

모각소 5주차

소프트웨어학과 202220771 김민준

최종 결과

모든 데이터셋을 사용하여 학습과 테스트를 진행하였다. 크기가 132인 numpy배열에서 33개의 visibility 값을 제외하고 총 99개의 데이터만을 사용하도록 커스텀 데이터셋을 구현했다. 전체 데이터셋에서 학습과 테스트의 비율을 8대2로 설정했다. 신경망은 다음과 같이 구축하였다.

```
import torch.nn as nn

class Classification(nn.Module):
    def __init__(self):
        super().__init__()
        self.fc1 = nn.Linear(99, 64)
        self.fc2 = nn.Linear(64, 32)
        self.fc3 = nn.Linear(32, 3)

    def forward(self, x):
        x = nn.Identity()(self.fc1(x))
        x = nn.Identity()(self.fc2(x))
        x = self.fc3(x)
        return x
```

실시간으로 동영상을 입력으로 넣었을 때, 어떠한 운동인지를 출력하기 때문에 매 순간 어떤 운동인지를 확인하므로 1프레임 단위로 학습을 진행했다. 입력층의 크기는 99로 1프레임에서 추출한 랜드마크의 좌표를 모두 입력층에 넣는다. 학습을 진행한 3가지의 운동을 분류하기 위해 마지막 출력층의 크기는 3이다. 또한, 좌표의 값은 크기에 따라 값의 중요도가 바뀌는 것이 아니기에 sigmoid나 relu함수를 사용하지 않고 항등함수를 활성화 함수로 사용했다. optimizer로는 Adam을 선택했고, 학습률은 0.001로 설정했다. 손실함수는 다중 클래스 분류를 위해 CrossEntropyLoss를 선택했다.

분류 성능을 확인하기 위해 약 13,000개의 프레임에서 추출한 랜드마크로 10번의 epoch으로 학습을 진행했다. 또한, push up, squat, pull up 외의 다른 운동 영상으로도 성능을 측정했다. 학습한 운동 외의 다른 자세가 감지되는 것을 방지하기 위해 softmax함수를 통해 각각의 확률을 구했고, 가장 큰 확률이 0.99를 넘지 못하면 감지하지 못한 것으로 분류했다.

결과

학습한 모델을 통해 push up, squat, pull up을 분류할 수 있다. Macro-average를 사용하여 F1-score를 구해 성능을 측정했다

	Precision	Recall	F1-score
Push up	0.89	1.00	0.94
Squat	0.28	1.00	0.44
Pull up	0.30	1.00	0.46
Macro avg	0.62	0.80	0.54

분류 모델의 성능 결과를 통해 모든 운동에 대해 높은 재현율을 보인다. 반면 squat와 pull up은 정밀도가 상대적으로 낮은 것을 알 수 있다. 이러한 현상은 실제 운동을 정확하게 분류하는 데에는 문제가 없지만, squat와 pull up의 경우 다른 운동과 혼동될 가능성이 높다는 것을 의미한다.

이러한 차이는 운동 자세의 특성 때문에 발생할 수 있다. 예를 들어, push up은 신체가 가로로 누워 있는 자세에서 수행되기 때문에 다른 운동과 구분이 비교적 쉽다. 반면에, squat와 pull up은 앉아있거나 서서 진행되는 운동이기 때문에 다른 운동과 혼동될 가능성이 높을 수 있다.