

Wrap-up Report

Sketch Image Classification

(CV-20) 김경수 김성규 김유경 김재진 박수영 이단유

1. 프로젝트 개요

컴퓨터 비전 분야에서는 다양한 형태의 이미지 데이터에 대해 인식과 분류와 같은 문제를 해결하고 있습니다. 여러 형태의 이미지 중 스케치 데이터는 사진과 다르게 사물에 대한 개념적 이해를 바탕으로 추상적이고 단순화된 형태의 데이터입니다. 스케치 데이터의 특성상 색상, 질감 등 다른 이미지 데이터에는 세부적인 특성들이 결여되어 있고, 대상의 형태와 구조에 초점을 맞춘 경우가 많습니다.

본 프로젝트에서는 이러한 스케치 데이터의 특성을 이해하고, 다양한 모델이 객체의 특징을 학습하여 적합한 클래스로 분류하는 것을 목적으로 하고 있습니다. 스케치 데이터를 활용하는 인공지능 모델은 디지털 예술, 게임 개발, 교육 콘텐츠 생성 등 다양한 분야에 응용될 수 있습니다. 다만, 보통의 이미지 데이터와 달리 세부적인 특성이 결여되어 있는 스케치 데이터의 특성을 반영하여 데이터 증강 등을 고려할 필요가 있습니다.

이번 프로젝트에서는 스케치 데이터셋을 살펴보고, 그 특성에 맞는 모델을 실험하였습니다. 또한, 실험 결과의 해석 및 환류를 통해 보다 나은 분류 모델을 구축하였습니다.

2. 프로젝트 팀 구성 및 역할

프로젝트 수행을 위한 팀 구성과 구성원별 역할은 다음 표와 같이 진행했습니다.

이름	역할
김경수	데이터 전처리, 협업툴 관리
김성규	데이터 전처리, 모델 적용
김유경	데이터 증강, 모델 적용
김재진	개발 환경 세팅, 모델 적용
박수영	데이터 증강, 모델 적용
이단유	데이터셋 분석, 모델 적용

3. 프로젝트 수행 절차 및 방법

프로젝트의 목적 달성을 위해 다음과 같은 절차와 방법으로 진행하였습니다.

- 관련 연구 조사: 이미지 분류에 적합한 다양한 모델을 조사하고, 데이터 증강을 위한 기법 조사
- 데이터셋 분석: 프로젝트에 앞서 적합한 모델 및 데이터 증강 방법을 고민하기 위한 탐색적 데이터 분석을 수행
- 데이터 전처리: 데이터셋 분석 과정에서 발견한 사실에 기반하여 데이터 전처리
- 모델 실험: 조사된 다양한 모델을 적용하여 현재 데이터셋에 적합한 모델 선정
- 데이터 증강 실험: 최신 기법 및 현재 데이터셋 특성에 맞게끔 데이터 증강 수행
- 모델 튜닝: 모델의 하이퍼 파라미터를 수정하면서 실험 진행

4. 관련 연구

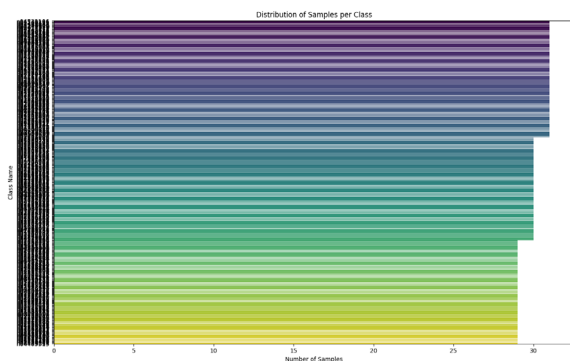
이미지 분류 문제에서 컴퓨터 비전 분야에서 오랜 연구 역사를 가지고 있으며, 다양한 방법론이 제시되어 왔습니다. 그 중에서도 합성곱 신경망(Convolutional Neural Network, CNN) 모델은 이미지 분류 문제를 해결하는데 있어 뛰어난 성능을 보여주었습니다. 보다 나은 성능 달성을 위해 구조가 깊어짐에 따라 발생하는 문제를 해결하기 위해 잔차 연결을 포함한 ResNet, 이미지 패치를 입력으로 받아 트랜스포머를 통한 분류를 수행하는 비전 트랜스포머(Vision Transformer, ViT), ViT에서 효과적이었던 부분을 차용해 CNN을 재설계한 ConvNeXt까지 다양한 모델이 연구되어 문제 해결에 적용되고 있습니다.[1]

