MobileCLIP: Fast Image-Text Models through Multi-Modal Reinforced Training

Pavan Kumar Anasosalu Vasu, et al. CVPR2024

2024.7.24

Lab Seminar

Harim Noh

Duksung Women's University
Dept. of Computer Engineering

Introduction

- CLIP 모델은 대형 트랜스포머 기반 인코더를 사용해 메모리와 지연시간이 큼
- 모바일 기기에서의 배포 어려움이 존재함
- 본 논문의 목표는 모바일 장치에 적합한 새로운 이미지-텍스트 인코더를 설계하는 것

Multi-Modal Reinforced Training

methodology

- Mobile CLIP에서 사용한 학습 전략

1. Dataset Reinforcement

   - 합성 캡션을 통한 이미지 캡션 모델의 knowledge를 활용

2. Knowledge distillation

   - 사전 학습된 CLIP 모델의 앙상블로부터 이미지-텍스트 정렬의 knowledge distillation

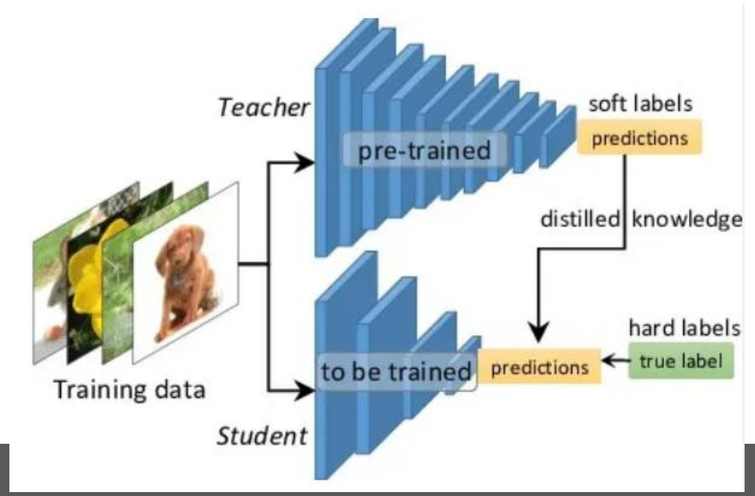
Multi-Modal Reinforced Training

methodology

- Knowledge distillation

  - 잘 학습된 큰 네트워크의 지식을 작은 네트워크에 전달하는 것

- Teacher 모델 : 복잡하고 큰 네트워크 모델이 주어진 데이터셋에서 학습
- Soft Labels : Teacher 모델은 학습 데이터에 대해 예측을 하게 됨. 이때 나온 예측 값으로 각각의 클래스에 대한 확률값
- Student 모델 학습 : Soft Labels를 사용하여 학습. Soft Labels는 Student 모델이 Teacher 모델의 예측 방식을 배울 수 있도록 도움



Dataset Reinforcement

methodology

- **Synthetic captions**
- CLIP 모델 훈련에 사용되는 이미지-텍스트 데이터셋은 대부분 웹에서 수집되며, 본질적으로 노이즈가 많음

