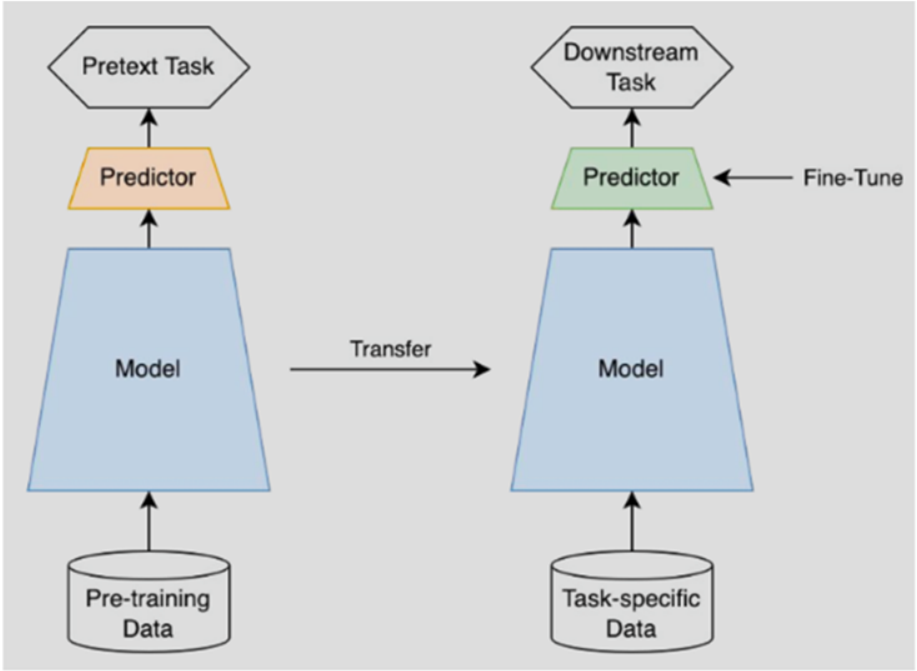
2024.04.09 (2회차) 미팅

2021105600 박지후
2018101819 김세한
2014110450 윤영근

label이 존재하는 데이터셋 얻기 힘들어
-> 상대적으로 얻기 쉬운 label없는 이미지로 target에 대해서 미리 모델 학습시키면 적은 label 데이터 셋으로도 효율적으로 학습 가능 : **unsupervised learning, self-supervised learning(SSL)**

Unlabelled image로 미리 모델을 어떻게 학습시킬 것인가? -> 그걸로 우리가 문제(pretext task)와 정답을 만들어서 그걸 학습시킴 : **self-supervised learning** (unsupervised과의 차이)



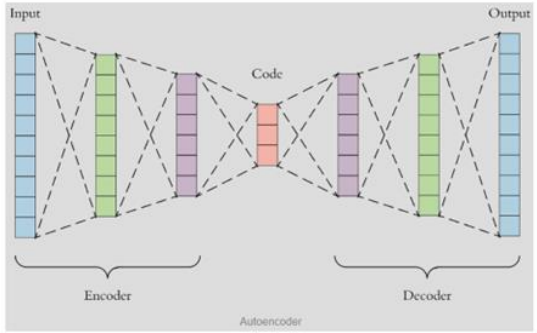
어떤 pretext task를 만들 거냐?

1. 한 샘플의 일부를 통해 그 샘플의 나머지를 예측하도록(intra) : self-prediction, 즉 하나의 완성형 이미지가 label 자체

1) 아예 없는 부분을 reconstruction해서 완성형 예측(generative learning) : autoencoder(encoder+decoder)필요, 즉 둘 다 학습 시켜야 함(계산 cost 큼). Classification일 경우 decoder는 downstream에서 필요 없음

ex) BERT, MAE…

2) 변형된 일부를 통해 완성형 예측(proxy task) : decoder필요없음.



2. 두 샘플 간의 관계를 예측하도록(inter) : constrative learning

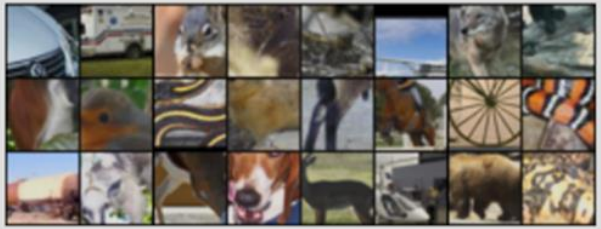


Fig. 1. Exemplary patches sampled from the STL unlabeled dataset which are later augmented by various transformations to obtain surrogate data for the CNN training.