5. 프로젝트 수행 결과

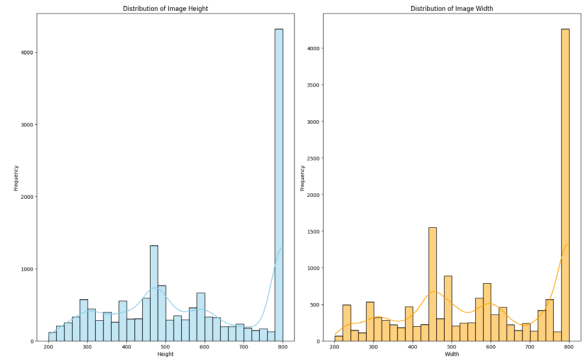
5.1. 데이터셋 분석

본 프로젝트에서 사용된 데이터셋은 ImageNet Sketch 데이터셋 중 500개 클래스에 해당하는 총 25,035개의 데이터로 구성되어 있습니다. 이 중 15,021개는 학습 데이터로 10,014개의 데이터는 평가 데이터로 나뉘어 있습니다.

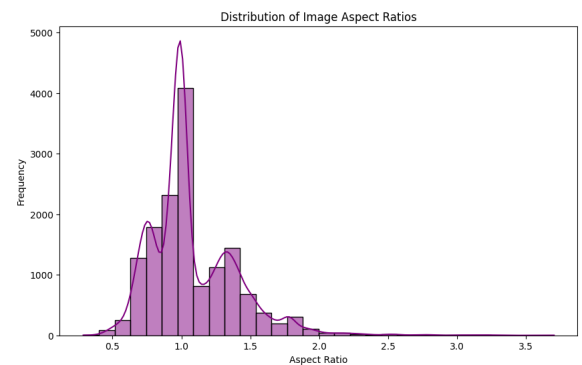
훈련 데이터셋은 클래스별로 29~31개로 균등하게 클래스가 분포하고 있습니다.



이미지 데이터들의 크기는 400~800 픽셀 사이에 대부분 분포하고 있으며, 특히 800 픽셀에 가까운 이미지가 많은 것을 확인할 수 있습니다.



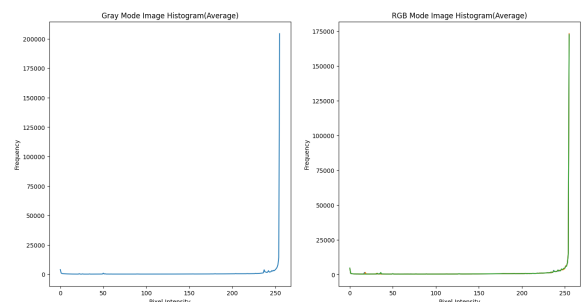
또한, 이미지 데이터의 절대적인 크기에 더해, 가로-세로 비율 역시 0.8 ~1.2 사이의 정방형에 가까운 이미지가 많음을 확인하였습니다.



한편, 훈련 데이터셋의 이미지 모드는 RGB는 12,443개, Gray 모드는 2,578개로 분포됨을 확인 하였습니다.

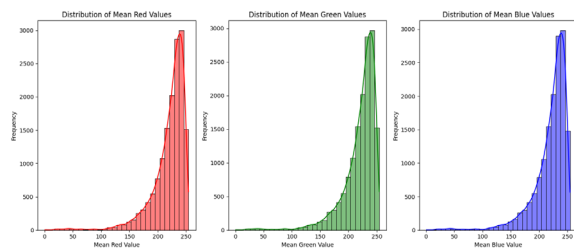
Mode	RGB	Gray
이미지(개)	12,443	2,578

각 이미지 모드(Gray-좌, RGB-우)의 이미지 히스토그램은 다음 차트와 같이 한쪽으로 치우쳐 있음을 확인할 수 있었습니다.



훈련 데이터셋의 RGB 평균값은 200~250에 분포

하고 있으며, 이미지 데이터들의 밝기가 전반적으로 높습니다.



데이터셋 전체의 관점 이외에 개별 훈련 데이터를 살펴보았을 때, 다음과 같은 사항을 발견하였습니다.

