Momentum Contrast for Unsupervised Visual Representation Learning

Kaiming He   Haoqi Fan   Yuxin Wu   Saining Xie   Ross Girshick

Facebook AI Research (FAIR)

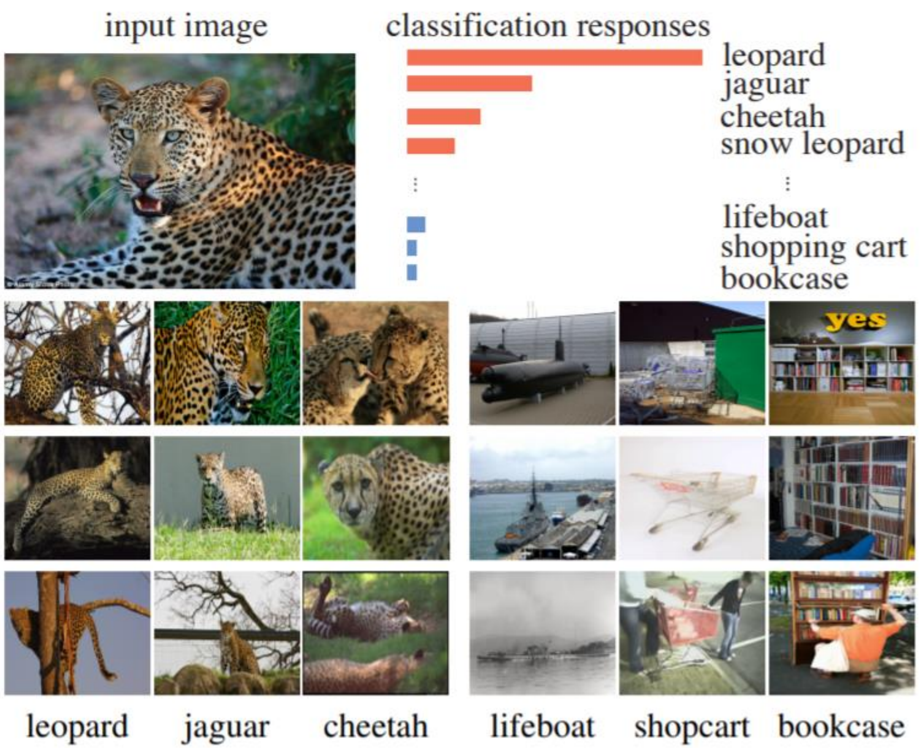
Presented by Yonggyu Kim

2020.03.19

Prerequisite

Unsupervised Feature Learning via Non-Parametric Instance Discrimination

Motivation



- A classifier can automatically <u>dis cover apparent similarity</u> among semantic categories, <u>without se mantic annotation.</u>

- Can we learn a good feature rep resentation that captures appare nt <u>similarity among instances, in stead of classes</u>, by merely askin g the feature to be discriminativ e of individual instances?

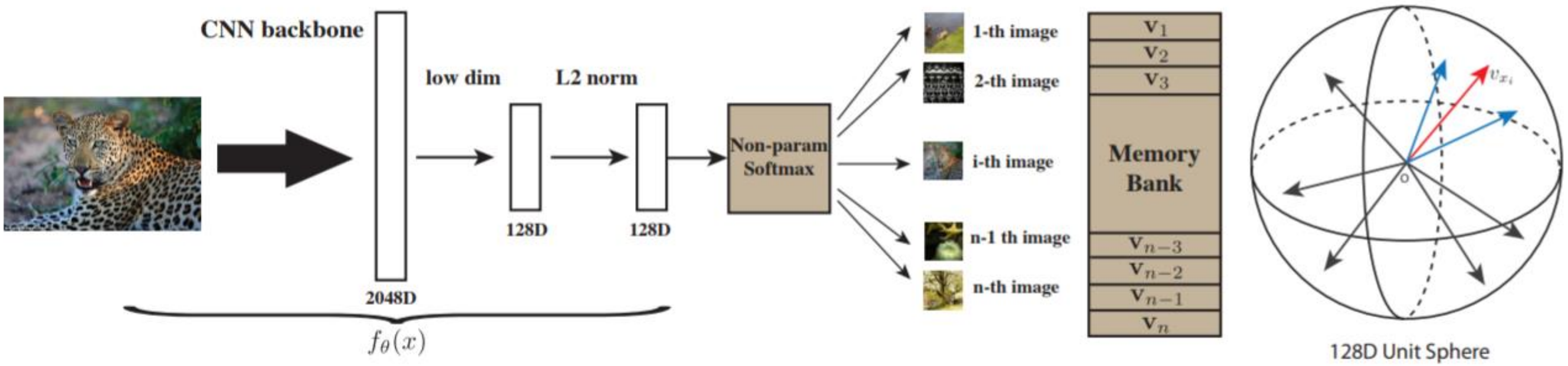
- Unsupervised learning setting에서 feature를 학습하고 transfer 함으로써 downstream task의 성능을 높이고자 하는 게 목표

