

Presentation

2024.10.17

FASHION-HOW 시즌5

(Sub-Task1, 2)



CONTENTS

1. Sub-Task 1 Info
2. Code Review
3. Sub-Task 2 Info
4. Code Review
5. Fin.



곽명빈

- EDA
- Data Preprocessing
- Modeling
- Etc.

Sub-Task 1

Sub-Task 1 Information

Multi-task classification

- Multi-task classification은 하나의 모델이 여러 개의 클래스를 동시에 예측하는 방법으로, 다양한 태스크를 동시에 해결 가능

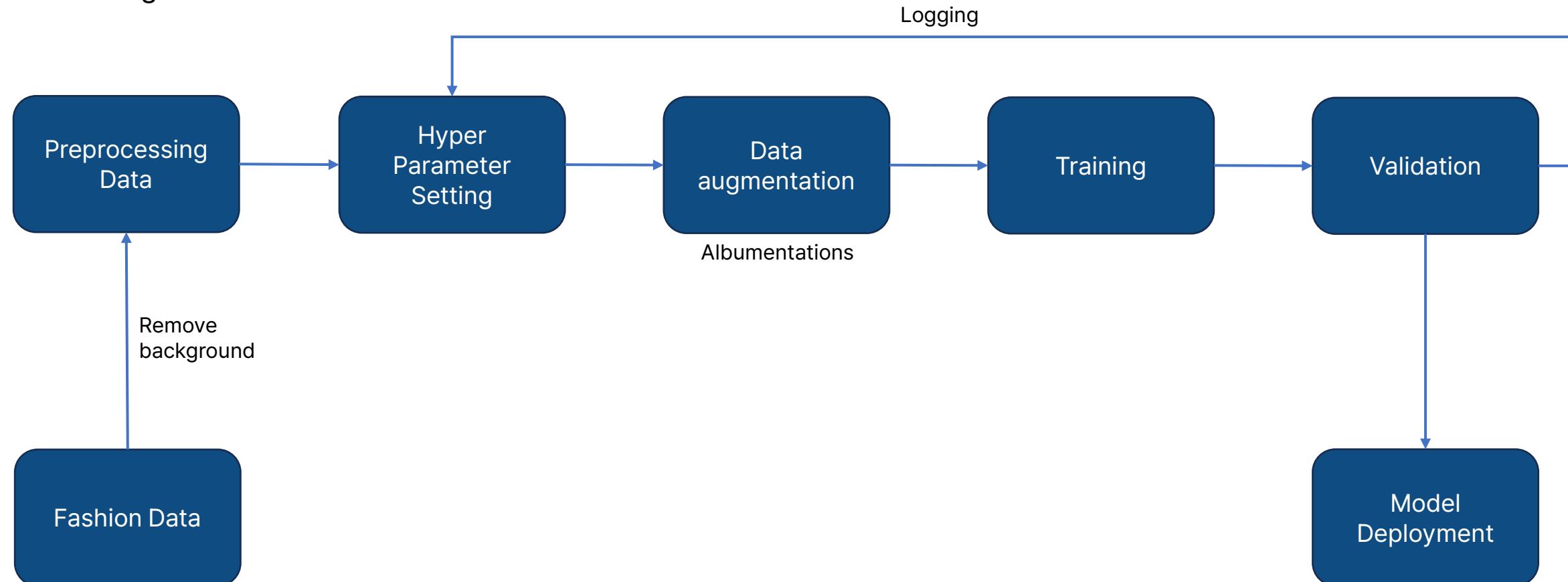
Fashion image attribute classification

- 입력된 패션 이미지를 미리 정해진 속성 카테고리들(daily, gender, embellishment)에서 각각의 라벨들로 분류하는 문제
- 하나의 이미지에서 여러 속성 카테고리들이 정의 될 수 있어 여러 개의 속성으로 분류 필요

Sub-task	Augmentation	Ensemble	Memory (과거 데이터셋)	Multimodal	외부 데이터셋	외부 사전학습 모델
1		불허	해당사항 없음	허용	허용	허용
2		허용	해당사항 없음	불허	불허	불허
3	허용	허용	불허	허용	불허	허용
4		허용	해당사항 없음	허용	허용	허용