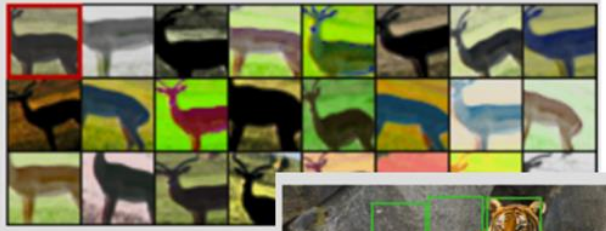
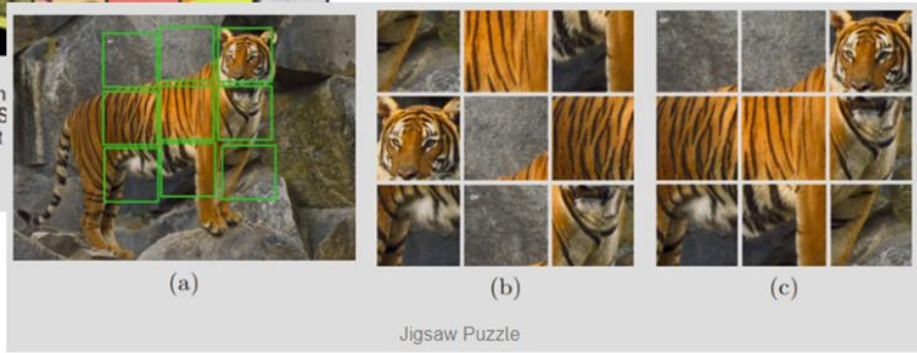


Fig. 2. Several random tran
patches extracted from the S
('seed') patch is in the top left