- 동일 클래스에 동일한 이미지가 flip, resize된 사례 관찰
- 하나 이미지에 여러 객체가 포함 이미지



동일 이미지(좌) 여러 객체 포함(우)

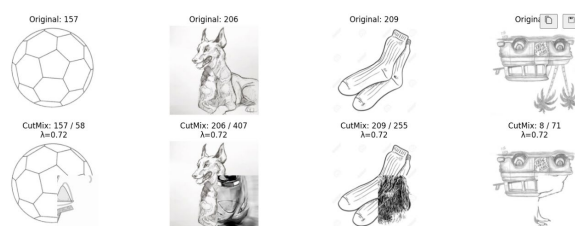
5.2. 데이터 전처리

데이터셋 분석 과정에서 관찰한 발견사항들은 이미지 분류 모델에서 해당 클래스의 다양한 형태를 학습하지 못하거나, 잘못된 특징을 학습할 우려가 있다고 판단하여 전처리를 진행하였습니다. 우선, 동일한 이미지를 제거한 데이터셋(del flip)과 여러 객체가 포함된 이미지를 분리한 데이터셋(object split)을 생성합니다. 그리고 제외된 이미지로 인해 부족해진 클래스별 분포 숫자를 맞춰 주기 위해 증강을 수행한 데이터셋(del flip + transform)과 여러 객체를 분리한 것을 추가한 데

이터셋(del flip + object split)을 준비하였습니다.

5.3. 데이터 증강

(CutMix) Yun et al.에 따르면, 서로 다른 두 이미지를 섞어서 모델의 성능 향상되어 효과적인 증강 기법임을 보였습니다.[2] 본 프로젝트에서는 클래스의 종류 개수 대비 적은 데이터를 증강이 필요하기 때문에 여러 이미지의 다양한 특징을 학습하고, 모델의 강건성을 향상시키기 위한 데이터 증강 방법으로 CutMix를 활용하였습니다.



CutMix 적용한 스케치 이미지 예시

(Edge Detection) Gerihos et al.에 따르면, CNN 모델은 형태만 유지한 경우 낮은 정확도를 보이는 문제를 해결하기 위해 객체의 형태에만 집중할 수 있도록 데이터셋을 구성하여 모델의 정확도와 강건성을 향상 시켰습니다.[3] 본 프로젝트에서는 사용하고 있는 스케치 데이터셋의 형태에 집중하기 위해 Canny 엣지 검출 알고리즘을 활용하였습니다.

(기하학적 변환) 스케치 데이터는 모양과 구조, 스트로크 순서가 중요한 이미지 특성입니다. 이러한 특성을 고려했을 때 스케치 방향과 크기를 학습할 수 있도록 Affine 변환이나 스트로크의 위치와 두께 변동을 시뮬레이션할 수 있도록 Elastic deformation을 증강 기법으로 고려할 수 있습니다.

(Noise & Blur) Gaussian noise를 더해 이미지에 존재하는 노이즈나 불완전한 스케치를 처리하고, Motion Blur를 통해 스케치에서 흔히 볼 수 있는 부정확한 스트로크 효과를 시뮬레이션 합니다.

기하학적 변환 및 Noise & Blur를 추가한 데이터 증강을 Sketch Alumentations (Sketch ab) 이라 정의하고 사용하였습니다.

5.4. 모델 적용

스케치 데이터셋 분류를 위해 torchvision과 timm에서 제공하고 있는 다양한 모델들을 실험하였고, 또한 제공된 훈련 데이터셋이 클래스명과 이미지가 함께 제공된다는 점에서 착안하여 CLIP 모델을 분류 문제 해결을 시도하였습니다. 실험했던 모델 중 주요 모델은 다음과 같습니다.

모델	설명
ResNet	잔차 연결을 통해 더욱 깊은 신경망 모델 구현
EfficientNet	컴파운드 스케일링을 통해 네트워크 효율성을 높인 모델
ConvNeXt	CNN을 ViT의 교훈에 기반해 재설계한 모델
CLIP	텍스트-이미지를 함께 대조하여 학습하는 멀티모달 모델

5.5. 결과

우선 이미지 분류에 사용할 모델을 선정하기 위해 모델별 정확도 비교했으며, 학습률 3.00E-04를 기준으로 적합한 학습률을 찾아가며 훈련 데이터셋에 모델을 학습시킨 결과입니다.