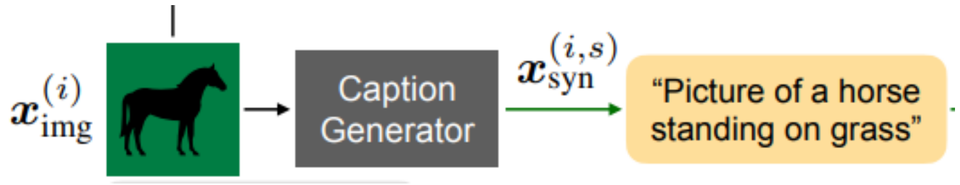
⇒ 광범위한 필터링 메커니즘을 사용해 데이터셋 품질을 개선

⇒ 노이즈가 적지만, 캡션은 충분히 설명적이지 않음

- 본 논문에서는 **CoCa(Contrastive Captioner) 모델**을 사용해 각 이미지에 대해 여러 개의 합성 캡션을 생성함

Dataset Reinforcement

methodology



- **CoCa (Contrastive Captioner)**
- 목표 : 이미지-텍스트 쌍을 처리하여 이미지에 대한 다양한 합성 캡션을 생성함. 데이터셋의 시각적 설명력 향상
- 대조 학습 : 대조 학습을 활용해 이미지와 텍스트 간의 유사성을 학습함
- 합성 캡션 : 이미지에 대해 여러 개의 합성 캡션을 생성하여, 데이터셋의 설명력을 높이고 노이즈를 줄임

Real caption:

“One of the replacement Fairfax stones.”

Synthetic captions:

“a large stone stack in the middle of a green field”

“an area with some old ruins , a tree , and grass”

Real caption:

“A four bedroom town house 20 paces from the beach - Appartement”

Synthetic captions:

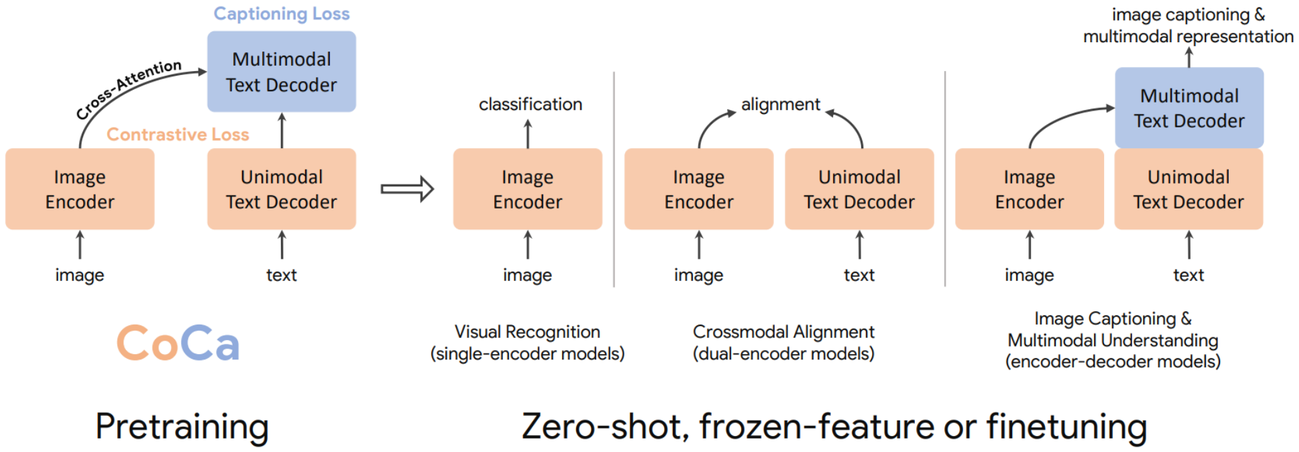
“a view of the beach with a path by it and bushes in the foreground”

“a walkway on the beach for walkers to get up and go”

Dataset Reinforcement

methodology

- **CoCa (Contrastive Captioner)**
- 트랜스포머 아키텍처를 기반으로 구축
- 이미지와 텍스트를 각각 인코딩하여 latent representation으로 변환
  - 대조 학습 태스크 : 이미지와 텍스트 간의 일치 여부를 판단하는 작업
  - 캡셔닝 태스크 : 이미지를 기반으로 자연어 캡션을 생성하는 작업



