



SimMIM: a Simple Framework for Masked Image Modeling



SimMIM

- masked image modeling을 위한 단순한 framework
- discrete VAE 혹은 clustering을 통한 block-wise masking과 tokenization 없이 간단하게 접근

디자인

1)

큰 masked patch size(32)로 random masking 하는 것이 강력한 pre-text task를 만들 수 있었음

2)

direct regression을 통해서 raw pixels의 RGB 값을 예측하는 것은 복잡한 디자인의 patch classification 보다 나쁘지 않은 성능을 보임

3) prediction head는

linear layer와 같은 가벼운 것이 무거운 것보다 나쁘지 않음

1. Introduction

- 해당 논문은 시각 신호를 마스크하여 이를 예측하는 "마스크된 신호 모델링" 작업에 대한 간단한 프레임워크를 제시함
 - 이 접근 방식은 시각 신호의 특성에 잘 부합하며, 복잡한 디자인 없이도 이전의 더 복잡한 방법과 유사하거나 더 나은 표현을 학습할 수 있음을 입증
- vision vs language
 1. image는 locality 가 조금 더 강함
→ pixel들이 근처와 밀접하고 강하게 상관관계를 맺고 있음
 2. visual signals는 raw and low-level이고, text tokens는 human-generated high level임

3. visual signal은 continuous, text token은 discrete

- 핵심 디자인과 통찰:
 1. 이미지 패치에 무작위 마스킹을 적용하여 간단하고 효과적인 방법을 제공
 2. 원시 픽셀 회귀 작업은 시각 신호의 연속성에 잘 부합하며, 복잡한 분류 접근 방식과 유사한 성능을 보임
 3. 매우 가벼운 예측 헤드를 사용하여 사전 훈련 속도를 향상시키고 전이 성능을 유지
- 이러한 간단한 접근 방식은 표현 학습에 매우 효과적이며, 대규모 모델에서도 잘 확장됨
 - 이를 통해 ImageNet-1K에서 최고의 정확도를 달성하고, Google의 JFT-3B 데이터셋보다 훨씬 적은 데이터로 큰 모델을 훈련할 수 있었음
 - 이러한 결과는 self-supervised learning 학습을 사용하여 증가하는 데이터 요구량 문제를 해결하는 데 도움

2. Related Work

Masked Language Modeling(MLM)

- 자연어 처리 분야에서 주요한 자기 지도 학습 방법 중 하나로, 가시적인 토큰을 제공하여 보이지 않는 토큰을 예측함으로써 표현을 학습
- 거대한 데이터를 활용하여 매우 큰 언어 모델을 학습하고 다양한 언어 이해 및 생성 작업에 잘 일반화 됨

Masked Image Modeling(MIM)

- 컴퓨터 비전 분야에서 주로 진행되어 온 작업으로, 이미지의 일부를 가려서 누락된 픽셀을 예측함으로써 작동함
- 이는 self-supervised task로, image inpainting이라 하는 고전적인 컴퓨터 비전 문제와 관련이 있음
- 이 논문의 접근 방식은 압축 감지와 관련이 있으며, 우리가 획득한 데이터의 대부분을 거의 손실 없이 버릴 수 있다는 점을 확인함
 - 마스크된 이미지 모델링은 image inpainting을 강력한 self-supervised 전문 작업으로 제안되며, 이는 아래쪽 작업에 대한 강력한 성능을 가짐
- 자기 지도 학습 접근 방식의 다양한 전문 작업 중 일부는 마스크된 이미지 모델링과는 매우 다르지만, 일부 작업은 신호의 보이지 않는 부분을 예측하는 철학을 따르며 관련이 있음

3. Approach

3-1. A Masked Image Modeling Framework

- SimMIM은 입력 이미지 신호의 일부를 마스킹하고 마스킹된 영역에서 원래 신호를 예측하는 **masked image modeling**을 통해 표현을 학습
- framework는 다음 4개의 주요 구성 요소로 구성됨

