다중 해상도를 활용한 조류 이미지 분류

2024.05.06

김태형, 유정화

목차

1. 문제 정의 및 데이터 설명

저해상도 조류 이미지 분류

- 입력으로 들어오는 64x64 크기의 저해상도 조류 이미지로부터 종을 분류하는 AI 알고리즘을 개발함.
- 학습 데이터는 64x64 크기의 15,834개의 저해상도 조류 이미지임.
- 평가 데이터는 64x64 크기의 6,786개의 저해상도 조류 이미지임.
- 추가적으로 학습 데이터와 1:1 쌍으로 구성된 256x256 크기의 고해상도 조류 이미지를 제공함.
- 총 25개의 범주를 가짐.

Train

Upscale Train

Test

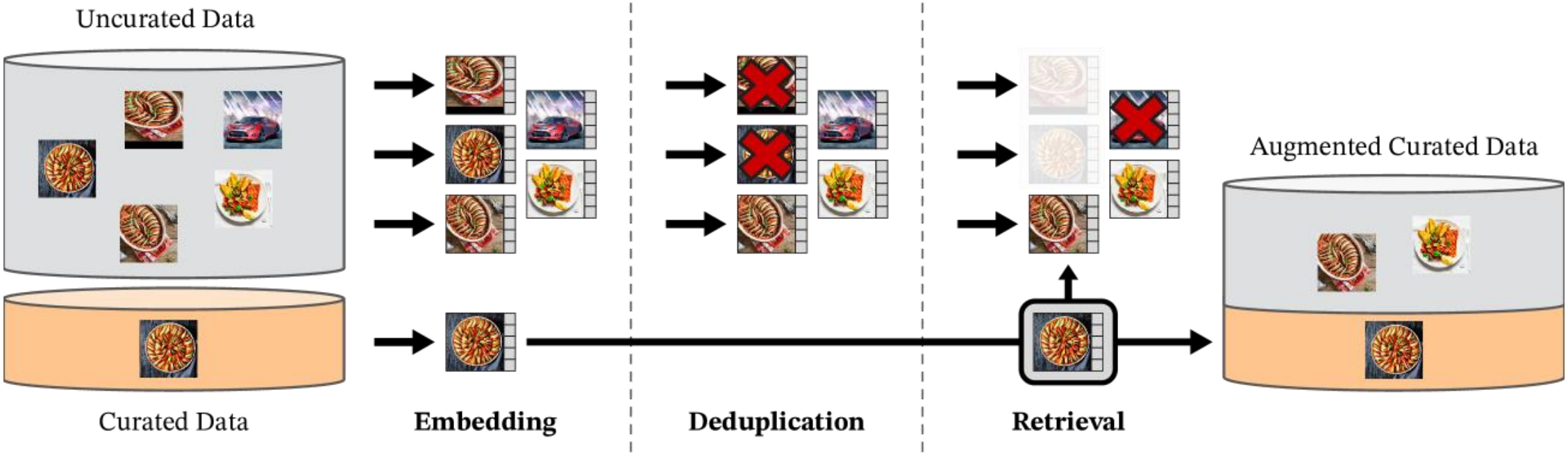
2. 다중 해상도 이미지 처리

Multiresolution Image Data Loader

- 저해상도 이미지와 고해상도 이미지를 모두 학습에 사용함.
- 검증과 평가는 저해상도 이미지만 사용함.
- 색인을 통해 다중 해상도 쌍으로 구성된 데이터의 누수를 방지함.

