 약 12000장 가량, 그리고 Kaggle에서 총 10000장 이상의 이미지 데이터를 수집하였으나 국내 품종 견에 대한 데이터가 많이 부족할 것이라는 생각이 들어서 국내 품종 견 데이터 수집을 위한 크롤링을 하였다.

크롤링의 타겟 사이트는 구글 검색창이다. 구글 이미지 검색의 키워드를 각각 시바견, 웰시, 진돗개의 샘플 데이터를 크롤링하였다. 크롤링에 사용한 코드는 파이썬을 이용하였고, 프레임워크로 Selenium을 사용하였다. 아래는 그 코드이다.



셀레니움은 브라우저를 명령어로 조작할 수 있게 해주는 프레임워크이다. 구글 이미지 검색창은 총 100개의 이미지를 보여주고 스크롤을 아래로 쭉 내리는 동작을 하면 이미지를 더 보여준다. 이를 위해 명령어를 통해 스크롤을 내려준 뒤 이미지들이 가지고 있는 rg\_i를 크롤링해준다.

아래는 시바견의 크롤링 데이터이다.



믹스견 품종 분류를 위하여 사용되는 주 학습 데이터는 Stanford dog dataset과 kaggle에서 수집한 데이터가 될 예정이다. 1) 모델 학습을 위한 신뢰성이 보장되며, 2) 데이터의 분류가 명확하고, 클래스 레이블 대입이 간단하다. 하지만 위에서도 말했듯이 국내 강아지 품종에 대한 데이터가 많이 부족하기 때문에 크롤링을 이용하여 필요한 데이터를 보충하기로 했다.

◈ 데이터 전처리 및 분석 계획

확보한 사진 데이터를 딥러닝 모델에 학습시키기 위해서는 데이터를 전처리 할 필요가 있다고 생각하였다. 데이터 전처리 과정을 통하여 첫째, 학습에 맞지 않는 모호하거나 틀린 이미지 데이터를 제거한다. 둘째, 부족한 이미지 데이터를 일정한 규칙을 통해 부풀려서 학습 데이터의 확보를 충분히 한다. 셋째, 사진 데이터를 학습 모델이 받아들일 수 있는 일정한 사이즈이 RGB 형태 행렬로 표현한다. 넷째, 사진 각각의 파일마다 대응되는 분류값(클래스) 레이블의 array를 만들어 냄으로써 딥러닝 모델에 사용할 수 있는 통합된 학습 데이터를 만들어 낸다.

1. 수작업 분류

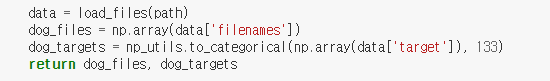
Stanford Dog dataset과 Kaggle의 데이터는 엄선된 데이터를 제공하고 있으므로 올바른 학습을 유도할 수 있지만, 크롤링을 통해 무작위 수집한 데이터들의 경우에는 일부 부적절한 데이터가 섞여 있으므로 걸러내야할 필요가 있다. 걸러내는 작업에서는 다른 머신러닝 모델을 사용하여, 인식하고 분류하는 방식을 사용할 수 있지만 이번 모델에서 학습하는 데이터의 양이 크지 않으므로 수작업 분류가 더 효율적으로 작용할 것이라 생각된다.

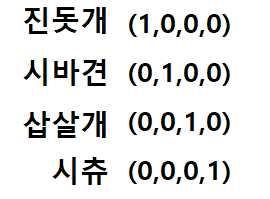
* 구글 이미지 검색을 통한 ‘진돗개’ 크롤링 수집 데이터들이다.

1. 이미지 데이터 클래스 레이블 대입

가장 먼저 이미지 데이터를 불러오게 되면, 사진 데이터는 RGB의 수치 값을 저장하는 하나의 행렬로 저장될 수 있다. 이를 통해 먼저 이미지 파일로 존재하는 데이터를 전부 학습 데이터로 만들 수 있다.

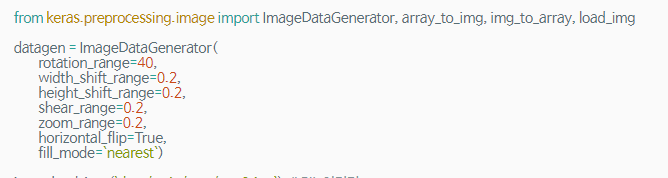


이미지 데이터 마다 일정한 분류 값(품종) 이 존재하므로, 이를 학습 데이터로 사용하기 위해서는 이미지에 대응되는 정답 데이터를 만들어 주어야 한다. 즉, X라는 벡터에 이미지 데이터가 나열돼 있다면 Y라는 벡터에는 같은 인덱스의 데이터가 어떤 품종에 속하는지를 알려줄 수 있어야한다. 이때, 클래스가 다수이므로 Y에 클래스 레이블을 할당하는 과정에서 원핫 인코딩 방식을 사용하여야 한다.



