

# DATA CREATOR CAMP

2024 데이터 크리에이터 캠프

대학부 팀명 :

SIGCOM(시그컴)





충북대학교 정보통신공학부  
**TEAM SIGCOM**



**이규하**

팀장, 21학번



**곽도현**

팀원, 21학번



**강찬솔**

팀원, 23학번



**허강민**

팀원, 21학번



충북대학교 정보통신공학부  
**TEAM SIGCOM**

우리는 데이터와 인공지능 분야에 대한 깊은 관심과 열정을 가지고 있으며, 이번 캠프를 통해 실력을 더욱 키우고자 한다. 인공지능과 데이터 분석은 현재 가장 혁신적이고 중요한 분야로, 다양한 산업에서 큰 변화를 이끌고 있다. 우리 동아리도 이러한 변화의 중심에서 사회에 긍정적인 영향을 미치고자 하며, 이를 위해서는 전문적인 지식뿐만 아니라 관련 실무 경험이 반드시 필요하다고 생각한다.



충북대학교 정보통신공학부  
**TEAM SIGCOM**

특히, 이번 캠프는 단순히 강의만 듣고 끝나는 것이 아니라, 실제 과제와 멘토링을 통해 실전 능력을 쌓을 기회를 제공한다고 하여 저희에게 큰 매력으로 다가왔다. 실무와 가까운 환경에서 전문가들의 조언과 함께 문제를 해결하는 과정을 통해, 저희의 지식과 기술을 더욱 발전시키고 이를 실제 현장에서 활용할 수 있는 능력으로 키우고자 한다.



충북대학교 정보통신공학부  
**TEAM SIGCOM**

또한, 다양한 시각을 가진 팀원들과 협업하며 서로의 이해력을 넓히고 창의적인 문제 해결 능력을 기를 수 있을 것으로 기대한다. 이러한 경험은 팀원 개개인의 경력에도 큰 도움이 될 것이며, 더 나아가 데이터와 인공지능 분야에서 의미 있는 성과를 이루는 중요한 밑거름이 될 것이라 생각하여 이번 2024 데이터 크리에이터 캠프에 참가하게 되었다.



## 문제. 인공지능은 사람의 마음을 이해할 수 있을까?

### ■ 연도별 패션 스타일 이미지 및 추천 데이터

2022

#의류 이미지 # 패션 # 포즈 # 쉐이프리스의류

NEW 연도별 패션 선호도 파악 및 추천 데이터

분야 영상이미지 유형 이미지

구축년도 : 2022    갱신년월 : 2023-10    조회수 : 4,960    다운로드 : 379    용량 : 333.22 GB

다운로드 샘플 데이터 ?

### ■ 데이터셋 외 코드 제공되지 않음



## Mission 1. 패션 스타일 이미지 분류

- 1-1. 주어진 이미지 데이터의 파일 명은 아래와 같은 형식이다. “{W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}.jpg”이에 기반하여 “이미지ID”  
수기준으로 “성별 & 스타일”통계치를 아래 표 형식으로 기입한다.

※ Training, Validation 데이터에 대해서 각각 통계표를 작성한다.

성별	스타일	이미지 수
여성	feminine	
	classic	
	minimal	
	popart	
	...	
남성	ivy	
	mods	
	hippie	
	bold	
	...	

W_96469_60_minimal_W.jpg
W_96472_19_lounge_W.jpg
W_96487_19_lounge_W.jpg
W_96507_19_lounge_W.jpg
W_96514_60_minimal_W.jpg
W_96538_60_minimal_W.jpg
W_96574_19_normcore_W.jpg
W_96596_60_popart_W.jpg
W_96599_60_popart_W.jpg
W_96600_60_minimal_W.jpg
W_96606_60_popart_W.jpg
W_96607_60_popart_W.jpg
W_96612_60_popart_W.jpg
W_96616_60_minimal_W.jpg
W_96617_60_minimal_W.jpg
W_96619_60_minimal_W.jpg
W_96625_60_minimal_W.jpg
W_96626_60_minimal_W.jpg
W_96632_60_minimal_W.jpg
W_96634_60_minimal_W.jpg
W_96637_60_minimal_W.jpg
W_96643_60_minimal_W.jpg





# Overall Process to Solving the Problem

목표: 주어진 이미지 데이터(Training 및 Validation)를 성별 및 스타일별로 분석한 통계표 작성

데이터 형식: 이미지 데이터들의 파일 명이 {W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}.jpg 로 구성.

조건: Training, Validation 데이터에 대해서 각각 통계표를 작성

## [Dataset 구성 확인]

Dataset 내 폴더의 트리 구조, 이미지 및 라벨링 파일 명, 이미지 및 라벨링 총 개수 확인

## [성별 및 스타일 통계 결과]

Training 이미지 데이터에 대한 통계표 작성

Validation 이미지 데이터에 대한 통계표 작성

통계표를 시각화 그래프로 표현



# [Dataset 구성 확인]

Dataset 내 폴더의 트리 구조, 이미지 및 라벨링 파일 명, 이미지 및 라벨링 총 개수 확인

## Dataset 내 폴더의 트리 구조 확인

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치하세요! https://aka.ms/PSWindows

개인 및 시스템 프로필을 로드하는 데 520ms가 걸렸습니다.
[GYUHA LEE@KONGSUNI_LAPTOP] D:/학교/2학년/2학기/대외활동 및 공모전/2. 진행 중/2024 데이터 크리에이터 캠프/문제 데이터셋/dataset
$ ls

디렉터리: D:\학교\2학년\2학기\대외활동 및 공모전\2. 진행 중\2024 데이터 크리에이터 캠프\문제 데이터셋\dataset

Mode                LastWriteTime       Length Name
----                -----          ---- 
d----
```

Dataset 폴더 내  
training\_image  
training\_label  
validation\_image  
validation\_label



# [Dataset 구성 확인]

Dataset 내 폴더의 트리 구조, 이미지 및 라벨링 파일 명, 이미지 및 라벨링 총 개수 확인

## training\_image 폴더 내 파일 명 확인 (일부)

```
└─(GYUHA LEE@KONGSUNI_LAPTOP)-[D:/학교/2학년/2학기/대외활동 및 공모전/2. 진행 중/2024 데이터 크리에이터 캠프/문제 데이터셋/dataset]
  └─$ cd training_image
  └─(GYUHA LEE@KONGSUNI_LAPTOP)-[D:/학교/2학년/2학기/대외활동 및 공모전/2. 진행 중/2024 데이터 크리에이터 캠프/문제 데이터셋/dataset/training_image]
    └─$ ls
```

디렉터리: D:\학교\2학년\2학기\대외활동 및 공모전\2. 진행 중\2024 데이터 크리에이터 캠프\문제 데이터셋\dataset\train  
ing\_image

{W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}.jpg

Mode	LastWriteTime	Length	Name
-a---	2024-07-04 오후 3:00	7586397	T_00253_60_popart_W.jpg
-a---	2024-07-04 오후 3:00	4948202	T_00456_10_sportivecasual_M.jpg
-a---	2024-07-04 오후 3:00	4897800	T_00588_10_sportivecasual_M.jpg
-a---	2024-07-04 오후 3:00	3476217	T_00770_60_minimal_W.jpg
-a---	2024-07-04 오후 3:00	4338203	T_00893_90_hiphop_W.jpg
-a---	2024-07-04 오후 3:00	4529143	T_01322_19_normcore_M.jpg
-a---	2024-07-04 오후 3:00	3097712	T_01883_10_sportivecasual_M.jpg
-a---	2024-07-04 오후 3:00	4647067	T_02527_10_sportivecasual_M.jpg
-a---	2024-07-04 오후 3:00	1403261	T_02558_19_normcore_M.jpg
-a---	2024-07-04 오후 3:00	3220547	T_02820_10_sportivecasual_W.jpg
-a---	2024-07-04 오후 3:00	2964781	T_02958_19_normcore_M.jpg
-a---	2024-07-04 오후 3:00	2294315	T_03772_90_hiphop_M.jpg
-a---	2024-07-04 오후 3:00	2685501	T_03876_90_hiphop_M.jpg
-a---	2024-07-04 오후 3:00	3232817	T_04310_19_normcore_W.jpg
-a---	2024-07-04 오후 3:00	5950950	T_04656_19_genderless_W.jpg
-a---	2024-07-04 오후 3:00	4018877	T_05088_19_normcore_W.jpg
-a---	2024-07-04 오후 3:00	2124122	T_05224_19_normcore_W.jpg
-a---	2024-07-04 오후 3:00	2671572	T_05759_19_lounge_W.jpg
-a---	2024-07-04 오후 3:00	3648893	T_06076_60_mods_M.jpg

4,070개 항목



# [Dataset 구성 확인]

Dataset 내 폴더의 트리 구조, 이미지 및 라벨링 파일 명, 이미지 및 라벨링 총 개수 확인

training\_label 폴더 내 파일 명 확인 (일부)

```
[GYUHA LEE@KONGSUNI_LAPTOP] D:/학교/2학년/2학기/대외활동 및 공모전/2. 진행 중/2024 데이터 크리에이터 캠프/문제 데이터셋/dataset/training_label
$ ls
```

디렉터리: D:\학교\2학년\2학기\대외활동 및 공모전\2. 진행 중\2024 데이터 크리에이터 캠프\문제 데이터셋\dataset\train  
ing\_label

Mode	LastWriteTime	Length	Name
-a---	2024-07-04 오전 11:27	753	T_00001_50_ivy_M_041317.json
-a---	2024-07-04 오전 11:27	755	T_00001_50_ivy_M_197195.json
-a---	2024-07-04 오전 11:27	756	T_00001_50_ivy_M_208803.json
-a---	2024-07-04 오전 11:27	753	T_00001_50_ivy_M_211499.json
-a---	2024-07-04 오전 11:27	754	T_00001_50_ivy_M_227846.json
-a---	2024-07-04 오전 11:27	757	T_00001_50_ivy_M_234404.json
-a---	2024-07-04 오전 11:27	762	T_00002_90_hiphop_M_209187.json
-a---	2024-07-04 오전 11:27	761	T_00002_90_hiphop_M_212545.json
-a---	2024-07-04 오전 11:27	765	T_00002_90_hiphop_M_218632.json
-a---	2024-07-04 오전 11:27	771	T_00003_19_normcore_M_201566.json
-a---	2024-07-04 오전 11:27	761	T_00004_90_hiphop_M_071538.json
-a---	2024-07-04 오전 11:27	763	T_00004_90_hiphop_M_203010.json
-a---	2024-07-04 오전 11:27	764	T_00004_90_hiphop_M_206164.json
-a---	2024-07-04 오전 11:27	754	T_00005_80_bold_M_085175.json
-a---	2024-07-04 오전 11:27	757	T_00005_80_bold_M_198422.json
-a---	2024-07-04 오전 11:27	755	T_00005_80_bold_M_204489.json
-a---	2024-07-04 오전 11:27	754	T_00006_50_ivy_M_027160.json
-a---	2024-07-04 오전 11:27	752	T_00006_50_ivy_M_041325.json
-a---	2024-07-04 오전 11:27	755	T_00006_50_ivy_M_220260.json
-a---	2024-07-04 오전 11:27	755	T_00006_50_ivy_M_223769.json

{W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}\_{설문ID}.json

211,346개 항목



# [Dataset 구성 확인]

Dataset 내 폴더의 트리 구조, 이미지 및 라벨링 파일 명, 이미지 및 라벨링 총 개수 확인

## Validation\_image 폴더 내 파일 명 확인 (일부)

```
└─(GYUHA LEE@KONGSUNI_LAPTOP)-[D:/학교/2학년/2학기/대외활동 및 공모전/2. 진행 중/2024 데이터 크리에이터 캠프/문제 데이터셋/dataset/validation_image]
$ ls
```

디렉터리: D:\학교\2학년\2학기\대외활동 및 공모전\2. 진행 중\2024 데이터 크리에이터 캠프\문제 데이터셋\dataset\validation\_image

Mode	LastWriteTime	Length	Name
-a---	2024-07-04	오후 3:00	7586397 T_00253_60_popart_W.jpg
-a---	2024-07-04	오후 3:00	4948202 T_00456_10_sportivecasual_M.jpg
-a---	2024-07-04	오후 3:00	6174219 T_01123_90_hiphop_M.jpg
-a---	2024-07-04	오후 3:00	2367189 T_01514_50_ivy_M.jpg
-a---	2024-07-04	오후 3:00	2816677 T_06910_50_classic_W.jpg
-a---	2024-07-04	오후 3:00	3363539 T_07990_60_mods_M.jpg
-a---	2024-07-04	오후 3:00	4682325 T_14538_00_cityglam_W.jpg
-a---	2024-07-04	오후 3:00	4300569 T_21986_70_hippie_M.jpg
-a---	2024-07-04	오후 3:00	4528714 T_21988_70_hippie_M.jpg
-a---	2024-07-04	오후 3:00	4559388 T_21992_70_hippie_M.jpg
-a---	2024-07-04	오후 3:00	1589110 W_00004_50_ivy_M.jpg
-a---	2024-07-04	오후 3:00	2406922 W_00012_50_ivy_M.jpg
-a---	2024-07-04	오후 3:00	5747707 W_00028_50_ivy_M.jpg
-a---	2024-07-04	오후 3:00	8030404 W_00033_60_mods_M.jpg
-a---	2024-07-04	오후 3:00	3628666 W_00073_50_ivy_M.jpg
-a---	2024-07-04	오후 3:00	2872704 W_00103_70_hippie_M.jpg
-a---	2024-07-04	오후 3:00	3808101 W_00117_19_normcore_M.jpg
-a---	2024-07-04	오후 3:00	2959216 W_00152_50_feminine_W.jpg
-a---	2024-07-04	오후 3:00	1734361 W_00161_60_space_W.jpg
-a---	2024-07-04	오후 3:00	4372735 W_00191_10_sportivecasual_W.jpg
-a---	2024-07-04	오후 3:00	5604824 W_00299_50_feminine_W.jpg
-a---	2024-07-04	오후 3:00	3677570 W_00351_70_hippie_W.jpg
-a---	2024-07-04	오후 3:00	4612842 W_00359_90_grunge_W.jpg
-a---	2024-07-04	오후 3:00	1579995 W_00366_60_minimal_W.jpg
-a---	2024-07-04	오후 3:00	2894425 W_00402_50_ivy_M.jpg

{W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}.jpg

951개 항목



# [Dataset 구성 확인]

Dataset 내 폴더의 트리 구조, 이미지 및 라벨링 파일 명, 이미지 및 라벨링 총 개수 확인

Validation\_label 폴더 내 파일 명 확인 (일부)

```
(GYUHA LEE@KONGSUNI_LAPTOP)-[D:/학교/2학년/2학기/대외활동 및 공모전/2. 진행 중/2024 데이터 크리에이터 캠프/문제 데이터셋/dataset/validation_label]
$ ls

디렉터리: D:\학교\2학년\2학기\대외활동 및 공모전\2. 진행 중\2024 데이터 크리에이터 캠프\문제 데이터셋\dataset\validation_label