Sub-Task 1 Code Review

Training Process



Sub-Task 1 Code Review

Remove Background

<https://github.com/danielgatis/rembg>

```
subtask1 - remove_background.py

1  from rembg import remove
2
3  for label in os.listdir(input_folder):
4      label_path = os.path.join(input_folder, label)
5
6      if os.path.isdir(label_path):
7          output_label_path = os.path.join(output_folder, label)
8
9      if not os.path.exists(output_label_path):
10         os.makedirs(output_label_path)
11
12     for image_file in tqdm(os.listdir(label_path)):
13         input_image_path = os.path.join(label_path, image_file)
14         output_image_path = os.path.join(output_label_path, image_file)
15
16         input_image = cv2.imread(input_image_path)
17
18         output_image = remove(input_image)    Remove Background
19         output_image = output_image[:, :, :3]
20
21         cv2.imwrite(output_image_path, output_image)
```



Sub-Task 1 Code Review

Data Augmentation

<https://github.com/albumentations-team/albumentations>

```
subtask1 - dataset.py

1  class ClassificationDataset(Dataset):
2      def __init__(self, root, df, transform=None, crop=True):
3          self.root = root
4          self.dataset = df
5          self.transform = transform
6          self.crop = crop
7
8      def __len__(self):
9          return len(self.dataset)
10
11     def __getitem__(self, idx):
12         data = self.dataset.iloc[idx]
13
14         image_path = data["image_name"]
15         image_path = os.path.join(self.root, image_path)
16
17         image = cv2.imread(image_path)
18         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
19
20         if self.crop and random.random() < CROP_PROB:
21             image = self.bbox_crop(image, data)
22
23         if self.transform is not None:
24             image = self.transform(image=image)["image"]
25
26         daily = data["Daily"]
27         gender = data["Gender"]
28         embellishment = data["Embellishment"]
29
30         return image, daily, gender, embellishment
```



Albumentations

- Blur Transform
- Resize Transform
- Dropout Transform
- Geometric Transform
- Color Transform

Sub-Task 1 Code Review

Training Info

<https://github.com/huggingface/pytorch-image-models>

```

subtask1 - model.py

1 class EVATiny224(nn.Module):
2     def __init__(self):
3         super(EVATiny224, self).__init__()
4
5         self.backbone = timm.create_model("eva02_tiny_patch14_224.mim_in22k", pretrained=True)
6         self.daily = nn.Linear(192, 6)
7         self.gender = nn.Linear(192, 5)
8         self.embellishment = nn.Linear(192, 3)
9
10    def forward(self, x):
11        x = self.backbone(x)
12        daily = self.daily(x)
13        gender = self.gender(x)
14        embellishment = self.embellishment(x)
15
16        return daily, gender, embellishment

```

convit_tiny.fb_in1k	224	73.126	26.874	91.69	8.31	5.71	0.875	bicubic
coat_lite_tiny.in1k	224	77.548	22.452	93.924	6.076	5.72	0.9	bicubic
vit_tiny_patch16_224.augreg_in21k_ft_in1k	224	75.452	24.548	92.854	7.146	5.72	0.9	bicubic
deit_tiny_patch16_224.fb_in1k	224	72.196	27.804	91.1	8.9	5.72	0.9	bicubic
eva02_tiny_patch14_336.mim_in22k_ft_in1k	336	80.672	19.328	95.526	4.474	5.76	1	bicubic
vit_tiny_patch16_384.augreg_in21k_ft_in1k	384	78.446	21.554	94.552	5.448	5.79	1	bicubic
mobilenetv2_120d.ra_in1k	224	77.298	22.702	93.516	6.484	5.83	0.875	bicubic
deit_tiny_distilled_patch16_224.fb_in1k	224	74.54	25.46	91.894	8.106	5.91	0.9	bicubic
hgnetv2_b0.ssld_stage2_ft_in1k	288	78.598	21.402	94.38	5.62	6	1	bicubic
hgnetv2_b0.ssld_stage1_in22k_in1k	288	78.034	21.966	94.248	5.752	6	1	bicubic
hgnetv2_b0.ssld_stage2_ft_in1k	224	77.342	22.658	93.786	6.214	6	0.965	bicubic
hgnetv2_b0.ssld_stage1_in22k_in1k	224	76.844	23.156	93.612	6.388	6	0.965	bicubic
regnety_006.pycls_in1k	224	75.262	24.738	92.526	7.474	6.06	0.875	bicubic
tf_efficientnet_lite2.in1k	260	77.472	22.528	93.744	6.256	6.09	0.89	bicubic
mobilenetv2_140.ra_in1k	224	76.528	23.472	93.008	6.992	6.11	0.875	bicubic
ghostnetv2_100.in1k	224	75.176	24.824	92.334	7.666	6.16	0.875	bicubic
efficientformerv2_s1.snap_dist_in1k	224	79.684	20.316	94.69	5.31	6.19	0.95	bicubic
tinynet_a.in1k	192	77.664	22.336	93.532	6.468	6.19	0.875	bicubic
regnety_006.pycls_in1k	224	73.884	26.116	91.666	8.334	6.2	0.875	bicubic
regnety_008.pycls_in1k	224	76.334	23.666	93.06	6.94	6.26	0.875	bicubic