ConvNeXt는 3x3 사이즈의 커널을 사용하는 기존 CNN 모델 대비 7x7 사이즈의 더 커진 커널을 사용해 더 많은 정보를 수용하고, 더 많은 합성곱층을 쌓아 더 복잡한 표현을 학습할 수 있도록 설계되어 높은 베이스라인 성능을 보임을 확인할 수 있었습니다.

모델	정확도
ResNet 18	0.4329
ResNet 50	0.6279

ResNet 101	0.5956
DenseNet 121	0.4552
Mobile v2	0.5760
ViT b 16	0.6575
Efficient b1	0.7823
ConvNeXt	0.8224

앞서 지도학습 기반으로 수행한 모델 외에도, 대규모 데이터로 사전 학습된 CLIP에 대해 text freezing(True, False), multi-prompt(True, False) 조건으로 실험을 수행했습니다.

Text Freezing	Multi-prompt	Learning rate	정확도
O	X	5.00E-06	0.7973
O	X	3.00E-06	0.8003
X	X	1.00E-05	0.8190
X	X	5.00E-06	0.8170
X	O	3.00E-05	0.7604
X	O	3.00E-06	0.7717
X	O	7.00E-06	0.7810

사전 학습이 진행된, 자가 지도 학습(Self-supervised Learning) 모델인 DINO에 대한 다양한 classifier를 대입하여 실험을 수행했습니다.

Classifier	Aug	Learning rate	정확도
Linear	X	3.00E-04	0.7477
	X	8.00E-04	0.7462
	X	3.00E-05	0.5557
Mlp	X	3.00E-05	0.6013
Attention	X	3.00E-05	0.7225
	O	1.00E-03	0.6805

베이스라인 기준으로 가장 좋은 성능을 보였던 ConvNeXt를 활용해 다양한 전처리를 수행하여

모델이 객체의 다양한 특징을 더 학습할 수 있도록 실험을 수행했고, 그 결과는 아래 표와 같습니다. 다만, 프로젝트 수행 기간의 제한으로 인해 모든 전처리 방법의 조합에 대한 실험을 진행하지는 못하였습니다.

동일한 조건 하에 전처리에 따라 정확도가 개선됨을 확인할 수 있었습니다. 특히, Sketch ab의 전처리가 가장 효율적으로 성능 향상에 기여함을 알 수 있었습니다. 이는 현재 주어진 데이터셋에서 스케치 이미지의 특성을 가장 효과적으로 반영한 전처리 방법이 Sketch ab 임을 추론할 수 있습니다.

전처리 방법				정확도
Sketch ab	Edge	CutMix	Del flip	
X	X	X	X	0.8224
O	X	X	X	0.8840
X	O	X	X	0.8690
O	O	X	X	0.8770
O	X	O	X	0.8880
O	X	O	O	0.8850

마지막으로 여러 학습된 모델들의 앙상블을 통해 최적의 추론 조합을 찾는 실험을 수행하였습니다. 여러 모델들을 결합한 경우 앞서 단일 모델의 경우보다는 성능이 일부 향상된 것을 확인할 수 있습니다. 프로젝트 수행 기간의 제한으로 인해 모든 경우의 수에 대해 실험을 진행하지는 못하였습니다.

Method	Model	전처리	정확도
Soft voting-1	ConvNeXt	Sketch ab	0.8940
		Edge	
	ConvNeXt	Sketch ab	
		Edge	
Soft voting-2	ConvNeXt	Sketch ab	0.8920
	ConvNeXt	Del flip	
		Edge	
		CutMix	
		Sketch ab	
Genetic	ConvNeXt	Sketch ab	0.8420
	CLIP	Basic	
	DINO	Basic	
	Efficient	Basic	
Bayesian	ConvNeXt	Sketch ab	0.8820
	CLIP	Basic	
	DINO	Basic	
	Efficient	Basic	
Random Forest	ConvNeXt	Sketch ab	0.8440
	CLIP	Basic	
	DINO	Basic	
	Efficient	Basic	

6. 평가 및 의견

(잘했던 점)

- 개별적으로 시도해보고 싶은 모델, 증강 기법 등을 다양하게 실험해보았습니다.
- 피어세션이나 데일리 스크럼 시간을 활용하여 진행 상황을 공유하였고, 질문이 있을 때마다 곧바로 질문하여 팀원들과 의문을 해결하였습니다.