Mode LastWriteTime Length Name
---- --:--:-- ----:--:--:-- {W/T}_{이미지ID}_{시대별}_{스타일별}_{성별}_{설문ID}.json
-a--- 2024-07-04 오전 11:27 768 T_00017_19_normcore_M_148650.json
-a--- 2024-07-04 오전 11:27 789 T_00031_10_sportivecasual_M_167884.json
-a--- 2024-07-04 오전 11:27 787 T_00031_10_sportivecasual_M_208711.json
-a--- 2024-07-04 오전 11:27 771 T_00035_19_normcore_M_214128.json
-a--- 2024-07-04 오전 11:27 788 T_00037_10_sportivecasual_M_167876.json
-a--- 2024-07-04 오전 11:27 786 T_00037_10_sportivecasual_M_201817.json
-a--- 2024-07-04 오전 11:27 786 T_00042_10_sportivecasual_M_030340.json
-a--- 2024-07-04 오전 11:27 788 T_00042_10_sportivecasual_M_208863.json
-a--- 2024-07-04 오전 11:27 768 T_00054_19_normcore_M_214044.json
-a--- 2024-07-04 오전 11:27 785 T_00063_10_sportivecasual_M_005860.json
-a--- 2024-07-04 오전 11:27 787 T_00063_10_sportivecasual_M_014467.json
-a--- 2024-07-04 오전 11:27 787 T_00066_10_sportivecasual_M_027233.json
-a--- 2024-07-04 오전 11:27 787 T_00066_10_sportivecasual_M_221066.json
-a--- 2024-07-04 오전 11:27 785 T_00070_10_sportivecasual_M_070306.json
-a--- 2024-07-04 오전 11:27 787 T_00070_10_sportivecasual_M_229279.json
-a--- 2024-07-04 오전 11:27 769 T_00074_19_normcore_M_204757.json
-a--- 2024-07-04 오전 11:27 783 T_00081_10_sportivecasual_M_009810.json
-a--- 2024-07-04 오전 11:27 789 T_00081_10_sportivecasual_M_235382.json
-a--- 2024-07-04 오전 11:27 786 T_00088_10_sportivecasual_M_196896.json
-a--- 2024-07-04 오전 11:27 788 T_00088_10_sportivecasual_M_229296.json
-a--- 2024-07-04 오전 11:27 769 T_00090_19_normcore_M_204959.json
```

{W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}\_{설문ID}.json

36,383개 항목



# [성별 및 스타일 통계 결과]

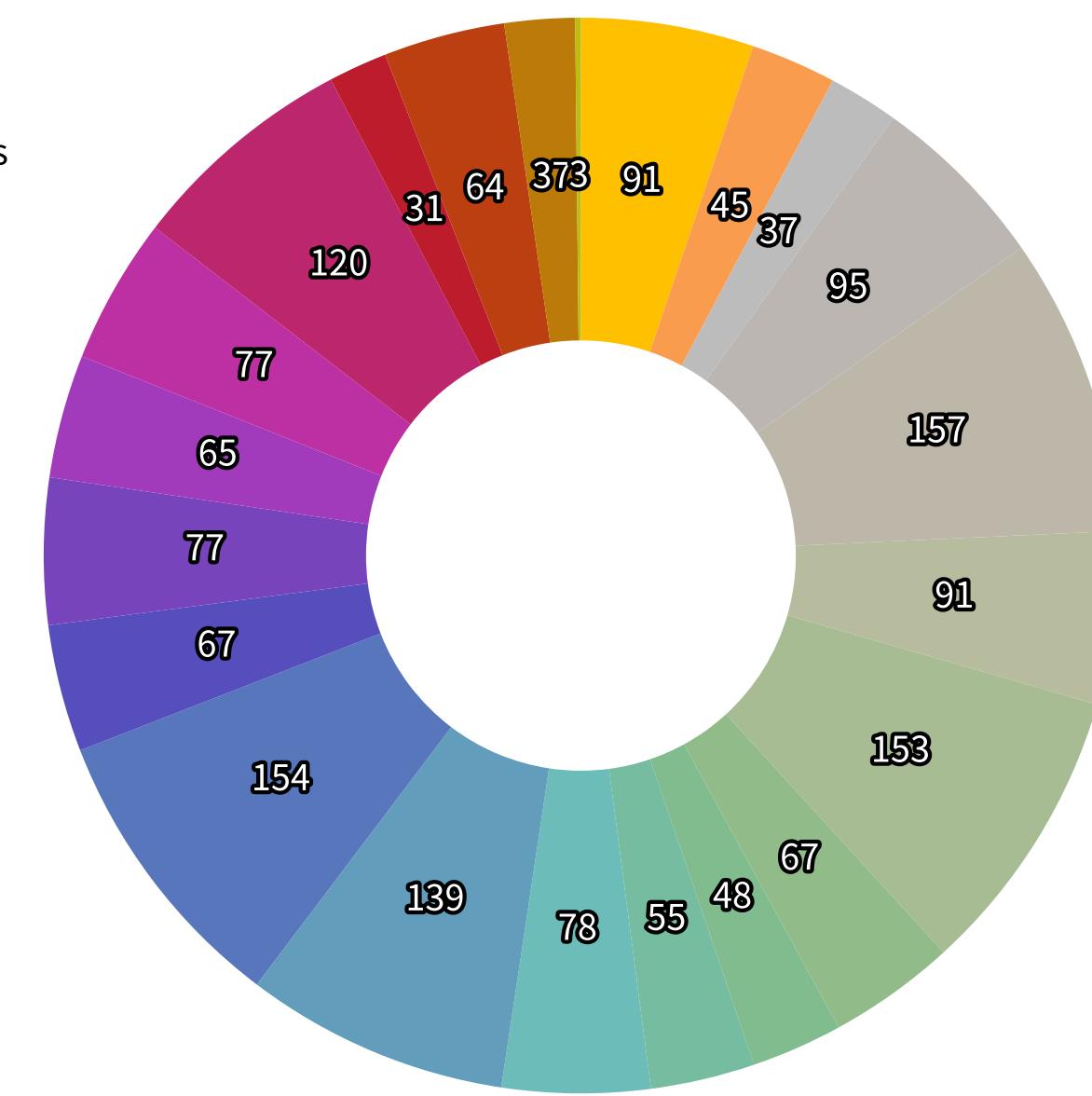
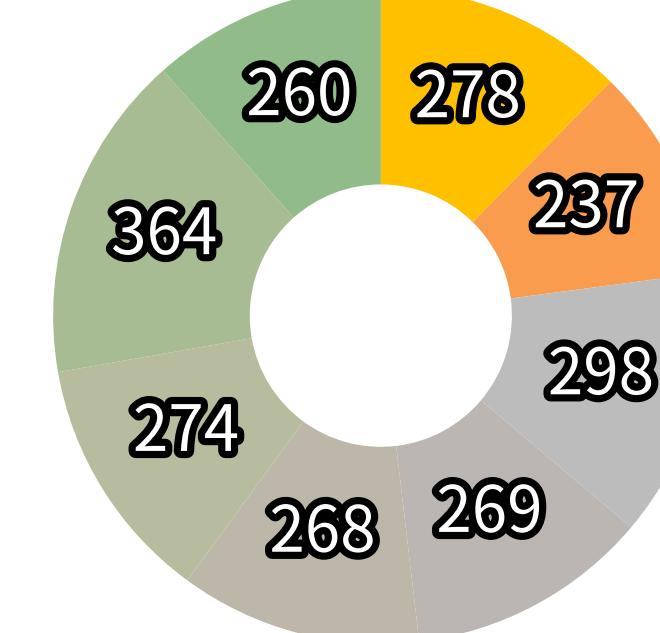
Training 이미지 데이터에 대한 통계표 작성

	성별	스타일	이미지 수
0	남성	metrosexual	278
1	남성	ivy	237
2	남성	sportivecasual	298
3	남성	mods	269
4	남성	bold	268
5	남성	hiphop	274
6	남성	normcore	364
7	남성	hippie	260
8	여성	kitsch	91
9	여성	lounge	45
10	여성	disco	37
11	여성	bodyconscious	95
12	여성	sportivecasual	157
13	여성	hippie	91
14	여성	normcore	153
15	여성	athleisure	67
16	여성	hiphop	48
17	여성	lingerie	55
18	여성	oriental	78
19	여성	minimal	139
20	여성	feminine	154
21	여성	cityglam	67
22	여성	classic	77
23	여성	punk	65
24	여성	genderless	77
25	여성	powersuit	120
26	여성	grunge	31
27	여성	ecology	64
28	여성	space	37
29	여성	military	33
30	여성	popart	41



# [성별 및 스타일 통계 결과]

Training 이미지 데이터에 대한 통계표를 시각화 그래프로 표현





# [성별 및 스타일 통계 결과]

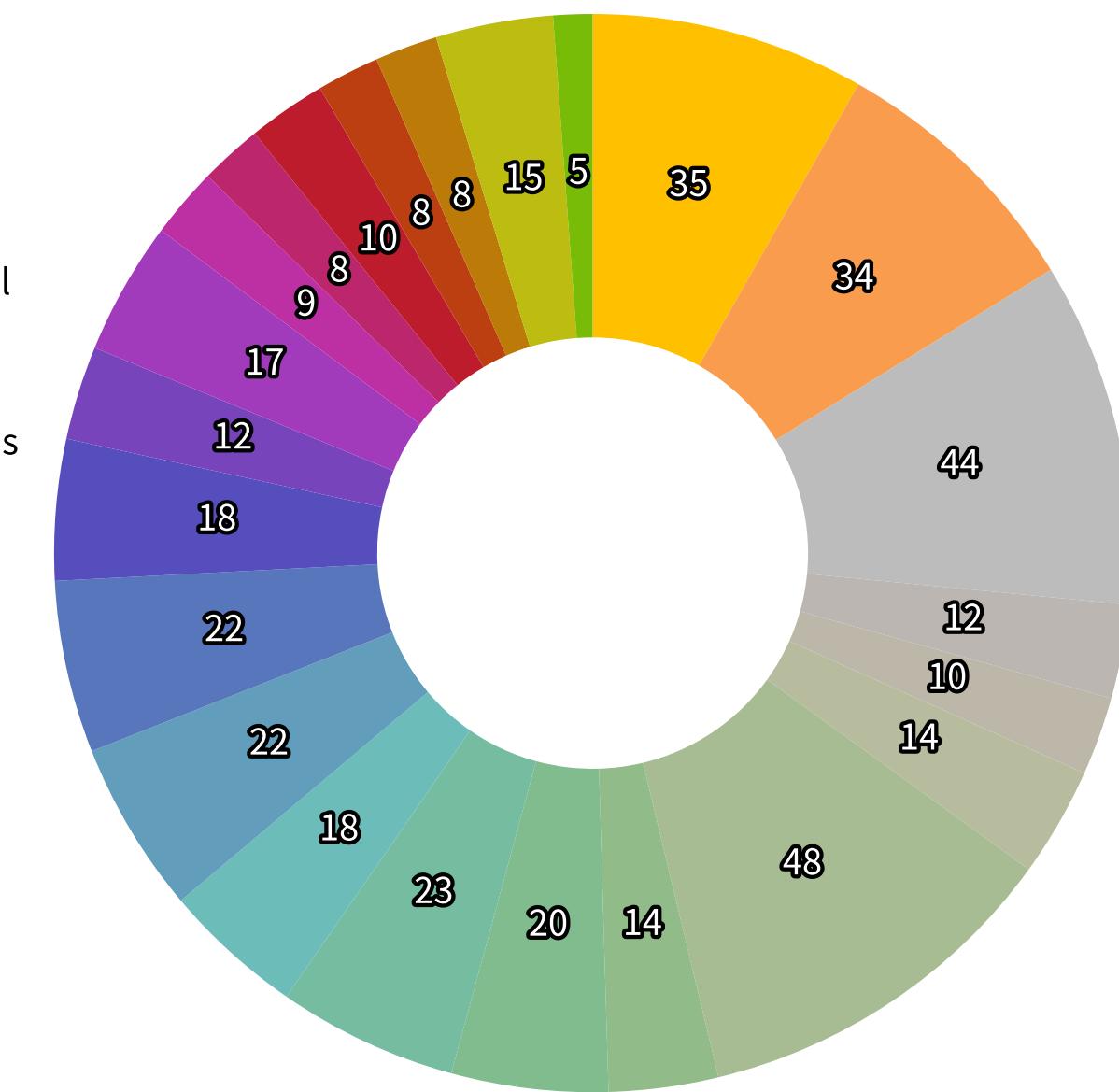
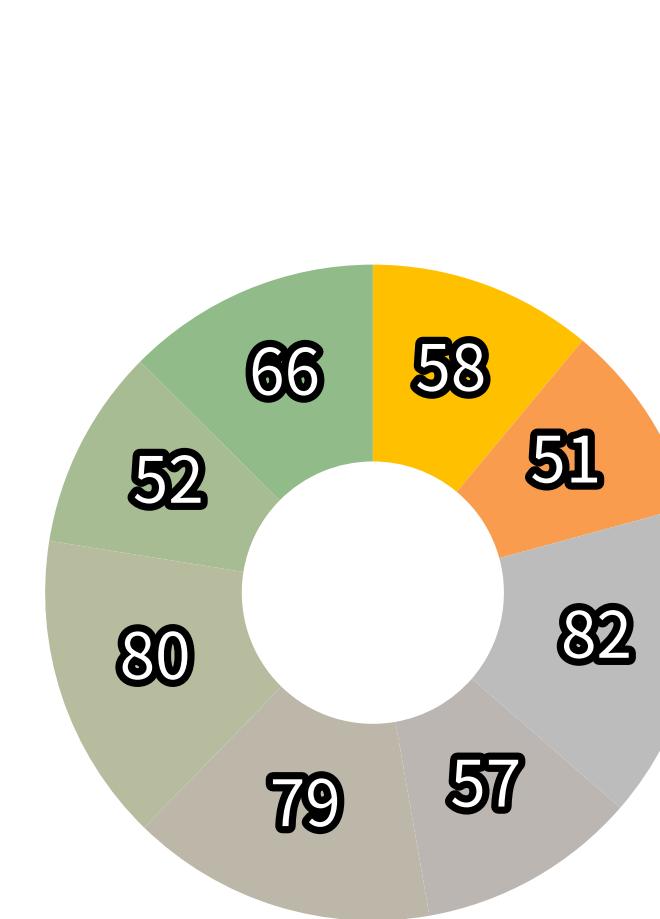
Validation 이미지 데이터에 대한 통계표 작성

	성별	스타일	이미지 수
0	남성	metrosexual	58
1	남성	normcore	51
2	남성	hippie	82
3	남성	bold	57
4	남성	ivy	79
5	남성	mods	80
6	남성	sportivecasual	52
7	남성	hiphop	66
8	여성	minimal	35
9	여성	powersuit	34
10	여성	feminine	44
11	여성	genderless	12
12	여성	disco	10
13	여성	hippie	14
14	여성	sportivecasual	48
15	여성	athleisure	14
16	여성	normcore	20
17	여성	bodyconscious	23
18	여성	cityglam	18
19	여성	kitsch	22
20	여성	classic	22
21	여성	oriental	18
22	여성	punk	12
23	여성	ecology	17
24	여성	military	9
25	여성	lounge	8
26	여성	grunge	10
27	여성	hiphop	8
28	여성	popart	8
29	여성	space	15
30	여성	lingerie	5



# [성별 및 스타일 통계 결과]

Validation 이미지 데이터에 대한 통계표를 시각화 그래프로 표현





## Mission 1. 패션 스타일 이미지 분류

- 1-2. ResNet-18를 활용하여 “성별 & 스타일” 단위로 클래스 분류를 수행하고 Validation 데이터에 대한 정확도를 제시한다.

- ResNet-18의 parameters는 무작위로 초기화하여 사용한다.  
(즉, pretrained weights는 사용할 수 없음)
- 성능을 높이기 위해 object detection, image cropping 등의 다양한 데이터 전처리 기법을 활용해도 무방하다.  
(데이터 전처리 단계에 한해서는 외부 라이브러리 활용 가능)



여성 에콜로지룩



여성 오리엔탈룩



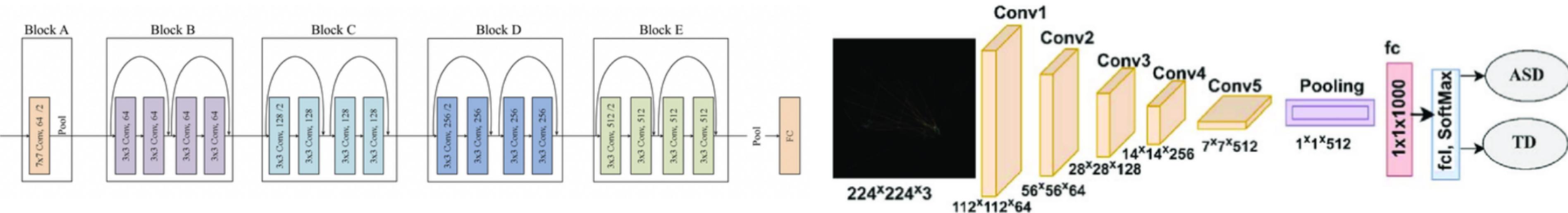
여성 놈코어룩



여성 클래식룩



# What is ResNet-18?



Convolutional Neural Network (CNN)의 한 종류로,  
깊은 신경망에서 발생하는 학습의 어려움을 해결하기 위해 설계된 모델이다.  
잔차 연결 (Residual Connection)을 통해 이전 층의 출력을 다음 층에 직접 전달함으로써  
정보 손실을 줄이고 학습 안정성을 높인다.  
총 ResNet-18은 명칭에서 알 수 있듯이, 18개의 신경망 층으로 구성되어 있다.

이번 문제에서는 Pre-trained Weights 없이 무에서 유의 방향으로 옛 스타일별 분류 기준을  
학습하는 것이 목표이며, 다음 슬라이드부터는 학습 모델을 도출하기까지의 과정 및 결과들을 풀어보겠다.



# Overall Process to Solving the Problem

목표: 주어진 이미지 데이터(Training 및 Validation)를 성별 및 스타일별로 분류하여 정확도를 측정.

데이터 형식: 이미지 데이터들의 파일 명이 {W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}.jpg 로 구성.

조건: 모델은 사전에 학습된 가중치를 사용하지 않으며, 모든 가중치는 처음부터 학습.

성능을 높이기 위해 object detection, image cropping 등의 다양한 데이터 전처리 기법을 활용해도 무방. (데이터 전처리 단계에 한해서는 외부 라이브러리 활용 가능)

## [데이터 전처리]

1. 더 나은 정확도 향상을 위해 이미지 데이터의 배경을 제거
2. 1-1에서 도출한 클래스 분류 통계표를 바탕으로 Custom Dataset Class를 정의

## [모델 학습]

1. Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습
2. Validation 이미지 파일을 매 Epoch마다 학습한 모델에 적용하여 분류 정확도 확인



# [데이터 전처리]

더 나은 정확도 향상을 위해 이미지 데이터의 배경을 제거

성능을 높이기 위해 object detection, image cropping 등의 다양한 외부 라이브러리를 활용 하여  
데이터 전처리 기법을 해도 된다는 문제 조건에 따라,  
우리는 원본 Dataset 이미지들의 배경 제거를 먼저 하기로 생각했다.

우리는 배경 제거를 위해 3가지의 라이브러리를 적용시켜보았다.

- DeepLabv3
- YOLOv5
- RemBG



# [데이터 전처리]

더 나은 정확도 향상을 위해 이미지 데이터의 배경을 제거

성능을 높이기 위해 object detection, image cropping 등의 다양한 외부 라이브러리를 활용하여 데이터 전처리 기법을 해도 된다는 문제 조건 및  
멘토링 시간을 통해 사진 내 옷이 아닌 다른 요소로 인해 정확도가 좌우될 수 있다는 조언을 통해  
우리는 원본 Dataset 이미지들의 배경 제거를 먼저 하기로 생각했다.

Google Colab 특성상 런타임 세션이 종료되면 파일들이 사라지는 특성이 있었고,  
분류 모델을 학습하는 부분과 배경 제거 부분을 같이 넣어버리게 되면  
매 Epoch 당 전체 사진 제거를 하게되어 학습 시간 증가는 당연,  
급격한 리소스 증가 및 Google Cloud 오류로 런타임이 죽어버리는 현상이 빈번하게 있었다.  
따라서 우리는 팀원 두 대의 Local PC에도 따로 환경을 구축한 후,  
독립적으로 배경 제거 데이터 전처리 작업을 수행하였다.



# [데이터 전처리]

더 나은 정확도 향상을 위해 이미지 데이터의 배경을 제거

가장 먼저 사용하였던 모델은 'DeepLabv3' 이었다.

이미지 분할(Semantic Segmentation)을 수행하기 위해 설계된 딥러닝 모델로,  
이미지 내의 각 픽셀을 특정 클래스(예: 배경, 사람, 물체 등)에 할당하여 이미지의 영역을 구분하고,  
원하는 객체만 남겨 그 외에는 검은색 마스킹으로 배경을 제거할 수 있었다.



하지만 실제 결과와 같이 옷 영역에 대한 헤손이 있고 배경 제거할 때 Local 리소스를 많이 잡아먹어서  
이는 적절하지 못하다는 판단을 내렸다.

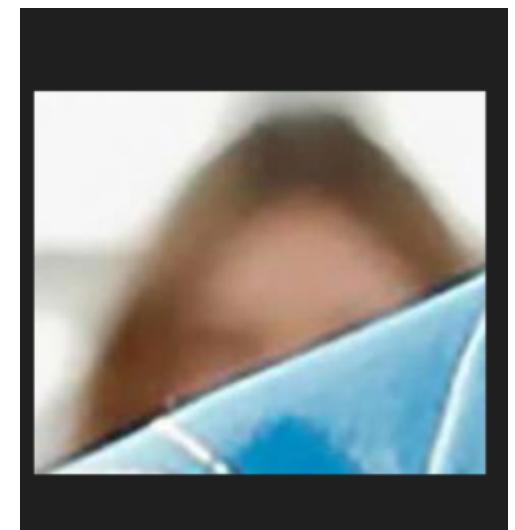
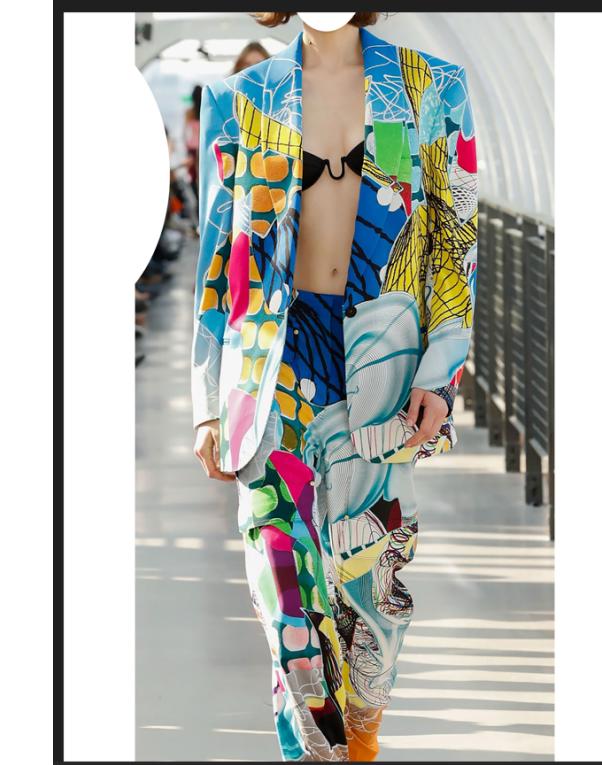


## [데이터 전처리]

더 나은 정확도 향상을 위해 이미지 데이터의 배경을 제거

그 다음 접근했던 모델은 'YOLOv5' 이었다.

실시간 객체 탐지(Object Detection)를 위해 이미지에서 다양한 객체를 빠르게 탐지하고 각 객체의 위치와 클래스 정보를 동시에 예측할 수 있도록 설계된 딥러닝 모델로, 이미지에서 특정 객체를 탐지하고 해당 객체를 분리하여 배경을 제거할 수 있었다.



하지만 실제 결과와 같이 배경을 완전히 지우지 못하였고, 가끔 탐지 자체를 잘못해서 적절하게 분리하지 못하는 경우도 있어 이 모델 또한 적절하지 못하다는 판단을 내렸다.

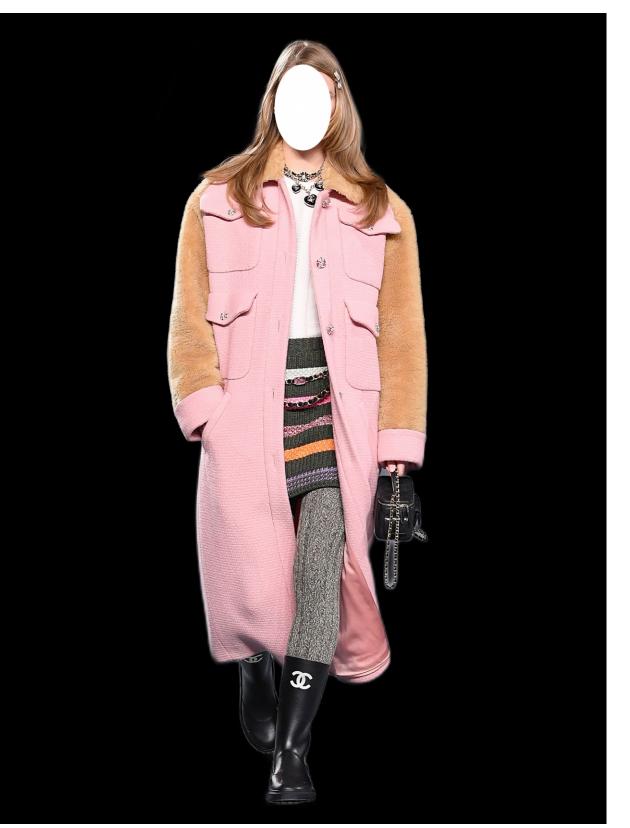


## [데이터 전처리]

더 나은 정확도 향상을 위해 이미지 데이터의 배경을 제거

마지막으로 접근했던 모델은 'RemBG' 이었다.

배경을 자동으로 제거하는 데 사용되는 Python 라이브러리이며, U^2-Net 딥러닝 모델을 이용하여 이미지에서 사람, 물체 등의 주요 객체를 구분하고, 해당 객체 외의 배경을 제거할 수 있었다.



사람이 옷을 입은 영역까지를 제외하고 나머지 영역이 검은색으로 잘 대체되어있는 점과,  
준수한 전처리 시간을 고려하여 이 모델로 최종 결정하였다.



## [데이터 전처리]

1-1에서 도출한 클래스 분류 통계표를 바탕으로 Custom Dataset Class를 정의

ResNet-18을 사용해 이미지 분류를 수행하려면,

모델이 이미지를 입력받았을 때 어떤 클래스에 속하는지 예측하기 위해서

클래스 정의(class definition)가 필요하다.

처음 시도하였을 때는 파일 명에 스타일이 나와있으니까 클래스를 나눠야 할 필요성을 인지하지 못해 파일을 바로 불러오는 식으로 바로 시도하였지만 결과는 당연히 오류투성이었다.

클래스(스타일)별로 폴더를 만들어놓고 원본을 옮긴 후 학습을 시도하였지만, 멘토링 시간에 이거 관련한 질의응답을 통해 '커스텀 클래스'를 지정해서 하면 물리적으로 디렉토리를 만들 필요가 없다는 조언도 받게 되었다.

따라서 클래스가 코드에 사전 정의가 되어야 한다는 점, 그리고 커스텀으로 클래스를 만들어서 적용할 수 있다는 점을 깨닫게 되었다.



## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

우리는 단순히 모델 학습만 하지 않고,  
최선의 Validation Values를 도출하기 위해 다양한 시각으로 학습을 진행해보았다.