1. 이미지 정규화 및 데이터 증강

케라스 (Keras) 라이브러리는 이미지를 정규화 하고, 다양한 규칙(회전, RGB 값 조절) 등을 통해 이미지 데이터를 다양하게 늘려주는 ImageDataGenerator 함수를 제공한다. 때문에 품종 별 150개 정도에 불과했던 학습 데이터를 수 백장까지 늘릴 수 있으며, 이와 같은 기법을 활용하면 딥러닝 모델의 정확도를 증가시킬 수 있다. 또한, 이미지를 늘려줄 뿐만 아니라 일정한 규칙에 의해서 이미지의 RGB를 Scaling하고 이미지의 사이즈 등을 정규화하여 효율적인 학습 데이터로 만들어준다.





전처리 된 학습 데이터를 딥러닝 모델에 학습시키고 그 모델을 통해 새로운 사진을 분석할 수 있도록 한다. 이때, 결과 값은 해당 품종(클래스)에 속할 확률을 표시하게 된다. 이와 같은 출력 값을 활용하여, 분석을 시도한 믹스견의 사진에 어떤 품종이 섞여 있는지를 대략적으로 확인할 수 있다.

1. Convolutional Neural Network, CNN을 통한 사진 데이터 분석과 분류

커널의 합성곱을 활용하여 이미지 데이터를 효과적으로 분류해내는 딥러닝 모델인 CNN을 이용하여 이미지를 분석한다.

일반적인 CNN은 다음 두 가지 파트로 구성되어 있다.

Convolutional base

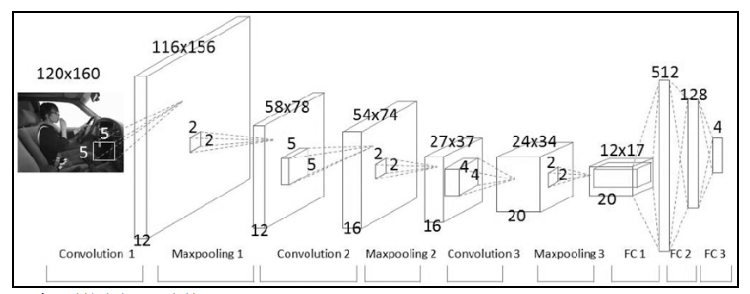
합성곱층과 풀링층이 여러겹 쌓여있는 부분, convolutional base의 목표는 이미지로부터 특징을 효과적으로 추출하는 것 (feature extraction).

Classifier

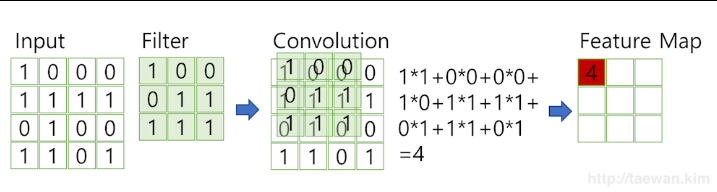
주로 완전 연결 계층 (fully connected layer)로 이루어져 있다. 완전 연결 계층이란 모든 계층의 뉴런이 이전 층의 출력 노드와 하나도 빠짐없이 모두 연결되어 있는 층을 말한다. 분류기(Classifier)의 최종 목표는 추출된 특징을 잘 학습해서 이미지를 알맞은 카테고리로 분류하는 것 (image classification).

이미지 분류(CNN에서 Classifier에 해당하는) 문제에서 표준으로 쓰이는 방법 중 하나는 완전 연결 계층을 쌓은 후 마지막에 소프트맥스 활성화함수(softmax activated layer) 계층을 놓는 것이다. 소프트맥스 계층은 주어진 카테고리 각각에 대해 그 카테고리일 확률값을 출력한다.

품종 분석을 목적으로 CNN 모델을 사용하기 위해서 Tensorflow와 keras 라이브러리에서 제공되는 함수를 이용할 계획이며, 대부분의 CNN 모델은 Convolution층 3개와 Pooling층 3개로 설계된다. 모델의 구조를 그림으로 살펴보면 다음과 같다.