1. Masking Strategy

- 입력 이미지를 input으로 받아 마스킹 할 영역을 선택하는 방법과 선택한 영역의 마스킹을 구현하는 방법을 설계
- 마스킹 후 변환된 이미지가 새로운 입력으로 사용됨

2. Encoder architecture

- 마스킹된 이미지에 대한 latent feature 표현을 추출한 다음 마스킹된 영역에서 원래 신호를 예측하는 데 사용됨
- 해당 논문에서는 주로 Vanilla ViT와 Swin Transformer의 일반적인 두 가지 ViT 아키텍처를 고려함

▼ Vanilla ViT

- 컴퓨터 비전 작업을 위한 효과적인 딥러닝 모델 중 하나
- 기존의 컨볼루션 신경망(CNN) 대신 트랜스포머(Transformer) 아키텍처를 사용하여 이미지를 처리
- 입력 이미지를 작은 패치들로 분할하고, 각 패치에 대한 임베딩을 생성한 후 이를 트랜스포머의 인코더에 전달하여 시퀀스 데이터로 처리

▼ Swin Transformer

- 기존의 트랜스포머 아키텍처를 확장하여 시각적인 특징을 효과적으로 캡처 할 수 있도록 설계된 딥러닝 아키텍처
- 입력 이미지를 임베딩한 후, 계층적인 구조로 정보를 처리하여 전역적인 문맥을 고려
 - 이미지 내의 장거리 의존성을 효과적으로 모델링할 수 있게 함

3. Prediction Head

- latent feature 표현에 적용되어 마스킹된 영역에서 원래 신호의 한 형태를 생성

4. Prediction target

- 예측할 원래 신호의 형태를 정의
 - 픽셀 값이거나 픽셀의 변환 등
- cross-entropy classification loss와 ℓ_1 또는 ℓ_2 regression loss를 포함한 일반적인 옵션으로 loss 유형을 정의

3-2. Masking Strategy

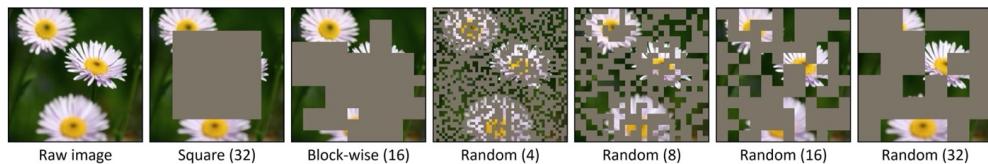


Figure 2. Illustration of masking area generated by different masking strategies using a same mask ratio of 0.6: square masking [38], block-wise masking [1] apply on 16-sized patches, and our simple random masking strategy on different patch sizes (e.g., 4, 8, 16 and 32).

- 마스킹된 영역의 입력 변환을 위해 각 마스킹된 패치를 대체하기 위해 학습 가능한 마스크 토큰 벡터를 사용
 - 토큰 벡터 차원은 패치 임베딩 후 다른 보이는 패치 표현과 동일하게 설정됨
- 마스킹 영역 선택을 위해 저자들은 다음 마스킹 전략을 연구하였음

Patch-aligned random masking

- 이미지 패치는 ViT의 기본 처리 단위로, 패치가 완전히 보이거나 완전히 마스킹되는 패치 레벨에서 마스킹을 작동하는 것이 편리함
 - Swin Transformer
 - 서로 다른 해상도 단계의 등가 패치 크기인 $4 \times 4 \sim 32 \times 32$ 를 고려
 - 기본적으로 마지막 단계의 패치 크기인 32×32 를 채택
 - ViT
 - 기본 마스크 패치 크기로 32×32 채택

Other masking strategies

- **중앙 영역 마스킹 전략:** 이미지에서 랜덤으로 움직일 수 있도록 완화
- **복잡한 블록별 마스킹 전략:** 16×16 와 32×32 의 두 마스크 패치 크기에서 위의 마스크 전략을 시도