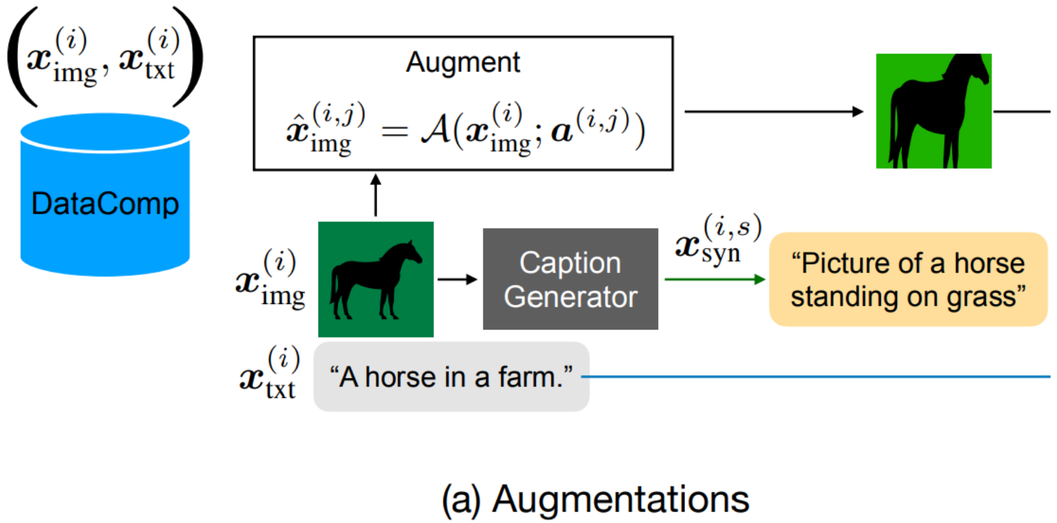
Dataset Reinforcement

methodology

- **Image augmentations**
- 각 이미지$x_{img}^{(i)}$에 augmentation 방법을 이용한 다수의 augmented image $\hat{x}_{img}^{(i,i)}$ 생성

$$\hat{\boldsymbol{x}}_{\text{img}}^{(i,j)} = \mathcal{A}(\boldsymbol{x}_{\text{img}}^{(i)}; \boldsymbol{a}^{(i,j)})$$

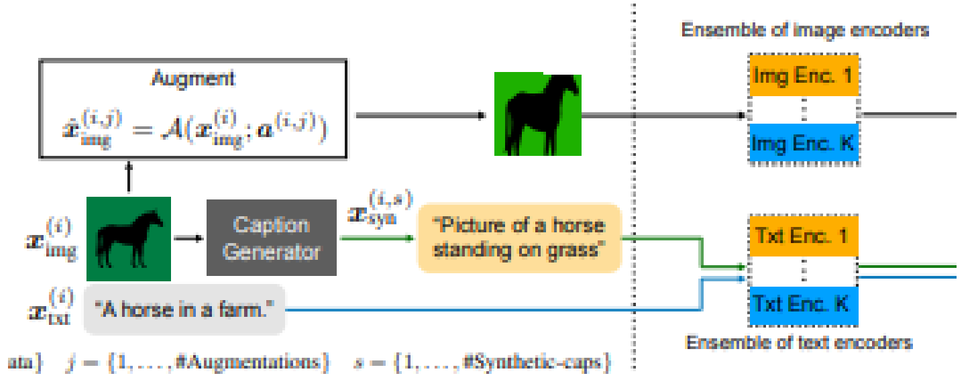
증강 파라미터



(a) Augmentations

Dataset Reinforcement

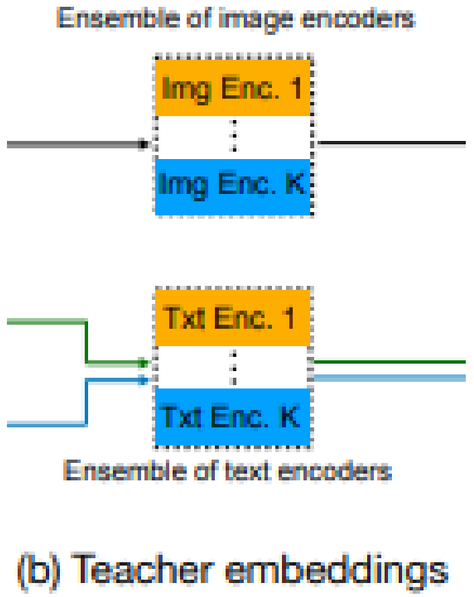
methodology



- **Ensemble teacher**
- K 개의 CLIP 모델을 이용해 Synthesized 이미지와 Synthesized 텍스트의 embedding 추출 $\Psi_{img}^{(i,s,k)}$, $\Psi_{syn}^{(i,s,k)}$
- 원본 정답 캡션에 대해서도 feature embedding 추출 $\Psi_{txt}^{(i,s,k)}$