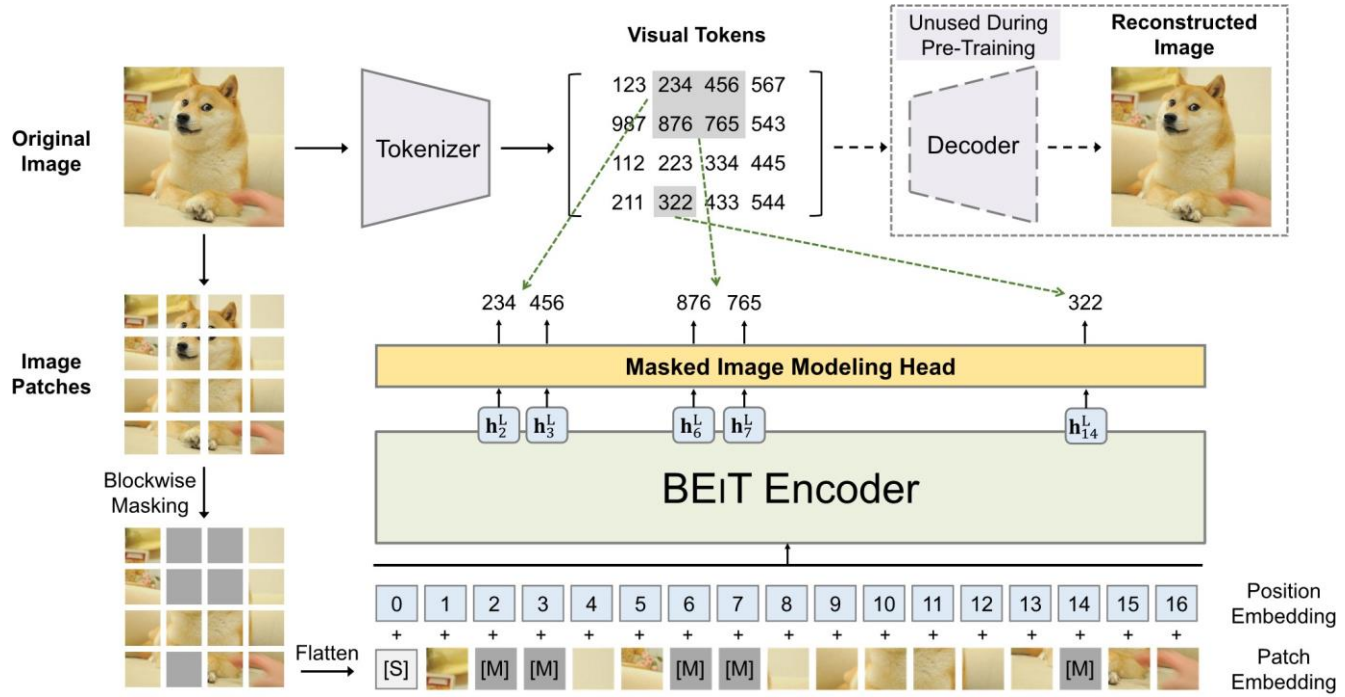
3. 방법론

DINOv2: Learning Robust Visual Features without Supervision (TMLR 2024.01)



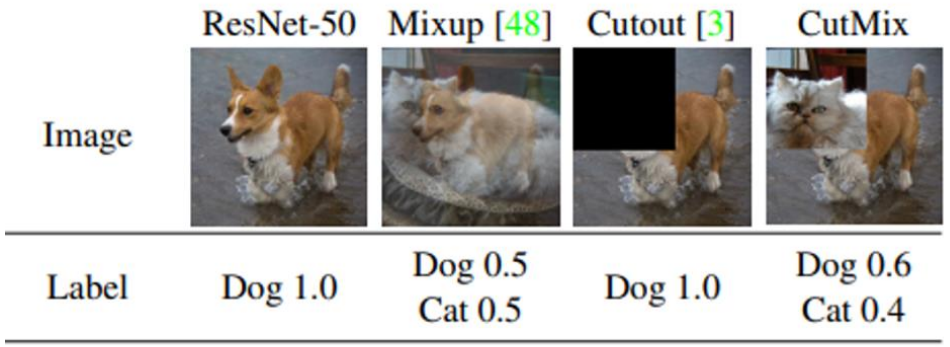
- 대량의 선별된 데이터에 대해 다양한 Vision Transformers (ViT) 아키텍처로 사전 학습된 모델임. (self-supervised learning 방법으로 학습)
- Embedding: 선별된 여러 데이터셋의 이미지와 선별되지 않은 대규모 이미지에 대해 Vit-H/16 네트워크를 사용하여 임베딩을 계산함.
- Duplication: 선별되지 않은 데이터에 대해 중복 감지 파이프라인을 적용하고 중복이라고 판단되는 이미지를 제거함.
- Retrieval: 선별된 소스 이미지에 가까운 선별되지 않은 이미지를 검색함. (k-mean 클러스터링, nearest neighbor 방법 이용)
- 검색된 선별되지 않은 이미지를 초기 데이터셋에 합쳐서 확장된 선별된 데이터(augmented curated data)를 구축함.