3-3. Prediction Head

- 입력이 인코더 출력과 일치하고 출력이 예측 타겟을 달성하는 한 임의의 형식과 용량을 가질 수 있음
- 일부 초기 연구들은 무거운 prediction head (디코더)를 사용하기 위해 오토인코더를 사용
- 본 논문에서는 prediction head가 linear layer 정도로 매우 가벼워질 수 있음을 입증 함
- 또한 해당 논문의 저자들은 2-layer MLP, inverse Swin-T, inverse Swin-B와 같이 더 무거운 head들도 시도해 봄

3-4. Prediction Targets

Raw pixel value regression

- 픽셀 값은 색상 공간에서 연속적으로 분포함
⇒ 회귀를 통해 마스킹된 영역의 픽셀을 예측하는 쪽으로 작동
- 일반적으로 비전 아키텍처는 ViT에서 16배, 대부분의 다른 아키텍처에서 32배와 같이 다운샘플링된 해상도의 feature map을 생성함
 - 입력 이미지의 전체 해상도에서 모든 픽셀 값을 예측하기 위해 feature map의 각 feature 벡터를 원래 해상도로 다시 매핑하고, 해당 벡터가 해당 픽셀의 예측을 담당하도록 함
- masking 된 pixel에 L1-loss 적용

$$L = \frac{1}{\Omega(\mathbf{x}_M)} \left\| \mathbf{y}_M^{\text{prediction}} - \mathbf{x}_M^{\text{input (RGB)}} \right\|_1,$$

- L1-loss 외에도 L2-loss와 smooth L1-loss 또한 고려

Other prediction targets

- 이전 접근 방식은 대부분 마스킹된 신호를 클러스터 또는 클래스로 변환한 다음, 마스킹된 이미지 예측을 위한 classification task를 수행하였음
- **Color clustering**

- iGPT에서 RGB 값은 많은 양의 자연 이미지를 사용하여 kmeans로 512개의 클러스터로 그룹화됨
 - 그런 다음 각 픽셀은 가장 가까운 클러스터 센터에 할당됨
 - 해당 방법을 사용하려면 9 bit 색상 팔레트를 생성하기 위한 추가 클러스터링 단계가 필요
- 실험에서는 iGPT에서 학습한 512개의 클러스터 센터를 사용
- **Vision tokenization**
 - BEiT에서는 discrete VAE (dVAE) 네트워크를 사용하여 이미지 패치를 dVAE 토대로 변환
 - 토큰 ID는 classification 타겟으로 사용됨
 - 해당 접근 방식에서는 추가 dVAE 네트워크에 대한 사전 학습이 요구됨
- **Channel-wise bin color discretization**
 - R, G, B 채널은 개별적으로 분류되며 각 채널은 동일한 bin으로 discretize 됨

4. Experiments

4-1. Ablation Study

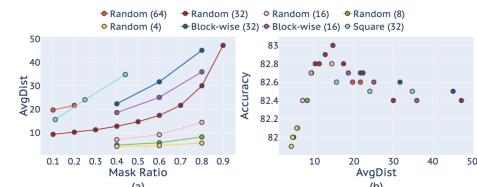
4-1-1. Settings

- **Swin-B** 를 기본 백본으로 선택
 - self-supervised 표현을 객체 감지 및 시멘틱 세그멘테이션과 같은 하위 작업에서 평가
 - 실험 부담을 줄이기 위해 입력 이미지 크기를 1922로 설정하고, 윈도우 크기를 6으로 조정
- 사전 훈련 및 미세 조정에는 ImageNet-1K 이미지 분류 데이터셋을 사용
- self-supervised pre-training에서는 **AdamW** 옵티마이저와 코사인 학습률 스케줄러를 사용하여 100개 에폭 동안 훈련
 - 가벼운 데이터 증강 전략을 선택
- **SimMIM**의 기본 구성 요소
 - 무작위 마스킹 전략, linear prediction head, L1-loss임

- 미세 조정에서도 AdamW 옵티마이저를 사용하며, 100개 에폭의 훈련과 코사인 학습률 스케줄러와 10 에폭의 웜업을 채택
- 데이터 증강에는 RandAug, Mixup, Cutmix, 라벨 스무딩, 및 랜덤 이레이징이 사용됨