- 즉, 기존의 좋은 성능을 가진 모델을 통해서 임베딩을 저장함

⇒ Student model이 학습할 때 teacher embedding을 가지고 유사도를 알 수 있음

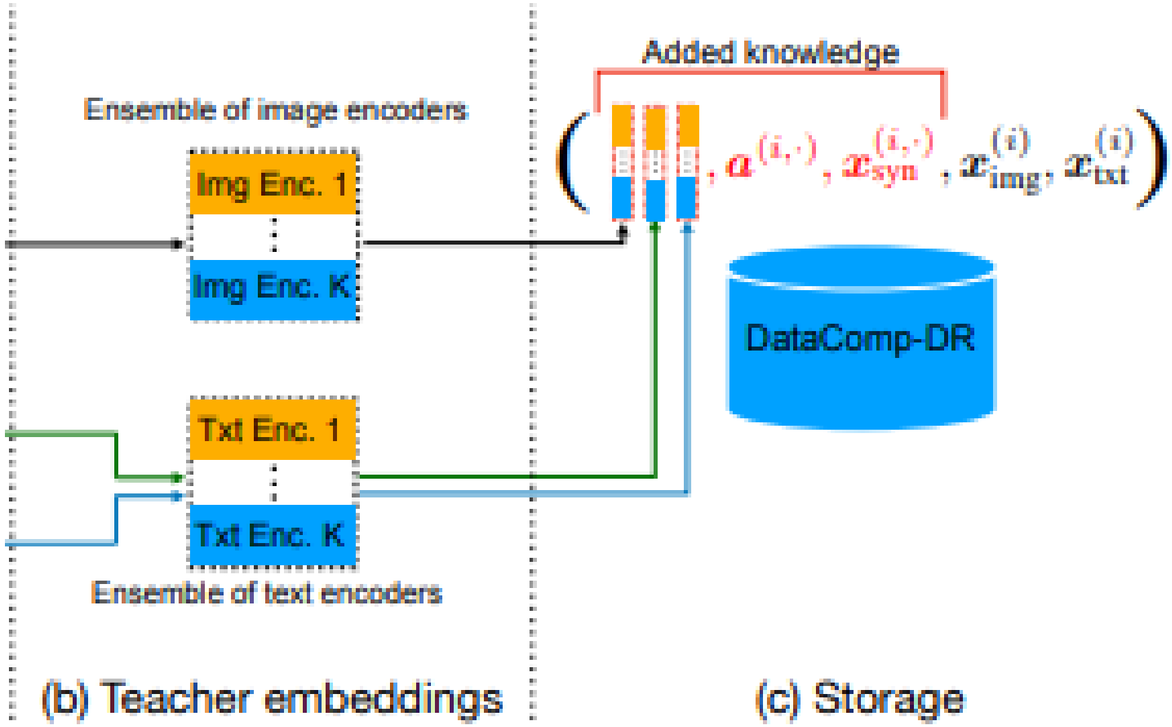


(b) Teacher embeddings

Dataset Reinforcement

methodology

- **Reinforced dataset**
- 데이터셋의 추가적인 knowledge 저장
- Image augmentation 시 사용한 $a^{(i,j)}$
- Synthetic captions $x_{syn}^{(i,s)}$
- Feature embedding $\Psi_{img}^{(i,s,k)}$, $\Psi_{syn}^{(i,s,k)}$, $\Psi_{txt}^{(i,s,k)}$
- 원본 이미지 $x_{img}^{(i)}$, 원본 캡션 $x_{txt}^{(i)}$