- Colab 환경의 런타임 모델에 따른 Validation Values
- Local 환경에서 Batch Size에 따른 Validation Accuracy와 그것이 안정화된 Epoch 단계
  - Local 환경에서 Batch Size에 따른 GPU Utilization
- 동일 환경에서 ResNet-18의 Pretrained 사용 여부에 따른 Validation Values
- Learning Rate에 따른 Validation Values

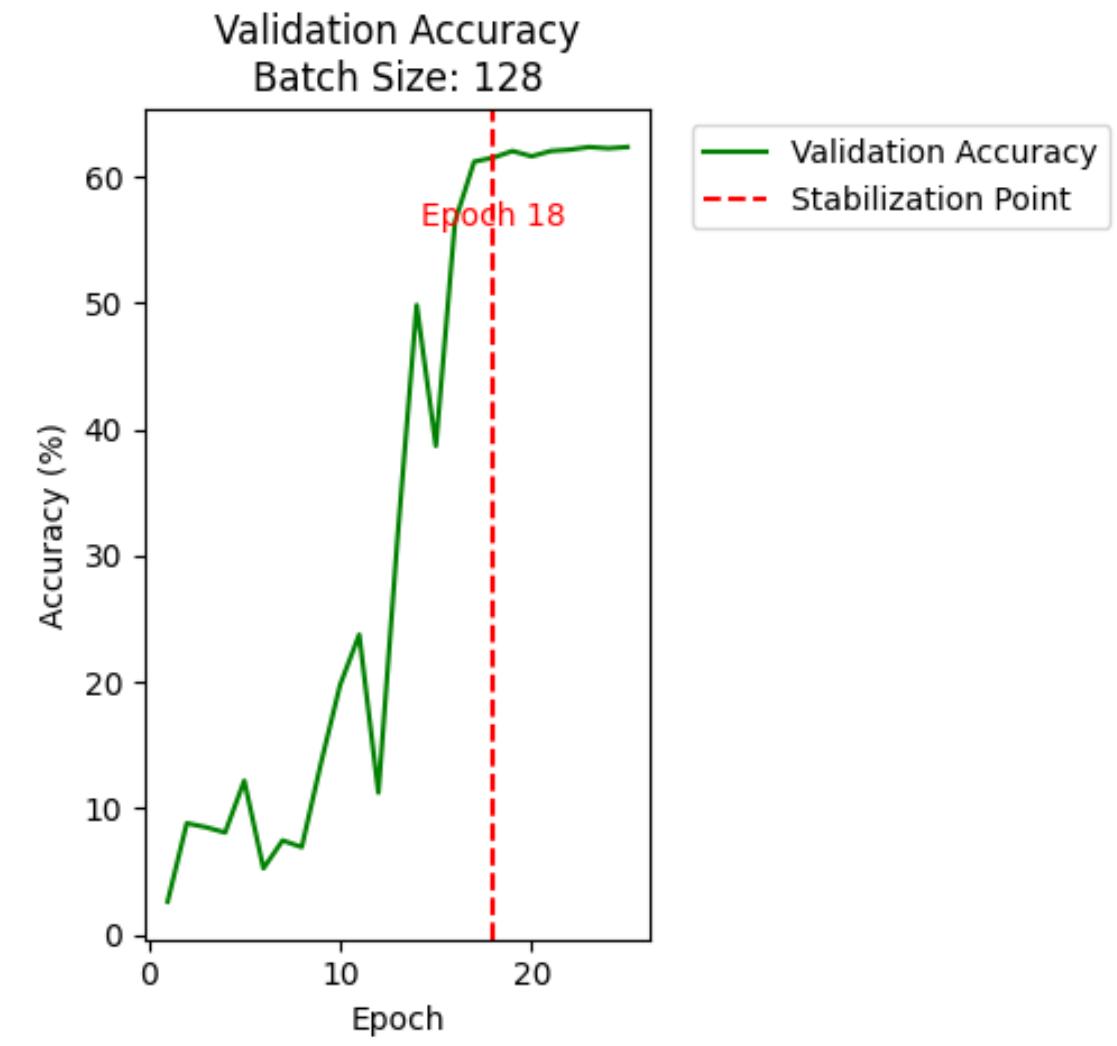
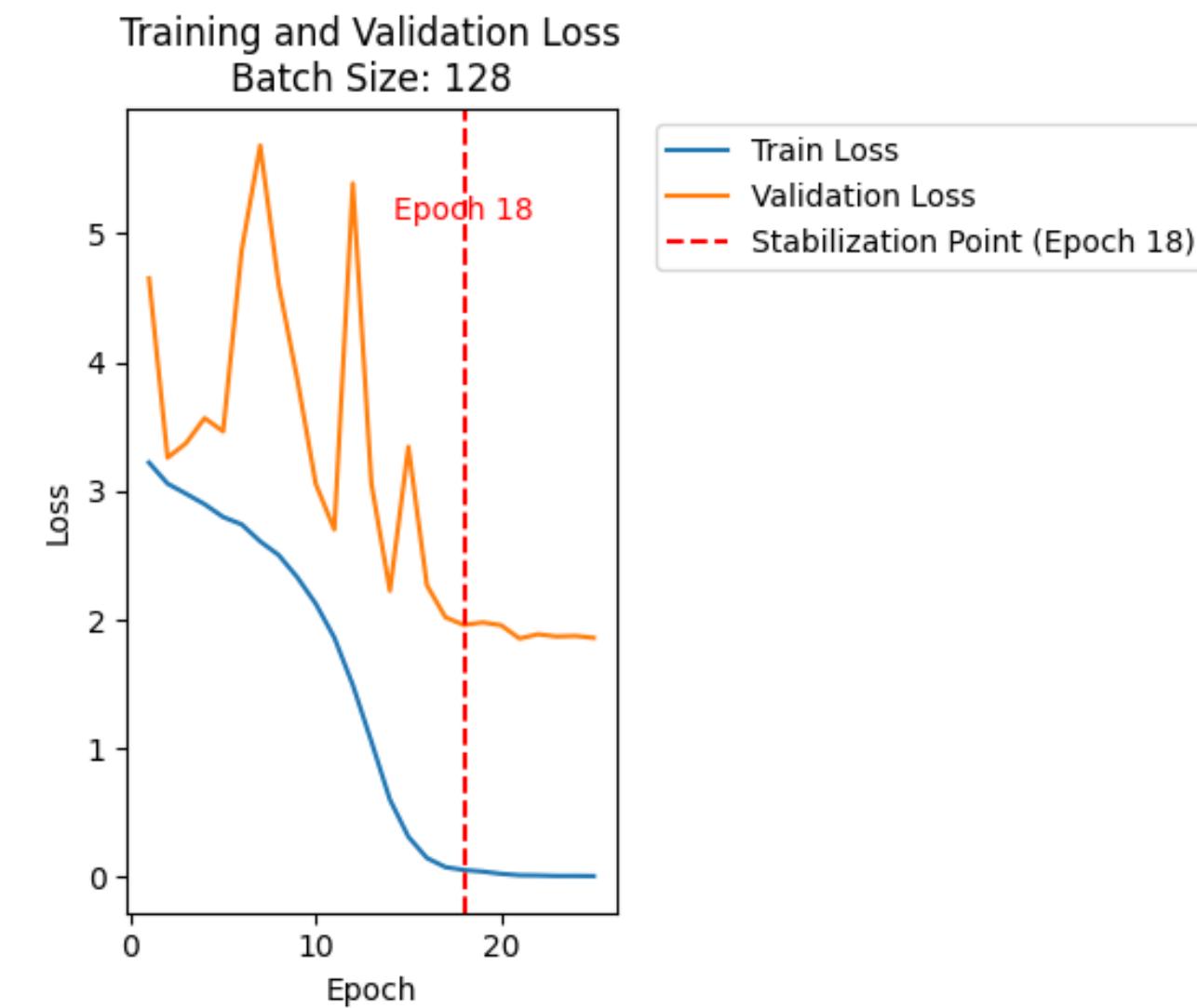


# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 1. Colab 환경의 런타임 모델에 따른 Validation Values (Epoch: 25, Batch Size: 128, Learning Rate: 0.001)

...  
Epoch 21/25 시작  
Epoch 21/25] 학습 완료, Average Loss: 0.0148  
Validation Loss: 1.8536, Validation Accuracy: 62.04%  
Epoch 22/25 시작  
Epoch 22/25] 학습 완료, Average Loss: 0.0132  
Validation Loss: 1.8872, Validation Accuracy: 62.15%  
Epoch 23/25 시작  
Epoch 23/25] 학습 완료, Average Loss: 0.0095  
Validation Loss: 1.8695, Validation Accuracy: 62.36%  
Epoch 24/25 시작  
Epoch 24/25] 학습 완료, Average Loss: 0.0095  
Validation Loss: 1.8743, Validation Accuracy: 62.25%  
Epoch 25/25 시작  
Epoch 25/25] 학습 완료, Average Loss: 0.0081  
Validation Loss: 1.8591, Validation Accuracy: 62.36%



(L4 with High Capacity Ram)

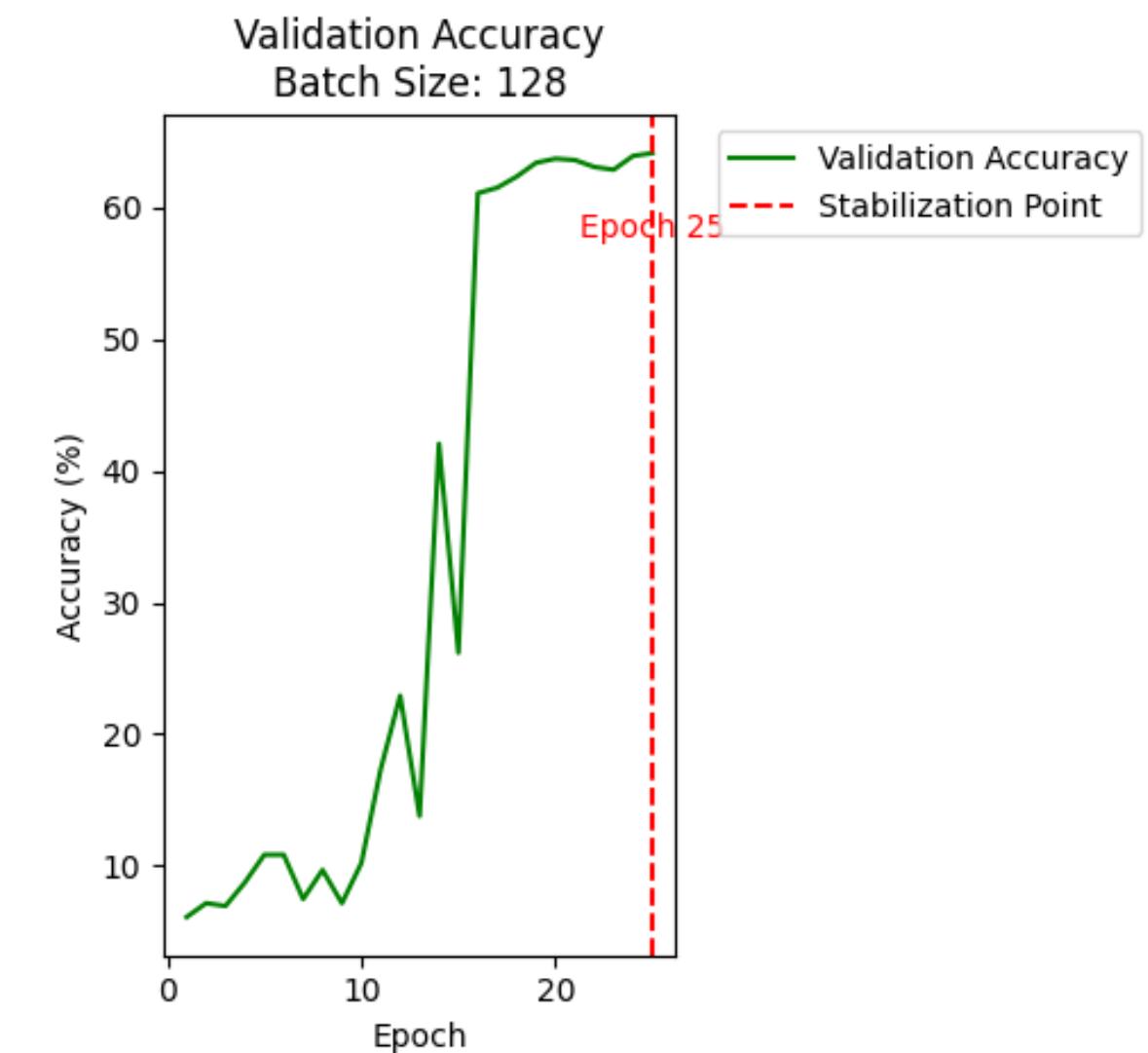
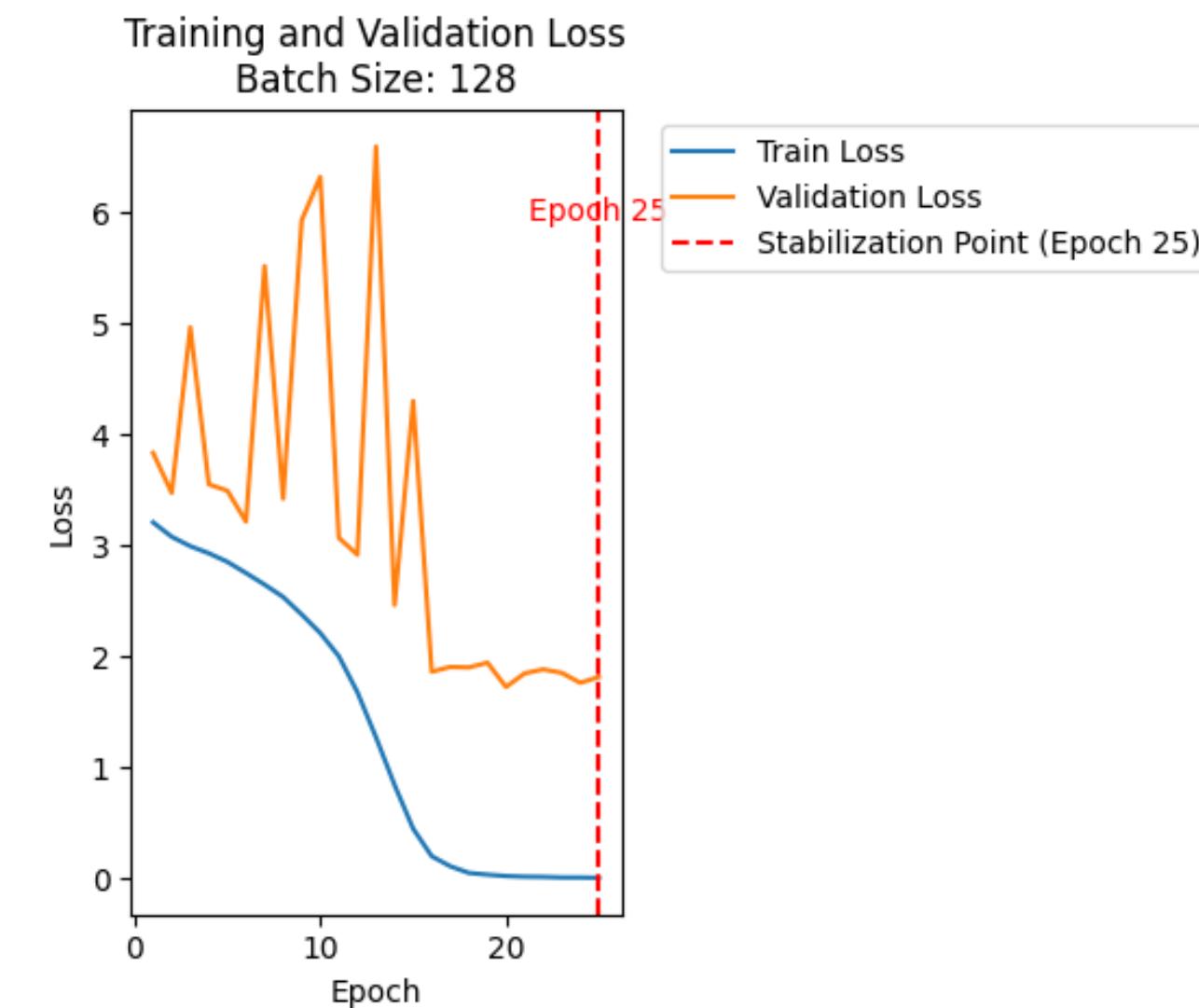


# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 1. Colab 환경의 런타임 모델에 따른 Validation Values (Epoch: 25, Batch Size: 128, Learning Rate: 0.001)

...  
Epoch 21/25 시작  
Epoch [21/25] 학습 완료, Average Loss: 0.0186  
Validation Loss: 1.8488, Validation Accuracy: 63.62%  
Epoch 22/25 시작  
Epoch [22/25] 학습 완료, Average Loss: 0.0165  
Validation Loss: 1.8845, Validation Accuracy: 63.09%  
Epoch 23/25 시작  
Epoch [23/25] 학습 완료, Average Loss: 0.0115  
Validation Loss: 1.8534, Validation Accuracy: 62.88%  
Epoch 24/25 시작  
Epoch [24/25] 학습 완료, Average Loss: 0.0112  
Validation Loss: 1.7636, Validation Accuracy: 63.93%  
Epoch 25/25 시작  
Epoch [25/25] 학습 완료, Average Loss: 0.0081  
Validation Loss: 1.8160, Validation Accuracy: 64.14%



(A100)



# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

2. Local 환경에서 Batch Size에 따른 Validation Accuracy와 그것이 안정화된 Epoch 단계  
(Epoch: 50, Learning Rate: 0.001, GPU: 4070 Super)

...

Epoch 46/50 시작

Epoch [46/50] 학습 완료, Average Loss: 0.0290

Validation Loss: 2.9433, Validation Accuracy: 61.51%

Epoch 47/50 시작

Epoch [47/50] 학습 완료, Average Loss: 0.0171

Validation Loss: 2.8916, Validation Accuracy: 60.78%

Epoch 48/50 시작

Epoch [48/50] 학습 완료, Average Loss: 0.0203

Validation Loss: 2.9917, Validation Accuracy: 61.41%

Epoch 49/50 시작

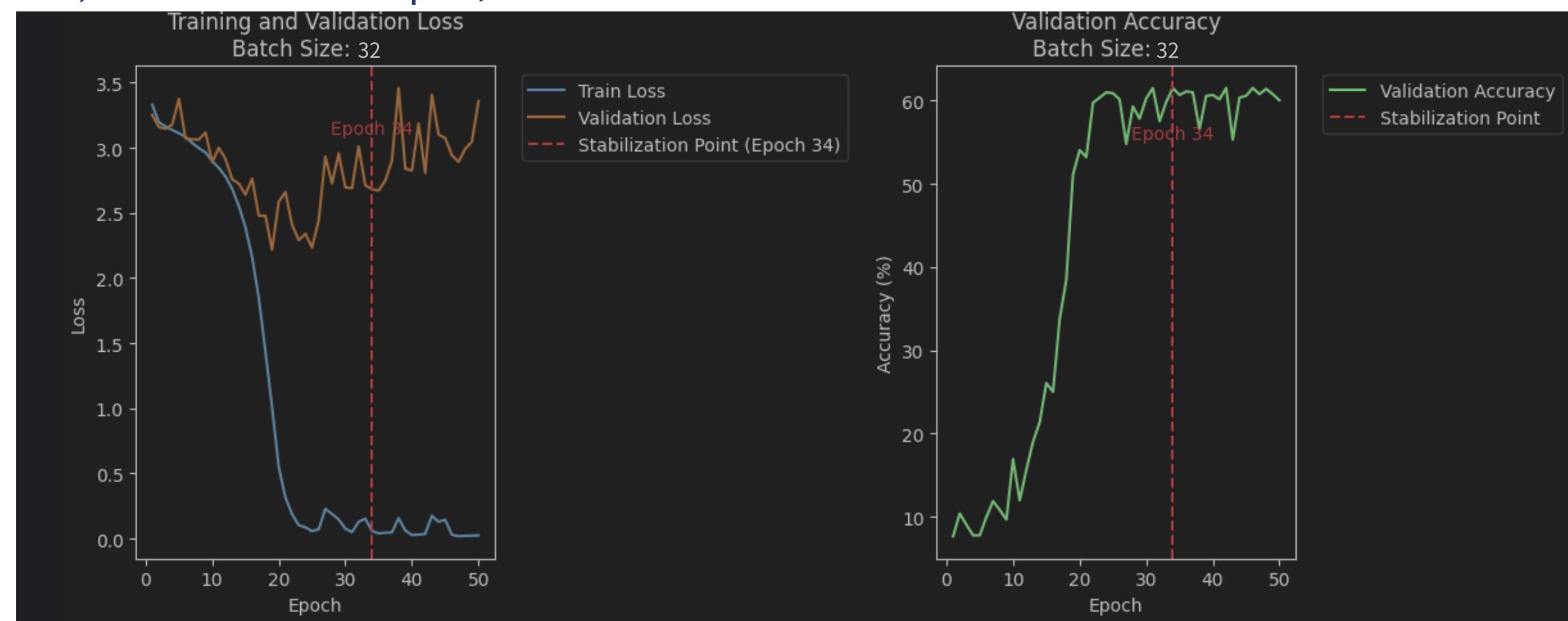
Epoch [49/50] 학습 완료, Average Loss: 0.0215

Validation Loss: 3.0455, Validation Accuracy: 60.78%

Epoch 50/50 시작

Epoch [50/50] 학습 완료, Average Loss: 0.0237

Validation Loss: 3.3591, Validation Accuracy: 60.04%



소요시간: 7H 27M

(Batch Size: 32)

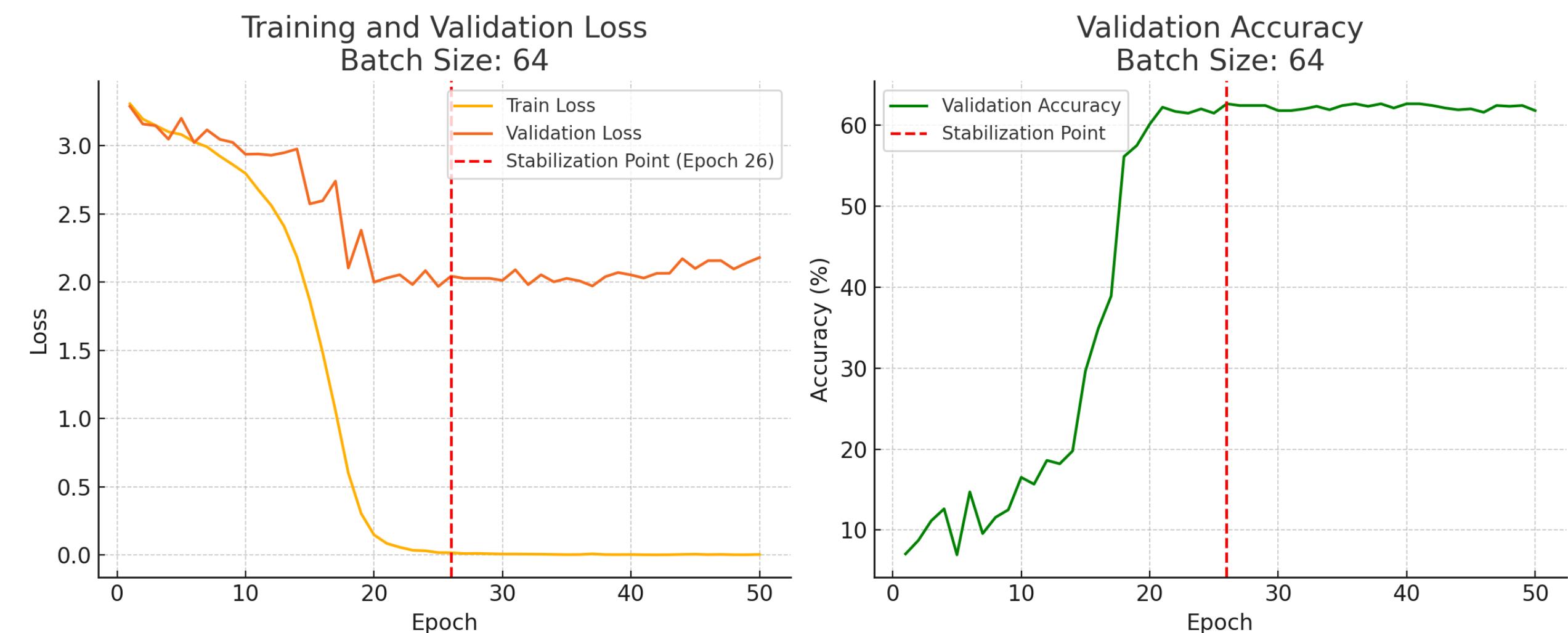


## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

2. Local 환경에서 Batch Size에 따른 Validation Accuracy와 그것이 안정화된 Epoch 단계  
(Epoch: 50, Learning Rate: 0.001, GPU: 4070 Super)

...  
Epoch 46/50 시작  
Epoch [46/50] 학습 완료, Average Loss: 0.0040  
Validation Loss: 2.1580, Validation Accuracy: 61.62%  
Epoch 47/50 시작  
Epoch [47/50] 학습 완료, Average Loss: 0.0054  
Validation Loss: 2.1587, Validation Accuracy: 62.46%  
Epoch 48/50 시작  
Epoch [48/50] 학습 완료, Average Loss: 0.0032  
Validation Loss: 2.0969, Validation Accuracy: 62.36%  
Epoch 49/50 시작  
Epoch [49/50] 학습 완료, Average Loss: 0.0032  
Validation Loss: 2.1423, Validation Accuracy: 62.46%  
Epoch 50/50 시작  
Epoch [50/50] 학습 완료, Average Loss: 0.0050  
Validation Loss: 2.1807, Validation Accuracy: 61.83%



소요시간: 7H 45M

(Batch Size: 64)

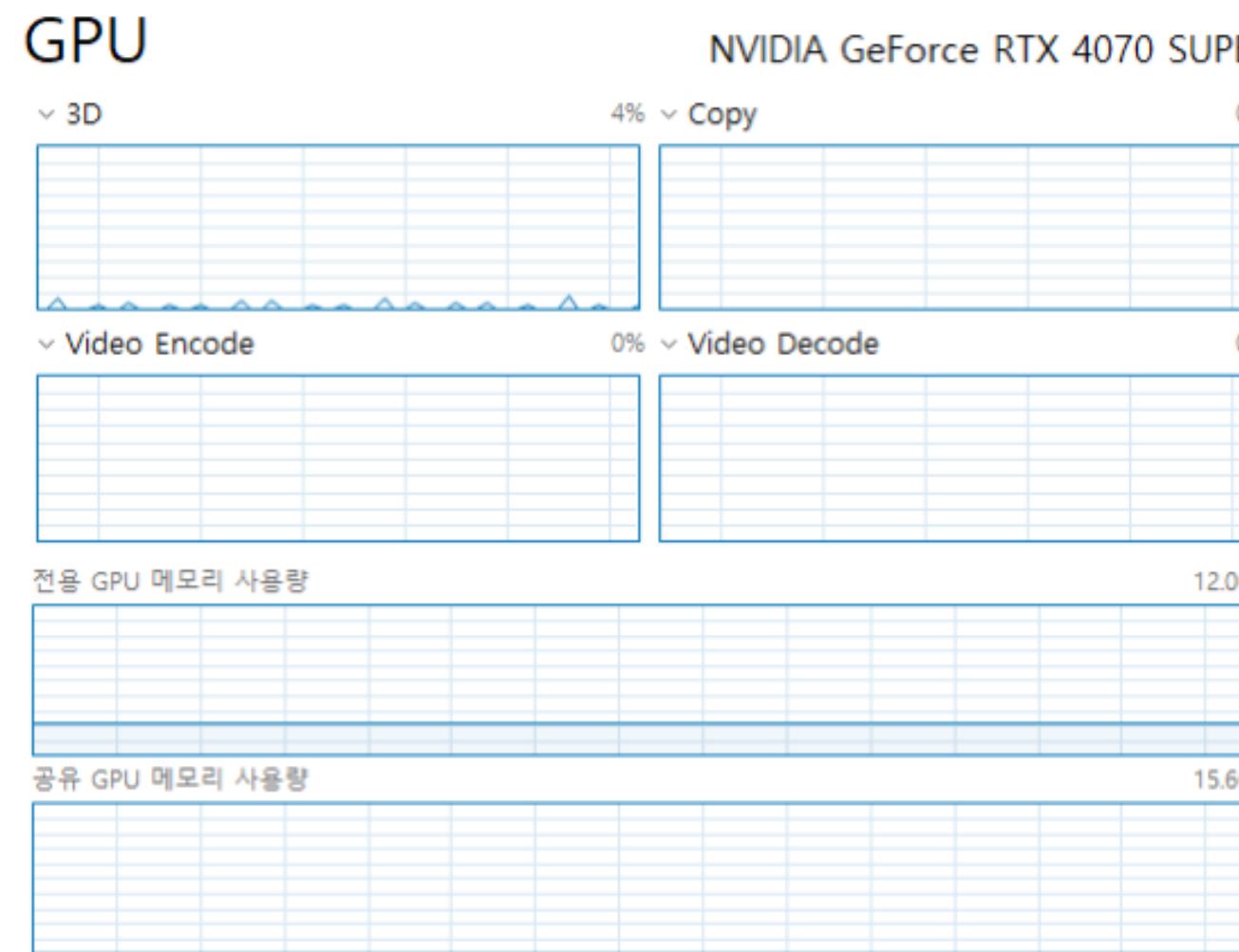


# [모델 학습]

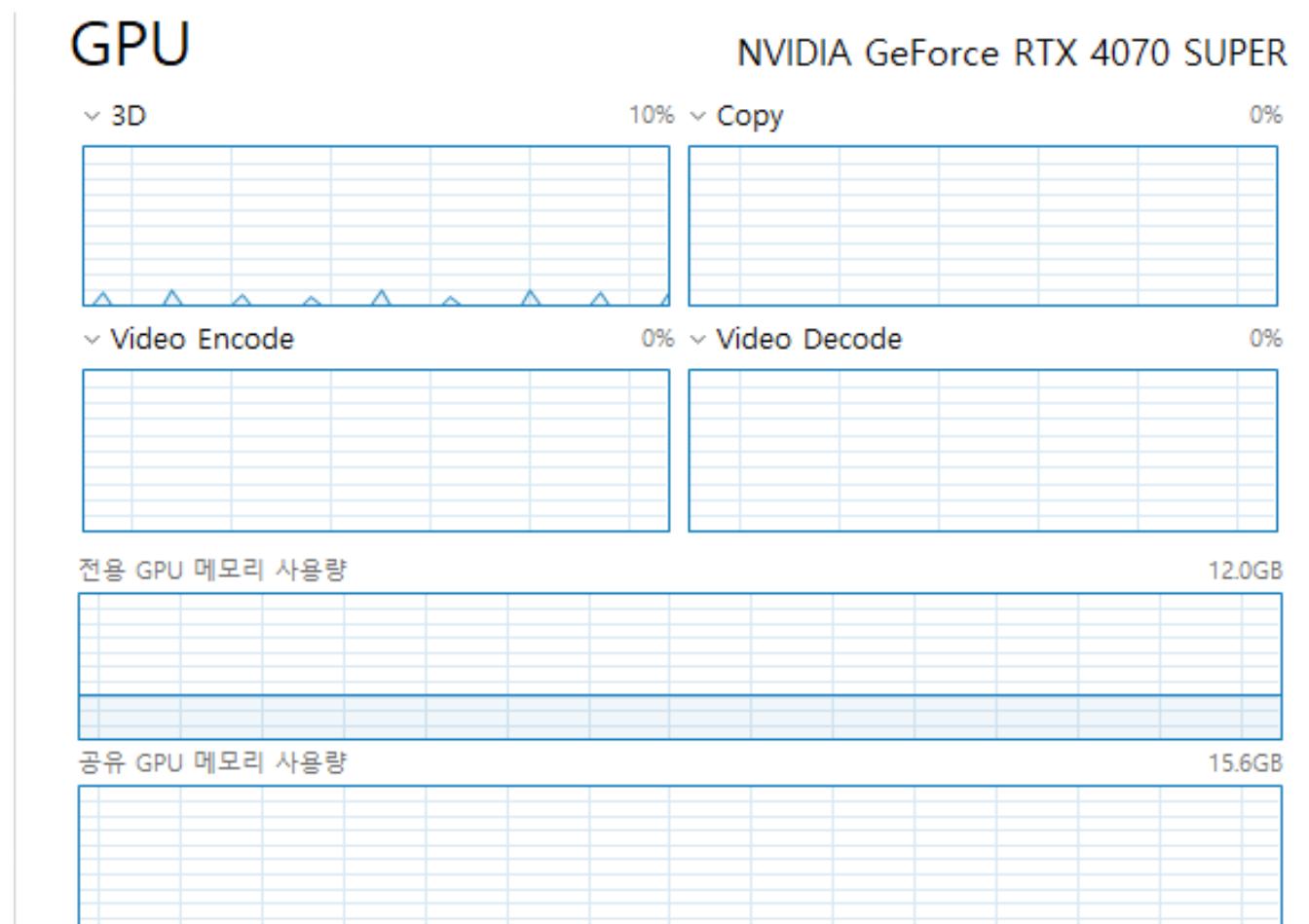
Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 2-1. Local 환경에서 Batch Size에 따른 GPU Utilization

(Epoch: 50, Learning Rate: 0.001, GPU: 4070 Super)



(Batch Size: 32)



(Batch Size: 64)

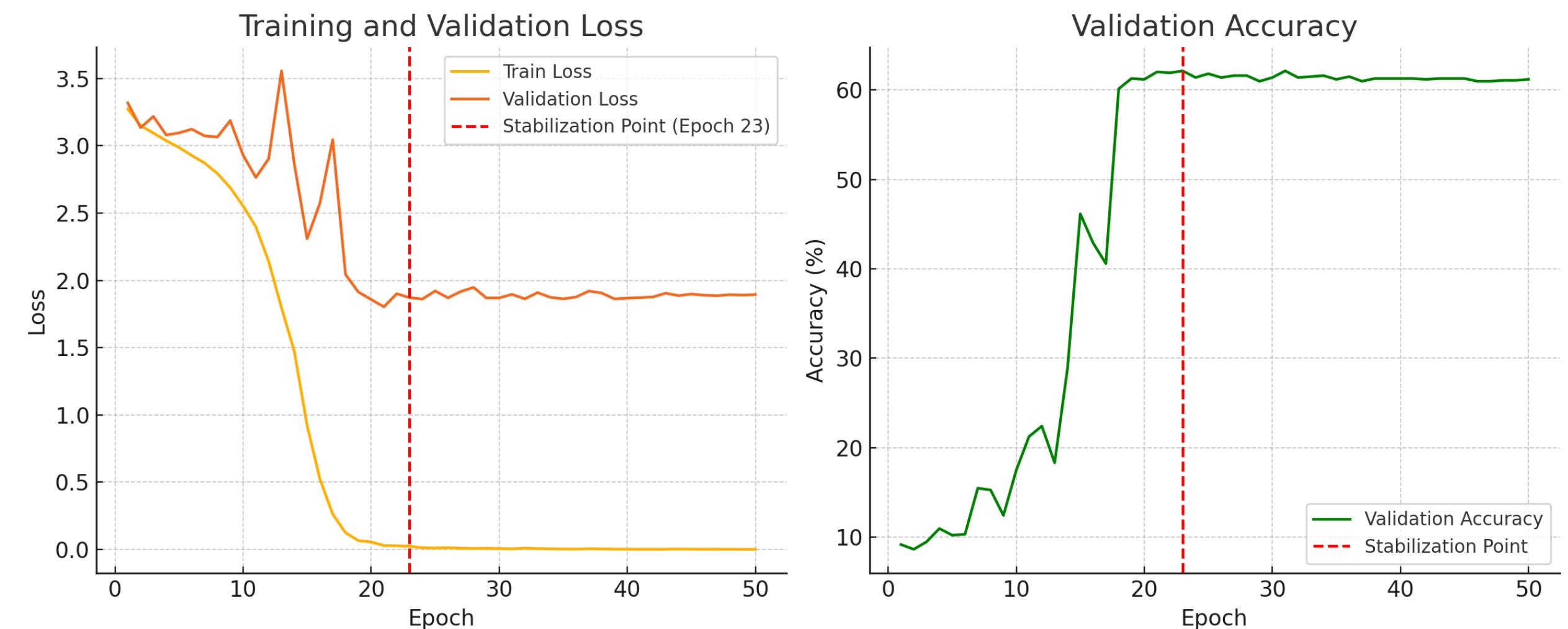


## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

2. Local 환경에서 Batch Size에 따른 Validation Accuracy와 그것이 안정화된 Epoch 단계  
(Epoch: 50, Learning Rate: 0.001, GPU: 4070 Super)

...  
Epoch 46/50 시작  
Epoch [46/50] 학습 완료, Average Loss: 0.0012  
Validation Loss: 1.8898, Validation Accuracy: 60.99%  
Epoch 47/50 시작  
Epoch [47/50] 학습 완료, Average Loss: 0.0014  
Validation Loss: 1.8861, Validation Accuracy: 60.99%  
Epoch 48/50 시작  
Epoch [48/50] 학습 완료, Average Loss: 0.0009  
Validation Loss: 1.8934, Validation Accuracy: 61.09%  
Epoch 49/50 시작  
Epoch [49/50] 학습 완료, Average Loss: 0.0006  
Validation Loss: 1.8906, Validation Accuracy: 61.09%  
Epoch 50/50 시작  
Epoch [50/50] 학습 완료, Average Loss: 0.0005  
Validation Loss: 1.8938, Validation Accuracy: 61.20%



소요시간: 7H 40M

(Batch Size: 128)

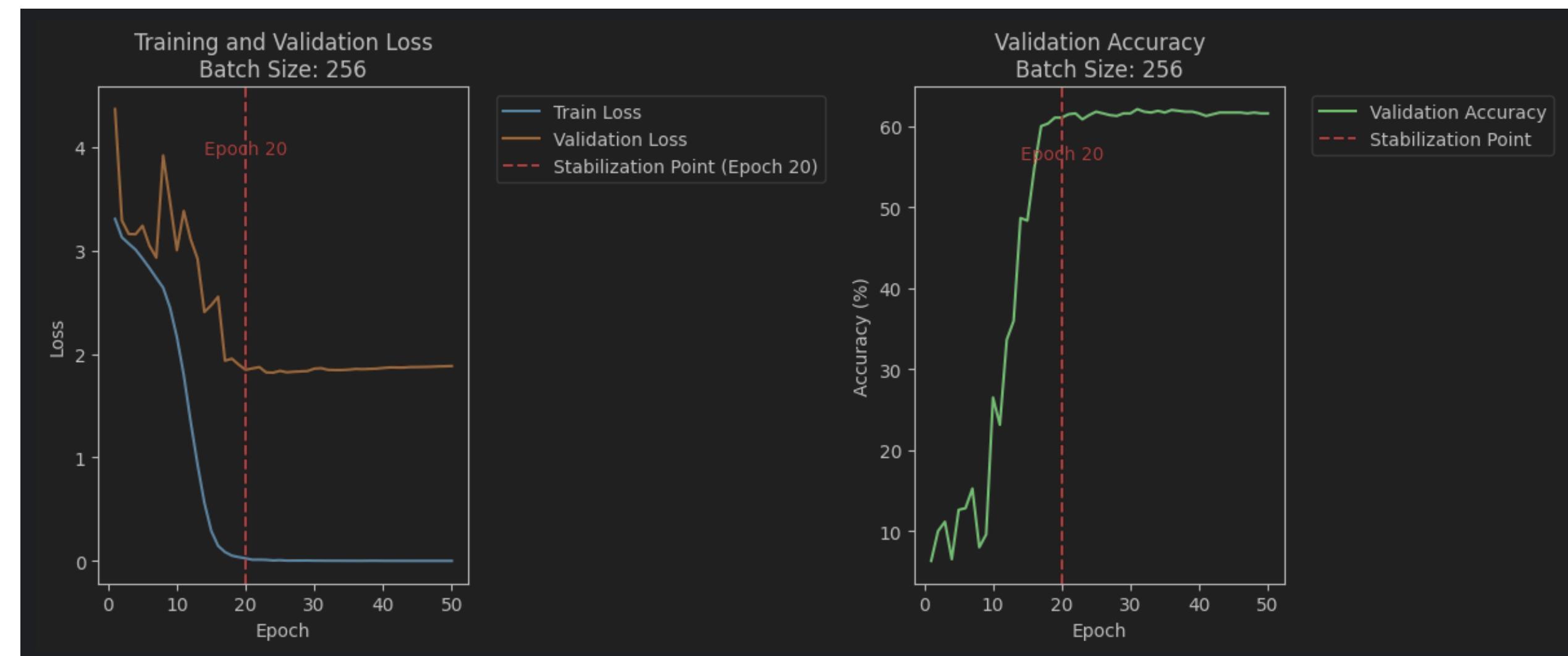


# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

2. Local 환경에서 Batch Size에 따른 Validation Accuracy와 그것이 안정화된 Epoch 단계  
(Epoch: 50, Learning Rate: 0.001, GPU: 4070 Super)

...  
Epoch 46/50 시작  
Epoch [46/50] 학습 완료, Average Loss: 0.0005  
Validation Loss: 1.8752, Validation Accuracy: 61.72%  
Epoch 47/50 시작  
Epoch [47/50] 학습 완료, Average Loss: 0.0005  
Validation Loss: 1.8769, Validation Accuracy: 61.62%  
Epoch 48/50 시작  
Epoch [48/50] 학습 완료, Average Loss: 0.0004  
Validation Loss: 1.8798, Validation Accuracy: 61.72%  
Epoch 49/50 시작  
Epoch [49/50] 학습 완료, Average Loss: 0.0004  
Validation Loss: 1.8816, Validation Accuracy: 61.62%  
Epoch 50/50 시작  
Epoch [50/50] 학습 완료, Average Loss: 0.0004  
Validation Loss: 1.8833, Validation Accuracy: 61.62%



소요시간: 7H 43M

(Batch Size: 256)

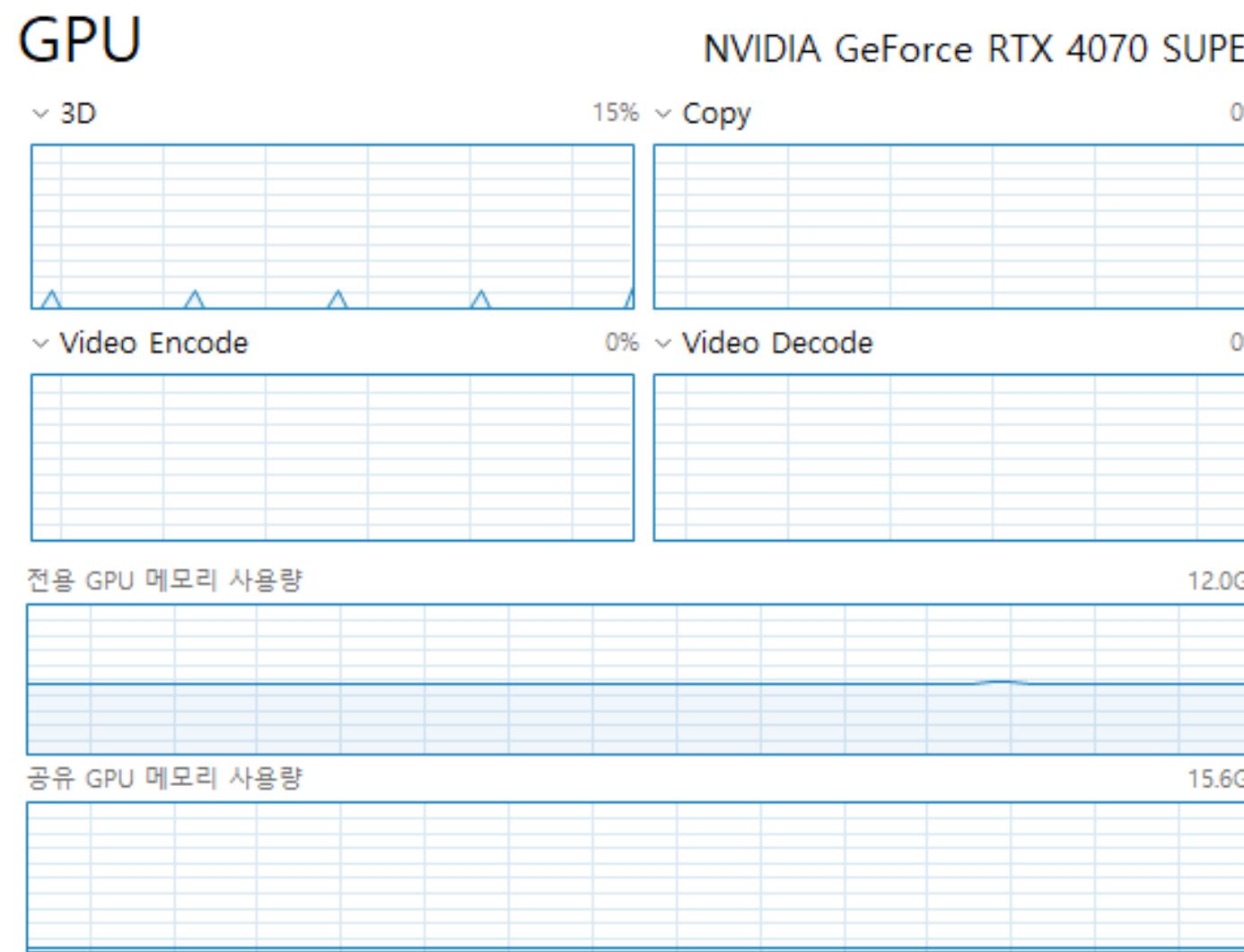


# [모델 학습]

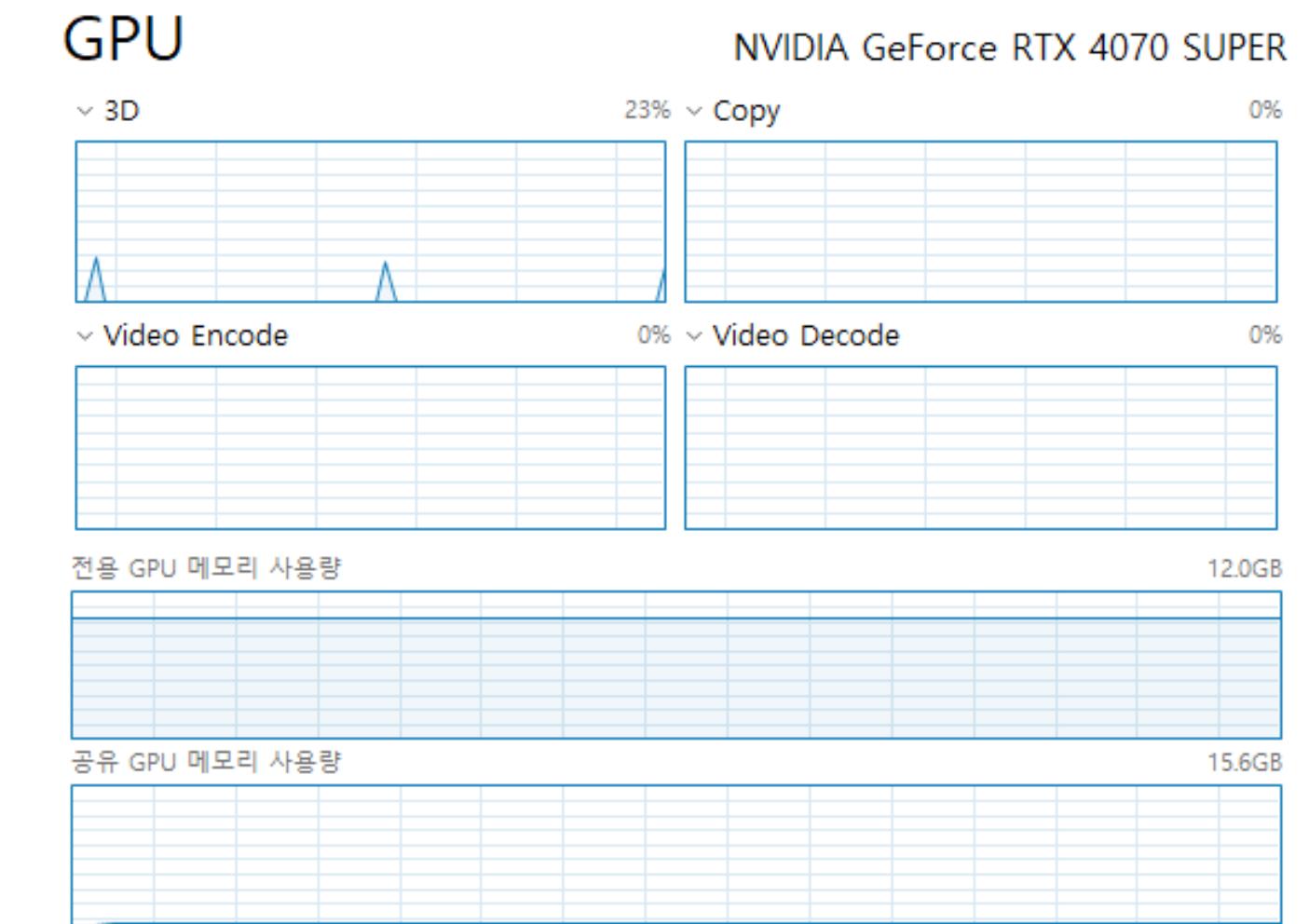
Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 2-1. Local 환경에서 Batch Size에 따른 GPU Utilization

(Epoch: 50, Learning Rate: 0.001, GPU: 4070 Super)



(Batch Size: 128)



(Batch Size: 256)



# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 3. 동일 환경에서 ResNet-18의 Pretrained 사용 여부에 따른 Validation Values (Epoch: 25, Batch Size: 128, Learning Rate: 0.001)

...

Epoch 21/25 시작

Epoch [21/25] 학습 완료, Average Loss: 0.0024

Validation Loss: 1.5817, Validation Accuracy: 66.56%

Epoch 22/25 시작

Epoch [22/25] 학습 완료, Average Loss: 0.0010

Validation Loss: 1.5729, Validation Accuracy: 66.14%

Epoch 23/25 시작

Epoch [23/25] 학습 완료, Average Loss: 0.0007

Validation Loss: 1.5704, Validation Accuracy: 66.14%

Epoch 24/25 시작

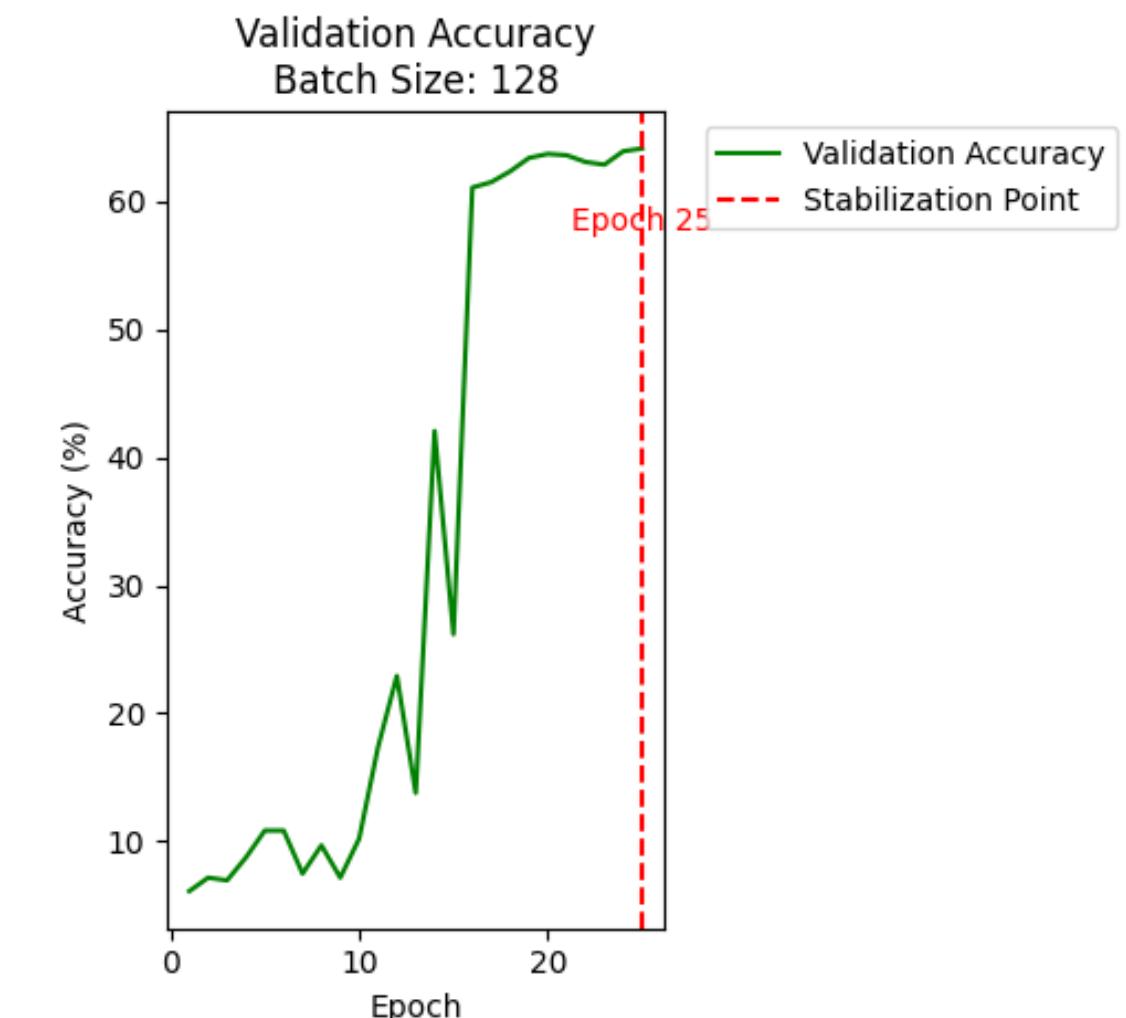
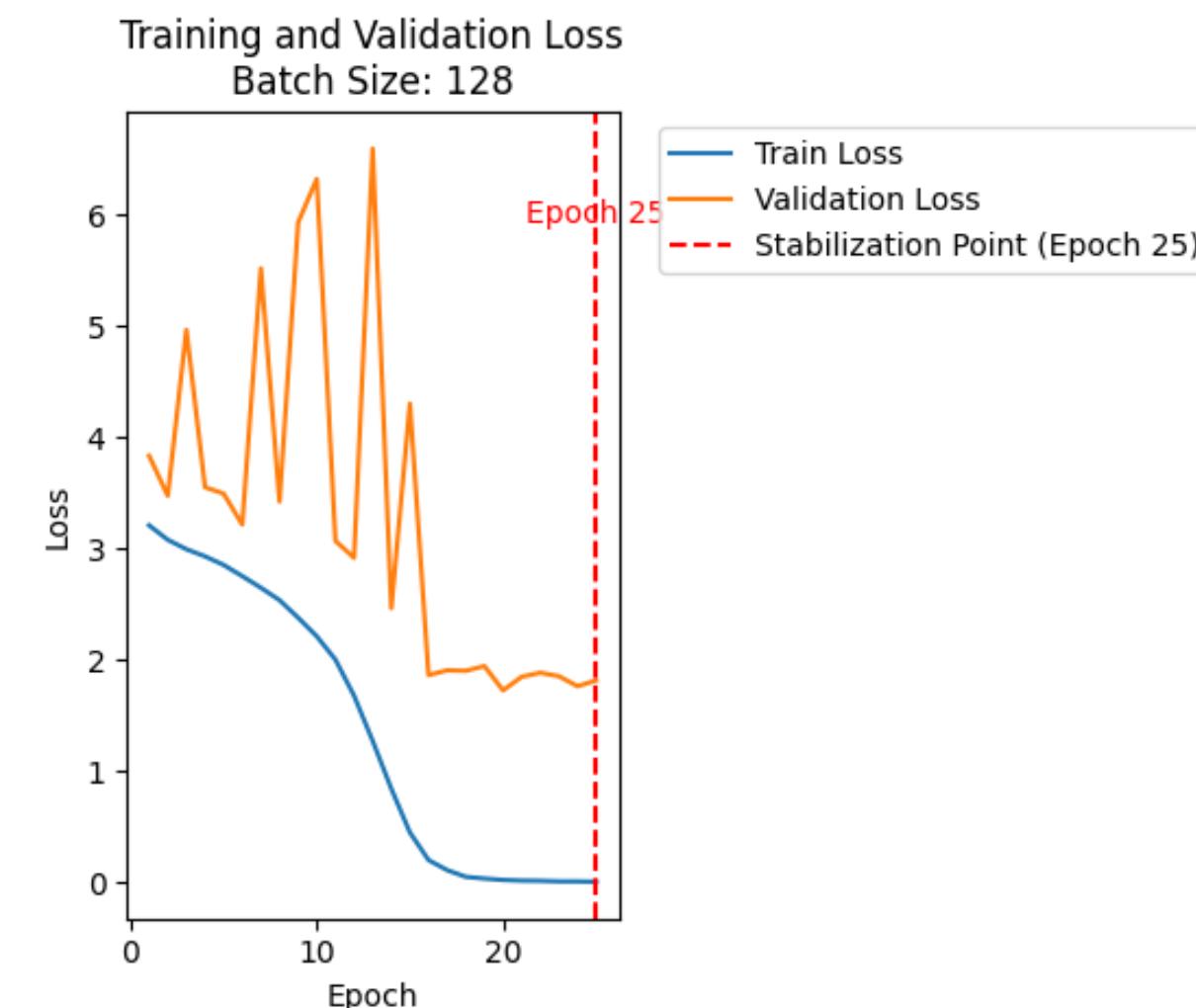
Epoch [24/25] 학습 완료, Average Loss: 0.0006

Validation Loss: 1.5729, Validation Accuracy: 65.72%

Epoch 25/25 시작

Epoch [25/25] 학습 완료, Average Loss: 0.0005

Validation Loss: 1.5768, Validation Accuracy: 66.35%



(Pretrained 사용: YES)

문제 조건 상에는 이를 사용하지 않고 하는 것으로 되어있었지만,  
Pretrained 사용하면 어느정도 차이가 발생하는지 궁금하게 되어 이런 접근도 해 보았다.

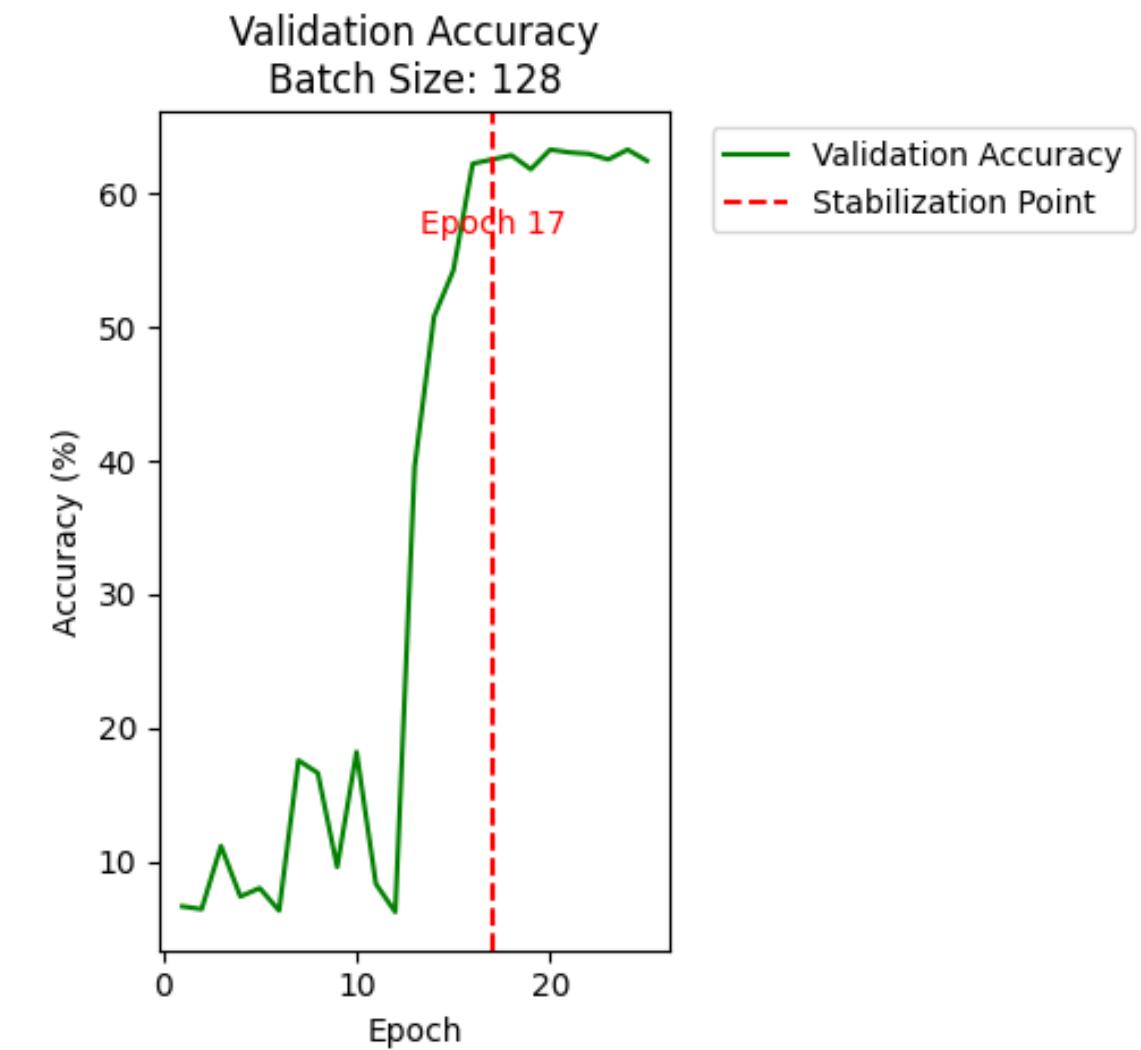
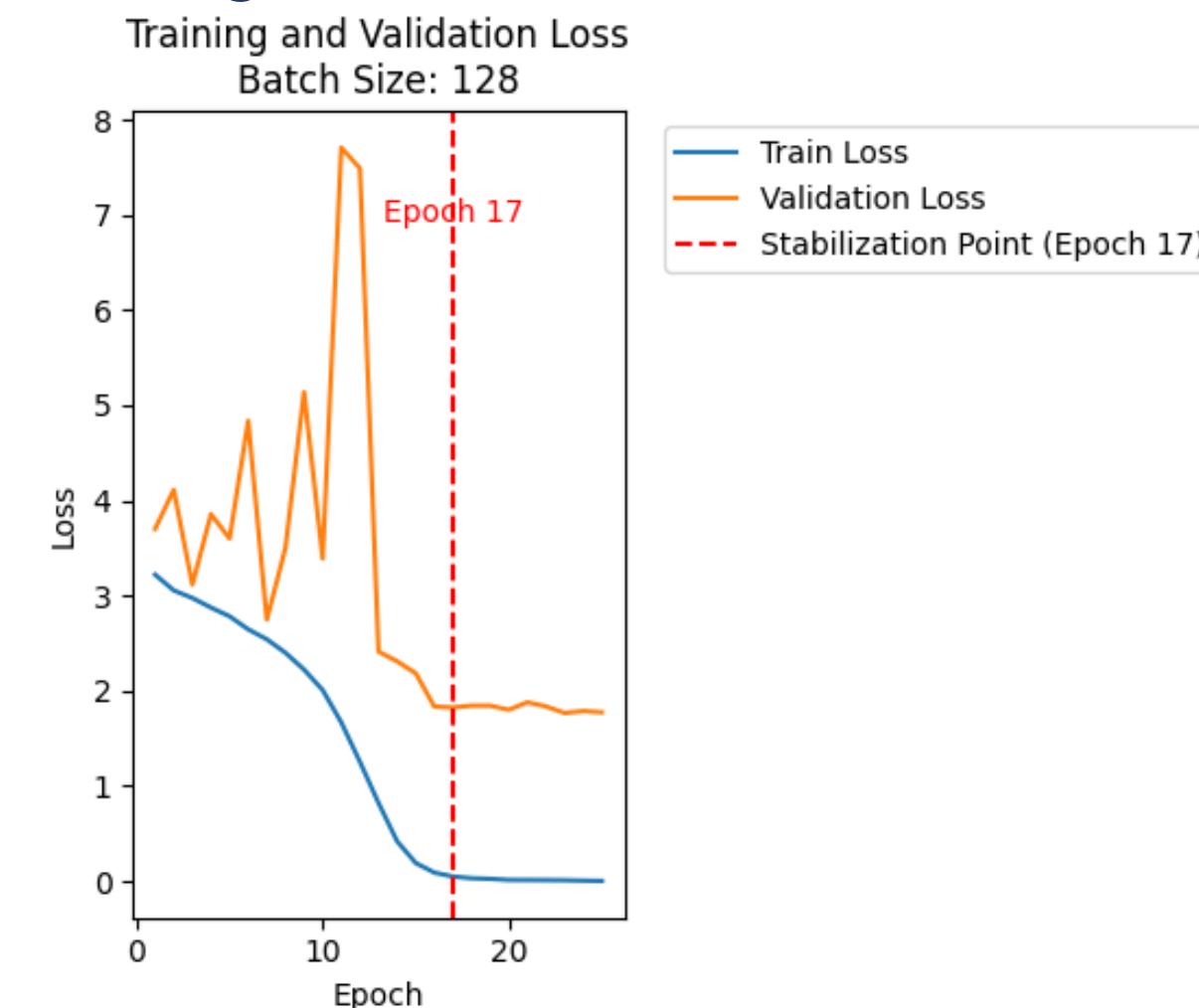


# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 3. 동일 환경에서 ResNet-18의 Pretrained 사용 여부에 따른 Validation Values (Epoch: 25, Batch Size: 128, Learning Rate: 0.001)

...  
Epoch 21/25 시작  
Epoch [21/25] 학습 완료, Average Loss: 0.0137  
Validation Loss: 1.8802, Validation Accuracy: 63.09%  
Epoch 22/25 시작  
Epoch [22/25] 학습 완료, Average Loss: 0.0128  
Validation Loss: 1.8358, Validation Accuracy: 62.99%  
Epoch 23/25 시작  
Epoch [23/25] 학습 완료, Average Loss: 0.0112  
Validation Loss: 1.7674, Validation Accuracy: 62.57%  
Epoch 24/25 시작  
Epoch [24/25] 학습 완료, Average Loss: 0.0067  
Validation Loss: 1.7881, Validation Accuracy: 63.30%  
Epoch 25/25 시작  
Epoch [25/25] 학습 완료, Average Loss: 0.0040  
Validation Loss: 1.7751, Validation Accuracy: 62.46%



(Pretrained 사용: NO)

문제 조건 상에는 이를 사용하지 않고 하는 것으로 되어있었지만,  
Pretrained 사용하면 어느정도 차이가 발생하는지 궁금하게 되어 이런 접근도 해 보았다.

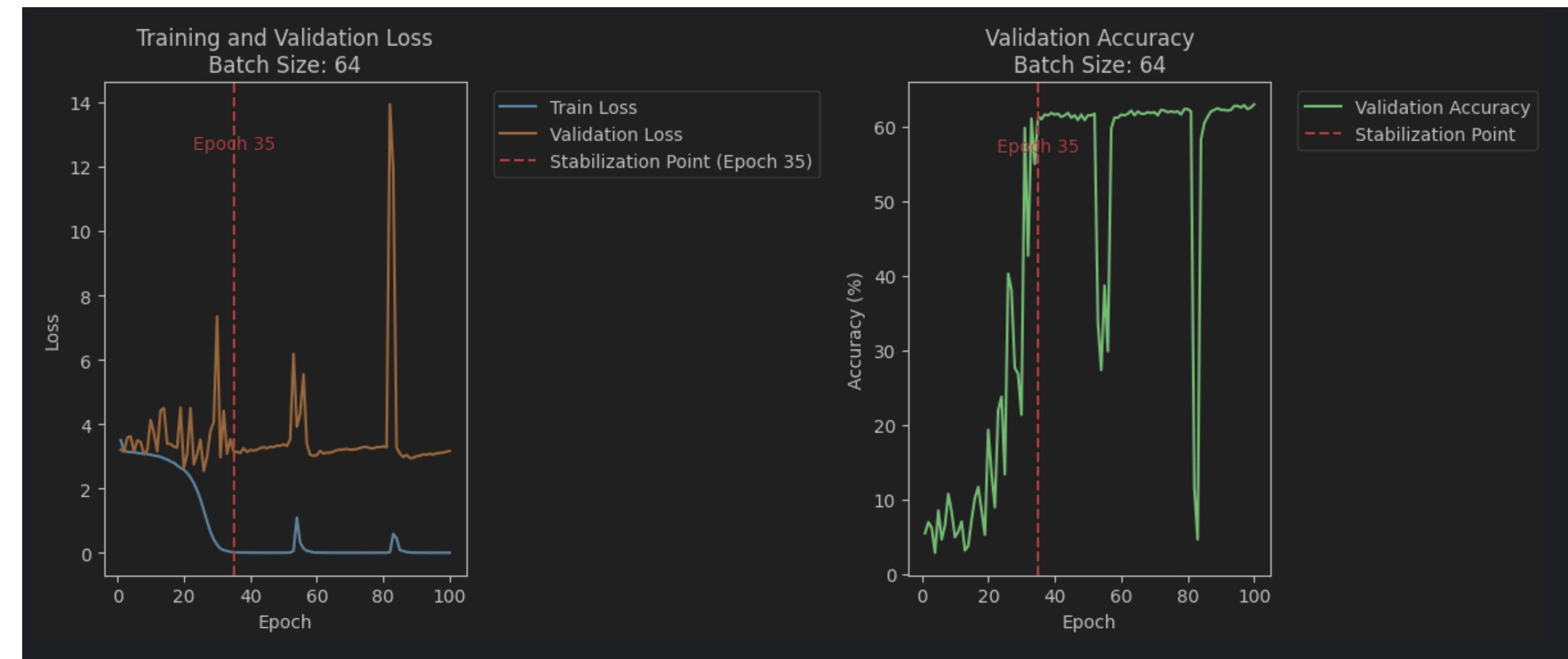


# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 4. 동일 환경에서 Learning Rate에 따른 Validation Values (Epoch: 100, Batch Size: 64)

...  
Epoch 96/100 시작  
Epoch [96/100] 학습 완료, Average Loss: 0.0006,  
Training Accuracy: 100.00%  
Validation Loss: 3.0907, Validation Accuracy: 62.57%  
Epoch 97/100 시작  
Epoch [97/100] 학습 완료, Average Loss: 0.0007,  
Training Accuracy: 99.98%  
Validation Loss: 3.0967, Validation Accuracy: 62.88%  
Epoch 98/100 시작  
Epoch [98/100] 학습 완료, Average Loss: 0.0008,  
Training Accuracy: 99.98%  
Validation Loss: 3.1153, Validation Accuracy: 62.36%  
Epoch 99/100 시작  
Epoch [99/100] 학습 완료, Average Loss: 0.0011,  
Training Accuracy: 99.98%  
Validation Loss: 3.1377, Validation Accuracy: 62.57%  
Epoch 100/100 시작  
Epoch [100/100] 학습 완료, Average Loss: 0.0005,  
Training Accuracy: 100.00%  
Validation Loss: 3.1652, Validation Accuracy: 62.99%



(Learning Rate: 0.1)

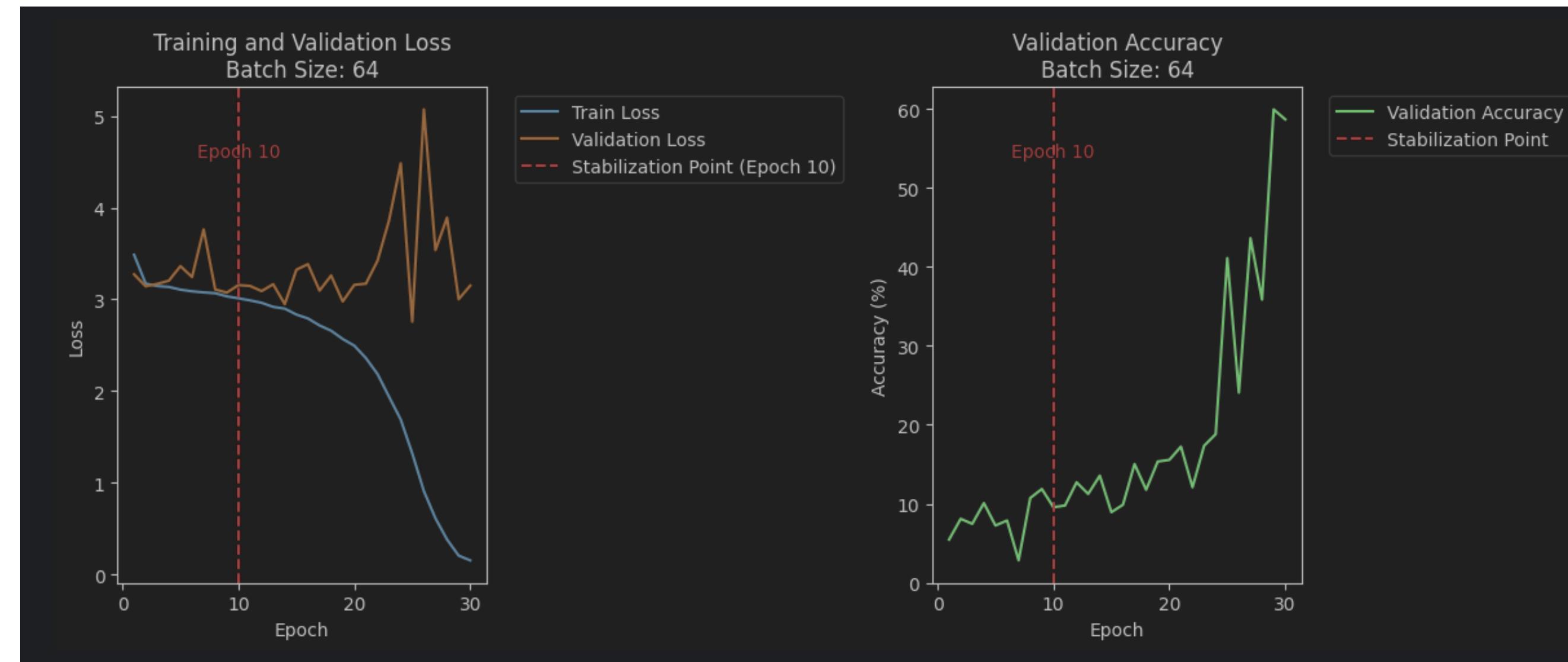


# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 4. 동일 환경에서 Learning Rate에 따른 Validation Values (Epoch: 30, Batch Size: 64)

...  
Epoch 26/30 시작  
Epoch [26/30] 학습 완료, Average Loss: 0.9083,  
Training Accuracy: 71.20%  
Validation Loss: 5.0730, Validation Accuracy: 24.08%  
Epoch 27/30 시작  
Epoch [27/30] 학습 완료, Average Loss: 0.6107,  
Training Accuracy: 81.20%  
Validation Loss: 3.5379, Validation Accuracy: 43.64%  
Epoch 28/30 시작  
Epoch [28/30] 학습 완료, Average Loss: 0.3795,  
Training Accuracy: 88.11%  
Validation Loss: 3.8918, Validation Accuracy: 35.86%  
Epoch 29/30 시작  
Epoch [29/30] 학습 완료, Average Loss: 0.2039,  
Training Accuracy: 94.25%  
Validation Loss: 3.0016, Validation Accuracy: 59.94%  
Epoch 30/30 시작  
Epoch [30/30] 학습 완료, Average Loss: 0.1517,  
Training Accuracy: 96.27%  
Validation Loss: 3.1502, Validation Accuracy: 58.68%



(Learning Rate: 0.01)

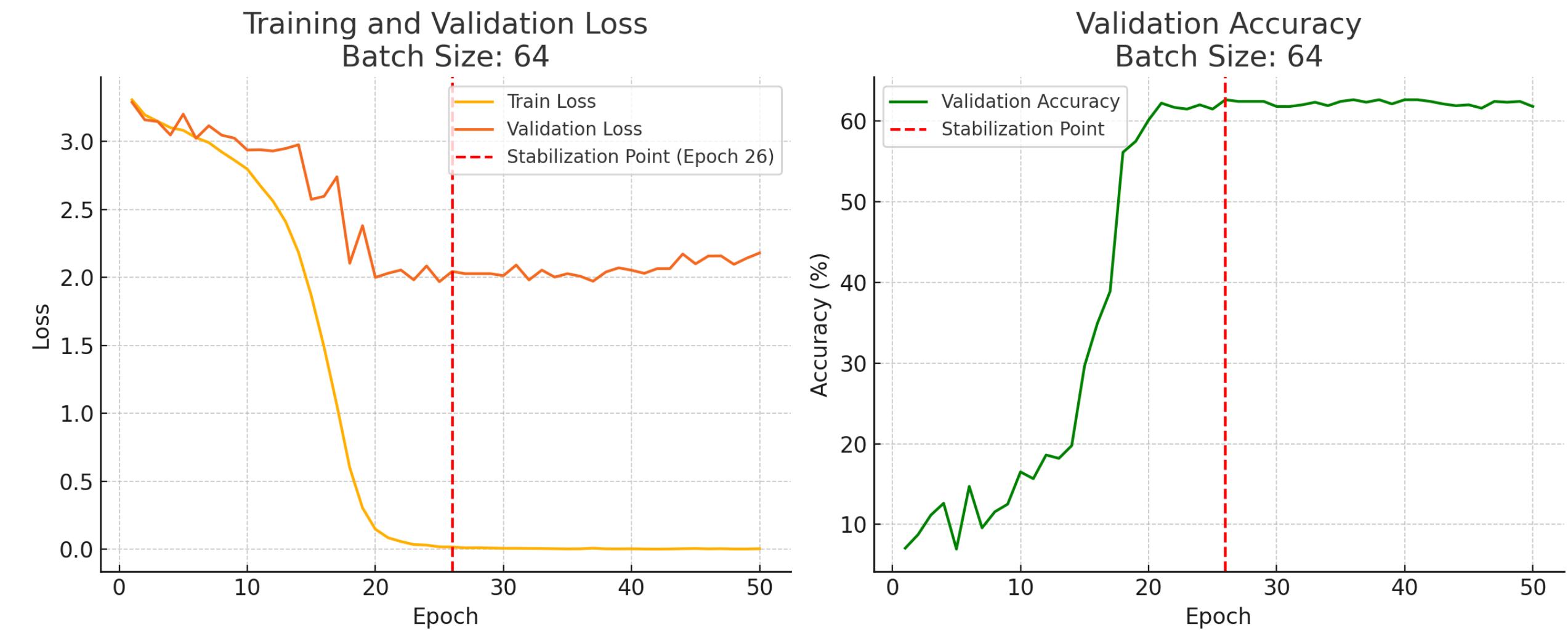


## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

### 4. 동일 환경에서 Learning Rate에 따른 Validation Values (Epoch: 50, Batch Size: 64)

...  
Epoch 46/50 시작  
Epoch [46/50] 학습 완료, Average Loss: 0.0040  
Validation Loss: 2.1580, Validation Accuracy: 61.62%  
Epoch 47/50 시작  
Epoch [47/50] 학습 완료, Average Loss: 0.0054  
Validation Loss: 2.1587, Validation Accuracy: 62.46%  
Epoch 48/50 시작  
Epoch [48/50] 학습 완료, Average Loss: 0.0032  
Validation Loss: 2.0969, Validation Accuracy: 62.36%  
Epoch 49/50 시작  
Epoch [49/50] 학습 완료, Average Loss: 0.0032  
Validation Loss: 2.1423, Validation Accuracy: 62.46%  
Epoch 50/50 시작  
Epoch [50/50] 학습 완료, Average Loss: 0.0050  
Validation Loss: 2.1807, Validation Accuracy: 61.83%



(Learning Rate: 0.001)

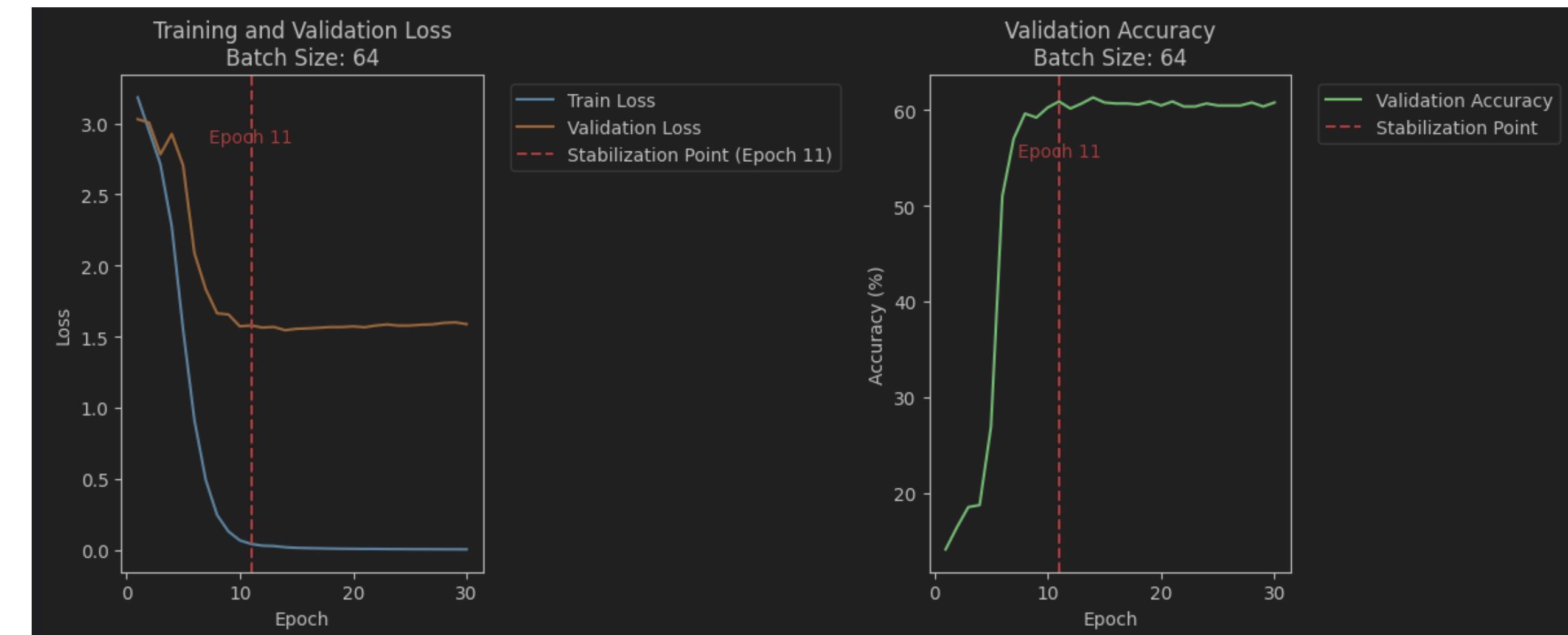


# [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

## 4. 동일 환경에서 Learning Rate에 따른 Validation Values (Epoch: 30, Batch Size: 64)

...  
Epoch 26/30 시작  
Epoch [26/30] 학습 완료, Average Loss: 0.0038,  
Training Accuracy: 100.00%  
Validation Loss: 1.5832, Validation Accuracy: 60.46%  
Epoch 27/30 시작  
Epoch [27/30] 학습 완료, Average Loss: 0.0036,  
Training Accuracy: 100.00%  
Validation Loss: 1.5857, Validation Accuracy: 60.46%  
Epoch 28/30 시작  
Epoch [28/30] 학습 완료, Average Loss: 0.0032,  
Training Accuracy: 100.00%  
Validation Loss: 1.5962, Validation Accuracy: 60.78%  
Epoch 29/30 시작  
Epoch [29/30] 학습 완료, Average Loss: 0.0031,  
Training Accuracy: 100.00%  
Validation Loss: 1.6000, Validation Accuracy: 60.36%  
Epoch 30/30 시작  
Epoch [30/30] 학습 완료, Average Loss: 0.0028,  
Training Accuracy: 100.00%  
Validation Loss: 1.5872, Validation Accuracy: 60.78%



(Learning Rate: 0.0001)



## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

다양한 각도로 모델 학습을 하면서 알게 된 내용은 다음과 같다.

- 대부분의 Case에서는 Validation Accracy Value가 60 ~ 63% 정도 나오게 되었다.
- Colab에서 학습할 때, 런타임 종류(L4, A100)에 따라 Validation Accracy Value가 대부분의 경우보다 살짝 향상된 바(A100) 있으나 보통 런타임의 차이는 학습 수행 속도의 차이가 더 뚜렷하게 나타났었다.
- Batch Size에 변화를 주고 학습하는 것은, Validation Accracy Value 및 학습 수행 속도에 차이를 보이기보다 일괄적으로 처리되는 데이터 양의 차이가 있어 메모리 리소스의 증가 및 데이터가 들어가는 타이밍이 늦어지게 되었다. (물론 이는 비례할 것 같았다.)

Batch Size가 크면 한번에 학습하는 이미지의 수가 많고 각 Epoch마다의 Validation Accuracy 상승 폭이 커졌다. → Validation Accuracy 60%에 빨리 도달하게 되었고 데이터 안정화가 더 빠른 시간에(더 적은 Epoch에) 도달할 수 있다는 것을 알게되었다.



## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

다양한 각도로 모델 학습을 하면서 알게 된 내용은 다음과 같다.

- Pretrained 사용을 해서 학습을 시도 한 것은 좋았던 것 같다. 사전 모델이 있으니 확실하게 큰 폭으로 Validation Accracy Value가 높아지게 됨과 학습 한 모든 경우에 대해서 가장 높은 Value를 확인할 수 있었다. 더불어서, Pretrained 사용하지 않고 만든 모델의 Value와 차이을 알 수 있었다.
- Learning Rate Value에 변화를 줄 때, LR Value가 커질수록 학습 안정화에 필요한 Epoch의 수가 증가하였다. 이 사항이 더 궁금해서 좀 더 알아본 내용은 다음과 같다.

'학습률이 클수록 Validation accuracy가 안정화되는 Epoch 값이 커지는 경향이 있는 이유는, 학습률이 크면 모델이 빠르게 갱신되며 파라미터가 급격하게 변화하게 되어, 최적화 과정에서 진동하거나 손실 함수의 최저점을 지나칠 가능성이 커집니다. 이런 경우, 모델이 수렴하는 데 시간이 더 걸리고, 충분히 학습이 이루어질 때까지 더 많은 epoch이 필요할 수 있습니다.'



## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

추가적으로 이 사항은 모델 학습 결과로써 알게된 내용과는 살짝 거리가 있긴 하지만,

ResNet-18의 이미지 입력을 .jpg 파일로 해야하는 것을 알게 되었다.

RGB 이미지를 입력으로 사용하며 이는 세 개의 채널(R, G, B)을 포함하는 .jpg 파일이 그것이다.

이 이야기를 하게된 것은 데이터 전처리 중 배경을 제거하는 과정에서 RemBG를 이용하였고

그 결과로 .png 파일을 내보냈었다.

.png 파일들을 먼저 작성했었던 분석 학습에 투입하였는데 학습이 시작조차 되지 않았고

그 이유를 알아보니 위에 말한대로 ResNet-18은 이미지 기본 입력이 3채널 RGB 이미지였다.

.png는 3채널 RGB에 투명 정도를 나타내는 A 채널까지 있다고 한다.

(뒷 페이지에 이어서...)



## [모델 학습]

Dataset 파일 중 Training 이미지 파일을 바탕으로 각 Class(스타일)을 분류하는 모델을 학습

구글링을 통해 여러 정보를 찾아본 끝에, .png 사진을 불러와서

`image = Image.open(img_path).convert('RGB')` 와 같은 코드를 거치면 된다는 것을 알게 되었다.

하지만 이어지는 Mission 2에서 다루게되는 JSON 파일 내에는 다 .jpg로 되어있기에 연속성을 위해 그 파일 확장자로 배경 제거를 다시 수행했었다.

이 과정을 통해서 .png와 .jpg의 차이점을 확실하게 알 수 있었다.



## Mission 2. 패션 스타일 선호 여부 예측

- 2-1. 주어진 라벨링 데이터의 파일 명은 아래와 같은 형식이다.“{W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}\_{설문ID}.json”이에 기반하여 “설문ID” 수 기준으로 “성별 & 스타일” 통계치를 아래 표 형식으로 기입한다.

- ※ 이때 주어진 이미지 데이터에 존재하는 “이미지ID”를 식별하여 유효한 라벨링 데이터 대상으로만 통계치를 구해야 한다. (이미지ID 기준으로 라벨링 데이터에는 있지만, 이미지 데이터에는 없는 경우가 있음)
- ※ Training, Validation 데이터에 대해서 각각 통계표를 작성한다.

성별	스타일	이미지 수
여성	feminine	
	classic	
	minimal	
	popart	
	...	
남성	ivy	
	mods	
	hippie	
	bold	
	...	