Model Size: $25 \text{ MB} = 25 \times 1024 \times 1024 = 26,214,400 \text{ bytes}$

Parameter Size: $n = \frac{26,214,400}{4} \approx 6,553,600$

Sub-Task 1 Code Review

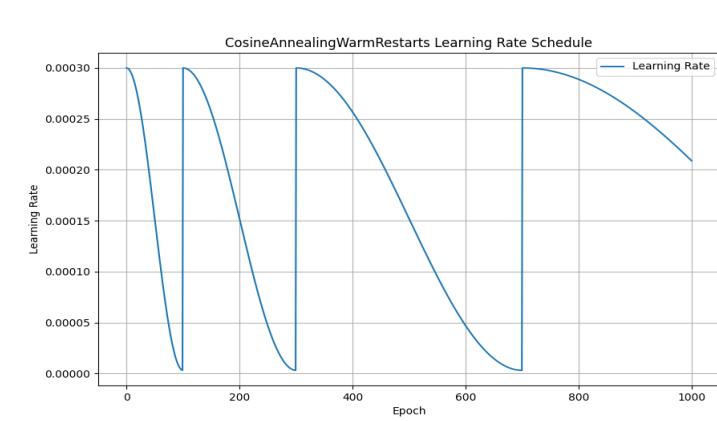
Training Info

subtask1 - main.py

```
1 model = EVATiny224()
2 scaler = torch.cuda.amp.GradScaler()
3 criterion = nn.CrossEntropyLoss(label_smoothing=0.1)
4 optimizer = AdamW(model.parameters(), lr=CFG.LEARNING_RATE)
5 scheduler = CosineAnnealingWarmRestarts(optimizer, T_0=30, T_mult=2, eta_min=CFG.LEARNING_RATE / 100)
```