(아쉬운 점)

- 개별적으로 실험을 수행했지만, 협업 툴을 적극적으로 활용하지 않아 결과 공유가 부족했습니다.

- 리더보드 점수에 지나치게 집중하다 보니 모델의 세부 구조를 깊이 있게 탐구하지 못했고, 이로 인해 논리적인 흐름으로 진행하지 못했던 점이 아쉬웠습니다.
- Git을 이용한 협업에서는 다양한 기능을 충분히 활용하지 못한 점에 개선이 필요하다고 느꼈습니다.

(배운 점)

- 원격 서버 환경에서 VS Code 작업하는 방법에 대해서 배웠습니다. 특히, 같은 서버에서 여러 팀원이 사용하게 될 경우 브랜치 생성이나 커밋과 같은 git 작업에 더 신경써야 한다는 점을 배웠습니다. 이를 통해 각 팀원의 작업을 명확히 분리하고, 충돌을 방지하는 협업 방식의 중요성을 실감하게 되었습니다.
- 데이터 전처리, 모델 학습, 평가, 결과 분석까지 프로젝트의 모든 과정을 직접 경험하며, 전체적으로 조망하는 능력을 키울 수 있었습니다.
- 다양한 모델을 앙상블하여 성능을 향상시키는 방법과 전이 학습을 통해 기존에 학습된 모델을 새로운 데이터셋에 활용하는 방법을 익혔습니다. 이러한 기법들이 모델 성능을 극대화할 수 있다는 점을 체감하여, 유용성을 확인할 수 있었습니

다.

(도전할 점)

- main 으로 사용할 베이스라인 코드를 미리 정리하여 초기 작업을 간소화하고, 모든 팀원이 이를 활용하여 작업할 수 있게끔 구조화하겠습니다.
- 사용 모델들의 논문을 통해 목적과 구조를 학습하여 해당 아키텍처에 이해를 바탕으로 진행할 것입니다.

7. 참고 문헌

- [1] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie. (2022). A ConvNet for the 2020s.
- [2] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, Youngjoon Yoo. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features.
- [3] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, Wieland Brendel. (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.

개인 회고

박수영

1. 나는 내 학습목표 달성을 위해 무엇을 했는가?

- 학습목표 1) 서버 환경에서 다양한 모델을 적용 → VGG, ResNet, Inception, ConvNeXt 등 제공되는 모델들을 실험해봤습니다.
- 학습목표 2) 이미지 데이터의 특성을 살펴보는 방법 공부 → PIL, OpenCV 등 이미지 처리 라이브러리를 학습하였고, 이미지 히스토그램과 같이 이미지 데이터의 특성을 살펴보는 방법을 공부했습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

- 객체를 인식하고 분류하는데 형태와 윤곽선이 더 중요할 수 있다고 하여 edge만 따로 추출하는 방식으로 훈련 데이터를 증강하는 실험을 진행했습니다. 학습률에 대한 모델의 성능 개선을 이끌기 위해 초기 학습률을 조정해보면 실험을 수행했습니다.

3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- 단순히 모델의 변경만으로도 비약적인 성능 향상을 경험할 수 있기 때문에 가장 적합한 모델을 찾는 노력이 우선임을 확인했습니다. 데이터 전처리만으로도 정확도를 향상시킬 수 있음을 확인(0.8224 → 0.86~0.88)하였고, 특히 훈련하려는 데이터의 특성에 맞는 전처리를 적용하는 것이 효과적임을 알 수 있었습니다. 초기 하이퍼파라미터의 수정만으로도 약간의 성능 향상(+0.01)이 되는 것을 확인하였고, 대회 마지막 시점에서는 하이퍼 파라미터 튜닝을 통해 성능 향상을 이끌 수 있음을 확인했습니다.

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 서버에서 백그라운드로 실행하는 것을 배워서 적용했는데, 여러 프로세스를 실험하거나, 메모리를 확인하면서 여러 프로세스를 시도하는 것이 효율적이었던 것 같습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 머신러닝 프로젝트 강의로 제공된 내용(웹 스크립트, Streamlit 등)들을 다양하게 활용하지는 못했습니다. 또한, GAN이 훈련을 위한 새로운 이미지를 생성하는데 도움이 될 수 있다고 읽은 것을 참고해 시도했으나, 생각보다 잘 생성되지 않았습니다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 다음 프로젝트는 유사한 학습 사례들이 있을 것으로 추측되기 때문에 관련된 프로젝트를 찾아보고, 전이학습을 다양하게 시도해볼 계획입니다. 그리고 shell script로 조금 더 자동화해서 주어진 메모리 환경에서 최대한 효율적으로 프로젝트를 진행해볼 계획입니다.