```

W_96626_60_minimal_W_008455.json
W_96626_60_minimal_W_234988.json
W_96632_60_minimal_W_008463.json
W_96632_60_minimal_W_234973.json
W_96634_60_minimal_W_008471.json
W_96634_60_minimal_W_234981.json
W_96637_60_minimal_W_008472.json
W_96637_60_minimal_W_234989.json
W_96643_60_minimal_W_018557.json
W_96643_60_minimal_W_234974.json
W_96645_60_minimal_W_018565.json
W_96645_60_minimal_W_234982.json
W_96646_60_minimal_W_018573.json
W_96646_60_minimal_W_234990.json

```



# Overall Process to Solving the Problem

목표: 주어진 라벨 데이터(Training 및 Validation)를 성별 및 스타일별로 분석한 통계표 작성

데이터 형식: 이미지 데이터들의 파일 명이 {W/T}\_{이미지ID}\_{시대별}\_{스타일별}\_{성별}\_{설문ID}.json로 구성.

조건: 이미지 데이터에 존재하는 “이미지ID”를 식별하여 유효한 라벨링 데이터 대상으로만 통계치를 구해야 한다.  
(이미지ID 기준으로 라벨링 데이터에는 있지만, 이미지 데이터에는 없는 경우가 있음)

Training, Validation 데이터에 대해서 각각 통계표를 작성

## [라벨링 파일의 구조]

JSON 파일 내 정보 구조 확인

문제 조건 확립

## [성별 및 스타일 통계 결과]

Training 라벨링 데이터에 대한 통계표 작성

Validation 라벨링 데이터에 대한 통계표 작성

통계표를 시각화 그래프로 표현



# [라벨링 파일의 구조]

## JSON 파일 내 정보 구조 확인

이미지 데이터에 존재하는 “이미지ID”를 식별하여 유효한 라벨링 데이터 대상으로만 통계치를 구해야 한다는 조건이 있기에 (이미지ID 기준으로 라벨링 데이터에는 있지만, 이미지 데이터에는 없는 경우가 있음), JSON 파일 구조에 대해서 잘 알아야 한다고 생각한다.

### 최상위 키:

"E\_id": 해당 레코드의 고유 식별자 (예: 234975).

"imgName": 이미지 파일명 (예: "W\_96649\_60\_space\_W.jpg").

### "item" 객체: 이미지와 관련된 세부 정보.

"imgName": 이미지 파일명. 최상위 "imgName"과 동일한 값을 가짐 (예: "W\_96649\_60\_space\_W.jpg").

"era": 이미지의 시대 정보 (예: 1960).

"style": 이미지의 스타일 정보 (예: "space").

"gender": 이미지의 성별 ("W"는 여성, "M"은 남성).

"survey": 설문 조사 항목에 대한 응답을 포함.

"Q1", "Q2", "Q3" 등: 각 설문 문항에 대한 응답 값. 숫자 형태로 응답 결과가 저장됨.

"Q5": 스타일 선호 여부 (예: 2).

스타일 선호도는 "1: 비선호", "2: 선호"로 표현됨.

"user" 객체: 설문에 참여한 응답자 정보.

"R\_id": 응답자 ID (예: 68254).

"r\_gender": 응답자 성별 (1: 남성, 2: 여성).

"age": 응답자의 연령대 (1: 20대 이하, 2: 30대, 등).

"mar": 결혼 여부 (1: 미혼, 2: 기혼).

"job": 직업군 (예: 4).

"income": 소득 수준 (1: 낮음, 2: 중간, 3: 높음).

"r\_style1", "r\_style2", ... "r\_style5": 응답자가 선호하는 스타일에 대한 정보. 각 값은 스타일에 대한 선호도 수준을 나타냄.



## [라벨링 파일의 구조]

### 문제 조건 확립

주어진 라벨링 데이터에서 이미지ID가 실제 이미지 데이터에 존재하는 경우에만 해당 데이터를 사용하여 통계치를 계산하여야 한다. 불필요하거나 잘못된 라벨 데이터를 걸러내고, 학습 및 평가에 적합한 정확한 라벨링 데이터를 사용하기 위한 준비과정이라 생각한다.

따라서, JSON 파일 내 imgName 항목의 값을 바탕으로 Training 및 Validation 이미지 데이터와 매칭이 되는 파일만 카운트 하여 통계치를 구하도록 하여야 한다.



# [성별 및 스타일 통계 결과]

Training 라벨링 데이터에 대한 통계표 작성

	성별	스타일	이미지 수	
1	M	ivy	1719	
2	M	sportivecasual	1657	
3	M	mods	1615	
4	M	hippie	1571	
5	M	hiphop	1321	
6	M	metrosexual	1226	
7	M	bold	1144	
8	M	normcore	1096	
9	W	sportivecasual	1143	
10	W	minimal	780	
11	W	feminine	762	
12	W	normcore	646	
13	W	powersuit	623	
14	W	classic	504	
15	W	bodyconscious	500	
16	W	athleisure	424	
17	W	kitsch	374	
18	W	oriental	369	
19	W	hippie	339	
20	W	cityglam	329	
21	W	ecology	295	
22	W	hiphop	243	
23	W	punk	237	
24	W	genderless	234	
25	W	lounge	207	
26	W	popart	195	
27	W	lingerie	188	
28	W	space	184	
29	W	disco	141	
30	W	grunge	122	
31	W	military	115	

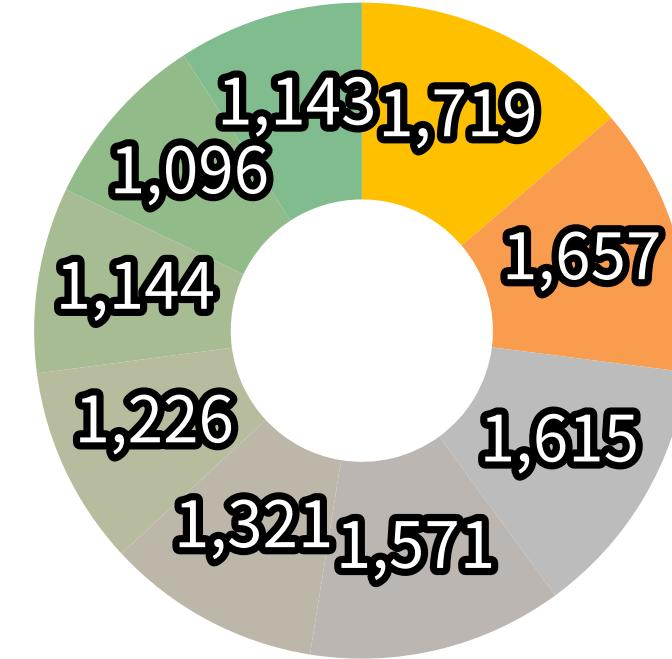




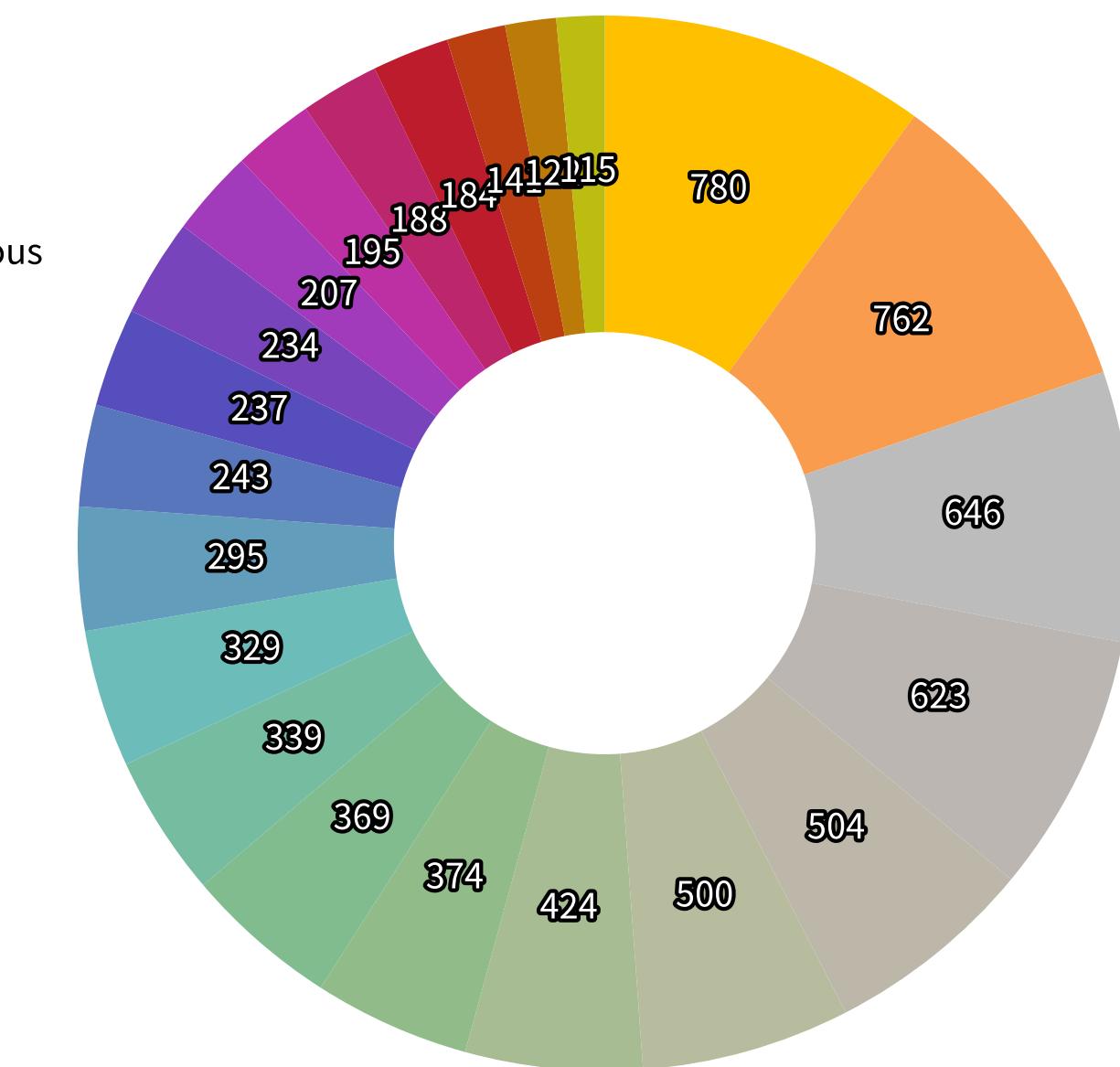
# [성별 및 스타일 통계 결과]

Training 라벨링 데이터에 대한 통계표를 시각화 그래프로 표현

- █ ivy
- █ sportivecasual
- █ mods
- █ hippie
- █ hiphop
- █ metrosexual
- █ bold
- █ normcore
- █ sportivecasual



- █ minimal
- █ feminine
- █ normcore
- █ powersuit
- █ classic
- █ bodyconscious
- █ athleisure
- █ kitsch
- █ oriental
- █ hippie
- █ cityglam
- █ ecology
- █ hiphop
- █ punk
- █ genderless
- █ lounge
- █ popart
- █ lingerie
- █ space
- █ disco
- █ grunge
- █ military





# [성별 및 스타일 통계 결과]

Validation 라벨링 데이터에 대한 통계표 작성

번호	성별	스타일	이미지 수
1	M	ivy	537
2	M	hippie	474
3	M	mods	438
4	M	hiphop	259
5	M	metrosexual	224
6	M	bold	221
7	M	sportivecasual	180
8	M	normcore	102
9	W	sportivecasual	278
10	W	feminine	208
11	W	minimal	171
12	W	powersuit	160
13	W	classic	117
14	W	bodyconscious	114
15	W	athleisure	80
16	W	space	75

17	W	oriental	71
18	W	ecology	70
19	W	cityglam	65
20	W	kitsch	61
21	W	normcore	50
22	W	hippie	46
23	W	popart	38
24	W	punk	38
25	W	military	32
26	W	hiphop	32
27	W	disco	31
28	W	grunge	29
29	W	genderless	24
30	W	lounge	20
31	W	lingerie	17



# [성별 및 스타일 통계 결과]

Validation 라벨링 데이터에 대한 통계표를 시각화 그래프로 표현





## Mission 2. 패션 스타일 선호 여부 예측

- 2-2. 2-1에서 구한 유효한 라벨링 데이터만 따로 분리하여 아래와 같이 100명 응답자의 “스타일 선호 정보표”를 구한다. 파일은 json 포맷으로 되어 있으며 json 필드 중, “응답자ID”는 “user>R\_id”로 알 수 있고, “스타일 선호 여부”는 “item>survey>Q5”로 알 수 있다.

※ 스타일 선호도 값은 “1: 비선호”, “2: 선호”이다.

응답자 ID	Training		Validation	
	스타일 선호	스타일 비선호	스타일 선호	스타일 비선호
64747	W_07894_00_cityglam_W.jpg	W_44386_80_powersuit_W.jpg	W_05628_00_cityglam_W.jpg	W_34024_10_sportivecasual_W.jpg
	W_37160_70_punk_W.jpg	W_34573_10_sportivecasual_W.jpg	W_37491_70_military_W.jpg	W_11610_90_grunge_W.jpg
	W_39725_19_normcore_W.jpg	W_40876_70_punk_W.jpg	W_38588_19_genderless_W.jpg	W_47169_70_hippie_W.jpg
	...	...	...	...
...				