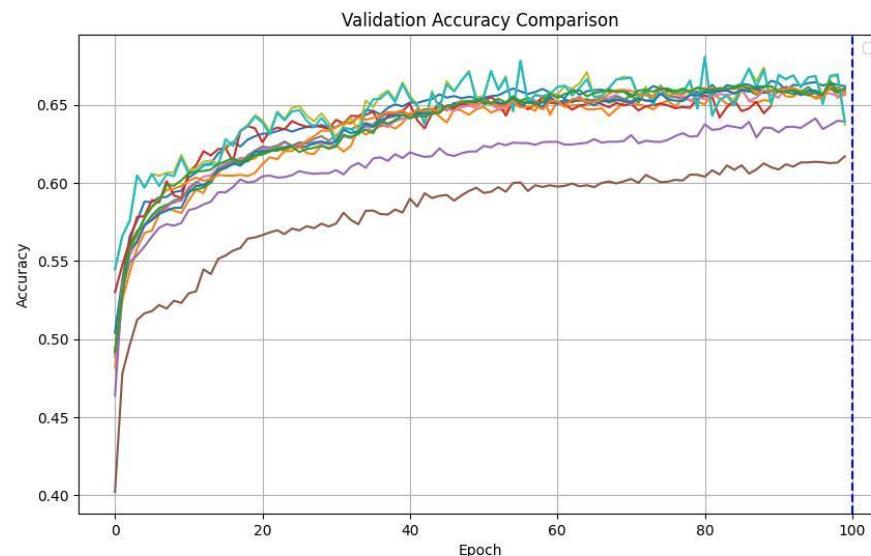
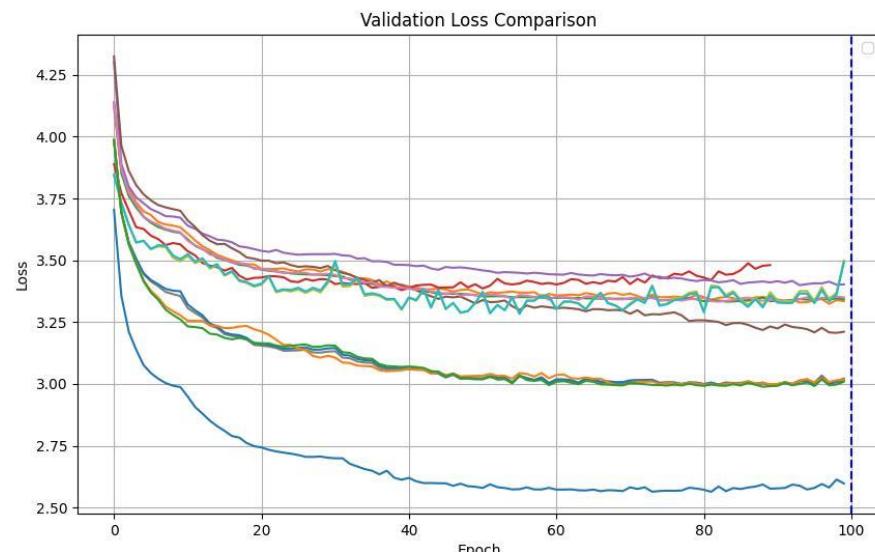
Train Info

- EVA02
- automatic mixed precision
- CrossEntropyLoss label smoothing: 0.1
- AdamW optimizer LR: 5e-6
- CosineAnnealingWramRestarts eta_min: 5e-8



Sub-Task 1 Code Review

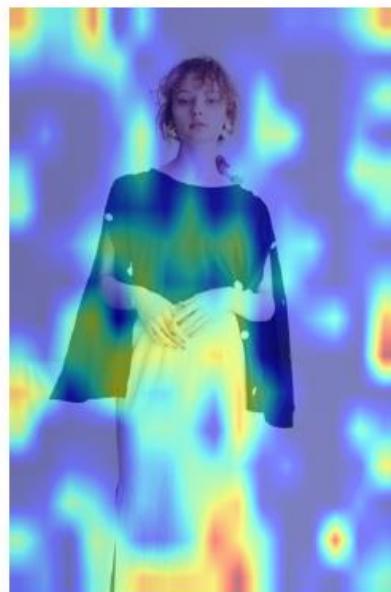
Experiments



- 다양한 Augmentation이 효과가 있었음.
- 이미지를 확률적으로 Crop하여 데이터에 다양성을 주는 테스트도 하였지만 효과가 크지 않았음. (Gender를 판단하는데, BBox 바깥 부분이 중요하다고 생각)
- Remove Background를 단일로 했을 때 유의미한 성능 향상이 있었음.
- Remove Background + Raw Data를 한번에 학습하는 테스트도 하였지만, 효과가 크지 않고 기존 Rembg 데이터만 가지고 학습했을 때 보다 성능이 좋지 않았음.
- Cutmix를 할 경우 학습시간이 크게 증가하지만 유의미한 성능 향상이 있지 않았음.

Sub-Task 1 Code Review

Challenges



- Gender의 Accuracy가 특히 낮게 나왔는데, 모델이 판단할 때 데이터의 외적인 부분을 보며 Gender class를 판단하는 경우가 많았음
- Gender Acc가 높아질 수록 데이터의 외적인 부분을 기준으로 판단하는 경향을 보임