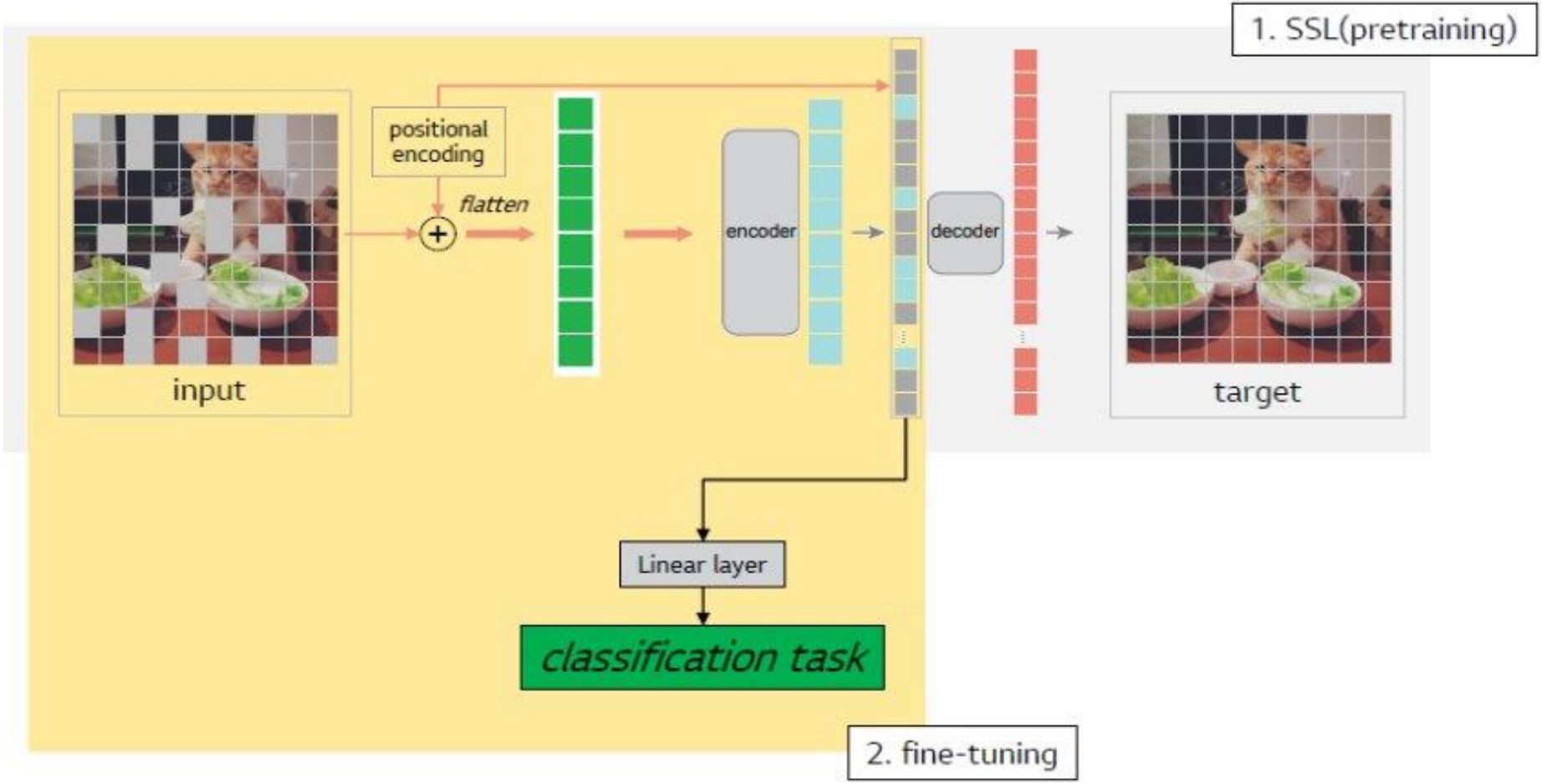
Exemplar



1. **Masking** : 이미지를 patch로 나눈 뒤, **uniform 분포** 따르는 **random** sampling으로 **75%** masking
    ->masking 비율이 높은 이유 : word token하나하나가 의미 정보가 dense한 언어와 다르게 이미지는 인접한 patch끼리는 중복이 많아 interpolation만 해도 어느 정도 될 정도로 의미 정보가 sparse함, 난이도를 어렵게 해서 학습을 더 잘 시키기 위해
    -> uniform 분포를 따르는 이유 : center부분만 학습을 잘 하는 문제(center bias문제)를 방지
2. **Positional encoding**
3. **Encoder :** visible patch만 image representation(encoded visible token) 추출, ViT사용

4. **Decoder :** encoded visible token에 embed token(Encoded visible token의 차원과 decoder의 차원을 맞춰주기 위한 linear projection)
    mask token(index정보)추가해 unshuffled
    mask token은 embedding 적용X라서 추가적인 positional embedding
    transformer block : patch가 아닌 pixel level reconstruction(이미지는 의미론적인 분해가 가능한 게 아니라서)
    predictor projection(Loss 계산을 위해 target image와 shape을 맞춰주기 위한 linear projection)
    class token삭제
5. **Masked patch MSE계산**(visible patch에 대해서는 진행 X), backpropagation(pre-training)

9. 해당 model에서 decoder 삭제 후 MLP추가(fine-tuning)


1. SSL(pretraining)
positional encoding
flatten
encoder
decoder
input
target
Linear layer
classification task
2. fine-tuning

MAE - 특징

blocks	ft	lin
1	84.8	65.5
2	**84.9**	70.0
4	**84.9**	71.9
8	**84.9**	**73.5**
12	84.4	73.3

(a) **Decoder depth**. A deep decoder can improve linear probing accuracy.

dim	ft	lin
128	**84.9**	69.1
256	84.8	71.3
512	**84.9**	**73.5**
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width**. The decoder can be narrower than the encoder (1024-d).

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	**84.9**	**73.5**	**1**×

(c) **Mask token**. An encoder without mask tokens is more accurate and faster (Table 2).

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	**85.4**	**73.9**
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target**. Pixels as reconstruction targets are effective.

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

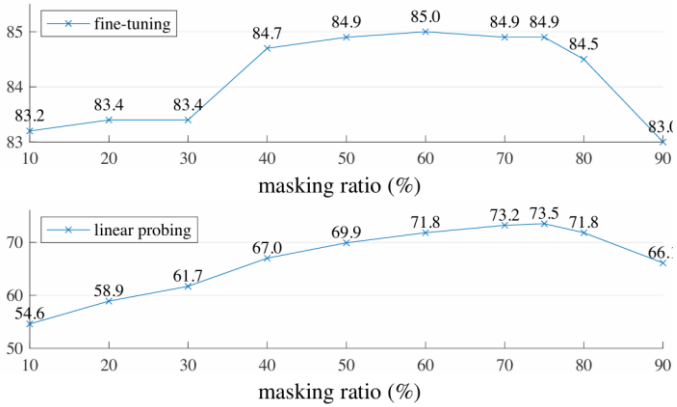
(e) **Data augmentation**. Our MAE works with minimal or no augmentation.