```

{
  "E_id": 18566,
  "imgName": "W_96651_60_minimal_W.jpg",
  "item": {
    "imgName": "W_96651_60_minimal_W.jpg",
    "era": 1960,
    "style": "minimal",
    "gender": "W",
    "survey": {
      "Q1": 2,
      "Q2": 1,
      "Q3": 1,
      "Q411": 1,
      "Q412": 2,
      "Q413": 2,
      "Q414": 2,
      "Q4201": 0,
      "Q4202": 0,
      "Q4203": 0,
      "Q4204": 4,
      "Q4205": 0,
      "Q4206": 0,
      "Q4207": 0,
      "Q4208": 0,
      "Q4209": 0,
      "Q4210": 0,
      "Q4211": 0,
      "Q4212": 0,
      "Q4213": 0,
      "Q4214": 14,
      "Q4215": 0,
      "Q4216": 0,
      "Q5": 2
    }
  },
  "user": {
    "R_id": 14762,
    "r_gender": 2,
    "age": 1,
    "mar": 1,
    "job": 5,
    "income": 1,
    "r_style1": 1,
    "r_style2": 1,
    "r_style3": 2,
    "r_style4": 2,
    "r_style5": 1
  }
}

```



# Overall Process to Solving the Problem

목표: Mission 2-1에서 추출한 유효한 라벨링 데이터를 바탕으로 100명의 응답자에 대한  
스타일 선호 정보를 작성

데이터 형식: .json 라벨링 파일의 필드 중, “user>R\_id”인 응답자와, “item>survey>Q5”인  
“스타일 선호 여부”를에 대한 정보로 구성

조건: 문제에 제시된 표 형식 최대한 유지 (멘토링 시간 때 전달받은 사항!)

## [라벨링 파일을 통해 이미지 파일 선정]

JSON 파일 내 user>R\_id, item>survey>Q5 요소를 참조하여

각 사용자의 Training 및 Validation 이미지의 선호, 비선호를 분류

## [분류된 정보 내보내기]

Mission 3의 3-2의, item-based filtering을 활용하여 이미지 간 유사도 측정을 위해 통계 정보 내보내기

## [내보냈던 JSON 파일을 바탕으로 상위 100명 응답자에 대한 스타일 정보표 작성]

상위 100명 응답자에 대한 스타일 정보표 작성



# [라벨링 파일을 통해 이미지 파일 선정]

JSON 파일 내 user>R\_id, item>survey>Q5 요소를 참조하여  
각 사용자의 Training 및 Validation 이미지의 선호, 비선호를 분류

## 1. 유효 이미지 ID 수집

- Training 이미지와 Validation 이미지에서 유효 이미지 ID 수집
- collect\_image\_ids() 함수를 사용하여 유효한 이미지 ID 추출

### [collect\_image\_ids()]

이미징 디렉토리 순회 → 이미지 ID 수집



# [라벨링 파일을 통해 이미지 파일 선정]

JSON 파일 내 user>R\_id, item>survey>Q5 요소를 참조하여  
각 사용자의 Training 및 Validation 이미지의 선호, 비선호를 분류

## 2. 선호/비선호 데이터 수집

- Training 데이터와 Validation 데이터의 라벨링 파일을 순회하며  
 유효 이미지 ID가 포함된 파일에 대해 선호 데이터를 수집
- collect\_preference\_data() 함수를 사용
- 각 JSON 파일을 열어 응답자 ID, 스타일, 선호 여부 수집
- 응답자 ID를 기준으로 선호/비선호 스타일 목록에 저장

## [collect\_image\_data()]

라벨링 디렉토리 순회 → 이미지 ID 확인 → JSON 데이터 읽기  
→ 데이터 추출 → 선호/비선호 스타일 분류  
→ 선호('선호') 또는 비선호('비선호') 여부에 따라 해당 스타일을  
 응답자 ID에 저장



# [라벨링 파일을 통해 이미지 파일 선정]

JSON 파일 내 user>R\_id, item>survey>Q5 요소를 참조하여  
각 사용자의 Training 및 Validation 이미지의 선호, 비선호를 분류

## 3. Training 및 Validation 데이터로부터 공통된 응답자 찾기

- Training 데이터와 Validation 데이터의 응답자 ID를 비교하여 공통된 응답자 추출
- 공통된 응답자 ID 목록 생성 후 응답자별 총 응답 수 계산
- 각 응답자에 대해 Training과 Validation 데이터에서 선호/비선호 스타일 수를 합산하여 총 응답 수 계산
- 총 응답수 기준 상위 100명 응답자 선택

## 4. 총 응답 수 기준으로 상위 100명의 응답자를 추출

- 응답자 리스트 top\_100\_respondents 생성



## [분류된 정보 내보내기]

Mission 3의 3-2의, item-based filtering을 활용하여 이미지 간 유사도 측정을 위해 통계 정보 내보내기

### 5. 상위 100명에 대한 선호 데이터 생성

- generate\_preference\_json() 함수를 사용하여 상위 100명의 응답자에 대한 선호/비선호 데이터를 JSON 구조로 생성
- Training 및 Validation 데이터에서 각 응답자에 대한 선호 정보를 정리

#### [generate\_preference\_json()]

응답자 순회 → 응답자 데이터 추가

### 6. JSON 파일로 저장

- 생성된 JSON 데이터를 .ipynb 파일이 위치한 디렉토리에 'top\_100\_preference.json' 파일로 저장



# [분류된 정보 내보내기]

Mission 3의 3-2의, item-based filtering을 활용하여 이미지 간 유사도 측정을 위해 통계 정보 내보내기

```

top_100_preference.json X
mission2 > {} top_100_preference.json > {} 64747 > {} Training > [ ]선호 > 11
  2   "64747": {
  3     "Training": {
  4       "선호": [
  5         "W_46907_80_powersuit_W.jpg",
  6         "W_14901_90_kitsch_W.jpg",
  7         "W_31416_70_hippie_W.jpg",
  8         "W_44330_10_sportivecasual_W.jpg",
  9         "W_03194_50_classic_W.jpg",
 10        "W_30399_19_genderless_W.jpg",
 11        "W_20598_70_military_W.jpg",
 12        "W_07894_00_cityglam_W.jpg",
 13        "W_38629_80_powersuit_W.jpg",
 14        "W_38421_10_athleisure_W.jpg",
 15        "W_21483_19_genderless_W.jpg",
 16        "W_34636_00_oriental_W.jpg",
 17        "W_21223_80_powersuit_W.jpg",
 18        "W_22057_19_genderless_W.jpg",
 19        "W_30454_60_minimal_W.jpg",
 20        "W_35674_60_minimal_W.jpg",
 21        "W_39725_19_normcore_W.jpg",
 22        "W_30434_60_minimal_W.jpg",
 23        "W_04972_90_kitsch_W.jpg",
 24        "W_48628_19_genderless_W.jpg",
 25        "W_22510_80_powersuit_W.jpg",
 26        "W_37160_70_punk_W.jpg",
 27        "W_05628_00_cityglam_W.jpg"
 28      ],
 29      "비선호": [
 30        "W_36644_00_oriental_W.jpg",
 31        "W_34024_10_sportivecasual_W.jpg",
 32        "W_14102_50_feminine_W.jpg",
 33        "W_40876_70_punk_W.jpg",
 34        "W_40690_80_bodyconscious_W.jpg",
 35        "W_02247_50_classic_W.jpg",
 36        "W_13904_50_feminine_W.jpg",
 37        "W_18951_50_feminine_W.jpg",
 38        "W_47169_70_hippie_W.jpg",
 39        "W_33026_90_hiphop_W.jpg",
 40        "W_02498_50_feminine_W.jpg",
 41        "W_37025_19_lounge_W.jpg",
 42      ]
    }
  }
}

