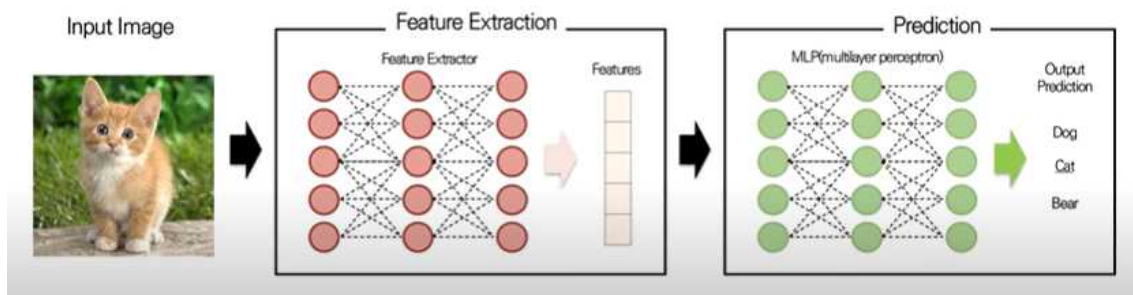


백본 정의



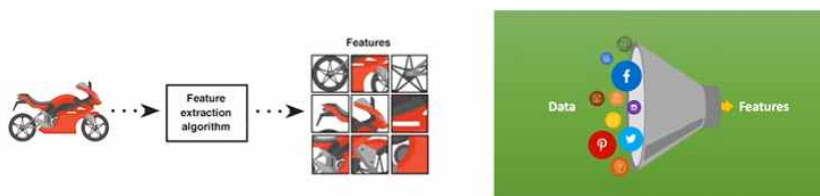
위 그림은 일반적인 이미지 분류 인공신경망이다.

1. Input Image를 넣은 뒤 그 Image의 Feature를 추출한다.
2. 추출한 이미지의 Feature값을 예측 모델에 넣어 예측한다.

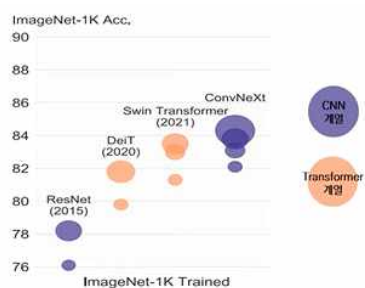
여기서 Feature Extractor가 흔히 말하는 백본 네트워크이다. 헛갈릴 거 없이 이게 백본이다.

백본을 사용하는 이유

1. 입력 데이터의 특징을 잘 추출해야됨으로
2. 입력데이터의 계산 복잡도를 낮추기 위해
3. 이미지 분류 분야에 사용이 됨



백본의 종류 (최신 버전)



Transformer 계열과 CNN계열로 나뉘
그 외 종류 : AlexNet, VGG, ResNet

U-net은 백본이 될 수 없나?

백본에서도 CNN기반으로 파생된 백본이 있어 헛갈릴 수 있지만 UNet은 백본 역할을 할 수 없다. Unet은 세그멘테이션 네트워크로 엑셀 단위로 분할하여 작업하는 딥러닝 네트워크이다. 즉 백본은 아까 말했듯 특징만 추출해주는 역할이고 UNet은 예측을 해주는 역할이다. 하지만 기존의 있는 백본과 결합하여 Unet의 인코더 부분에 백본을 이용할 수 있다. Unet과 ResNet을 결합한 것이 ResUNet 네트워크이다.

그리고 이번 모델 구현에서 제가 구현해본게 ResUNet 네트워크

```
class ResNetBackbone(nn.Module):
    def __init__(self):
        super(ResNetBackbone, self).__init__()

        resnet = models.resnet50(pretrained=True)

        self.features = nn.Sequential(*list(resnet.children())[:-2])
        self.upsample = nn.Upsample(size=(224, 224), mode='bilinear', align_corners=True)
        self.res_down1 = double_conv(2048, 64)

    def forward(self, x):
        features = self.features(x)
        features = self.upsample(features)
        features = self.res_down1(features)
        return features
```

ResNet50을 사용함

-> 여기서 문제가 나오는 feature값이 항상 (2048,7,7)과 같이 작은 크기의 사이즈로 나와 feature값을 그대로 UNet에 학습시키면 값이 너무작아져서 학습이 안되는 오류 발생

-> 그래서 Upsample으로 이미지 크기를 넓혀준 뒤 UNet으로 학습을 시켜줌

159	Conv2d-156	[-1, 512, 7, 7]	2,359,296
160	BatchNorm2d-157	[-1, 512, 7, 7]	1,024
161	ReLU-158	[-1, 512, 7, 7]	0
162	Conv2d-159	[-1, 2048, 7, 7]	1,048,576
163	BatchNorm2d-160	[-1, 2048, 7, 7]	4,096
164	ReLU-161	[-1, 2048, 7, 7]	0
165	Bottleneck-162	[-1, 2048, 7, 7]	0
166	Conv2d-163	[-1, 512, 7, 7]	1,048,576
167	BatchNorm2d-164	[-1, 512, 7, 7]	1,024
168	ReLU-165	[-1, 512, 7, 7]	0
169	Conv2d-166	[-1, 512, 7, 7]	2,359,296
170	BatchNorm2d-167	[-1, 512, 7, 7]	1,024
171	ReLU-168	[-1, 512, 7, 7]	0
172	Conv2d-169	[-1, 2048, 7, 7]	1,048,576
173	BatchNorm2d-170	[-1, 2048, 7, 7]	4,096
174	ReLU-171	[-1, 2048, 7, 7]	0
175	Bottleneck-172	[-1, 2048, 7, 7]	0
176	Upsample-173	[-1, 2048, 224, 224]	0
177	Conv2d-174	[-1, 64, 224, 224]	1,179,712
178	ReLU-175	[-1, 64, 224, 224]	0
179	Conv2d-176	[-1, 64, 224, 224]	36,928
180	ReLU-177	[-1, 64, 224, 224]	0
181	=====		

ResNet backbone summary 마지막 부분 feature값을 (2048,244,244) 형태로 출력함
그 후 이 값을 주어진 UNet에다가 학습!