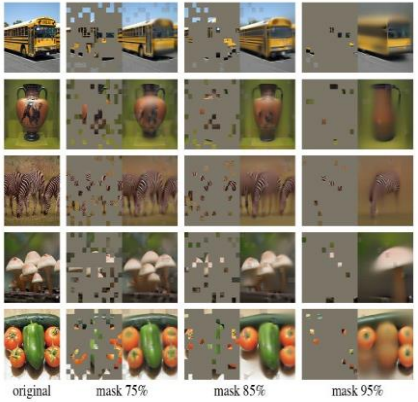
case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

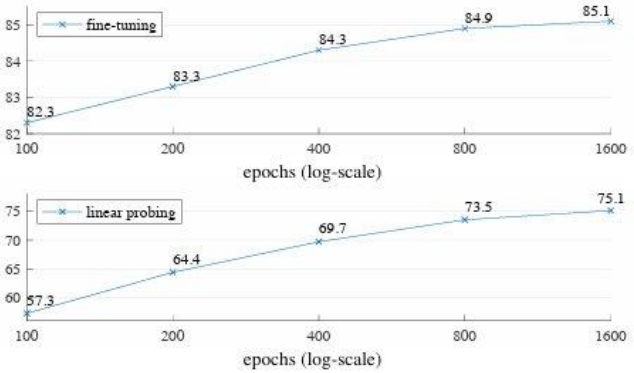
(f) **Mask sampling**. Random sampling works the best. See Figure 6 for visualizations.

random 75% block 50% grid 75%

MAE - 특징







=> FC Layer만 바꾸기(linear probing)보단 다시 학습시켜야 한다.

Natural Image MAE와 비교했을 때 X-ray Image MAE 시 고려해야 할 점

1. 일반 이미지보다 spatial consistency가 높음
   => Masking 비율 90%

2. 흉부 X-ray MAE 논문의 경우, 집중해야 할 질환 부분이 local하므로 작게 crop할 시 판단에 필수적인 부분이 사라질 수 있고, 질환이 여러 군데에 퍼져있을 경우를 고려
   => random resize crop 진행 시, 일반적인 이미지 crop(0.2~0.1)보다 더 큰 size의 image patch 사용 (0.5~1.0)

-> 모델이 집중해야 할 부위가 확실한 본 task에서는 crop size는 (0.5~1.0)으로 하되, random distribution을 center에 집중하도록 하는 게 더 좋지 않을까? + masking도 center 집중으로

Fine Tuning

lr schedular(1.5e-4부터cosine annealing strategy), optimizer(AdamW optimizer : β1 = 0.9,β1 = 0.95) 그대로
1.5e-4/5/6
Layer-wise LR decay 0.55
RandAug magnitude 6
DropPath rate 0.2
75 epochs

