

## 2024-1학기 창의학기제 주간학습보고서 (1주차)

창의과제	세종대학교 집현캠퍼스를 개선시킨 웹서비스 개발				
이름	이지민	학습기간	3월4일 ~ 3월 15일		
학번	23012127	학습주차	1	학습시간	3
학과(전공)	인공지능	과목명	자기주도 창의전공1	수강학점	3
※ 수강학점에 따른 회차별 학습시간 및 10주차 이상 학습 준수					
금주 학습목표	얼굴 분류 모델을 찾고 그 모델을 파이토치로 구현하기				
학습내용	<p>얼굴인식을 위한 인공지능을 만들기 위해 이미지 데이터를 학습시킬 때 사용되는 Convolutional Neural Network의 원리를 알아보았다.</p> <p>Convolutional Neural Network 줄여서 CNN은 이미지의 한 픽셀을 기준으로 그 픽셀이 주변 여덟 픽셀 혹은 그 이상의 픽셀과 어떤 관계를 가지고 있다는 이미지의 특징을 Neural Network(NN)에 적용하기 위해 만들어졌다.</p> <p>CNN은 convolution layer와 pooling layer로 이루어져 있다. convolution layer에서는 앞서 말했던 이미지의 특징을 NN에 적용하기 위해 receptive field를 정의하여 이미지와 convolution 시킨다. 그러면 해당 이미지 데이터의 특징을 지닌 feature map을 형성하게 된다.</p> <p>Pooling layer에서는 모델의 학습 속도를 향상하기 위해 이미지의 dimension을 줄인다.</p> <p>CNN을 통해 생성한 feature map을 Fully Connected Layer에 적용하여 최종적으로 분류해 낸다.</p> <div data-bbox="517 1420 1179 1886"> <p>Hand-drawn diagram illustrating the CNN architecture. It shows an input image (7x7) being convolved with a 3x3 filter (w) to produce a 5x5 feature map. This feature map is then passed through a fully connected layer to produce the final output. The diagram is labeled 'CNN' and includes handwritten notes in Korean explaining the process.</p> </div> <p>&lt;작성자가 이해한 CNN 원리&gt;</p>				



공부한 CNN의 원리를 가지고 얼굴인식 모델을 만들기 전에 CNN에 대한 이해도를 높이고 pre-train 된 모델의 특징을 알아보기 위해 pytorch에서 제공하는 MobileNetV3를 불러와 조원들의 얼굴 사진을 input data로 만들어 모델에 넣어 분류를 시도해 보았다.

<ImageTransform class code>

```
class ImageTransform():
    def __init__(self, resize, mean, std):
        self.data_transform = {
            'train': transforms.Compose([
                transforms.Resize((224,224)), # 이미지를 224*224 크기로 resize
                transforms.ToTensor()
            ]),
            'val': transforms.Compose([
                transforms.Resize((224,224)), # 이미지를 224*224 크기로 resize
                transforms.ToTensor()
            ])
        }

    def __call__(self, img, phase):
        return self.data_transform[phase](img)
```

전처리를 마친 이미지 데이터들은 pytorch에서 사용할 수 있도록 pytorch의 dataloader로 만든다.

pre-train된 MobileNetV3를 사용하는 모델 클래스를 작성하고 train함수를 작성하여 이전에 만든 dataloader를 모델에 넣어 학습시켜 본다.

<model class code>

```
from torchvision.models import mobilenet_v3_large
class SiameseNetwork(nn.Module):
    def __init__(self):
        super(SiameseNetwork, self).__init__()
        self.model=mobilenet_v3_large(pretrained=True) #pretrain된 모델을 불러옴
        self.fc = nn.Sequential(
            nn.Linear(960, 512),
            nn.ReLU(inplace=True),
            nn.Linear(512, 512),
            nn.ReLU(inplace=True),
            nn.Linear(512, 2))
        self.model.classifier=self.fc
    def forward(self, input1, input2):
        output1 = self.model(input1)
        output2 = self.model(input2)
        return output1, output2
```

학습방법

CNN과 관련된 여러 동영상을 찾아보았다.

CNN과 관련된 서적을 찾아보았다.

혼자 공부하는 머신러닝&딥러닝 책에서 학습한 내용을 바탕으로 딥러닝 모델을 구현했다.



학습성과 및 목표달성도	시행 결과 어느 정도의 분류를 해내는 것을 볼 수 있었지만, 만족스러운 결과는 나오지 않았다. 또한 label에 존재하지 않는 data를 넣었을 때 여러 label에 대한 softmax probabilities가 80% 이상 나오는 등 얼굴을 인식한다고는 볼 수 없다고 보았다.
참고자료 및 문헌	convolution에 관한 영상: <a href="https://www.youtube.com/watch?v=KuXjwB4LzSA">https://www.youtube.com/watch?v=KuXjwB4LzSA</a> 인공 신경망에 관한 영상: <a href="https://www.youtube.com/watch?v=aircAruvnKk">https://www.youtube.com/watch?v=aircAruvnKk</a> CNN 원리 참고 : 파이썬 딥러닝 파이토치/정보문화사/이경택, 방성수, 안성준 저 핸즈온 머신러닝/한빛미디어/오렐리앙 제룡 저
내주 계획	보안을 위한 얼굴인식을 수행하기 위한 신경망을 만들기 위해선 어떤 방법이 있는지 확인해보고 적용해본다. 정확도를 높이기 위해 얼굴의 특징만을 더 자세하게 따서 학습시키는 방법을 강구한다.

년 월 일

지도교수

(인)