Sub-Task 2

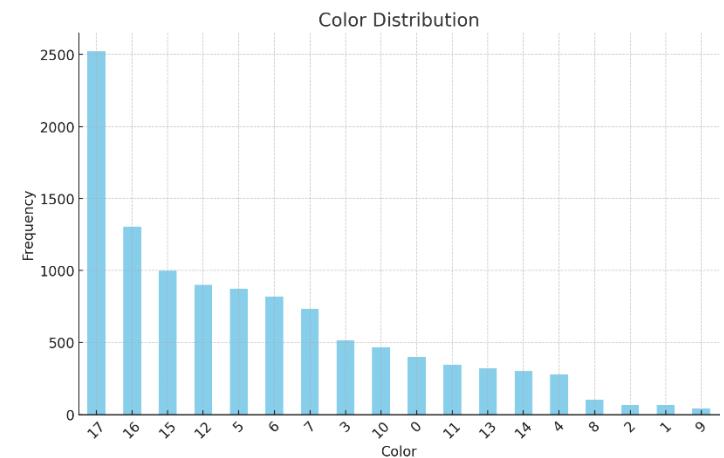
Sub-Task 2 Information

Imbalanced data

- Class의 데이터의 빈도 수가 차이가 큰 데이터 셋
- 데이터 수가 적은 Class에 대해서도 분류 정확도가 떨어지는 한계점을 개선

Fashion image color classification

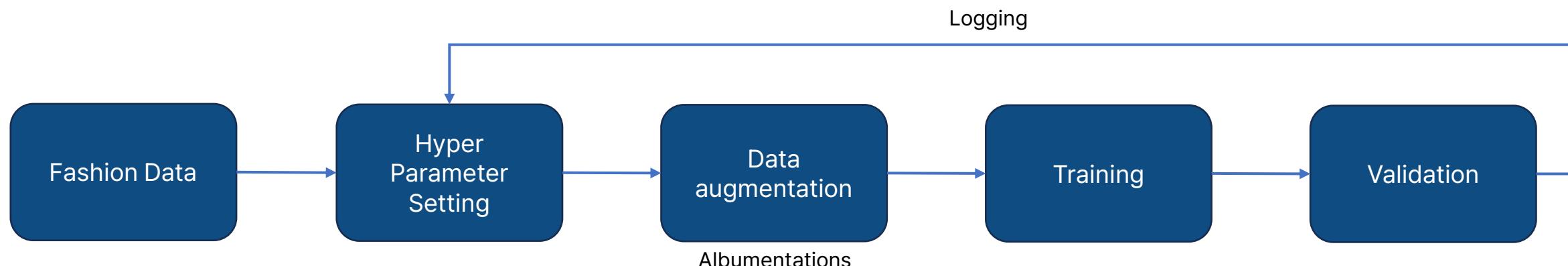
- 패션 이미지 내 불균형한 색상 카테고리 분류



Sub-task	Augmentation	Ensemble	Memory (과거 데이터셋)	Multimodal	외부 데이터셋	외부 사전학습 모델
1		불허	해당사항 없음	허용	허용	허용
2	허용	허용	해당사항 없음	불허	불허	불허
3	허용	허용	불허	허용	불허	허용
4		허용	해당사항 없음	허용	허용	허용

Sub-Task 2 Code Review

Training Process



Sub-Task 2 Code Review

Data Preprocessing

subtask2 - dataset.py

```

1 labels = df["Color"].values
2 class_counts = np.bincount(labels)
3 class_weights = 1.0 / class_counts
4 sample_weights = class_weights[labels]
5 logging.info(f"Class Weights: {class_weights}")
6 sampler = WeightedRandomSampler(weights=sample_weights, num_samples=len(sample_weights), replacement=True)

```

Sampler

- class weights : 각 클래스 빈도의 역수를 사용하여 가중치 부여
- Sample weight mapping : 클래스 가중치를 레이블에 매핑하여 샘플별 가중치 생성
- WeightedRandomSampler : 불균형한 클래스가 균등하게 학습되도록 가중치 기반 샘플링

⇒ Class가 비슷한 비율로 선택되도록 가중치를 조정, 모델 학습 시 특정 클래스에 편향되는 것을 방지

```

unique labels: tensor([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17],
device='cuda:0')
Counts: tensor([11, 11,  9,  7,  7,  6,  8,  6,  2,  8,  7,  4,  6,  9,  6,  7,  7,  7],
device='cuda:0')

