모델 학습 보고서

이나겸

<table>
<tr><td>set_seed</td><td>set_seed를 만들고 seed 값을 줘서 고정된 결과가 나올 수 있도록 만들어줬다.
(model 호출 후 eval 값이 안 바뀌고 계속 같은 결과가 나옴)</td></tr>
<tr><td>dataset</td><td>dataset :
- category는 “O”와 “R” 두 개의 라벨링
- split_data 함수
dataset 안에 sklearn의 train_test_split으로 데이터의 30%를 랜덤하게 샘플링
(random_state 값을 줘서 호출할 때마다 동일하게 분할하도록 만듬)
<pre>
# dataset
category = {"O" : 0, "R" : 1}
class MyCustomDataset(Dataset):
    def __init__(self, path, mode, transform=None):
        self.path = self.split_data(path, mode)
        self.mode = mode
        self.transform = transform

    def __getitem__(self, index):
        data = self.path[index]
        # ./DATASET/TRAIN\O\O_1.jpg
        img = Image.open(data).convert("RGB")

        if self.transform is not None :
            img = self.transform(img)
        label_temp = data.split('\\')[-2]
        label = category[label_temp]

        return img, label

    def __len__(self):
        return len(self.path)

    def split_data(self, path, mode):
        O_data = sorted(glob.glob(os.path.join(path, mode, "O", "*.jpg")))
        R_data = sorted(glob.glob(os.path.join(path, mode, "R", "*.jpg")))
        t1_data, _, _, _ = train_test_split(O_data, O_data, test_size=0.7, random_state=100)
        t2_data, _, _, _ = train_test_split(R_data, R_data, test_size=0.7, random_state=100)
        data = t1_data + t2_data
        return data
</pre></td></tr>
<tr><td>augmentation</td><td>Resize((224,224)),
RandomHorizontalFlip(p=0.2), RandomVerticalFlip(p=0.2),
Normalize([0.5, 0.5, 0.5], [0.2, 0.2, 0.2])</td></tr>
<tr><td>model</td><td>mobilenetV2</td></tr>
<tr><td>epoch</td><td>10</td></tr>
<tr><td>Loss</td><td>CrossEntropyLoss()</td></tr>
<tr><td>optimizer</td><td>SGD / lr = 0.01 / momentum = 0.9</td></tr>
<tr><td>best.pt</td><td>
<pre>
0it [00:00, ?it/s]Starting evaluation
12it [00:02, 4.45it/s]
Test acc for image : 753 ACC : 95.22
End test..
</pre></td></tr>
<tr><td>last.pt</td><td>
<pre>
Starting evaluation
12it [00:02, 4.41it/s]
Test acc for image : 753 ACC : 91.77
End test..
</pre></td></tr>
</table>

데이터의 10%만 샘플링해서 사용
learning rate 0.0025, momentum 0.9 일 때 대체로 accuracy 값이 좋았음.
RandomAdjustSharpeness를 쓰니까 다른 하이퍼 파라미터를 바꿔도 전반적으로 학습결과가 조금 떨어짐.
데이터의 30% 샘플링해서 학습한 것보다 더 적은 10%로 학습했을 때도 95% 이상 좋은 결과가 나왔음.

model	mobilenetV2
epoch	10
Loss	CrossEntropyLoss()

augmentation	Resize((224,224)), RandomHorizontalFlip(p=0.2), RandomVerticalFlip(p=0.2), Normalize([0.5, 0.5, 0.5], [0.2, 0.2, 0.2])
optimizer	SGD / lr = 0.01 / momentum = 0.9
best.pt	Starting evaluation 4it [00:01, 2.67it/s] Test acc for image : 251 ACC : 95.62 End test..
last.pt	0it [00:00, ?it/s]Starting evaluation 4it [00:01, 2.60it/s] Test acc for image : 251 ACC : 94.02 End test..

augmentation	Resize((224,224)), RandomHorizontalFlip(p=0.2), RandomVerticalFlip(p=0.2), **RandomAutocontrast(),** Normalize([0.5, 0.5, 0.5], [0.2, 0.2, 0.2])
optimizer	SGD / **lr = 0.0025** / momentum = 0.9
best.pt	0it [00:00, ?it/s]Starting evaluation 4it [00:01, 2.66it/s] Test acc for image : 251 ACC : 96.81 End test..
last.pt	0it [00:00, ?it/s]Starting evaluation 4it [00:01, 2.63it/s] Test acc for image : 251 ACC : 93.23 End test..

augmentation	Resize((224,224)), RandomHorizontalFlip(p=0.2), RandomVerticalFlip(p=0.2), **RandomAutocontrast(), RandomAdjstSharpness(1.5),** Normalize([0.5, 0.5, 0.5], [0.2, 0.2, 0.2])
optimizer	SGD / **lr = 0.005** / **momentum = 0.7**
best.pt	0it [00:00, ?it/s]Starting evaluation 4it [00:01, 2.63it/s] Test acc for image : 251 ACC : 96.02 End test..
last.pt	Starting evaluation 4it [00:01, 2.68it/s] Test acc for image : 251 ACC : 94.42 End test..