FFFFinal.ipynb  top_100_preference.json X mission3_json.ipynb mission3_json_knn3.ipynb missik ...
mission2 > {} top_100_preference.json > {} 63405 > {} Validation
  2   "64747": {
  3     "Validation": {
  4       "선호": [
  5         "W_44330_10_sportivecasual_W.jpg",
  6         "W_39164_00_oriental_W.jpg",
  7         "W_37491_70_military_W.jpg",
  8         "W_20598_70_military_W.jpg",
  9         "W_30988_90_kitsch_W.jpg",
 10        "W_38588_19_genderless_W.jpg",
 11        "W_22510_80_powersuit_W.jpg",
 12        "W_05628_00_cityglam_W.jpg"
 13      ],
 14      "비선호": [
 15        "W_34024_10_sportivecasual_W.jpg",
 16        "W_14102_50_feminine_W.jpg",
 17        "W_47169_70_hippie_W.jpg",
 18        "W_02498_50_feminine_W.jpg",
 19        "W_11610_90_grunge_W.jpg",
 20        "W_27828_60_minimal_W.jpg"
 21      ]
 22    }
 23  },
 24  "63405": {
 25    "Training": {
 26      "선호": [
 27        "W_00555_50_ivy_M.jpg",
 28        "W_04206_60_mods_M.jpg",
 29        "W_12551_19_normcore_M.jpg",
 30        "W_17849_00.metrosexual_M.jpg",
 31        "W_01853_60_mods_M.jpg",
 32        "W_06682_00.metrosexual_M.jpg",
 33        "W_06785_50_ivy_M.jpg",
 34        "W_17510_19_normcore_M.jpg",
 35        "W_15400_60_mods_M.jpg",
 36        "W_00023_60_mods_M.jpg",
 37        "W_02931_00.metrosexual_M.jpg",
 38        "W_01670_10_sportivecasual_M.jpg",
 39        "W_15294_50_ivy_M.jpg",
 40        "W_17909_60_mods_M.jpg",
 41        "W_02890_19_normcore_M.jpg"
 42      ]
    }
  }
}

```

줄 16, 열 29 | 공백: 4 | UTF-8 | LF | { JSON | Q }

top\_100\_preference.json



# [표 작성]

내보냈던 JSON 파일을 바탕으로 상위 100명 응답자에 대한 스타일 정보표 작성

## 7. JSON 데이터를 순회하며 응답자 데이터 수집

- 생성된 JSON 데이터를 순회하며 응답자 ID, Training 및 Validation 데이터의 선호/비선호 스타일을 수집
- 데이터를 리스트(rows)에 추가하여 각 응답자에 대한 스타일을 열로 분리

## 8. Pandas 데이터프레임 생성

- pandas를 사용하여 응답자 데이터를 정리한 데이터프레임(df) 생성
- 각 응답자의 Training 선호 스타일, Training 비선호 스타일, Validation 선호 스타일, Validation 비선호 스타일을 열로 정리
- 데이터프레임의 인덱스 설정 후 출력



# [표 작성]

## 내보냈던 JSON 파일을 바탕으로 상위 100명 응답자에 대한 스타일 정보표 작성

응답자 ID	Training 선호 스타일	Training 비선호 스타일	Validation 선호 스타일	Validation 비선호 스타일
1 64747	W_29783_10_sportivecasual_W.jpg, W_46907_80_powersuit_W.jpg, W_14901_90_kitsch_W.jpg, W_31416_70_hippie_W.jpg, W_44330_10_sportivecasual_W.jpg, W_03194_50_classic_W.jpg, W_30399_19_genderless_W.jpg, W_20598_70_military_W.jpg, W_07894_00_cityglam_W.jpg, W_38629_80_powersuit_W.jpg, W_38421_10_athleisure_W.jpg, W_21483_19_genderless_W.jpg, W_34636_00_oriental_W.jpg, W_21223_80_powersuit_W.jpg, W_22057_19_genderless_W.jpg, W_30454_60_minimal_W.jpg, W_35674_60_minimal_W.jpg, W_39725_19_normcore_W.jpg, W_30434_60_minimal_W.jpg, W_04972_90_kitsch_W.jpg, W_48628_19_genderless_W.jpg, W_22510_80_powersuit_W.jpg, W_37160_70_punk_W.jpg, W_05628_00_cityglam_W.jpg	W_36644_00_oriental_W.jpg, W_34024_10_sportivecasual_W.jpg, W_14102_50_feminine_W.jpg, W_40876_70_punk_W.jpg, W_40690_80_bodyconscious_W.jpg, W_02247_50_classic_W.jpg, W_13904_50_feminine_W.jpg, W_18951_50_feminine_W.jpg, W_47169_70_hippie_W.jpg, W_33026_90_hiphop_W.jpg, W_02498_50_feminine_W.jpg, W_37025_19_lounge_W.jpg, W_11610_90_grunge_W.jpg, W_27828_60_minimal_W.jpg, W_42595_60_popart_W.jpg, W_44386_80_powersuit_W.jpg, W_41633_60_space_W.jpg, W_38771_10_sportivecasual_W.jpg, W_48378_90_grunge_W.jpg, W_03643_00_cityglam_W.jpg, W_34573_10_sportivecasual_W.jpg	W_46907_80_powersuit_W.jpg, W_44330_10_sportivecasual_W.jpg, W_39164_00_oriental_W.jpg, W_37491_70_military_W.jpg, W_20598_70_military_W.jpg, W_30988_90_kitsch_W.jpg, W_38588_19_genderless_W.jpg, W_22510_80_powersuit_W.jpg, W_05628_00_cityglam_W.jpg	W_34024_10_sportivecasual_W.jpg, W_14102_50_feminine_W.jpg, W_47169_70_hippie_W.jpg, W_02498_50_feminine_W.jpg, W_11610_90_grunge_W.jpg, W_27828_60_minimal_W.jpg
2 63405	W_00555_50_ivy_M.jpg, W_04206_60_mods_M.jpg, W_12551_19_normcore_M.jpg, W_17849_00.metrosexual_M.jpg, W_01853_60_mods_M.jpg, W_06682_00.metrosexual_M.jpg, W_06785_50_ivy_M.jpg, W_17510_19_normcore_M.jpg, W_15400_60_mods_M.jpg, W_00023_60_mods_M.jpg, W_02931_00.metrosexual_M.jpg, W_01670_10_sportivecasual_M.jpg, W_15294_50_ivy_M.jpg, W_17909_60_mods_M.jpg, W_02890_19_normcore_M.jpg, W_07021_90_hiphop_M.jpg, W_06812_50_ivy_M.jpg, W_16084_80_bold_M.jpg, W_04684_90_hiphop_M.jpg, W_04522_50_ivy_M.jpg, W_12814_19_normcore_M.jpg, W_07198_10_sportivecasual_M.jpg	W_15472_70_hippie_M.jpg, W_00530_10_sportivecasual_M.jpg, W_17454_80_bold_M.jpg, W_12456_10_sportivecasual_M.jpg, W_16755_00.metrosexual_M.jpg, W_16736_00.metrosexual_M.jpg, W_17219_70_hippie_M.jpg, W_12383_80_bold_M.jpg, W_12904_50_ivy_M.jpg, W_17443_90_hiphop_M.jpg, W_17599_19_normcore_M.jpg, W_02714_00.metrosexual_M.jpg, W_16501_70_hippie_M.jpg, W_06691_10_sportivecasual_M.jpg, W_16919_19_normcore_M.jpg, W_15517_70_hippie_M.jpg, W_15471_10_sportivecasual_M.jpg, W_15134_80_bold_M.jpg, W_15140_80_bold_M.jpg, W_12443_90_hiphop_M.jpg, W_17108_19_normcore_M.jpg, W_15782_70_hippie_M.jpg	W_01853_60_mods_M.jpg, W_15294_50_ivy_M.jpg, W_02879_90_hiphop_M.jpg, W_02677_60_mods_M.jpg, W_06860_19_normcore_M.jpg, W_04684_90_hiphop_M.jpg, W_04522_50_ivy_M.jpg	W_16755_00.metrosexual_M.jpg, W_12304_80_bold_M.jpg, W_12904_50_ivy_M.jpg, W_17443_90_hiphop_M.jpg, W_16501_70_hippie_M.jpg, W_07187_70_hippie_M.jpg, W_15140_80_bold_M.jpg
3 64346	W_12527_70_hippie_M.jpg, W_16777_60_mods_M.jpg, W_00856_10_sportivecasual_M.jpg, W_26175_50_ivy_M.jpg, W_23922_50_ivy_M.jpg, W_21401_00.metrosexual_M.jpg, W_23920_19_normcore_M.jpg, W_16233_80_bold_M.jpg, W_23905_60_mods_M.jpg, W_24874_80_bold_M.jpg, W_12155_80_bold_M.jpg, W_24103_50_ivy_M.jpg, W_09289_19_normcore_M.jpg, W_24977_70_hippie_M.jpg, W_49510_00.metrosexual_M.jpg, W_00047_70_hippie_M.jpg, W_09233_60_mods_M.jpg, W_29990_90_hiphop_M.jpg, W_09856_10_sportivecasual_M.jpg, W_28693_10_sportivecasual_M.jpg, W_24268_00.metrosexual_M.jpg, W_30040_60_mods_M.jpg, W_02754_70_hippie_M.jpg, W_09154_50_ivy_M.jpg, W_09239_19_normcore_M.jpg, W_29918_19_normcore_M.jpg	W_16121_80_bold_M.jpg, W_07112_90_hiphop_M.jpg, W_16822_19_normcore_M.jpg, W_24250_90_hiphop_M.jpg, W_25721_90_hiphop_M.jpg, W_33015_19_normcore_M.jpg, W_25543_60_mods_M.jpg, W_15769_00.metrosexual_M.jpg, W_28908_90_hiphop_M.jpg, W_24243_70_hippie_M.jpg, W_24838_70_hippie_M.jpg, W_24931_50_ivy_M.jpg, W_25828_10_sportivecasual_M.jpg, W_25391_00.metrosexual_M.jpg, W_24832_80_bold_M.jpg, W_17172_19_normcore_M.jpg, W_02678_50_ivy_M.jpg, W_25065_10_sportivecasual_M.jpg, W_12268_80_bold_M.jpg, W_16390_10_sportivecasual_M.jpg	W_07316_00.metrosexual_M.jpg, W_24103_50_ivy_M.jpg, W_29990_90_hiphop_M.jpg, W_09154_50_ivy_M.jpg, W_29918_19_normcore_M.jpg	W_16430_90_hiphop_M.jpg, W_16121_80_bold_M.jpg, W_24250_90_hiphop_M.jpg, W_26099_19_normcore_M.jpg, W_24838_70_hippie_M.jpg, W_24931_50_ivy_M.jpg, W_00496_60_mods_M.jpg
	W_35091_80_powersuit_W.jpg, W_49853_00_oriental_W.jpg, W_18205_50_feminine_W.jpg, W_36601_19_normcore_W.jpg, W_34974_60_minimal_W.jpg, W_45035_10_sportivecasual_W.jpg	W_32870_90_hiphop_W.jpg, W_32939_70_military_W.jpg, W_41279_19_genderless_W.jpg, W_22943_10_athleisure_W.jpg, W_46417_70_military_W.jpg, W_08584_50_feminine_W.jpg,	W_35091_80_powersuit_W.jpg,	W_34024_10_sportivecasual_W.jpg,



# [표 작성]

내보냈던 JSON 파일을 바탕으로 상위 100명 응답자에 대한 스타일 정보표 작성

...

		W_20475_00_military_W.jpg, W_00050_50_normcore_W.jpg, W_33862_00_oriental_W.jpg, W_09961_50_feminine_W.jpg		
98	64662	W_34872_70_military_W.jpg, W_48434_10_sportivecasual_W.jpg, W_11500_50_feminine_W.jpg, W_22718_60_space_W.jpg, W_29108_90_kitsch_W.jpg, W_21898_80_bodyconscious_W.jpg, W_35009_10_sportivecasual_W.jpg, W_34808_10_sportivecasual_W.jpg, W_46905_80_powersuit_W.jpg, W_36130_00_ecology_W.jpg, W_20551_70_hippie_W.jpg, W_43283_90_grunge_W.jpg, W_40520_19_genderless_W.jpg, W_31559_90_kitsch_W.jpg, W_22067_19_normcore_W.jpg, W_40817_70_military_W.jpg, W_22180_00_oriental_W.jpg, W_37426_00_oriental_W.jpg, W_45749_90_grunge_W.jpg, W_46922_00_cityglam_W.jpg, W_14425_10_sportivecasual_W.jpg, W_10338_50_classic_W.jpg, W_43763_00_oriental_W.jpg, W_22856_60_minimal_W.jpg, W_07386_50_feminine_W.jpg, W_22210_70_disco_W.jpg, W_24433_19_genderless_W.jpg, W_41537_10_sportivecasual_W.jpg, W_20669_60_minimal_W.jpg, W_46562_80_powersuit_W.jpg, W_40322_60_space_W.jpg, W_47546_60_minimal_W.jpg, W_28695_90_kitsch_W.jpg, W_32299_80_powersuit_W.jpg	W_34808_10_sportivecasual_W.jpg, W_36130_00_ecology_W.jpg, W_43283_90_grunge_W.jpg, W_22623_00_cityglam_W.jpg, W_33876_60_popart_W.jpg, W_47546_60_minimal_W.jpg, W_18391_50_classic_W.jpg	
99	7658	W_19067_50_classic_W.jpg, W_13271_60_minimal_W.jpg, W_08382_19_normcore_W.jpg, W_19531_00_oriental_W.jpg, W_01050_80_bodyconscious_W.jpg, W_04877_00_oriental_W.jpg, W_00334_70_hippie_W.jpg, W_11169_19_normcore_W.jpg, W_13535_80_powersuit_W.jpg, W_08112_90_hiphop_W.jpg, W_05312_80_bodyconscious_W.jpg, W_08383_10_athleisure_W.jpg, W_02345_60_space_W.jpg, W_03201_50_feminine_W.jpg, W_15077_19_normcore_W.jpg, W_18658_80_powersuit_W.jpg, W_02456_10_sportivecasual_W.jpg, W_13772_70_hippie_W.jpg, W_07851_70_disco_W.jpg, W_14178_70_hippie_W.jpg, W_19543_00_oriental_W.jpg, W_05499_90_kitsch_W.jpg, W_13455_80_powersuit_W.jpg, W_07975_80_powersuit_W.jpg, W_14023_90_lingerie_W.jpg, W_13688_90_hiphop_W.jpg, W_08680_00_cityglam_W.jpg, W_14380_90_hiphop_W.jpg, W_13941_60_minimal_W.jpg, W_13266_19_normcore_W.jpg, W_18996_50_feminine_W.jpg, W_13081_50_classic_W.jpg, W_00344_90_grunge_W.jpg, W_10510_60_space_W.jpg, W_00682_70_punk_W.jpg	W_13535_80_powersuit_W.jpg, W_19003_50_feminine_W.jpg, W_05312_80_bodyconscious_W.jpg, W_02345_60_space_W.jpg, W_14923_60_minimal_W.jpg, W_13688_90_hiphop_W.jpg, W_10510_60_space_W.jpg	
100	64295	W_17151_19_normcore_M.jpg, W_01664_60_mods_M.jpg, W_03013_10_sportivecasual_M.jpg, W_31375_19_normcore_M.jpg, W_12803_70_hippie_M.jpg, W_25380_60_mods_M.jpg, W_15846_80_bold_M.jpg, W_28460_10_sportivecasual_M.jpg, W_24253_00_metrosexual_M.jpg, W_30027_50_ivy_M.jpg, W_02878_90_hiphop_M.jpg, W_16297_80_bold_M.jpg, W_28350_19_normcore_M.jpg, W_25465_00_metrosexual_M.jpg, W_10098_50_ivy_M.jpg	W_16515_10_sportivecasual_M.jpg, W_16374_10_sportivecasual_M.jpg, W_16150_50_ivy_M.jpg, W_12573_90_hiphop_M.jpg, W_04683_10_sportivecasual_M.jpg, W_25879_00_metrosexual_M.jpg, W_32008_80_bold_M.jpg, W_20873_70_hippie_M.jpg, W_25244_19_normcore_M.jpg, W_26062_80_bold_M.jpg, W_23900_50_ivy_M.jpg, W_31791_00_metrosexual_M.jpg, W_25280_90_hiphop_M.jpg, W_26874_70_hippie_M.jpg, W_12548_70_hippie_M.jpg, W_02726_00_metrosexual_M.jpg, W_29059_70_hippie_M.jpg, W_27750_60_mods_M.jpg, W_02742_60_mods_M.jpg, W_29487_80_bold_M.jpg, W_33169_00_metrosexual_M.jpg, W_25678_70_hippie_M.jpg, W_28645_10_sportivecasual_M.jpg, W_04237_60_mods_M.jpg, W_26196_50_ivy_M.jpg	W_16374_10_sportivecasual_M.jpg, W_25761_90_hiphop_M.jpg, W_15729_90_hiphop_M.jpg, W_32314_19_normcore_M.jpg, W_23900_50_ivy_M.jpg, W_27750_60_mods_M.jpg, W_24598_80_bold_M.jpg, W_04237_60_mods_M.jpg, W_31478_19_normcore_M.jpg



## Mission 3. 패션 스타일 선호 여부 예측

- 3-1. 추천 시스템의 기본인 협업 필터링 (Collaborative Filtering)은 크게 user-based filtering, item-based filtering 방식으로 나뉘어져 있다. 각각에 대해서 이해하고, 2-2에서 구해 본 응답자의 “스타일 선호 정보표”를 토대로 Validation 데이터 내 응답자의 “스타일 선호 여부 예측” 문제를 2가지 기법으로 어떻게 적용해 볼 수 있고, 서로 비교하여 어떤 장단점을 갖는지 설명한다.

※ 설명을 용이하게 하기 위해 응답자의 스타일 선호도 예시를 들어서 설명해도 무방하다.

- 3-2. 3-1에서 살펴 본 기법 중, item-based filtering을 직접 구현해본다. “이미지 간 유사도” (image2image)만을 활용하여 Validation 데이터 내 응답자의 “스타일 선호 여부 예측” 문제를 수행하고 성능을 측정한다.

※ Hint. 1-2에서 학습한 ResNet-18의 중간 layer 값을 활용하여 각 이미지의 feature vector를 구하고, 벡터 연산을 통해 이미지 간 유사도를 구해볼 수 있다.

※ 예측 문제에서 활용한 파라미터 및 임계 값 등의 수치를 정확하게 제시한다.



# Overall Process to Solving the Problem

목표: (3-1) 응답자의 스타일 선호 여부를 예측하기 위해 협업 필터링(Collaborative Filtering) 기법을 적용하고,  
User-based Filtering과 Item-based Filtering 방식의 장단점을 비교  
(3-2) Item-based Filtering을 직접 구현하여, Validation 데이터에서 응답자의 스타일 선호 여부를 예측

## [실제 예측 수행(Mission 3-2) 결과 제시 전에 하고자 하는 이런저런 이야기들]

1. User-based Filtering와 Item-based Filtering 방식의 차이
2. Mission 3-2를 User-based Filtering으로 적용하는 가정  
(서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)
3. Mission 3-2를 Item-based Filtering으로 적용하는 가정  
(스타일을 선호하는 응답자가 유사한 스타일 역시 선호할 가능성이 높다고 가정)
4. 유clidean 거리 기법이 아니라 코사인 유사도를 사용한 이유
5. 고차원의 저주



# Overall Process to Solving the Problem

목표: (3-1) 응답자의 스타일 선호 여부를 예측하기 위해 협업 필터링(Collaborative Filtering) 기법을 적용하고, User-based Filtering과 Item-based Filtering 방식의 장단점을 비교  
(3-2) Item-based Filtering을 직접 구현하여, Validation 데이터에서 응답자의 스타일 선호 여부를 예측

## [실제 예측 수행 결과 도출]

1. Mission 3-2의 전반적인 이해
2. 코사인 유사도 임계치 지정에 따른 결과
3. 문제 풀이 과정 속에서 알게 된 내용들



# 1. User-based Filtering와 Item-based Filtering 방식의 차이

## 협업 필터링의 개요

협업 필터링의 두 가지 접근 방식인 사용자 기반 필터링과 아이템 기반 필터링은 데이터 분석 및 추천 시스템에서 응답자 또는 아이템 간의 유사성을 활용해 예측을 수행하는 방법이다.

### User-based Filtering

스타일 선호도가 비슷한 응답자들 간의 유사도를 계산하여, 서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정하에 새로운 스타일 선호 여부를 예측한다.

예를 들어, 응답자 A와 B가 비슷한 스타일을 좋아하고 싫어하는 경향을 보인다면 A와 B가 새로운 스타일에 대해 동일한 선호도를 가질 가능성이 높다고 보는 것이다.

이를 위해 응답자 간의 유사도를 계산할 때, 코사인 유사도와 같은 방법을 사용할 수 있다.

이후 유사도가 높은 응답자들의 스타일 선호 패턴을 바탕으로 특정 응답자의 새로운 스타일에 대한 선호도를 예측한다.

사용자 기반 필터링의 장점은 응답자 개개인의 고유 취향을 반영할 수 있다는 점이다.

하지만 응답자들이 선호하는 스타일이 다양하거나 특정 스타일에 대한 데이터가 부족한 경우에는 성능이 저하될 수 있다.



# 1. User-based Filtering와 Item-based Filtering 방식의 차이

## Item-based 필터링

응답자 간의 유사도가 아닌 스타일 간 유사성에 중점을 둔다.

이 방식에서는 하나의 스타일을 선호하는 응답자가 유사한 스타일 역시 선호할 가능성이 높다고 가정한다.

예를 들어, ‘클래식’ 스타일을 좋아하는 응답자는 이와 비슷한 특징을 가진 ‘미니멀’ 스타일을 선호할 가능성이 높다고 예측하는 식이다.

아이템 기반 필터링에서는 스타일의 특징 벡터를 활용해 유사한 스타일을 계산하고, 유사한 스타일들 간의 관계를 바탕으로 예측을 수행한다.

특징 벡터는 ResNet-18 등의 모델을 통해 각 스타일의 이미지에서 추출한 데이터로,

이를 바탕으로 스타일 간의 유사도를 계산할 수 있다.

스타일 간 유사도를 계산하는 방법으로는 코사인 유사도, 유clidean 거리 등을 사용할 수 있다.

이 방식의 장점은 응답자별 선호도가 적은 상황에서도 스타일 유사성에 기반해 예측할 수 있다는 점이다.

하지만 응답자 개개인의 취향보다는 스타일 간 유사성에 기반한 예측이므로 특정 응답자의 고유한 취향을 반영하기는 어렵다는 한계가 있다.



## 2. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

### 구현 방법

각 응답자에 대해 ResNet-18의 중간 레이어에서 추출한 특징 벡터를 사용해 응답자들이 선호한 스타일 이미지를 나타내는 벡터 평균을 구하고, 이를 통해 응답자마다 선호하는 스타일의 대표 벡터를 생성한다.

### 중간레이어 근처에서 추출하는 이유와 마지막 혹은 첫 레이어가 아니라 중간레이어를 사용하는 이유

1. 중간 레이어에서 추출: 중간 레이어에 도달했을 때 해당 레이어의 활성화값을 특징 벡터로 바로 추출

- 장점
  - 효율성: 한 레이어의 출력값만 사용하기 때문에 계산이 간단하고 효율적이다.
  - 노이즈 최소화: 특정 레이어의 출력값만을 사용하므로 필요 이상의 정보가 배제되며, 상대적으로 중요한 특징만 포함되어 과적합을 방지할 수 있다.
  - 적절한 일반화 수준: 중간 레이어는 이미지의 저수준 정보와 고수준 정보가 균형 있게 포함된 특징을 제공하여, 일반화와 구체화가 균형 잡힌 벡터로서 활용될 수 있다.



## 2. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

- 단점
  - 정보 손실: 한 레이어의 출력만 사용하므로, 다른 레이어에서 추출할 수 있는 추가 정보가 반영되지 않아 세부적인 특징들이 누락된다.
  - 특정 스타일 표현 부족: 중간 레이어가 지나치게 일반화된 경우, 각 스타일의 특수성을 정확히 반영하지 못한다. 이로 인해 스타일 차이를 명확히 구별하는 데 한계가 있을 수 있다.



## 2. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

2. 중간 레이어까지 추출: 중간 레이어까지의 모든 특징을 누적하여 최종 벡터로 결합

- 장점
  - 다양한 정보 반영: 여러 레이어에서 얻은 다양한 수준의 특징을 종합하여 활용할 수 있어, 이미지에 대한 풍부한 정보를 포함한다.
  - 복합적인 스타일 표현: 초기 레이어의 저수준 특징과 중간 레이어의 고수준 특징이 모두 반영되어 복합적이고 세부적인 스타일 표현이 가능하다.  
이는 복잡한 스타일을 구분하거나 섬세한 차이를 반영할 수 있다.
- 단점
  - 계산 복잡성 증가: 여러 레이어의 출력을 결합해야 하므로 계산량이 크게 증가하고, 메모리 사용량도 늘어난다.
  - 노이즈 증가 가능성: 필요 이상의 세부 정보가 포함되어 노이즈가 증가할 수 있다.  
특히, 불필요한 세부 사항이 많아지면 중요한 특징이 묻힐 위험이 있다.
  - 과적합 가능성: 모든 정보를 포함하기 때문에 모델이 특정 데이터에 과적합될 가능성이 높아져 일반화 성능이 떨어질 수 있다. 복잡한 특징을 모두 반영하여 세부적으로 분석하려면 중간 레이어까지 추출하는 것이 좋고, 효율적인 계산과 적절한 일반화가 필요하다면 중간 레이어에서 추출하는 것이 유리하다.



## 2. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

### 3. 레이어 근처에서 특징 벡터를 추출하는 이유

- 일반화와 구체화의 균형: 중간 레이어에서는 초기 레이어에서 주로 감지하는 기본적인 형태(모서리, 텍스처, 색상)와, 마지막 레이어에서 감지하는 특정 분류를 위한 고유한 패턴 사이의 균형이 잡힌 특징 벡터를 얻을 수 있다. 이는 특정 스타일이나 이미지에 대한 일반화된 특징을 포함하면서도, 지나치게 구체적이지 않게 추상화된 정보가 유지되는 지점이다.  
즉, 다양한 스타일에 대해 효과적으로 응용될 수 있는 일반화된 특징을 추출할 수 있다.
- 노이즈 감소: 모델의 마지막 레이어는 분류를 위한 고유한 정보를 담고 있어 특정 클래스에 최적화되어 있다. 이는 다른 스타일 간 차이를 잘 구별하는데 적합하지 않을 수 있으며, 노이즈가 많이 포함될 수 있다.  
중간 레이어는 구체적인 클래스 정보가 아닌 다양한 스타일을 효과적으로 구분할 수 있는 노이즈가 적은 정보를 제공한다.
- 유사도 계산의 효율성: 중간 레이어의 벡터는 고유한 클래스에 맞춰 세분화되지 않고 다양한 스타일 간 일반적인 차이를 반영하므로 유사도를 계산할 때 과적합 없이 유효한 유사도 측정을 가능하게 한다.  
이는 여러 스타일 간 유사도를 효과적으로 측정해 특정 응답자의 선호 스타일을 예측하는 데 유리하다.



## 2. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

### 4. 마지막 레이어와 첫 레이어를 사용하지 않는 이유

- 초기 레이어: 이미지의 단순한 저수준 특징(예: 색상, 가장자리 등)을 추출하기 때문에 고차원적인 스타일 정보를 담기 어렵다. 따라서 스타일 간 미세한 차이를 파악하기 어렵고, 유사한 스타일도 정확히 구분하기 힘들다.
  - 이는 특히 응답자들의 특정 스타일 선호 여부를 파악하기 위해서는 지나치게 일반화된 특징 벡터가 되어버려, 스타일의 본질적인 차이를 반영하지 못한다.
  - 결론적으로, 중간 레이어를 사용하는 이유는 일반화와 구체화의 균형을 맞춘 특징 벡터를 얻기 위해서이다. 중간 레이어는 지나치게 구체적이거나 세분화된 정보는 배제하면서도 스타일을 구분하기에 충분히 유의미한 정보를 제공한다.
- 생성된 대표 벡터 간 유사도를 계산하여 유사한 응답자 그룹을 형성한다.  
예를 들어, 코사인 유사도를 사용해 응답자 간 스타일 선호의 유사도를 측정할 수 있다.
- Validation 데이터에서 새로운 스타일에 대한 선호 여부를 예측할 때, 유사한 응답자 그룹에서 해당 스타일을 선호하는 응답자 수를 바탕으로 예측을 수행한다.  
예를 들어, 유사한 응답자들이 주로 선호한 스타일이라면 그 스타일을 "선호"로 예측한다.



## 2. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

- 마지막 레이어: 마지막 레이어의 벡터는 주로 특정 클래스 분류를 위해 최적화되어 있어 각 스타일 간 고유한 특징을 지나치게 반영할 수 있다. 이런 경우, 유사한 스타일이라 하더라도 클래스가 다르면 벡터 간 유사도가 크게 낮아질 수 있고, 동일한 취향을 가진 응답자나 비슷한 스타일을 분류하기 어려워질 수 있다. 또한, 마지막 레이어는 분류 문제에 필요한 세부 정보가 너무 많아 지나치게 세분화된 특징을 포함한다. 이는 사용자 간의 일반적인 선호 패턴을 파악하는 데 오히려 방해가 될 수 있다.
- 장단점
  - 장점: 응답자 개개인의 취향을 고려하여 특정 응답자 맞춤형 예측을 수행할 수 있다.  
타일에 대한 선호 데이터가 충분히 쌓일수록 예측의 정확도가 증가할 수 있다.
  - 단점: 새로운 스타일에 대한 데이터가 부족한 경우 유사한 응답자 그룹이 형성되지 않아 예측이 어려울 수 있으며, 특정 응답자의 선호도가 전체 데이터에 많이 포함되어야 성능이 올라가는 경향이 있다.  
선호가 분산되어 있을 때는 효과가 떨어질 수 있다.



### 3. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

#### 구현 방법

- ResNet-18의 중간 레이어에서 추출한 특징 벡터를 사용해 스타일 이미지 간 유사도를 계산한다.  
코사인 유사도나 유클리드 거리 등을 사용해 유사도를 측정할 수 있다.
- 특정 응답자가 선호하는 스타일과 유사한 스타일을 찾기 위해 유사도가 높은 스타일들 간의 그룹을 형성한다.
- Validation 데이터에서 새로운 스타일에 대한 선호 여부를 예측할 때는 해당 응답자가 선호하는 스타일과  
유사한 스타일들의 선호 여부를 바탕으로 예측을 수행한다.  
예를 들어, 응답자가 ‘클래식’ 스타일을 선호한다면, ‘클래식’과 유사한 ‘미니멀’ 스타일을 선호할 가능성이 있다고  
예측하는 방식이다.



### 3. Mission 3-2를 User-based Filtering으로 적용하는 가정 (서로 유사한 응답자들이 비슷한 스타일을 선호할 것이라는 가정)

#### 장단점

- 장점: 특정 스타일에 대한 선호도가 적더라도 스타일 간 유사성을 바탕으로 선호도를 예측할 수 있어, 예측의 범용성을 높일 수 있다. 응답자별 선호 스타일에 대한 데이터가 부족해도 유사한 스타일 간 유사도만으로 예측이 가능해 빠르게 학습할 수 있다.
- 단점: 응답자 개개인의 취향보다는 스타일 간 유사성에 초점을 맞추므로 특정 응답자의 고유한 취향을 반영하기 어렵다. 스타일 간 유사도가 높다고 해서 반드시 선호도가 일치하지 않을 수 있어, 예측의 정확도가 떨어질 가능성이 있다.



## 4. 유클리드 거리 기법이 아니라 코사인 유사도를 사용한 이유

### 이유

코사인 유사도를 많이 사용하는 이유는 벡터의 크기보다는 방향에 초점을 맞추기 때문이다. 특히 추천 시스템이나 텍스트 마이닝과 같은 분야에서 벡터 간의 유사성을 비교할 때, 두 벡터가 가리키는 방향이 얼마나 비슷한지가 더 중요한 경우가 많다.

### 크기의 영향 제거

코사인 유사도는 벡터의 크기(길이)가 아닌 벡터의 방향을 비교하는 방법이다. 예를 들어, 두 벡터의 크기가 다르더라도 같은 방향을 향하고 있다면 유사도가 높게 나오므로, 벡터의 크기가 결과에 영향을 주지 않습니다. 추천 시스템에서는 두 사용자의 행동 패턴이 비슷한지 여부가 중요하지, 얼마나 많이 구매했는지 (크기)는 덜 중요할 수 있다.



## 4. 유클리드 거리 기법이 아니라 코사인 유사도를 사용한 이유

### 벡터의 희소성

코사인 유사도는 희소 벡터에서도 성능이 좋습니다.

예를 들어, 대형 이미지 또는 텍스트 데이터에서 만들어진 벡터는 많은 차원에 0이 포함된 희소한 형태를 가집니다.

코사인 유사도는 이러한 희소 벡터 간 비교에 효율적이며, 유클리드 거리보다 더 직관적인 결과를 제공합니다.

두 벡터가 일부 차원에서만 값을 공유해도 방향만 비슷하다면 유사도가 높게 나타나기 때문입니다.

### 고차원 데이터에서의 효율성

유클리드 거리는 고차원 데이터에서 차원의 저주(Curse of Dimensionality) 영향을 더 크게 받습니다.

즉, 차원이 커질수록 유클리드 거리가 실제 유사도를 나타내는 데 한계가 생깁니다.

반면, 코사인 유사도는 고차원에서도 벡터 방향에 초점을 맞추기 때문에 차원의 저주의 영향을 상대적으로 덜 받습니다.

요약해봤을 때 코사인 유사도는 크기를 무시하고, 방향만 비교함으로써 더 효율적인 유사도를 구할 수 있습니다.

특히 이번 경우처럼 고차원 데이터를 분석하기에 코사인 유사도를 사용하였습니다.



## 5. 고차원의 저주

### 고차원의 저주 (Curse of Dimensionality)

데이터 분석이나 기계 학습에서 발생하는 문제로, 차원이 증가할수록 데이터 공간의 크기가 기하급수적으로 커져 효과적인 분석이나 학습이 어려워지는 현상을 말한다.

이 개념은 주로 고차원 데이터를 다룰 때 나타나며, 차원이 증가할수록 알고리즘 성능이 급격히 저하될 수 있다.

### 코사인 유사도에서의 고차원의 저주

코사인 유사도는 두 벡터 간의 각도를 기준으로 유사성을 측정하는 방식이다.

코사인 유사도가 1에 가까울수록 두 벡터는 같은 방향을 가리키며, 0에 가까울수록 서로 직교한다.

고차원에서의 문제: 차원이 높아질수록 대부분의 벡터들이 서로 직교(orthogonal)에 가까워진다.

즉, 벡터 간의 각도가 비슷해지며 코사인 유사도를 기반으로 두 벡터를 구분하기 어려워진다.

이로 인해 고차원에서 벡터 간 유사성을 제대로 평가하지 못할 가능성이 커진다.



## 5. 고차원의 저주

### 유클리드 거리에서의 고차원의 저주

유클리드 거리는 두 벡터 간의 직선 거리를 측정하는 방식이다.

이는 벡터의 차이를 직접 계산하여 얼마나 떨어져 있는지 측정한다.

고차원에서의 문제: 차원이 증가할수록 유클리드 거리도 기하급수적으로 증가하여, 두 벡터 간 거리가 거의 비슷해진다. 고차원에서는 거의 모든 벡터가 서로 멀리 떨어져 있어, 거리 기반의 유사성 판단이 어려워지고 의미를 잃게 된다.

### 코사인 유사도가 유클리드 거리보다 고차원에서 유리한 이유

코사인 유사도는 고차원 벡터에서도 벡터의 크기가 아닌 방향을 기준으로 유사성을 측정하므로, 고차원에서 유클리드 거리보다 유리할 수 있다.

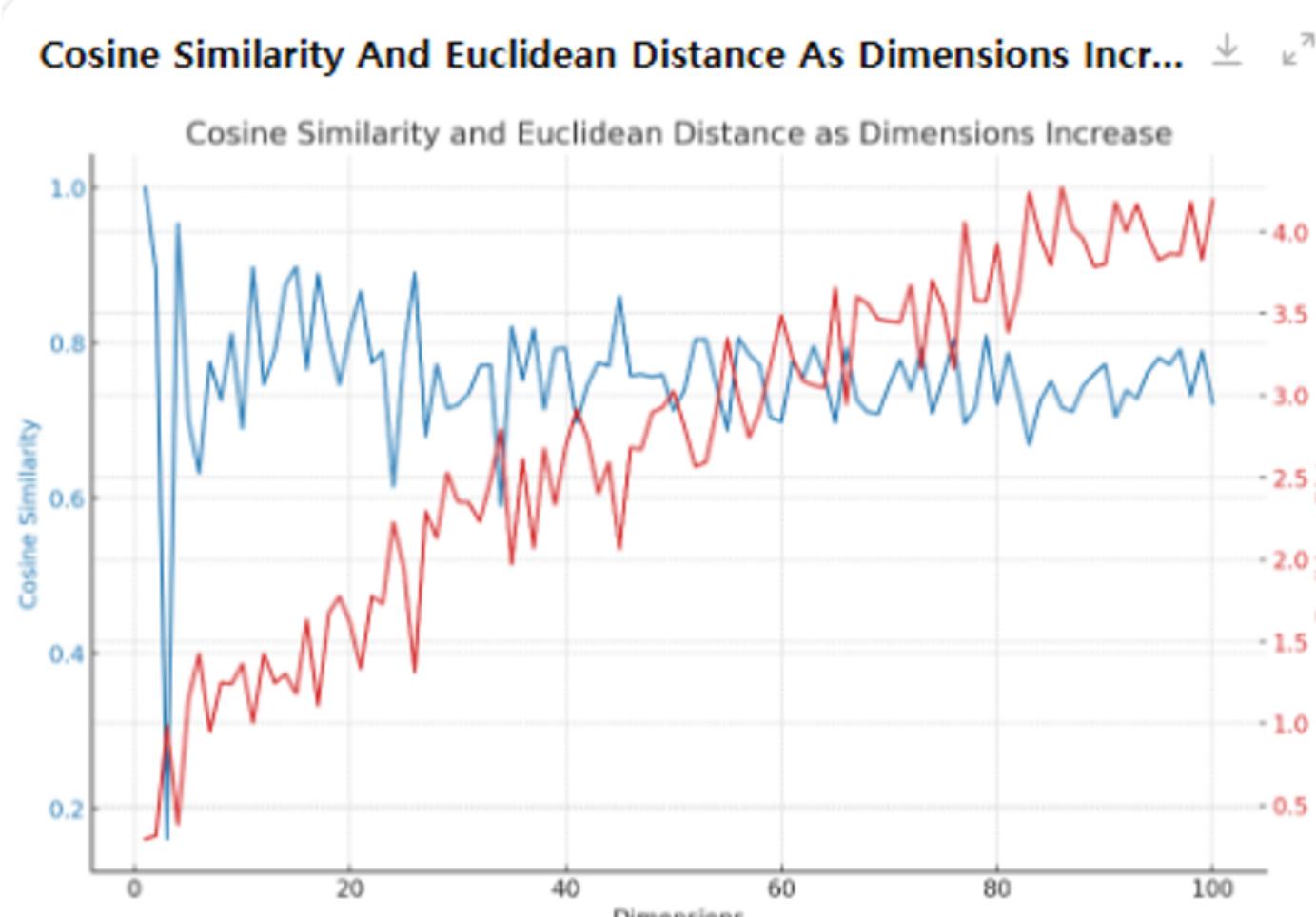
- 방향성을 중시: 코사인 유사도는 두 벡터의 방향성을 기준으로 하여 유사성을 측정한다.

이는 고차원에서도 벡터의 크기나 절대적인 거리와 관계없이 패턴이나 특징의 유사성을 측정하는 데 더 적합하다.



## 5. 고차원의 저주

- 크기 차이 무시: 문서 데이터나 이미지 특징 벡터와 같은 고차원 데이터에서는 벡터 간의 크기보다는 방향성이 더 중요하다. 코사인 유사도는 크기를 무시하고 벡터 간의 각도만을 계산하기 때문에, 고차원 벡터 간 유사성을 비교할 때 효과적이다.
- 유클리드 거리의 왜곡: 고차원에서 유클리드 거리는 벡터 간의 거리 차이를 왜곡하게 된다. 차원이 높아질수록 모든 벡터가 거의 동일한 거리를 가지게 되어, 유사성 측정에서 의미가 없어질 수 있다.



- 코사인 유사도 (파란색 선): 차원이 증가함에 따라, 초기에는 크게 변동하다가 차원이 높아질수록 점점 1에 가까워집니다. 이는 고차원에서는 벡터 간의 각도가 비슷해져 코사인 유사도 값이 거의 일정하게 유지되며, 벡터 간의 차이를 구분하기 어렵다는 것을 보여줍니다. 즉, 고차원에서는 거의 모든 벡터들이 직교에 가까워지는 문제를 나타냅니다.
- 유클리드 거리 (빨간색 선): 차원이 증가할수록 두 벡터 사이의 거리가 기하급수적으로 증가하는 모습을 보여줍니다. 차원이 높아질수록 벡터 간 거리가 점점 더 커지기 때문에, 고차원에서는 거의 모든 벡터가 서로 멀리 떨어지게 됩니다.



## 5. 고차원의 저주

### 결론

고차원의 저주에서 코사인 유사도는 방향성을 기준으로 유사성을 측정하기 때문에,

고차원 벡터에서 유클리드 거리보다 더 유리하다.

유클리드 거리는 차원이 증가할수록 벡터 간 거리가 비슷해져 의미 있는 유사성을 찾기 어려워지지만,

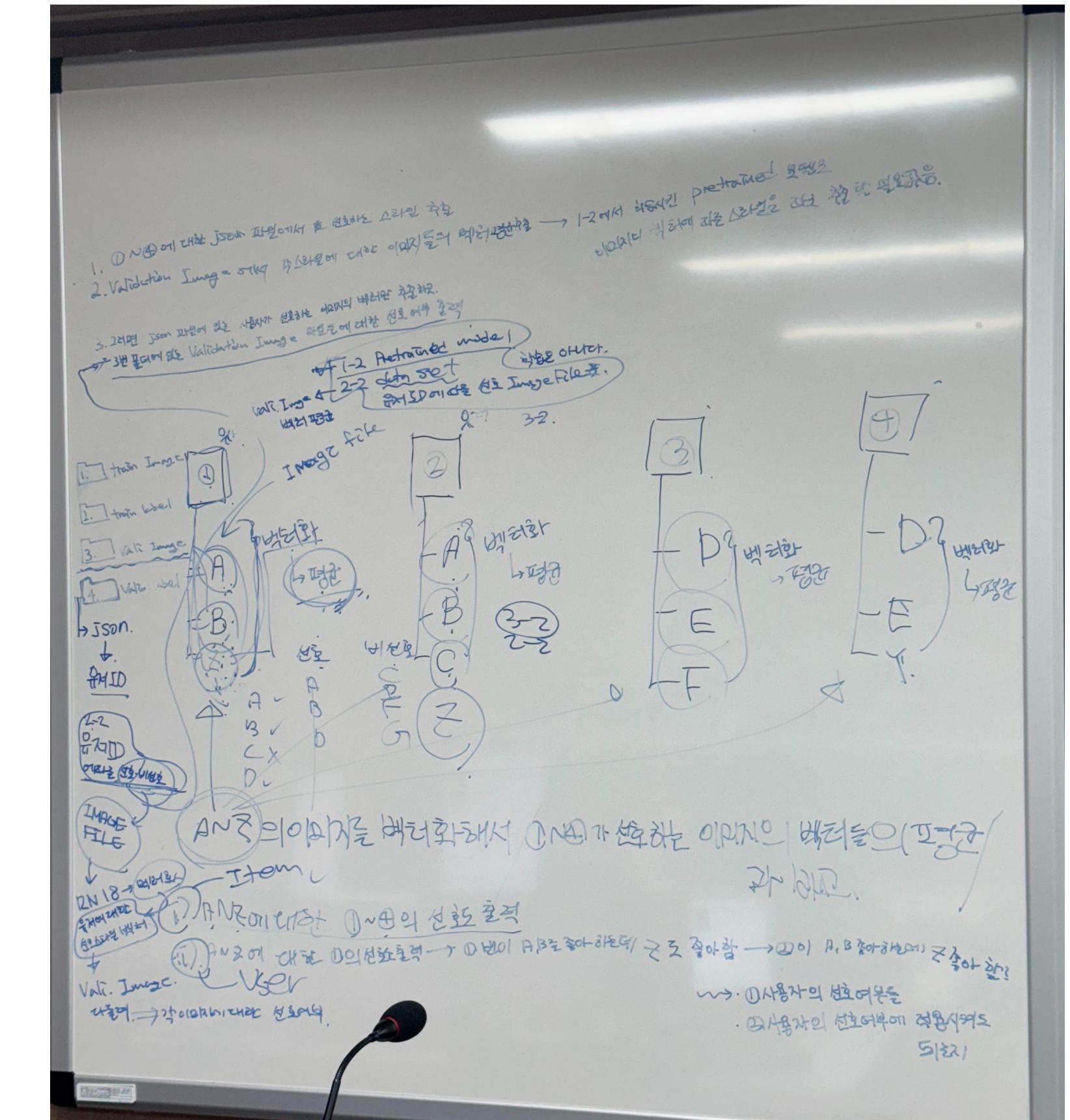
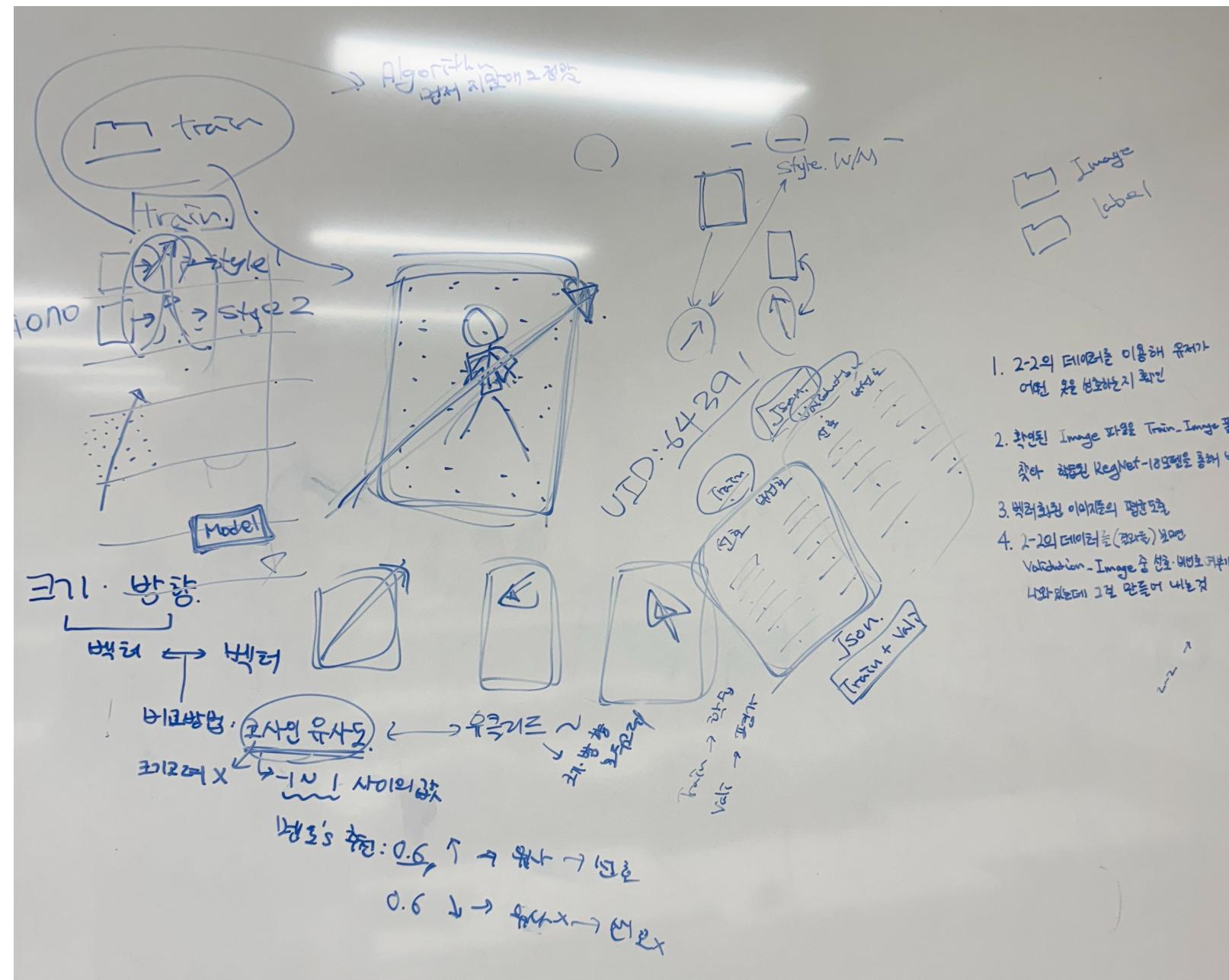
코사인 유사도는 벡터 간의 각도만을 고려하므로 고차원에서도 일정한 성능을 발휘할 수 있다.



# [실제 예측 수행 결과 도출]

Mission 3-2의 전반적인 이해

우리는 문제를 최대한으로 이해하기 위해  
여러번의 회의를 거쳤고, 만족스러운 성과를 낼 수 있었다.

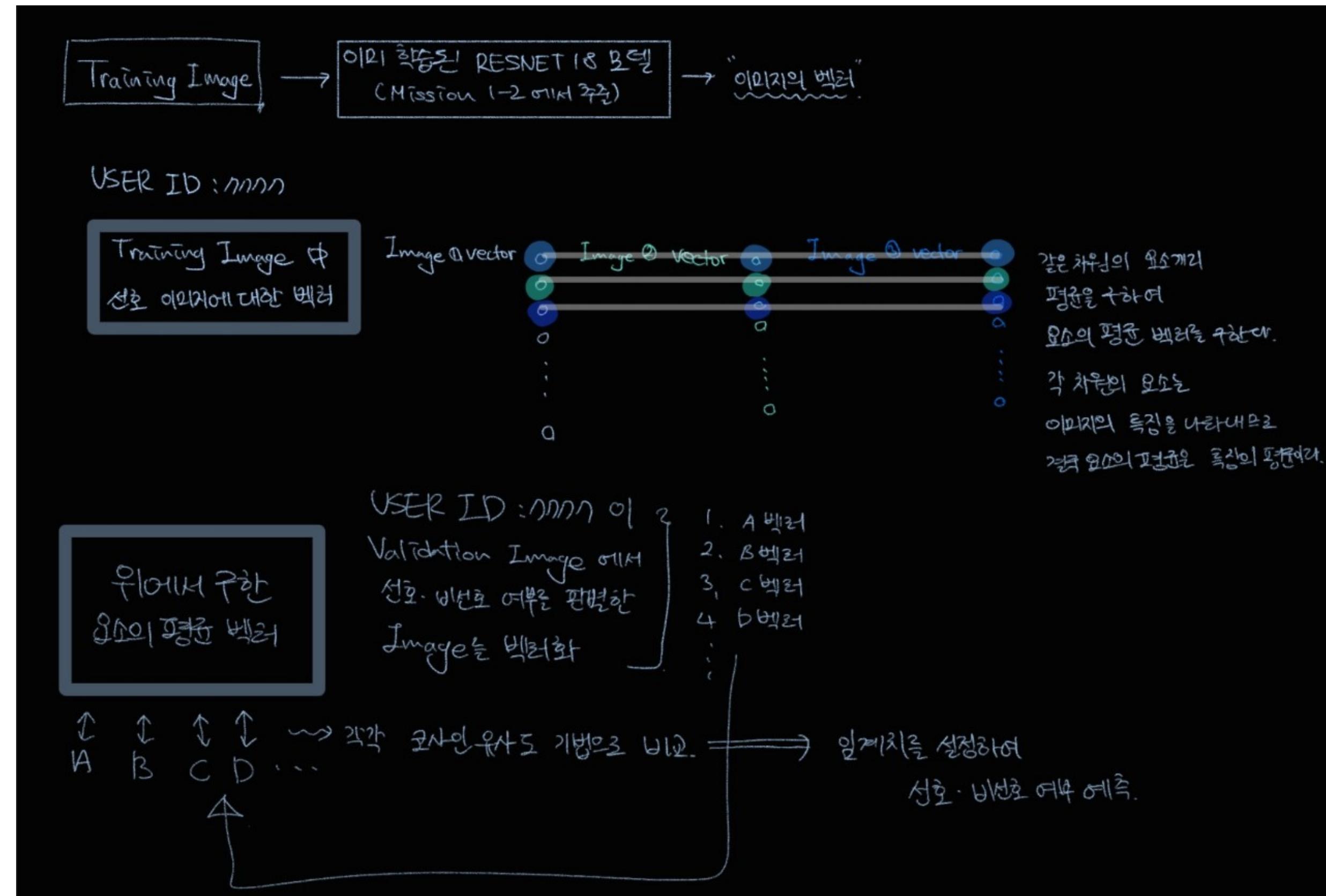




# [실제 예측 수행 결과 도출]

Mission 3-2의 전반적인 이해

우리가 이해한 내용을 최종적으로  
다음과 같이 정리해보았다.





## [실제 예측 수행 결과 도출]

Mission 3-2의 전반적인 이해

이전의 1-2에서 RESNET-18 모델을 학습시켰고, 이번 미션에서는 1-2에서 생성된 Pretrained model을 사용한다.

그리고 2-2에서 도출했던 분류표에 따라 각 User\_id 별 선호 또는 비선호하는 Training 이미지 파일과

Validation 이미지 파일을 이용할 것인데, 먼저 각 유저들이 선호하는 스타일의 Training 이미지를

모두 1-2에서 학습되어 생성된 Pretrained Model에 통과시켜 벡터화 한다.

이미지들의 특징 벡터들은 n차원의 벡터이고, 모든 벡터들은 같은 모델로 벡터화 하였으므로 같은 차원 값을 가진다.

따라서 각 특징 벡터의 동일한 위치에 있는 특징 요소의 평균을 구해 각 요소의 평균 벡터를 사용자마다 생성한다.

이 벡터는 사용자의 선호 벡터라고 볼 수 있다.



## [실제 예측 수행 결과 도출]

Mission 3-2의 전반적인 이해

앞의 과정에서 구한 사용자의 특징 벡터를 가지고 Validation 이미지의 선호, 비선호 여부를 예측하여야 한다.

Validation 이미지 또한 동일 모델로 벡터화 하고, 각 사용자의 특징 벡터를 평균 낸 벡터와 비교한다.

이때 비교 방법은 코사인 유사도 방식을 적용하였다.

코사인 유사도 방식은 두 벡터를 비교 했을때,  $-1 \sim 1$  값이 나오고, 1에 가까울 수록 유사하다고 볼 수 있다.

우리는 최선의 스타일 예측 결과를 내기 위해서, 코사인 유사도의 임계치를  $0.6 \sim 0.95$  사이  $0.05$  단위로

나누어 어떤 임계치가 가장 적합한지 확인하였다.



# [실제 예측 수행 결과 도출]

코사인 유사도 임계치 지정에 따른 결과

우리는 예측 수행 결과를 다음과 같이 알아 볼 수 있었다.