Training

methodology

Lambda

$$\mathcal{L}_{\text{Total}}(\mathcal{B}) = (1-\lambda)\mathcal{L}_{\text{CLIP}}(\mathcal{B}) + \lambda\mathcal{L}_{\text{Distill}}(\mathcal{B}),$$

배치 CLIP Loss Distill Loss

- **Loss function**
- 이미지-텍스트 teacher encoder에서 이미지-텍스트 쌍 간의 유사성 행렬을 training target 이미지-텍스트 인코더로 distillation

⇒ Teacher 모델이 생성한 유사성 정보를 student 모델에 전달하여 student 모델이 잘 학습하도록 돕는 과정

- CLIP Loss : CLIP에서 사용하는 contrastive 손실함수
- Distill Loss : teacher 모델을 불러와 유사도 계산
- Lambda : lambda를 통해 teacher를 얼마나 활용하는지 조절

텍스트-이미지 방향

$$\mathcal{L}_{\text{Distill}}(\mathcal{B}) = \frac{1}{2}\mathcal{L}^{\text{I2T}}_{\text{Distill}}(\mathcal{B}) + \frac{1}{2}\mathcal{L}^{\text{T2I}}_{\text{Distill}}(\mathcal{B}),$$

이미지-텍스트 방향

⇒ 이미지가 텍스트와 어떻게 연결되는지를 평가

Number of Teacher model Student Embeddings

$$\mathcal{L}^{\text{I2T}}_{\text{Distill}}(\mathcal{B}) = \frac{1}{bK}\sum_{k=1}^{K}\text{KL}(\mathcal{S}_{\tau_k}(\Psi^{(k)}_{\text{img}}, \Psi^{(k)}_{\text{txt}}) \| \mathcal{S}_{\hat{\tau}}(\Phi_{\text{img}}, \Phi_{\text{txt}})),$$

Teacher Embeddings

⇒ 교사 모델과 학생 모델의 임베딩을 바탕으로 유사성 행렬을 계산

⇒ 유사성 행렬 간의 KL(쿨백-라이블러) 발산을 계산

Training

methodology

- **쿨백-라이블러 발산**
- 두 확률 분포 간의 차이를 측정하는 방법
- 주로 한 분포가 다른 분포와 얼마나 다른지 정량화하는 데 사용

근사 분포

실제 분포의 확률 밀도 함수

$$D_{KL}(P \parallel Q) = \sum_{x} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

실제 분포

근사 분포의 확률 밀도 함수

Training

methodology

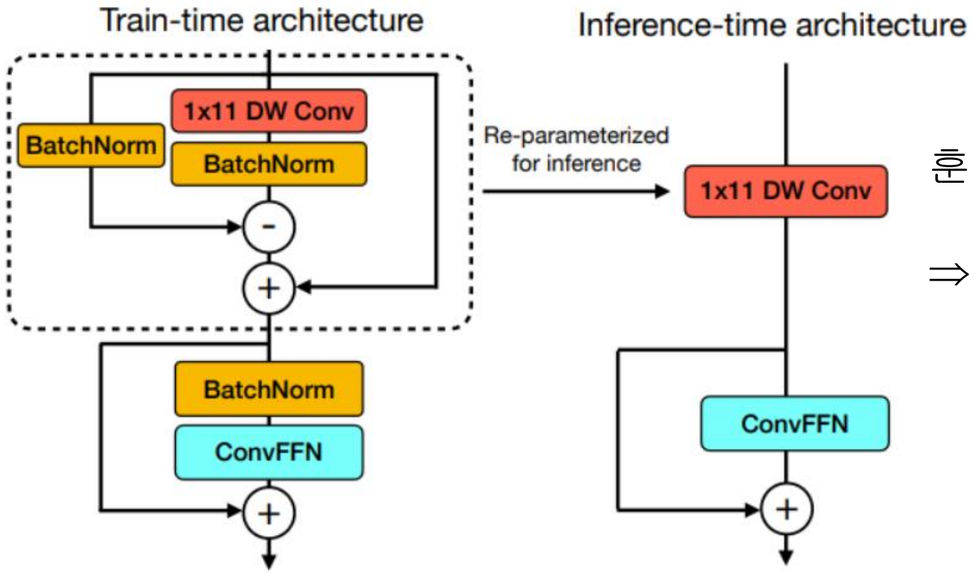
$$\sum_{\mathcal{B} \in \{B_{\text{real}}, B_{\text{syn}}\}} \mathcal{L}_{\text{Total}}(\mathcal{B}).$$