4-1-2. Masking Strategy

Mask Type	Masked patch size	Mask ratio	Top-1 acc (%)
square	32	0.11 (2×2)	82.6
	32	0.25 (3×3)	82.5
	32	0.44 (4×4)	82.5
block-wise	16/32	0.4	82.7/82.7
	16/32	0.6	82.6/82.6
	16/32	0.8	82.4/82.5
random	4/8/16/32	0.4	81.9/82.0/82.4/82.9
	4/8/16/32	0.6	82.0/82.1/82.7/82.8
	4/8/16/32	0.8	82.1/82.4/82.8/82.4
	64	0.1	82.6
	64	0.2	82.6
random	32	0.1	82.7
	32	0.2	82.8
	32	0.3	82.8
	32	0.4	82.9
	32	0.5	83.0
	32	0.6	82.8
	32	0.7	82.7
	32	0.8	82.4
	32	0.9	82.4



AvgDist에 따른 fine-tuning 성능

다양한 마스킹 비율에 따른 AvgDist
(마스킹된 pixel들에 대해 가장 가까이 보이
는 픽셀까지의 평균 거리)

- 다양한 마스킹 전략이 표현 학습에 미치는 영향을 연구
 - 간단한 무작위 마스킹 전략이 다른 특별히 설계된 전략보다 더 높은 정확도를 보이며, 특히 큰 마스크 패치 크기를 사용할 때 안정적으로 좋은 결과를 보임
- 마스킹 비율을 늘리거나 패치 크기를 크게 하는 것이 정확도에 긍정적인 영향을 미침
- 또한 새로운 AvgDist 지표는 마스킹된 픽셀 간의 평균 거리를 측정하여 효과적인 마스킹 전략을 평가하는 데 도움
- 실험에서는 보통 32의 패치 크기에서 0.6의 마스킹 비율을 사용하였음
→ 안정적인 성능을 보이기 때문
- 언어 영역에서 사용되는 작은 마스킹 비율과는 다르게, 컴퓨터 비전 작업에서는 다른 마스킹 전략과 비율이 사용됨

4-1-3. Prediction Head

Head	#params	Training costs	Top-1 acc (%)
Linear	89.9M	1×	82.8
2-layer MLP	90.9M	1.2×	82.8
inverse Swin-T	115.2M	1.7×	82.4
inverse Swin-B	174.8M	2.3×	82.5

- 더 무거운 헤드는 일반적으로 더 낮은 손실을 유발하지만, 다운스트림 ImageNet-1K 작업에서의 전이 성능은 낮음
 - 이는 더 강력한 인페인팅 기능이 반드시 더 나은 다운스트림 성능을 가져오지 않음을 시사
- 추가로, 단일 선형 레이어 헤드가 미세 조정 메트릭에서 경쟁력 있는, 최적의 전이 성능을 보여준다는 점을 입증
 - 이는 마스크 이미지 모델링의 경우 대조적 학습 접근 방식에서의 헤드 설계에 대한 중요한 탐색이 필요하지 않을 수 있다는 것을 시사

4-1-4. Prediction Resolution

Image size (ratio of inputs)	비율					
	6^2 (1/32)	12^2 (1/16)	24^2 (1/8)	48^2 (1/4)	96^2 (1/2)	192^2 (1/1)
	Top-1 acc (%)	82.3	82.7	82.8	82.7	82.8

Table 3. Ablation on different prediction resolutions. A moderately large resolution (no less than 1/16) all perform well.