개인 회고

김경수

나는 내 학습목표 달성을 위해 무엇을 했는가?

- 목표: 분류 모델을 이해하고, 학습하여 리더보드 평가 과정까지 흐름을 체득한다.
 1. 모듈 별로 구분되어 있는 베이스라인을 한 줄 한 줄 검색해서 어떤 역할을 하는 것인지 학습했습니다.
 2. 기능 개선을 위해 증강과 모델 선택의 중요성을 깨닫고, 팀원의 도움을 받아서 증강의 종류를 찾는 방법을 학습하고, 라이브러리 문서에서 코드를 찾고, 인자를 이해하는 방법을 학습했습니다.

나는 어떤 방식으로 모델을 개선했는가?

- 1. 기초적인 증강 방법들에 대해서 검색하고, 코드와 인자에 대해서 정리했습니다.
 2. 데이터 증강을 위해 수작업 진행을 서포트하는 업무를 수행했습니다.

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- 1. 수작업을 통해 증강을 한 데이터와 기존 데이터로 나누어 모델을 학습하면서, "동시에 여러 이미지가 섞인 이미지를 Crop하면 정확성이 올라갈 것이다"에 대한 가설 검증을 할 수 있었습니다. flip, edge 등의 기존 라이브러리의 증강과 비교해 큰 효율이 있지는 않았지만, 이 행동을 통해 그런 학습을 얻을 수 있었습니다.

전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 데이터로 모델을 학습하고 리더보드로 결과를 확인한다는 과정 자체를 이해하지 못하고 있었는데, 팀원들이 회의하고 직접 수정하는 것을 옆에서 지켜보며 그 과정을 이해할 수 있어서 성장하는 기회가 될 수 있었습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 함께 서버를 쓰면서, 코드, git, 데이터에 대한 이해가 부족해서, 어떤 부분을 만질 때 문제가 생기는지 직접 만져보며 경험했어야 하는데, 너무 조심스럽게 접근했던 것 같습니다.
- 과거 학습 내용을 잘 이해하고, 금번 학습 내용도 잘 이해하고 참여했다면 좋았을 텐데 이해도가 낮았던 것이 아쉬웠습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 교육 내용을 자세히 학습하고, 좀 더 도전적으로 코드를 수정하고 서버를 돌려보면서 실험적으로 학습을 하고 싶습니다.
- 가설을 세우고, 그 가설을 충족시키기 위해서는 어떤 증강 or 모델이 필요할 지 설계하고, A/B Test를 진행하고 인사이트를 도출해 보고 싶습니다.

개인 회고

이단유

- 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

내 목표는 Sketch image classification 대회에서 높은 정확도를 달성하는 것이었고 이를 위해 다양한 CNN과 트랜스포머 기반 모델을 실험했다. 특히 데이터를 잘 이해하기 위해 데이터 전처리와 증강에 신경을 썼다. 데이터가 적고 클래스가 많아서 Albumentations 라이브러리를 이용해 다양한 증강 기법을 적용했다.

- 나는 어떤 방식으로 모델을 개선했는가?

baseline에 ResNet18 모델을 사용한 후 ConvNeXt, ViT, CoAtNet, EfficientNet 등 다양한 모델을 도입하고 Dropout을 추가했다. 또한 AdamW 옵티마이저와 ReduceLROnPlateau 스케줄러를 적용하여 학습 안정성을 높였다. 이러한 변경을 통해 정확도가 향상되었다. 또한 데이터 전처리 부분에서 Object Detection을 사용하여 이미지 안의 객체를 추출하여 학습에 넣고자 하였다.

- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

이 과정에서 85%의 정확도를 기록할 수 있었다. 특히 증강 기법을 적절히 활용한 것과 데이터 특성에 맞는 모델 선택이 중요하다는 것을 깨달았다. 또한 스케줄러의 세심한 조정이 학습에 중요한 역할을 한다는 점도 배웠다.