- **Efficient Training**
- 샘플의 원본 이미지, 캡션 $x_{img}^{(i)}$ $x_{txt}^{(i)}$, Synthesized caption $x_{syn}^{(i,s)}$ 랜덤하게 load
- 저장된 augmentation 파라미터 $a^{(i,j)}$ 랜덤하게 load 후 $\hat{x}_{img}^{(i,i)}$ 생성
- Teacher 모델의 임베딩 load $\Psi_{img}^{(i,s,k)}$, $\Psi_{syn}^{(i,s,k)}$, $\Psi_{txt}^{(i,s,k)}$
- 2개의 데이터 배치 조합 생성 : (기존 자막, 증강 이미지) / (증강 자막, 증강 이미지)
- $L_{Total}(B)$ 계산
- Teacher 임베딩이 데이터셋의 일부로 저장되어 있어 teacher 모델에 추가적인 계산 없이 총 손실 계산 가능

Training

architecture

- 기존 CLIP 모델의 Text encoder는 Transformer의 방식
- Mobile CLIP : 1D Conv와 Self Attention 레이어를 혼합해서 사용하는 하이브리드 텍스트 인코더 사용
- **Text-RepMixer(reparameterizable convolutional token mixing)** 빠르게 추론하기 위해 학습과 추론 아키텍처를 분리



훈련 중 사용된 구조를 1X11 DW Conv 구조로 재파라미터화

⇒Skip-connection의 효과를 유지한 채, 연산량과 메모리 감소의 이점을 얻음

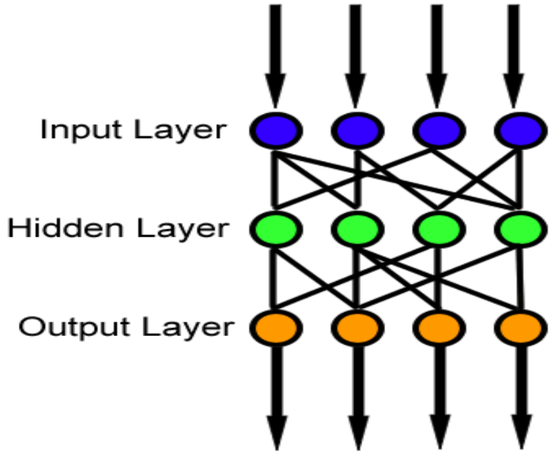
$$Y = \mathrm{BN}(\sigma(\mathrm{DWConv}(\mathrm{X}))) + \mathrm{X}$$

$$Y = \mathrm{DWConv}(\mathrm{BN}(\mathrm{X})) + \mathrm{X}$$

Training

architecture

- ConvFFN
- Transformer 모델의 기본 구조인 피드포워드 신경망(FFN) 레이어에 컨볼루션(Conv) 레이어를 결합한 형태
- Feed-Forward Network (피드포워드 네트워크)
  - 여러 층의 완전 연결된 신경망으로 구성
  - 입력 데이터를 선형 변환한 후 비선형 활성화 함수를 통해 출력으로 변환
- 컨볼루션 레이어에서 생성된 특징 맵은 FFN으로 전달되어, 더 높은 수준의 복잡한 패턴을 학습