BEiT: BERT Pre-Training of Image Transformers (ICLR 2022)



- 사전 학습된 DALL-E tokenizer를 이용해서 이미지 패치들을 토큰화 후 BERT 방식으로 훈련된 ViT 모델임.
- DALL-E tokenizer를 통해 이미지를 이산화 된 visual token으로 만듦.
- 이미지 패치 40%에 마스킹을 적용하고, 마스킹 된 패치들은 학습 가능한 임베딩으로 교체한 뒤 마스킹 되지 않은 패치와 함께 ViT에 입력함.
- 마스킹 되었던 패치들에 대해 DALL-E를 통해서 미리 생성 되었던 이산화 된 토큰을 예측하도록 ViT가 훈련됨.

CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features (ICCV 2019)



	Mixup	Cutout	CutMix
Usage of full image region	✔	✘	✔
Regional dropout	✘	✔	✔
Mixed image & label	✔	✘	✔

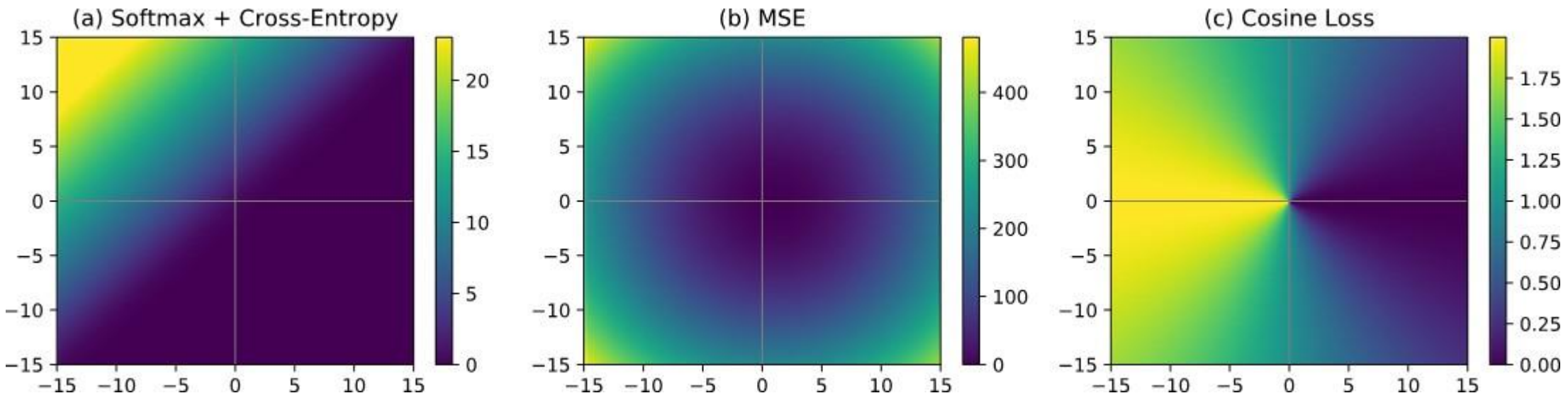
Table 2: Comparison among Mixup, Cutout, and CutMix.