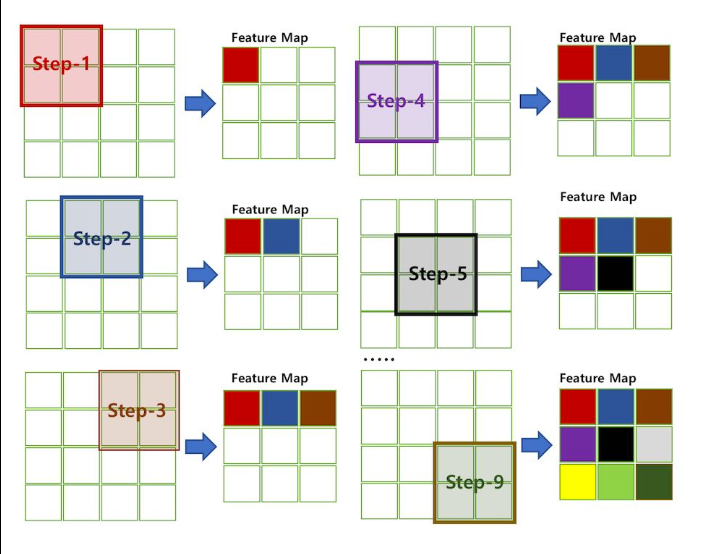


CNN은 필터를 사용해 특징을 잡아낸다. 필터는 CNN이 학습해야 할 데이터이다. CNN은 특징 잡는 필터를 학습하므로 다른 신경망 보다 더욱 장동화된 이미지 분류를 한다. 아래는 필터와 인풋데이터, 합성곱하여 피쳐맵이 만들어지는 과정이다.



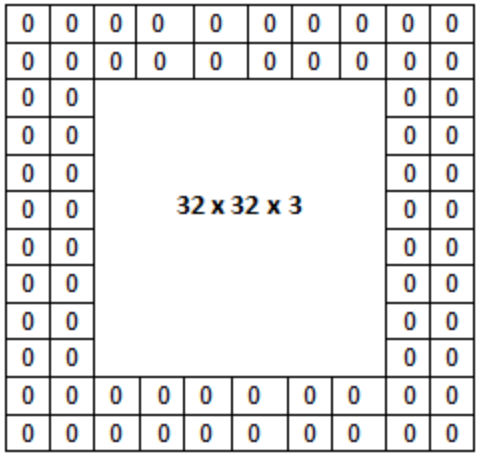
필터는 위 방법과 같이 정해 놓은 stride의 값만큼 이동하여 피쳐맵의 값을 채운다. 위 그림에서 stride가 1일 경우, 피쳐맵은 2X2가 된다.

아래 그림은 4X4의 인풋데이터를 2X2의 필터를 적용하여 1 stride로 3X3의 피쳐맵을 만든다.



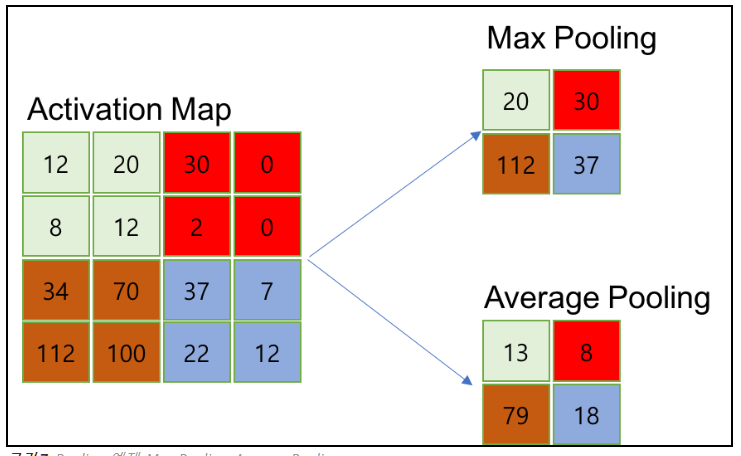
인풋 -> 피쳐맵으로 변환되며 줄어든 크기는 패딩을 통해 채운다. 보통 패딩값은 0으로 채운다.

패딩은 사용하지 않을 수 있으며 필요에 따라 달라진다.



다음으론 풀링이다.

풀링 레이어는 컨볼루션 레이어의 출력 데이터를 입력으로 받아서 출력 데이터(Activation Map)의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용된다. 그림은 Max pooling과 Average Pooling의 동작 방식을 설한다. 일반적으로 Pooling 크기와 Stride를 같은 크기로 설정하여 모든 원소가 한 번씩 처리되도록 설정한다.