- 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

ReduceLROnPlateau 스케줄러를 도입한 것이다. 이 스케줄러는 모델이 학습 과정에서 성능 향상이 없을 때 학습률을 자동으로 감소시켜 학습이 정체되는 지점을 해결할 수 있었다. 데이터 전처리 부분에서 Object Detection을 사용하여 이미지 안의 객체를 추출하여 학습에 넣고자 하였지만 결과가 좋지 않았다.

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

이번 프로젝트에서 시간 분배를 잘 하지 못한 탓에 시도해보고 싶었던 모델들의 앙상블을 못했다. 시도해봤다면 5점 이상을 올릴 수 있는 가설이었기 때문에 매우 아쉬웠다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

내가 처음 세운 가설들을 증명하기 위해서 시간 분배를 잘 하여 실험할 것이다. 또한 더 나은 모델 비교를 위해 자동화된 실험 환경을 구축하고 효율적인 하이퍼파라미터 튜닝 방식을 도입하고 싶다.

개인 회고

김성규

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

학습목표 : 이미지의 특성을 이해하고, 모델의 구조를 익히고 코드를 활용하자.

- 다른 데이터와는 다른 이미지 데이터의 특성을 이해하기 위해 학습에 조금 더 집중했습니다. EDA를 진행할 때 이미지는 어떤 부분에서 EDA를 해야하는지 학습하였습니다.
- 또한 기본적인 모델의 구조를 파악하기 위하여 한 줄씩 코드를 수행하며 이해하려 하였고, 이를 통해 다양한 모델을 학습하려 노력했습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

- 기본적인 모델을 수행하며, 데이터를 변환하여 증강하는 작업을 진행하였습니다. 또, 한 이미지 내에 여러 객체가 존재하는 이미지들에서 객체를 분리하는 작업을 통해 이미지 데이터를 증강하여 오버피팅을 줄이고 조금 더 다양한 이미지들을 구분할 수 있는 모델을 만들고자 했습니다.

3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- 제공된 이미지 데이터셋을 검증하고, 정리하여 조금 더 오버피팅을 방지하는 방향으로 데이터셋을 구성하는 것을 목적으로 진행하였습니다. competitions에서 높은 성적을 내기 위해서는 다양한 데이터에 적합한 모델을 수행하는 것도 중요하지만 competitions에서 요구하는 test 데이터셋에 더 집중하여 높은 결과를 도출해야한다는 것을 깨달았습니다.

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 모델을 깊게 이해하기 위해 논문과 코드 수행을 함께 하며 공부하는 방향으로 진행하였습니다. 확실히 논문에서 의도하는 목적과 그 목적을 이루기 위해 어떤 선택들을 하여 모델을 구성했는지 알 수 있었습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 이미지 데이터를 이해하고, 모델의 작동방식을 이해하는데 너무 많은 시간을 소모하였습니다.
- 시간적 한계가 커 다양한 모델을 수행해보지 못한 것이 아쉽습니다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- Git 등 협업 도구들을 조금 더 다양하게 활용하여 협업을 좀 더 원활하게 할 것입니다.
- 스스로 가설을 세우고 검증을 통해 문제를 정의하고, 해결방안을 찾는 과정을 학습할 것입니다.
- 한가지 모델을 깊게 파고드는 것도 좋지만 다양한 모델을 수행해볼 수 있도록 노력할 것입니다.

개인 회고

김재진

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- Sketch image classification 문제를 해결하기 위해 다양한 모델의 성능평가(CLIP, ConvNeXt, EfficientNet, Dino, Resnet 등등)와 실험, 데이터셋 수정 진행
- 기존 데이터셋에서는 클래스간 데이터 수를 맞추기 위해 단순 flip으로 up sampling이 진행되었는데 제거 후 Randaug기법으로 up sampling 진행

2. 나는 어떤 방식으로 모델을 개선했는가?

- CLIP에서는 기존 방식에서의 loss와 One-hot encoding label로의 loss의 비교, text encode weight freezing 여부에 따른 비교, 클래스당 1,80의 prompt를 제공했을 때의 성능을 평가
- Dino ViT의 마지막 분류 모델로 linear, mlp, attention으로 학습 진행

3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- 위에 1,2이외에도 ensemble, augmentation등을 하면서 정확도의 향상이 이루어지긴 했지만 그러한 지점에서 더 명확한 깨달음을 얻으려 하지 못하고 다음으로 진행만 한 것 아쉬웠으며, 5번 항목에서 후술