```

Sub-Task 2 Code Review

Data Augmentation

<https://github.com/albumentations-team/albumentations>

```
subtask2 - augmentation.py

1 def train_transform():
2     transform = A.Compose(
3         [
4             A.Resize(CFG.H, CFG.W, interpolation=cv2.INTER_CUBIC),
5             A.ShiftScaleRotate(
6                 shift_limit=0.1,
7                 scale_limit=0.2,
8                 rotate_limit=20,
9                 border_mode=4,
10                p=0.3,
11            ),
12            A.CoarseDropout(
13                min_holes=1,
14                max_holes=6,
15                min_width=4,
16                min_height=4,
17                max_height=int(CFG.H * 0.1),
18                max_width=int(CFG.W * 0.1),
19                p=0.5,
20            ),
21            A.Normalize(),
22            ToTensorV2(),
23        ]
24    )
25    return transform
26
```



Albumentations

- Color를 맞추는 간단한 분류라고 판단
- 색상에 영향을 줄 수 있는 Augmentation 최소화
- Cutout & ShiftScaleRotate 만 적용

Sub-Task 2 Code Review

Training Info

<https://github.com/huggingface/pytorch-image-models>

```

subtask2 - model.py

1  class ConvNeXt(nn.Module):
2      def __init__(self):
3          super(ConvNeXt, self).__init__()
4
4          self.backbone = timm.create_model("convnextv2_femto", pretrained=False)
5          self.backbone.head.fc = nn.Linear(self.backbone.head.fc.in_features, 18)
6
7      def forward(self, x):
8          x = self.backbone(x)
9          return x
10
11
12  class EVATiny(nn.Module):
13      def __init__(self):
14          super(EVATiny, self).__init__()
15
16          self.backbone = timm.create_model("eva02_tiny_patch14_224.mim_in22k", pretrained=False)
17          self.clf = nn.Linear(192, 18)
18
19      def forward(self, x):
20          x = self.backbone(x)
21          x = self.clf(x)
22          return x
23
24
25  class Mobile(nn.Module):
26      def __init__(self):
27          super(Mobile, self).__init__()
28
29          self.backbone = timm.create_model("mobilenetv4_conv_small", pretrained=False)
30          self.backbone.classifier = nn.Linear(self.backbone.classifier.in_features, 18)
31
32      def forward(self, x):
33          x = self.backbone(x)
34          return x
35

```

Model Size: $25 \text{ MB} = 25 \times 1024 \times 1024 = 26,214,400 \text{ bytes}$

$$\text{Parameter Size: } n = \frac{26,214,400}{4} \approx 6,553,600$$

Experiments

- Pre-Train이 허용되지 않는 환경
- 다양한 모델 테스트(ConvNeXt, ViT, MobilNet, etc...)

⇒ Pre-Train 없이도 ViT 기반 모델이 성능 우수

Sub-Task 2 Code Review

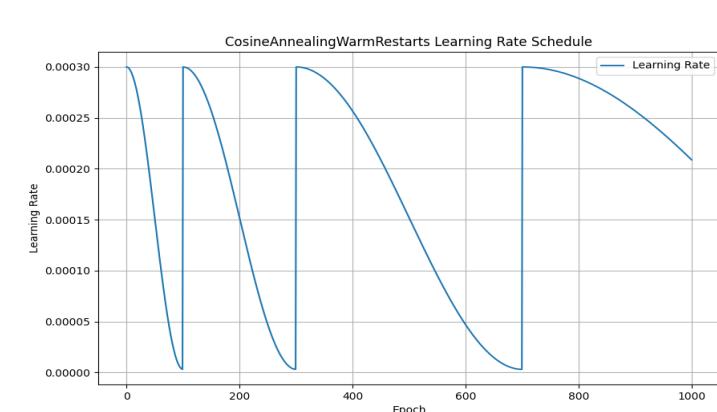
Training Info

subtask2 - main.py

```
1 model = EVATiny()
2 scaler = torch.cuda.amp.GradScaler()
3 criterion = nn.CrossEntropyLoss(label_smoothing=0.1)
4 optimizer = AdamW(model.parameters(), lr=CFG.LEARNING_RATE)
5 scheduler = CosineAnnealingWarmRestarts(optimizer, T_0=20, T_mult=2, eta_min=1e-5)
```