Training

architecture

- FastViT 아키텍처를 차용
  - FFN 블록에 대해 MLP(Multi-Layer Perceptron) expansion ratio는 4.0으로 사용
  - Expansion ratio : MLP의 중간 레이어에서의 뉴런 수가 입력 레이어에 비해 얼마나 확장되는지 나타냄
  - FFN 블록의 선형 레이어에서 중복성 초래함
- expansion ratio을 단순히 3.0으로 낮추고 아키텍처의 depth를 증가
  - 이미지 인코더의 동일한 parameter 수를 유지

Experiments

Variant	$\{C_1, C_2, C_3, C_4\}$	$\{L_1, L_2, L_3, L_4\}$
MCi0	$\{64, 128, 256, 512\}$	$\{2, 6, 10, 2\}$
MCi1	$\{64, 128, 256, 512\}$	$\{4, 12, 20, 4\}$
MCi2	$\{80, 160, 320, 640\}$	$\{4, 12, 24, 4\}$

(a) Configurations of MCi.

Name	Dataset	Seen Samples	Image Encoder	Text Encoder	Params (M) (img+txt)	Latency (ms) (img+txt)	Zero-shot CLS		Flickr30k Ret.		COCO Ret.		Avg. Perf. on 38
							IN-val	IN-shift	T→I	I→T	T→I	I→T	
Ensemble Teacher	DataComp-1B [18] OpenAI-400M [46]	-	ViT-L/14 ViT-L/14	Base Base	(-)	(-)	80.1	69.6	74.5	92.3	46.7	66.5	67.3
TinyCLIP-RN19M [67]	LAION-400M [50]	15.2B	ResNet-19M	Custom	18.6 + 44.8	1.9 + 1.9	56.3	43.6	58.0	75.4	30.9	47.8	48.3
TinyCLIP-RN30M [67]	LAION-400M [50]	15.2B	ResNet-30M	Custom	29.6 + 54.2	2.6 + 2.6	59.1	45.7	61.5	80.1	33.8	51.6	50.2
TinyCLIP-40M/32 [67]	LAION-400M [50]	15.2B	ViT-40M/32	Custom	39.7 + 44.5	3.0 + 1.9	59.8	46.5	59.1	76.1	33.5	48.7	51.2
MobileCLIP-S0	DataCompDR-1B	13B	MCi0	MCt	11.4 + 42.4	1.5 + 1.6	**67.8**	**55.1**	**67.7**	**85.9**	**40.4**	**58.7**	**58.1**
OpenAI-RN101	OpenAI-400M [46]	13B	ResNet-101	Base	56.3 + 63.4	4.3 + 3.3	62.3	48.5	58.0	79.0	30.7	49.8	50.3
OpenAI-B/32	OpenAI-400M [46]	13B					63.3	48.5	58.8	78.9	30.4	50.1	52.5
LAION-B/32	LAION-2B [51]	32B	ViT-B/32	Base	86.2 + 63.4	5.9 + 3.3	65.7	51.9	66.4	84.4	39.1	56.2	54.8
DataComp-B/32	DataComp-1B [18]	13B					69.2	55.2	61.1	79.0	37.1	53.5	58.0
DataComp-B/32-256	DataComp-1B [18]	34B	ViT-B/32-256	Base	86.2 + 63.4	6.2 + 3.3	72.8	58.7	64.9	84.8	39.9	57.9	60.9
MobileCLIP-S2	DataCompDR-1B	13B	MCi2	Base	35.7 + 63.4	3.6 + 3.3	**74.4**	**63.1**	**73.4**	**90.3**	**45.4**	**63.4**	**63.7**

- MobileCLIP-S0는 TinyCLIP과 같은 최근 작업보다 훨씬 우수한 성능을 보임
- ViT-B/32 모델과 유사한 성능을 보임
  - MobileCLIP-S0는 2.8배 작고 3배 빠름