```
Processing user 1/100 (ID: 64747)...
Validation (선후): W_05628_00_cityglam_W.jpg, Similarity: 0.88, Predicted: 선후, Result: 0
Validation (선후): W_20598_70_military_W.jpg, Similarity: 0.97, Predicted: 선후, Result: 0
Validation (선후): W_22510_80_powersuit_W.jpg, Similarity: 0.85, Predicted: 선후, Result: 0
Validation (선후): W_30988_90_kitsch_W.jpg, Similarity: 0.94, Predicted: 선후, Result: 0
Validation (선후): W_37491_70_military_W.jpg, Similarity: 0.97, Predicted: 선후, Result: 0
Validation (선후): W_38588_19_genderless_W.jpg, Similarity: 0.92, Predicted: 선후, Result: 0
Validation (선후): W_39164_00_oriental_W.jpg, Similarity: 0.96, Predicted: 선후, Result: 0
Validation (선후): W_44330_10_sportivecasual_W.jpg, Similarity: 0.94, Predicted: 선후, Result: 0
Validation (선후): W_46907_80_powersuit_W.jpg, Similarity: 0.91, Predicted: 선후, Result: 0
Validation (비선후): W_02498_50_feminine_W.jpg, Similarity: 0.92, Predicted: 선후, Result: X
Validation (비선후): W_11610_90_grunge_W.jpg, Similarity: 0.87, Predicted: 선후, Result: X
Validation (비선후): W_14102_50_feminine_W.jpg, Similarity: 0.93, Predicted: 선후, Result: X
Validation (비선후): W_27828_60_minimal_W.jpg, Similarity: 0.90, Predicted: 선후, Result: X
Validation (비선후): W_34024_10_sportivecasual_W.jpg, Similarity: 0.93, Predicted: 선후, Result: X
Validation (비선후): W_47169_70_hippie_W.jpg, Similarity: 0.89, Predicted: 선후, Result: X
...
Processing user 2/100 (ID: 63405)...
Validation (선후): W_01853_60_mods_M.jpg, Similarity: 0.97, Predicted: 선후, Result: 0
Validation (선후): W_02677_60_mods_M.jpg, Similarity: 0.91, Predicted: 선후, Result: 0
Validation (선후): W_02879_90_hiphop_M.jpg, Similarity: 0.96, Predicted: 선후, Result: 0
Validation (선후): W_04522_50_ivy_M.jpg, Similarity: 0.83, Predicted: 비선후, Result: X
Validation (선후): W_04684_90_hiphop_M.jpg, Similarity: 0.96, Predicted: 선후, Result: 0
Validation (선후): W_06860_19_normcore_M.jpg, Similarity: 0.94, Predicted: 선후, Result: 0
Validation (선후): W_15294_50_ivy_M.jpg, Similarity: 0.91, Predicted: 선후, Result: 0
Validation (비선후): W_07187_70_hippie_M.jpg, Similarity: 0.96, Predicted: 선후, Result: X
Validation (비선후): W_12304_80_bold_M.jpg, Similarity: 0.92, Predicted: 선후, Result: X
Validation (비선후): W_12904_50_ivy_M.jpg, Similarity: 0.94, Predicted: 선후, Result: X
Validation (비선후): W_15140_80_bold_M.jpg, Similarity: 0.79, Predicted: 비선후, Result: 0
Validation (비선후): W_16501_70_hippie_M.jpg, Similarity: 0.88, Predicted: 선후, Result: X
Validation (비선후): W_16755_00.metrosexual_M.jpg, Similarity: 0.84, Predicted: 비선후, Result: 0
Validation (비선후): W_17443_90_hiphop_M.jpg, Similarity: 0.84, Predicted: 비선후, Result: 0
Processing completed for all users.
```

```
Processing user 99/100 (ID: 7658)...
Validation (선후): W_01234_10_sportivecasual_W.jpg, Similarity: 0.94, Predicted: 선후, Result: 0
Validation (선후): W_04927_50_feminine_W.jpg, Similarity: 0.95, Predicted: 선후, Result: 0
Validation (선후): W_09731_19_genderless_W.jpg, Similarity: 0.92, Predicted: 선후, Result: 0
Validation (비선후): W_02345_60_space_W.jpg, Similarity: 0.92, Predicted: 선후, Result: X
Validation (비선후): W_05312_80_bodyconscious_W.jpg, Similarity: 0.93, Predicted: 선후, Result: X
Validation (비선후): W_10510_60_space_W.jpg, Similarity: 0.92, Predicted: 선후, Result: X
Validation (비선후): W_13535_80_powersuit_W.jpg, Similarity: 0.90, Predicted: 선후, Result: X
Validation (비선후): W_13688_90_hiphop_W.jpg, Similarity: 0.96, Predicted: 선후, Result: X
Validation (비선후): W_14923_60_minimal_W.jpg, Similarity: 0.95, Predicted: 선후, Result: X
Validation (비선후): W_19003_50_feminine_W.jpg, Similarity: 0.94, Predicted: 선후, Result: X
Processing user 100/100 (ID: 64295)...
Validation (선후): W_12803_70_hippie_M.jpg, Similarity: 0.96, Predicted: 선후, Result: 0
Validation (선후): W_28563_90_hiphop_M.jpg, Similarity: 0.93, Predicted: 선후, Result: 0
Validation (선후): W_30027_50_ivy_M.jpg, Similarity: 0.96, Predicted: 선후, Result: 0
Validation (선후): W_33329_50_ivy_M.jpg, Similarity: 0.96, Predicted: 선후, Result: 0
Validation (비선후): W_04237_60_mods_M.jpg, Similarity: 0.95, Predicted: 선후, Result: X
Validation (비선후): W_15729_90_hiphop_M.jpg, Similarity: 0.96, Predicted: 선후, Result: X
Validation (비선후): W_16374_10_sportivecasual_M.jpg, Similarity: 0.89, Predicted: 선후, Result: X
Validation (비선후): W_23900_50_ivy_M.jpg, Similarity: 0.94, Predicted: 선후, Result: X
Validation (비선후): W_24598_80_bold_M.jpg, Similarity: 0.88, Predicted: 선후, Result: X
Validation (비선후): W_25761_90_hiphop_M.jpg, Similarity: 0.94, Predicted: 선후, Result: X
Validation (비선후): W_27750_60_mods_M.jpg, Similarity: 0.97, Predicted: 선후, Result: X
Validation (비선후): W_31478_19_normcore_M.jpg, Similarity: 0.92, Predicted: 선후, Result: X
Validation (비선후): W_32314_19_normcore_M.jpg, Similarity: 0.80, Predicted: 비선후, Result: 0
Processing completed for all users.
```



# [실제 예측 수행 결과 도출]

코사인 유사도 임계치 지정에 따른 결과

우리는 예측 수행 결과의 수치를

다음과 같이 정의하였다.

- Accuracy (정확도)
- Precision (정밀도)
- Recall (재현율)
- F1 Score

A: 모델이 "선호"라고 예측하는 사건

B: 이미지가 실제 "선호"인 사건

정밀도: 모델이 "선호"라고 예측했을 때, 이미지가 실제로 "선호"일 확률.

A

B

$$\Rightarrow P(B|A)$$

재현율: 이미지가 실제로 "선호"일 때, 모델이 "선호"라고 예측할 확률

B

A

$$\Rightarrow P(A|B)$$

정확도: 전체 예측 중에서 올바르게 예측된 비율

F1 score: 정밀도와 재현율의 조화평균

예측: 우리가 학습시킨 RESNET18 모델의 정확도가 60% ~ 63% 정도이므로  
3-2에서의 정확도가 60% ~ 63%에 유효한 수로 신뢰성 있는 데이터라고 간주.



# [실제 예측 수행 결과 도출]

코사인 유사도 임계치 지정에 따른 결과

우리는 예측 수행 결과의 수치를

다음과 같이 정의하였다.

- Accuracy (정확도)
- Precision (정밀도)
- Recall (재현율)
- F1 Score

## 2. 정확도 (Accuracy):

- 정확도는 전체 예측 중에서 맞춘 비율을 나타냅니다. 즉, 전체 데이터 중에서 모델이 얼마나 잘 예측했는지를 나타내는 지표입니다.
- 수식:  $\text{Accuracy} = \frac{\text{맞게 예측한 샘플 수}}{\text{전체 샘플 수}}$

## 3. 정밀도 (Precision):

- 정밀도는 모델이 '긍정(positive)'으로 예측한 데이터 중에서 실제로도 긍정인 데이터의 비율입니다. 예를 들어, 모델이 '선호'로 예측한 이미지 중 실제로 '선호'에 해당하는 이미지의 비율을 나타냅니다.
- \*\*오탐(false positive)\*\*을 줄이는 것이 중요한 상황에서 유용한 지표입니다.
- 수식:  $\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$ 
  - TP: True Positives (맞게 긍정으로 예측한 수)
  - FP: False Positives (잘못 긍정으로 예측한 수)

## 4. 재현율 (Recall):

- 재현율은 실제로 긍정인 데이터 중에서 모델이 긍정으로 정확하게 예측한 비율입니다. 모든 긍정을 얼마나 잘 잡아냈는지를 나타내며, 민감도(sensitivity)라고도 합니다.
- \*\*놓친 긍정(false negative)\*\*을 줄이는 것이 중요한 상황에서 유용한 지표입니다.
- 수식:  $\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$ 
  - FN: False Negatives (잘못 부정으로 예측한 수)

## 5. F1 Score:

- F1 점수는 \*\*정밀도(Precision)\*\*와 \*\*재현율(Recall)\*\*의 조화 평균으로, 두 지표 간의 균형을 나타냅니다. 이 지표는 정밀도와 재현율 사이의 균형을 맞추는 데 유용합니다.
- 만약 정밀도와 재현율 간에 큰 차이가 있다면, F1 점수는 이를 조화롭게 평가해줍니다. 즉, 한쪽 지표에만 너무 치우치지 않고 전체적인 성능을 확인할 때 사용합니다.
- 수식:  $\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$



# [실제 예측 수행 결과 도출]

## 코사인 유사도 임계치 지정에 따른 결과

### ----- Model Performance Metrics -----

Accuracy (정확도): 0.58

Precision (정밀도): 0.45

Recall (재현율): 0.20

F1 Score: 0.28

임계치 0.95

### ----- Model Performance Metrics -----

Accuracy (정확도): 0.46

Precision (정밀도): 0.41

Recall (재현율): 0.74

F1 Score: 0.52

임계치 0.9

### ----- Model Performance Metrics -----

Accuracy (정확도): 0.42

Precision (정밀도): 0.40

Recall (재현율): 0.92

F1 Score: 0.56

임계치 0.85

### ----- Model Performance Metrics -----

Accuracy (정확도): 0.42

Precision (정밀도): 0.40

Recall (재현율): 0.97

F1 Score: 0.57

임계치 0.8

### ----- Model Performance Metrics -----

Accuracy (정확도): 0.40

Precision (정밀도): 0.40

Recall (재현율): 0.99

F1 Score: 0.57

임계치 0.75

### ----- Model Performance Metrics -----

Accuracy (정확도): 0.40

Precision (정밀도): 0.40

Recall (재현율): 1.00

F1 Score: 0.57

임계치 0.7



# [실제 예측 수행 결과 도출]

코사인 유사도 임계치 지정에 따른 결과

----- Model Performance Metrics -----

Accuracy (정확도): 0.40

Precision (정밀도): 0.40

Recall (재현율): 1.00

F1 Score: 0.57

임계치 0.65

----- Model Performance Metrics -----

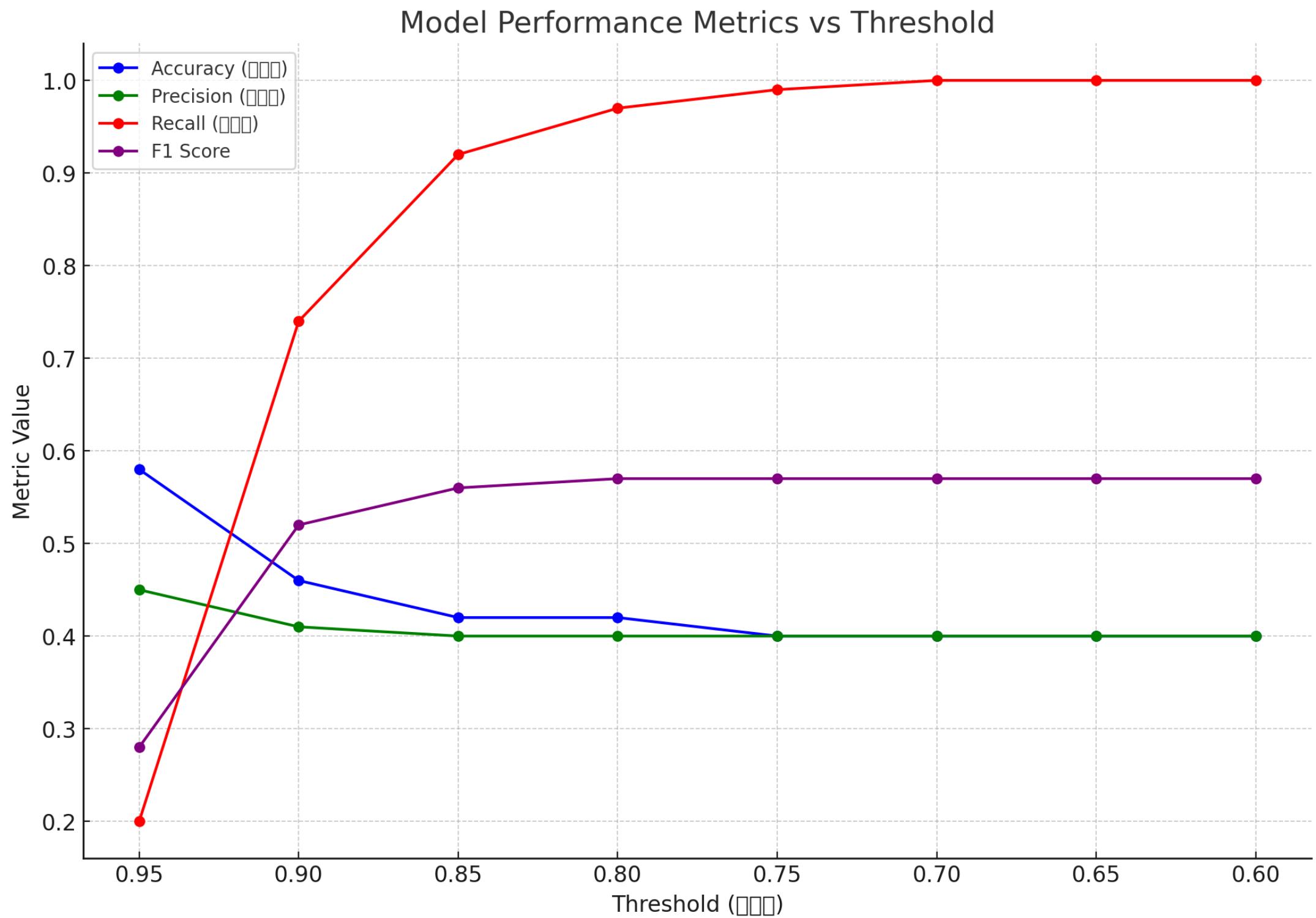
Accuracy (정확도): 0.40

Precision (정밀도): 0.40

Recall (재현율): 1.00

F1 Score: 0.57

임계치 0.6





## [실제 예측 수행 결과 도출]

문제 풀이 과정 속에서 알게 된 내용들

이번 문제를 풀면서 Item-based Filtering을 직접 구현해보고 예측 시스템을 만들어 볼 수 있었다.  
예측의 결과들을 코사인 유사도 값의 임계치를 조절해보면서 예측 성능 수치에 대해 자세히 알 수 있었다.

먼저 정확도를 보면, 임계치가 낮아질 수록 정확도 또한 떨어졌다.

정확도의 의미가 전체 샘플 중 맞게 예측한 샘플의 수인 것을 생각할때,  
선호 데이터이든, 비선호 데이터이든, 모두 꽤 높은 유사도를 가지고 있다는 것을 의미한다고 생각한다.

두번째로 정밀도에서는, 정확도와 비슷하게 임계치가 감소함에 따라 같이 감소하였다.

이는 임계치가 낮아질 수록 선호 이미지의 기준이 낮아져 실제로는 비선호 이미지 이지만,  
60%대 밖에 안되는 꽤 낮은 모델의 기준에 의해 선호로 잘못 분류된 데이터가 많았다고 생각한다.

이와 반대로 재현율은, 임계치가 감소할때, 재현율은 증가하는 양상을 보였다.

이것은 정밀도와는 반대로 임계치가 낮아질 수록 선호 이미지의 기준이 낮아져  
조금만 선호 벡터와 유사해도 선호로 분류하여, 어지간한 이미지는 모두 선호로 예측하게 되었다.



충북대학교 정보통신공학부  
**TEAM SIGCOM**



**이규하**

팀장, 21학번

데이터 크리에이터 캠프에는 내가 인공지능 분야로의 더 큰 관심을 가지고 나아갈 수 있게 해준 '시발점'이라 할 수 있다. 단순히 이론만으로 배우는 인공지능이 아니라 실습을 통해 실질적으로 체험하고 스스로 배워가고 다질 수 있는 기회라 시작 전부터 데이터 크리에이터 캠프의 활동이 많이 기다려졌다.

맡은 포지션이 팀장인지라 팀원들의 의견을 팀 프로젝트에 잘 녹일 수 있게 도왔다. 팀 내 서로 아는 부분과 모르는 부분이 제각각이고, 할 수 있는 역량의 영역도 그러하다. 그래도 팀원들이 아이디어를 유연하고 많이 낼 수 있도록 촉매 역할을 계속 해왔다. 이런 과정을 통해서 서로 놓치고 있었던 부분을 자연스럽게 채워나갈 수 있었다. 또한, 팀원들과 함께 부딪혀보면서 인공지능 분야에 대해 더 재밌고 쉽게 배울 수 있었던 것 같다. 일상의 주입식 강의보다는 실습을 통해 생각해보고 시행착오를 겪어보며 좀 더 나은 결과를 도출해내는 법을 깨닫게 되었다.



충북대학교 정보통신공학부  
**TEAM SIGCOM**



이번 활동을 통해서 앞으로 내 미래에 있을 '개발자'라는 목표에 다가갈때  
큰 힘이 될 인공지능 분야의 역량을 기를수 있었다. 평소에 듣기만 했던 분야인지라  
이번 미션들에 쉽게 접근하기가 어려웠는데, 멘토링을 통해 공부에 대한 방향성과  
자신감을 기를수 있었고 다양한 머신러닝의 기법들과 이론에 걸쳐서 실습까지를  
한번에 경험 할 수 있었다. 이 경험을 토대로 사회에 선한 영향을 미치게 하는  
인공지능 개발자로 성장하고자 하는 목표가 생겼다.

## 곽도현

팀원, 21학번



충북대학교 정보통신공학부  
**TEAM SIGCOM**



이번 데이터 크리에이터 캠프를 통해 머신러닝과 딥러닝 영역에 대해 많이 배울 수 있었다. 이런 활동이 거의 처음이라 많이 햇갈리고 뭘 어디서부터 시작해야할지 잘 몰랐었지만, 같이 생각을 나누는 팀원들이 있었고 매주 멘토링 활동 덕분에 방향과 목표를 확립해 나갈 수 있었다. 인공지능 분야로의 시작은 어려웠지만 이번 활동을 통해 첫 단추를 정확히 잠구었다고 생각하며, 앞으로도 또 다른 인공지능 분야 활동을 해낼 수 있는 자신감이 생겼다.

## 강찬솔

팀원, 23학번



충북대학교 정보통신공학부  
**TEAM SIGCOM**



**허강민**

팀원, 21학번

이번 데이터 크리에이터 캠프를 통해 기초적인 머신러닝의 이해와 더불어, 직접 학습을 시켜보며 조금 더 심화된 내용까지, 전반적인 인공지능의 내용에 대하여 몸소 체험할 수 있어 좋았다.

이론적인 부분과 학습 코드의 전체적인 흐름에 관련된 내용에서 내가 먼저 이해한 부분은 팀원에게 설명하며 중간중간 비어있던 개념을 채울 수 있었고, 내가 조금 늦게 이해한 내용은 팀원이 나를 도와주며 서로의 부족한 부분을 발전시켰다.

내가 주로 맡았던 부분은 전반적인 코드와 학습 과정 중 필요한 수치들을 변화시키며 결과에 어떤 영향을 미치는지, 학습 시간에 어떤 영향을 미치는지 확인하는 부분이었다.

이를 통해 학습에 관련된 여러 수치들의 역할을 알 수 있었고, 적절한 수치를 찾아나가는 과정 속에서 학습 자체의 흐름 또한 이해하는데 도움이 되었다.

나는 이번 데이터 크리에이터 캠프에서 인공지능의 전반적인 내용을 몸소 체험해보며 나를 한층 더 발전시킬 수 있어 좋았다.



본콘텐츠는 한국지능정보사회진흥원(NIA)의 동의없이 무단 사용할 수 없으며  
상업적 목적으로 이용을 금합니다.

# DATA CREATOR CAMP

2024 데이터 크리에이터 캠프

감사합니다