지나치게 낮은 해상도가 아닌 경우 거의 다 비슷한 성능을 보임

4-1-5. Prediction Target

Loss	Pred. Resolution	Top-1 acc (%)
Classification		
8-bin	192^2	82.7
8-bin	48^2	82.7
256-bin	192^2	N/A
256-bin	48^2	82.3
iGPT cluster	192^2	N/A
iGPT cluster	48^2	82.4
BEiT	-	82.7
Regression		
ℓ_2	192^2	82.7
smooth- ℓ_1	192^2	82.7
ℓ_1	192^2	82.8
ℓ_1	48^2	82.7
ℓ_1	6^2	82.3

- 다양한 예측 대상의 효과를 비교
 - L1, smooth-L1, 및 L2의 세 가지 손실이 유사한 성능을 보이며, 정교한 클래스 정의나 색상 클러스터링보다 우리의 접근 방식이 약간 더 나은 결과를 보임
 - 또한, 가시적 신호의 재구성과 가시적 신호의 예측에 대한 서로 다른 철학을 고려할 때, 마스킹된 영역만 예측하는 접근 방식이 모든 이미지 픽셀을 복구하는 것보다 우수한 성능을 보임
- ⇒ 예측 작업이 더 유망한 표현 학습 접근 방식일 수 있음을 시사

4-2. Comparison to Previous Approaches on ViT-B

- ViT-B 를 인코더로 사용할 때 시스템 수준의 비교 결과
- 사전 훈련에서는 800 epoch의 코사인 학습률 스케줄러와 20 epoch의 선형 웜업 절차를 사용했으며, 미세 조정에서는 0.65의 레이어별 학습률 감쇠를 채택하였음

Methods	Input Size	Fine-tuning Top-1 acc (%)	Linear eval Top-1 acc (%)	Pre-training costs
Sup. baseline [46]	224^2	81.8	-	-
DINO [5]	224^2	82.8	78.2	$2.0 \times$
MoCo v3 [9]	224^2	83.2	76.7	$1.8 \times$
ViT [15]	384^2	79.9	-	$\sim 4.0 \times$
BEiT [1]	224^2	83.2	56.7	$1.5 \times^\dagger$
Ours	224^2	83.8	56.7	$1.0 \times$

- 논문의 접근 방식은 이전 접근 방식보다 +0.6% 높은 83.8%의 최고 정확도를 미세 조정에서 달성하였음
- 또한, 우리의 접근 방식은 단순성 덕분에 더 높은 훈련 효율을 보였음
⇒ 해당 접근 방식이 미세 조정에 더 적합한 표현을 학습한다고 시사

4-3. Scaling Experiments with Swin Transformer

- Swin Transformer를 backbone으로 한 스케일링 실험

Methods	Pre-train	Fine-tune	Backbone	Top-1 acc (%)	Param
Sup.	192^2	224^2	Swin-B	83.3	88M
Sup.	192^2	224^2	Swin-L	83.5	197M
Sup.	192^2	224^2	SwinV2-H	83.3	658M
Ours	192^2	224^2	Swin-B	84.0	88M
Ours	192^2	224^2	Swin-L	85.4	197M
Ours	192^2	224^2	SwinV2-H	85.7	658M
Ours	192^2	512^2	SwinV2-H	87.1	658M
Ours	192^2	640^2	SwinV2-G	90.2	3.0B