Gradcam

1. 각 Feature map의 원소가 특정 클래스에 주는 영향력을계산 : gradiant이용

2. Feature map의 각 원소를 미분한값을 나눠서 구한 가중치의 평균값을 각 Feature map에 곱함

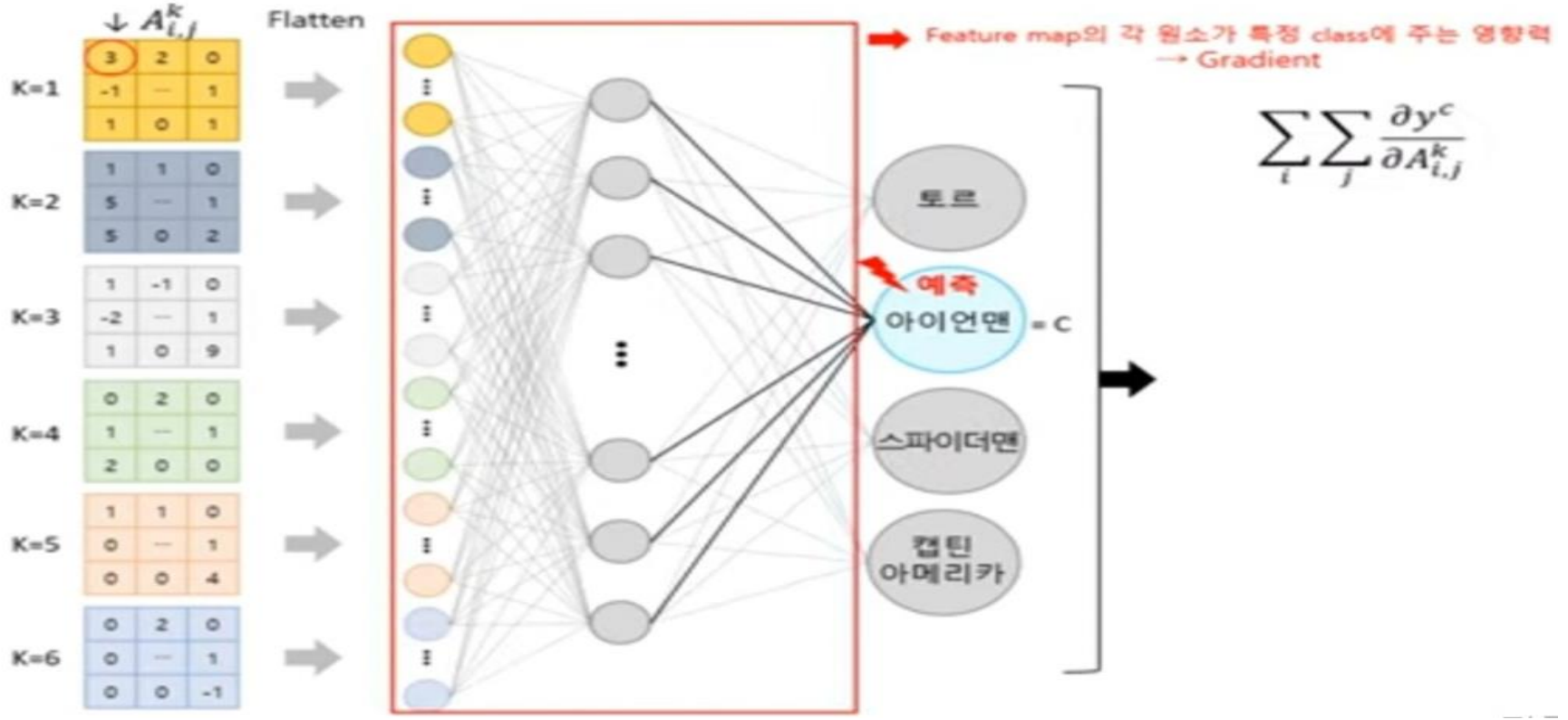
3. 각 Featuere 별 heat map 계산

C class에 대한
Grad-CAM Score

$$L^{c}_{Grad-CAM} = ReLU \sum_{k} a^{c}_{k} A^{k}$$

$$a^{c}_{k} = \frac{1}{Z} \sum_{i} \sum_{j} \frac{\partial y^{c}}{\partial A^{k}_{i,j}}$$

Gradcam


$\downarrow A^k_{i,j}$
Flatten
K=1
K=2
K=3
K=4
K=5
K=6
Feature map의 각 원소가 특정 class에 주는 영향력
→ Gradient
$\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{i,j}}$
토르
예측
아이언맨
= C
스파이더맨
캡틴
아메리카