4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- Wandb를 활용하여 학습과정을 시각적으로 간편하게 볼 수 있어 모델 AB test의 시각화, 하이퍼 파라미터 값에 따른 학습 추세 인지하기 용이

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 프로젝트를 함에 있어 명확한 목표 설정을 하지 못한 것이 난잡한 실험으로 이어지게 되며, 목표의 부재가 리더보드 성적에 감정적으로 연연하진 않지만 은연중에 유일한 목표가 되어 그것만 보고 실험만 하게 되었다. 여러 실험을 하기 위해 검색하며 기본적인 지식과 논문을 통한 결론을 얻긴 했으나, 직접 진행하면서 추세정도만 파악했지 결과들에 대한 원인 파악을 하려는 과정들을 소홀히 한 것이 매우 아쉬웠다
- 개인적으로도 목표의 부재가 문제가 되었듯, 팀적으로도 목표와 과정을 명확히 하지 못한 것이 원활한 협업에 대한 필요성을 퇴색되게 한 것 같다(개인적으론 다들 열심히 했지만, 협업 측면에서 아쉬움이 남음)
- 소스코드 평가기준표를 보니 너무 나이브하게 작성했다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 개인/팀 * 최종/구간별 목표 설정 - 대단한 목표가 아니더라도 설정 자체에 의미가 있고 뚜렷해 지는 과정이라고 생각한다.
- 느리더라도 제대로 하자! - 현재는 결과보단 과정의 인과를 명확히 배우는 것이 중요하다는 건 당연히 알고는 있지만 실천하지 못했다. 이제부터는 기록을 남기며 제대로 학습하려 하며, 아예 하루 중에 그럴 시간을 할당해서 진행할 예정

개인 회고

김유경

나는 내 학습목표 달성을 위해 무엇을 했는가?

- 데이터 증강 방법이 모델 성능에 얼마나 유의미한지 확인하고, 데이터셋 설정부터 모델의 학습 및 추론 과정등의 프로젝트 파이프라인을 명확히 이해하여 베이스라인 구조를 개선하는 것을 목표로 삼았습니다. 이를 위해 CutMix 증강 기법을 적용하였고, 기존 베이스라인을 리팩토링하여 보다 효율적인 실행 환경을 구축했습니다.

나는 어떤 방식으로 모델을 개선했는가?

- 다양한 데이터를 사용하여 모델의 일반화 성능을 높이기 위해 CutMix 증강 기법을 모델에 적용하였습니다. 그리고 soft voting 앙상블 기법을 통해 단일 모델에 의존하지 않고, 여러 모델의 예측 확률을 반영하여 모델의 불확실성을 개선하였습니다.

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- CutMix 기법을 적용으로 약 0.004의 정확도를 향상시켰습니다. 해당 증강 기법을 두 가지 방식으로 구현하여, 각 방법의 코드 효율성을 고민해 볼 수 있었습니다.

- 결과가 좋은 모델들을 기반으로 앙상블을 시도한 결과, 정확도가 약 0.01 증가했습니다. 하지만, 동일한 모델끼리만 앙상블을 진행해서 그 효과가 제한적이었던 점과 더 다양한 앙상블 기법을 추가로 시도해보지 못한 점에 대해 아쉬움이 남습니다.

전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- StepLR 스케줄러를 사용하여 일정한 간격으로 학습률을 감소시켰고, 에폭이 증가함에 따라 학습률을 점진적으로 낮춤으로써 모델이 보다 안정적으로 수렴할 수 있었습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- CNN계열 모델에만 의존하여 다양한 모델을 시도하지 못한 점이 아쉬웠습니다. 또한, 학습률, 배치 사이즈 외에 손실 함수나 학습률 스케줄러 같은 다양한 변화를 적용해보지 못한 점이 아쉬움이 남습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 시각화 툴을 활용하여 논리적인 단계별 분석을 수행하고, 모델의 임베딩 차원에서 이미지 속 어떤 부분에 attention이 집중되는지 자유자재로 시각화해보고 싶습니다.

- 코드 리팩토링 후에도 파라미터 변경에 따른 수작업의 한계를 느꼈기 때문에, 다음 프로젝트에서는 웹 스크립트를 활용한 자동화 작업을 시도할 것입니다.

- PR 규칙이나 커밋 규칙을 정하여 해당 rule 아래에서 git 협업을 적극적으로 활용하여 프로젝트 진행을 매끄럽게 하고 싶습니다.