Pooling 레이어는 Convolution 레이어와 비교하여 다음과 같은 특징이 있다.

학습대상 파라미터가 없음

Pooling 레이어를 통과하면 행렬의 크기 감소

Pooling 레이어를 통해서 채널 수 변경 없음

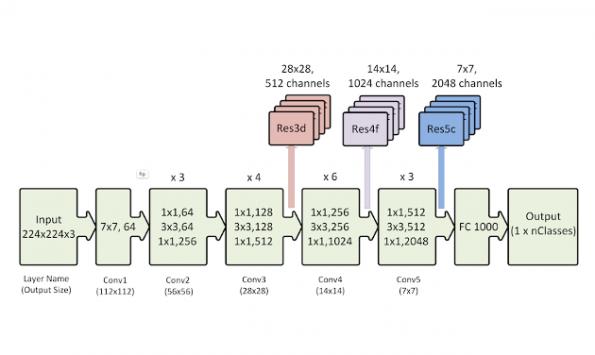
CNN에서는 주로 Max Pooling을 사용한다.

이런 방식으로 세번 정도의 Convolution Layer와 Max Pooling를 지나면 채널의 수는 많아지고 이미지 데이터의 크기는 점점 작아진다. 각 Convolution Layer의 학습 파라미터의 수는 입력채널, 출력채널, 필터 가로, 필터 세로크기의 곱이다.

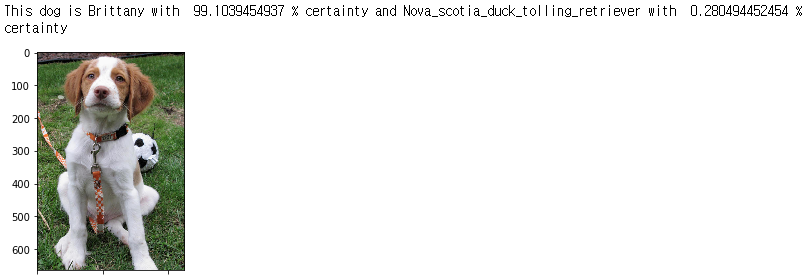
Convolution Layer와 Max Pooling들을 거쳐, Flatten Layer에 도착한다. Flatten Layer는 CNN의 데이터 타입을 Fully Connected Neural Network의 형태로 변경하는 레이어이다. Flatten 레이어에는 파라미터가 존재하지 않고, 입력 데이터의 Shape 변경만 수행한다. 마지막으로 데이터는 Softmax layer에 도착하고 분류된다.

1. Transfer learning using Pre-trained ResNet-50

상대적으로 매우 적은 수의 강아지 품종 별 데이터를 활용하여 강아지를 분류하기는 어렵다. 때문에 사전에 대규모 데이터를 통해 학습된 모델의 앞쪽 레이어를 불러와서 활용할 수 있도록 한다. 이와 같은 방법을 Transfer Learning이라고 한다. 즉, 강아지 사진 자체가 학습되어 있지 않은 모델을 처음부터 학습시키기보다는 강아지와 고양이, 사람, 물건 등을 구분할 수 있는 학습 모델에서, 강아지를 더 세부적으로 분류하는 모델을 도출하기가 쉽다는 원리이다. 이때, Transfer Learning에서 사용되는 모델은 15년도 Image-net에서 좋은 성적을 거둔 CNN 응용 모델인 Resnet-50을 이용할 계획이며, 해당 모델은 Keras 라이브러리에 미리 구현이 되어있으므로 쉽게 Transfer Learning이 가능하다. (Resnet-50 모델과 병목 특징을 불러오는 예제)



1. 결과 데이터를 이용하여 믹스견의 품종 예측



딥러닝 모델을 학습시키고 임의의 강아지 사진을 넣어서 예측하였을 때, 그 결과 값은 위의 그림과 같이 확률로 출력된다. 이때, 출력의 활성화 함수인 softmax 함수는 가장 정답에 근접할 것으로 여겨지는 레이블의 추정 확률을 상대적으로 강조하여 출력하게 되는 경향이 있기 때문에, 단일 품종을 제시하는 것에는 효율적이지만 다양한 품종의 가능성을 제시하는 것에는 부적절하다. 즉, 믹스견의 품종을 예측하는 것은 어렵다. 때문에 출력 결과 값의 2등, 3등 클래스의 확률을 조정하거나 혹은 출력 방식을 softmax에서 단순한 함수 값으로 바꾸어 결과를 만들 수 있는 방법을 생각할 필요가 있다.