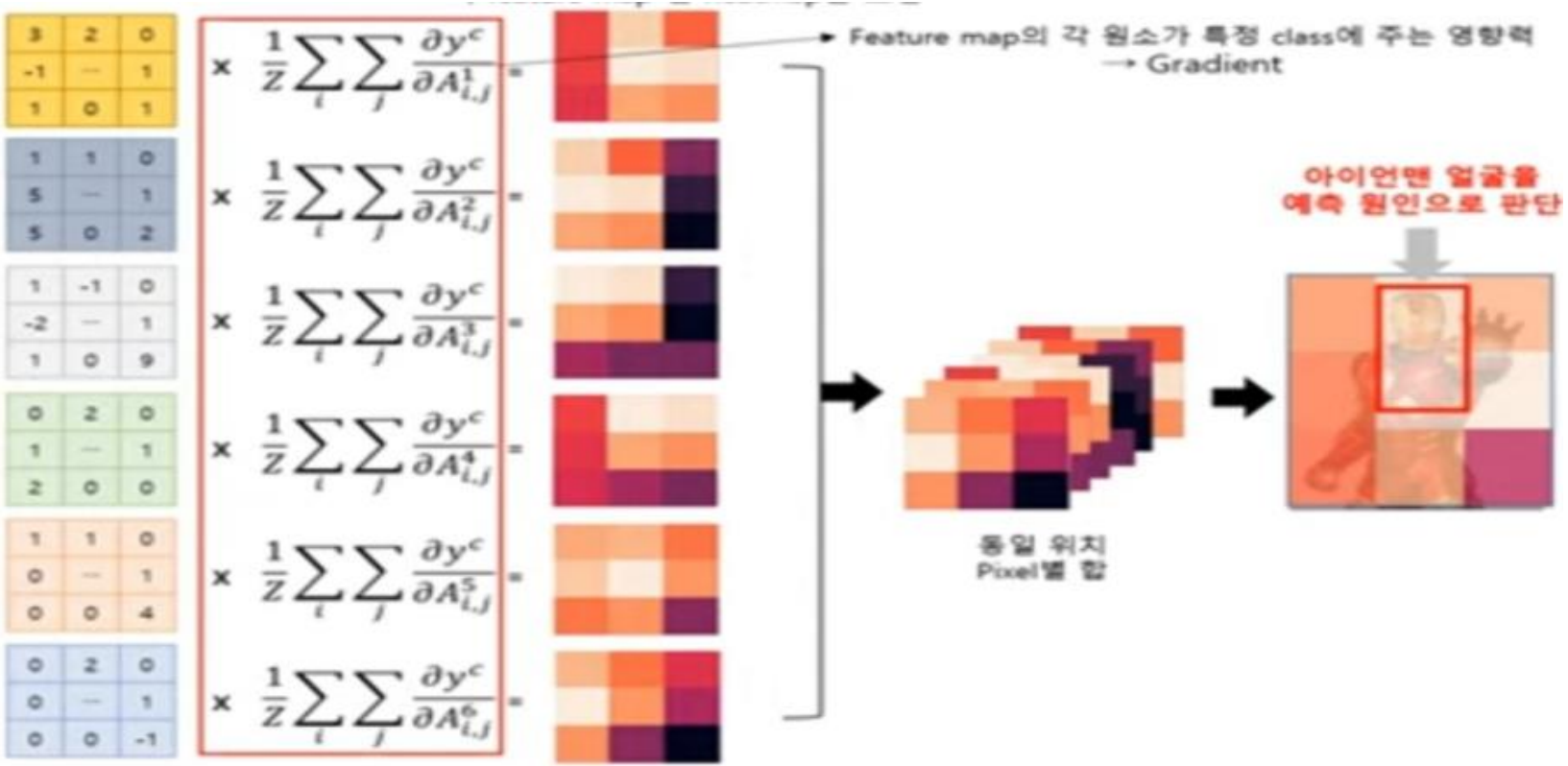
Gradcam의 아이디어

1. 모델에 특정입력을 넣었을때 나온 feature map을 weighted average 후 heatmap 구성

2. weight를 gradient를 map으로 backpropagation한후 map에서 해당 gradient의 합(average)을 구해 해당 map의 중요도를 계산

3. heatmap을 원본 이미지 크기로 resize한 뒤 오버레이

Gradcam


Feature map의 각 원소가 특정 class에 주는 영향력
Gradient
아이언맨 얼굴을
예측 원인으로 판단
동일 위치
Pixel별 합

Gradcam(ViT)

1. ViT의 경우 tokenization후 feature를 추출하는 모델로 feature map이 아님(바로적용 불가)

2. patch별로 추출했던 feature의 중요도를 2차원 상으로 다시 reshape

3. CNN방법과 달리 target class지정 필요

```
    # 타겟 클래스 설정
    target_class = labels[0]  # 첫 번째 이미지의 실제 라벨을 타겟으로 사용
    targets = [torch.tensor([target_class]).to(device)]
    # Generate the CAM mask
    grayscale_cam = cam(input_tensor=img)[0, :]
```

개발 방향

Imagenet -> 어깨 X-ray(CNN, ViT)

흉부 X-ray MAE -> 어깨 X-ray(CNN, ViT)

흉부 X-ray MAE -> 어깨 X-ray proxy -> 어깨 X-ray(CNN, ViT)

흉부 X-ray MAE -> 흉부 X-ray proxy -> 어깨 X-ray(CNN, ViT)

흉부 X-ray MAE -> 어깨 X-ray MAE(center/random) -> 어깨 X-ray(CNN, ViT)