Model	# Params	Top-1 Err (%)	Top-5 Err (%)
ResNet-101 (Baseline) [12]	44.6 M	21.87	6.29
ResNet-101 + Cutout [3]	44.6 M	20.72	5.51
ResNet-101 + Mixup [48]	44.6 M	20.52	5.28
ResNet-101 + CutMix	44.6 M	**20.17**	**5.24**
ResNeXt-101 (Baseline) [45]	44.1 M	21.18	5.57
ResNeXt-101 + CutMix	44.1 M	**19.47**	**5.03**

Table 4: Impact of CutMix on ImageNet classification for ResNet-101 and ResNext-101.

- 이미지 기반 태스크의 성능을 끌어 올리기 위한 데이터 증강법 중 하나임.
- Cut-and-paste 방법을 이용함.
- 두 개의 데이터에서 이미지의 일부를 컷(cut)한 후 섞고(mix), 레이블에 대해서도 알파 블렌딩을 함.
- Mixup: 두 이미지를 전체 영역에 대해 섞고, 섞은 비율 만큼 레이블에 대해서도 알파 블렌딩을 함.
- Cutout: 이미지의 일부를 컷함.

Deep Learning on Small Datasets without Pre-Training using Cosine Loss (WACV 2020)



- 데이터 셋이 작을 경우 크로스 엔트로피(Cross-Entropy)보다 코사인 로스(Cosine Loss)가 더 나은 성능을 보임.
- Cosine Loss : $\mathcal{L}_{\cos}(x, y) = 1 - \langle \varphi(y), \psi(f_\theta(x)) \rangle$
  - x: 데이터, y: 레이블
  - $f_\theta(x)$: 네트워크
  - φ: 임베딩된 feature를 prediction space로 변환하는 함수
  - ψ: 클래스를 prediction space로 변환하는 함수
  - $\langle . . \rangle$: 내적

Hyperparameter Setting & Results

- 최종 제출 모델은 stratified 5-fold를 활용한 BEiT와 DINOv2를 soft voting한 결과임.
- BEiT : beit-large-patch16-224-pt22k-ft22k
- DINOv2 : dinov2-large
- 하이퍼 파라미터는 다음과 같음.
- image_size : 224, optimizer : 'adamw,' learning rate : 0.00003(BeiT) / 0.00001(DINOv2), scheduler : 'cosine annealing'
- batch_size : 64, epochs : 10, mixed_presicion : 16
- A100 (40GB)로 실험함.
- 규칙에 위반되지 않는 BEiT만 사용한 모델의 Private Accuracy는 0.98208임.

Models	Private Accuracy	Public Accuracy
BEiT	0.98208	0.97961
DINOv2	**0.98303**	0.98122
BEiT+DINOv2	0.98274	**0.98313**

5. 결론

- 저해상도 이미지 분류를 위해 다양한 실험을 진행함.
- 64x64 사이즈의 이미지를 224 사이즈보다 큰 384, 512 사이즈로 리사이즈하여 학습하면 loss가 잘 수렴하지 않음.
- SRCNN, ESRGAN, Swin2SR, HAT 등 Super-Resolution 기법을 활용하여 데이터를 생성 후 학습함.
- SR 성능이 향상될수록 모델의 분류 성능이 올랐지만 64x64 원본 데이터로 학습한 모델의 성능이 더 좋음.
- 저해상도 이미지와 고해상도 이미지를 같이 학습에 사용하면 성능이 더 좋아짐.
- RandAugment, mixup 기법보다 cutmix의 성능이 좋음.
- Cosine Embedding Loss를 활용하면 성능이 더 좋아짐.
- Test Time Augmentation(TTA)는 성능이 하락함.

Thank you ☺

taehyeong93@korea.ac.kr