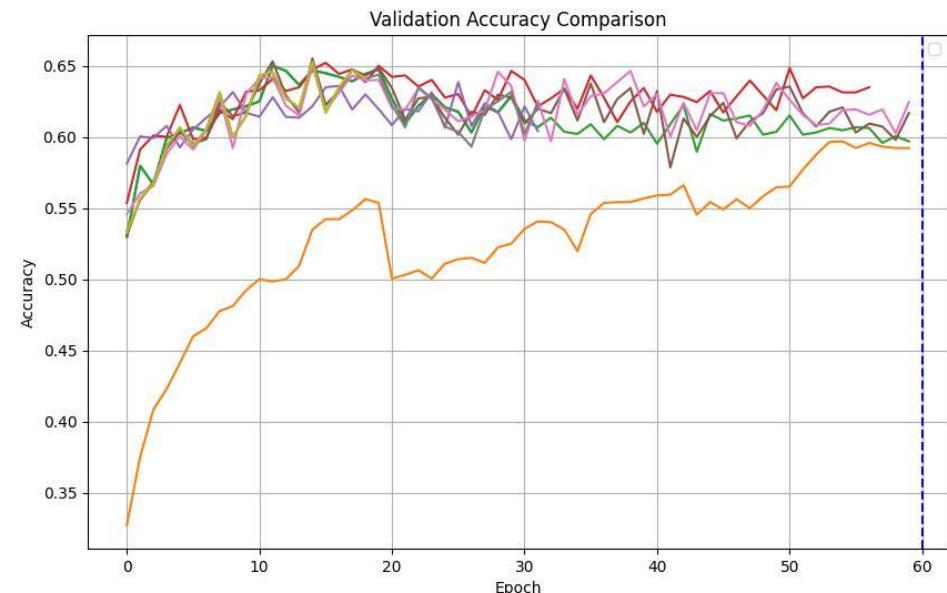
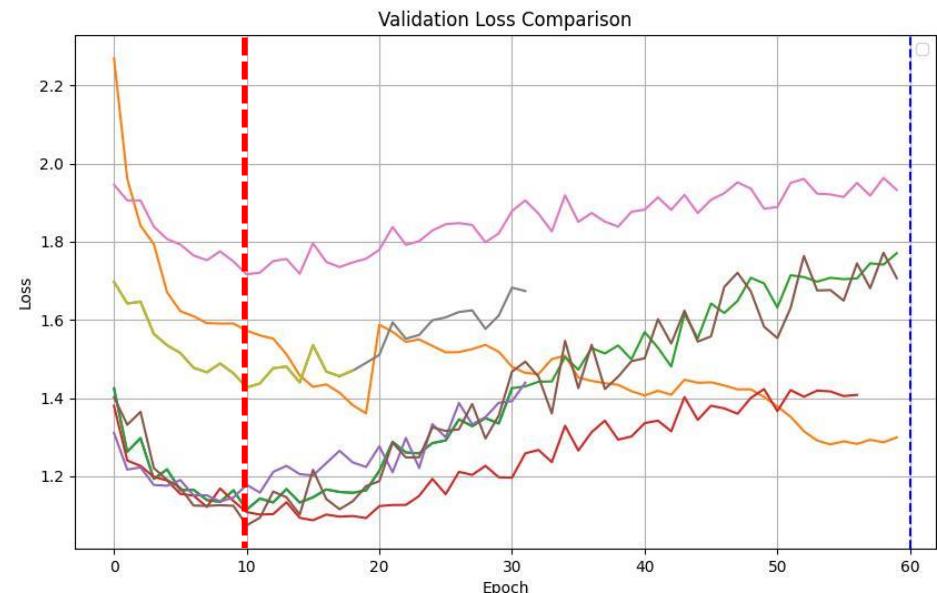
Train Info

- EVA02
- automatic mixed precision
- CrossEntropyLoss label smoothing: 0.1
- AdamW optimizer LR: 1e-4
- CosineAnnealingWramRestarts eta_min: 1e-5



Sub-Task 2 Code Review

Experiments



- 이미지를 확률적으로 Crop하여 해당 이미지의 색을 맞추는 테스트도 하였지만 효과가 크지 않았음.
- Remove Background가 Sub-Task1에서는 성능향상이 있었지만, 2에서는 있지 않았음. (class에 검정색 class가 있었기 때문이라고 판단)
- Remove Background + Raw Data를 한번에 학습하는 테스트도 하였지만, 효과가 크지 않았음.
- CutMix + Mixup을 적용해봤지만, 성능 향상에 좋지 않았음

감사합니다