Model	Pretrained Dataset	Method	Pretrained	Finetuned (NIH Chest X-ray)	mAUC
DenseNet-121	ImageNet	Categorization	torchvision official	google drive	82.2
ResNet-50	ImageNet	MoCo v2	google drive	google drive	80.9
ResNet-50	ImageNet	BYOL	google drive	google drive	81.0
ResNet-50	ImageNet	SwAV	google drive	google drive	81.5
DenseNet-121	X-rays (0.3M)	MoCo v2	google drive	google drive	80.6
DenseNet 121	X-rays (0.3M)	MAE	google drive	google drive	81.2
ViT-Small/16	ImageNet	Categorization	DeiT Official	google drive	79.6
ViT-Small/16	ImageNet	MAE	google drive	google drive	78.6
ViT-Small/16	X-rays (0.3M)	MAE	google drive	google drive	**82.3**
ViT-Base/16	X-rays (0.5M)	MAE	google drive	google drive	**83.0**

Model	Pretrained Dataset	Finetuned (Chest X-ray)	mAUC	Finetuned (CheXpert)	mAUC	Finetuned (COVIDx)	Accuracy
ViT-Small/16	X-rays (0.3M)	google drive	**82.3**	google drive	**89.2**	google drive	**95.2**
ViT-Base/16	X-rays (0.5M)	google drive	**83.0**	google drive	**89.3**	google drive	**95.3**

향후 계획

4/9 : gradCAM 이론 학습 + CNN gradCAM 구현 (영근),

main code 수정 + proxy/MAE 이론 학습 (지후),

proxy와 MAE 이론 학습 + 기본 code 구현(ImageNet->CIFAR10) (세한)

4/16 : gradCAM 통계 분석 방법 학습, ViT gradCAM 구현(영근)

어깨 X-ray proxy(rotation) -> 어깨 X-ray(CNN, ViT) (지후)

흉부 X-ray MAE -> 어깨 X-ray(CNN, ViT)
Imagenet MAE -> 어깨 X-ray(CNN, ViT) (세한)

4/23 : 흉부 X-ray MAE -> 흉부 X-ray proxy -> 어깨 X-ray(CNN, ViT)

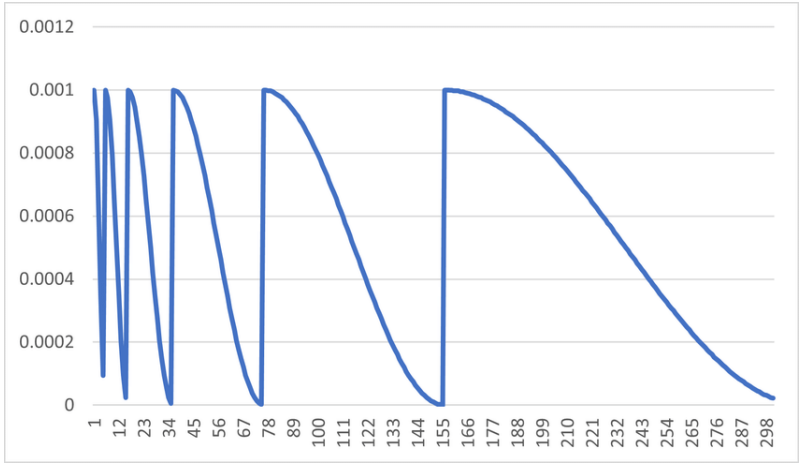
Imagenet MAE -> -> 흉부 X-ray proxy -> 어깨 X-ray(CNN, ViT)

각 모델 gradCAM 결과 확인

4/30 : 중간보고서 제출

질문 사항

1. rotation proxy는 이미지의 중요한 object 가 무엇인지와 그 방향/각도에 대해서 모델이 학습할 수 있도록 함. 근데 어깨 x-ray로 이를 진행했을 때 정작 중요한 중간 부분보다는 edge파트가 집중적으로 학습되는 건 아닌지? 그럼 우리 본 task에서 주는 긍정적 영향이 클까?
2. proxy task에서와 본 task에서 같은 어깨 x-ray 데이터를 사용해 학습?
3. 1.5e-4부터 cosine annealing strategy 어떻게?
4. MAE DenseNet121 사용?



weight decay = 0.05,
learning rate = 1.5e-4, batch size = 2048

RandomResizedCrop => image의 multi scale feature 학습 가능, training sample 부족으로 인한 과적합 방지