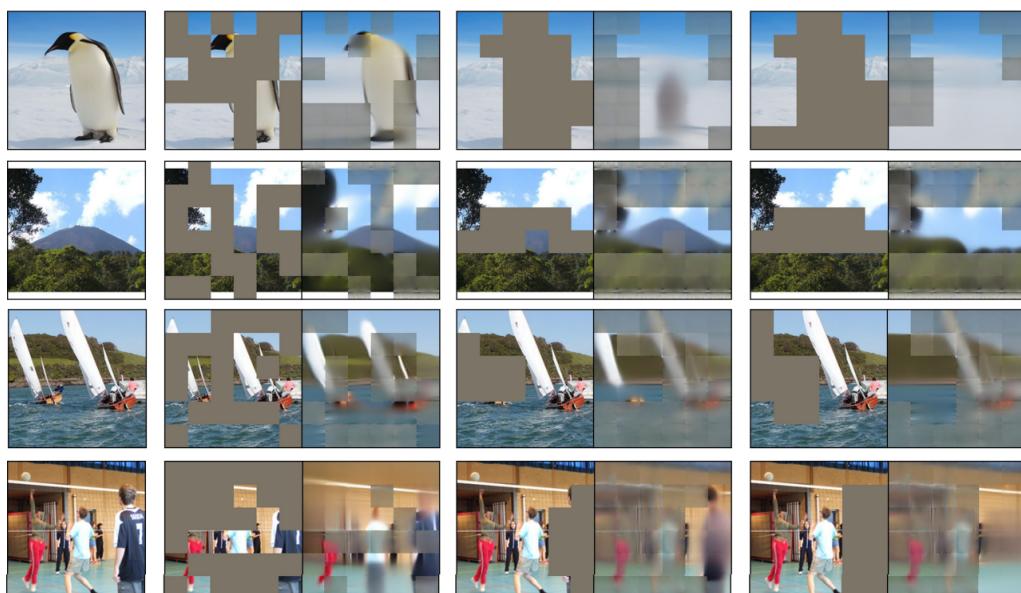
- Swin Transformer의 다양한 모델 크기를 실험에 활용하였음
 - Swin-B, Swin-L, SwinV2-H, 그리고 SwinV2-G를 활용하였음
 - 실험 부담을 줄이기 위해 사전 훈련에서는 더 작은 이미지 크기인 192^2 를 사용했고, 단계적 학습률 스케줄러를 채택하여 다른 훈련 기간의 실험에서 첫 번째 단계의 모델 훈련을 재사용하였음

- ImageNet-1K 데이터셋을 사용하여 사전 훈련하며, SwinV2-G는 ImageNet-22K-ext 데이터셋을 사용하였음
- 사전 훈련은 800 에폭 동안 진행되었고, 미세 조정에서는 더 큰 이미지 크기를 사용하여 100-에폭 또는 50-에폭을 사용하였음
- 결과적으로, Swin-B, Swin-L, 그리고 SwinV2-H는 감독된 대조군보다 더 높은 정확도를 달성했으며, SwinV2-H 모델의 경우 ImageNet-1K에서 87.1%의 최고 정확도를 달성하였음
- SimMIM 접근 방식은 JFT-3B 데이터 대신에 약 40배 작은 데이터를 사용하여 3B SwinV2-G 모델의 훈련을 돋는데, 이는 여러 대표적인 비전 벤치마크에서 강력한 성능을 보였음

4-4. Visualization

- 제안된 접근 방식과 주요 설계를 시각적으로 이해하고자 함
 - 예제 이미지로는 ImageNet-1K 검증 세트 활용

What capability is learned



사람이 디자인한 여러 마스크로 복구된 이미지

- 무작위 마스크부터 주요 객체의 대부분을 제거하는 마스크까지 사용되었음
- 관찰 결과, 적당한 부분의 마스킹은 모양과 질감을 잘 복원하는 반면, 대부분을 마스킹하면 객체의 존재를 예측할 수 있으며 완전히 마스킹하면 배경 질감으로 인페인팅 됨

Prediction vs reconstruction



- 후자의 접근 방식이 더 나은 외관을 보이지만, 이는 모델 용량이 복구 영역에 낭비될 수 있음을 시사

Effects of masked patch size



고정된 마스킹 비율에서 다른 마스크된 패치 크기로 이미지를 복원

- 작은 패치 크기는 세부 사항을 더 잘 복구하지만, 전이 성능이 감소하는 경향이 있음
 - 더 작은 패치 크기로 인해 예측 작업이 쉽게 이루어질 수 있기 때문임

5. Conclusion

- 해당 논문은 표현 학습을 위해 마스크된 이미지 모델링을 활용하는 간단하면서도 효과적인 self-supervised learning 학습 프레임워크인 **SimMIM**을 제안
- 구성
 1. 적당히 큰 마스크된 패치 크기로 무작위 마스킹 전략을 사용
 2. 직접 회귀 작업을 통해 RGB 값의 원시 픽셀을 예측

3. 예측 헤드는 선형 레이어와 같이 가벼운 구조를 채택할 수 있음