Prerequisite

Unsupervised Feature Learning via Non-Parametric Instance Discrimination

Method

Memory Bank = instance dictionary



$$P(i|\mathbf{v}) = \frac{\exp\left(\mathbf{w}_i^T \mathbf{v}\right)}{\sum_{j=1}^{n} \exp\left(\mathbf{w}_j^T \mathbf{v}\right)} \qquad P(i|\mathbf{v}) = \frac{\exp\left(\mathbf{v}_i^T \mathbf{v}/\tau\right)}{\sum_{j=1}^{n} \exp\left(\mathbf{v}_j^T \mathbf{v}/\tau\right)}$$

<u>Computational cost</u>

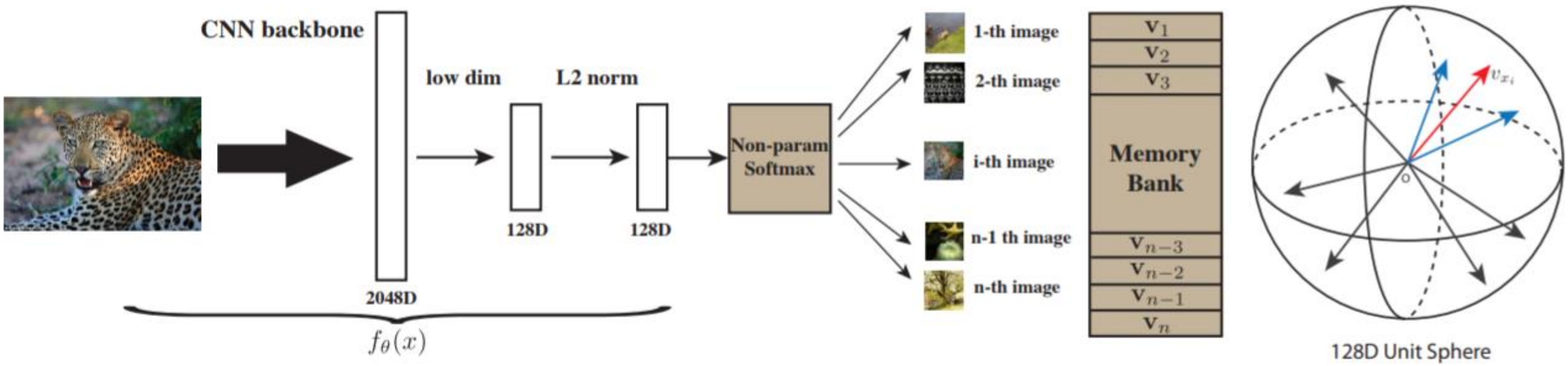
Negative contrastive estimation(NCE) performs binary classification task that is to <u>discriminate between data samples and noise samples</u>.

Prerequisite

Unsupervised Feature Learning via Non-Parametric Instance Discrimination

NCE

Memory Bank : instances dictionary



$$P(i|\mathbf{v}) = \frac{\exp(\mathbf{v}^T \mathbf{f}_i / \tau)}{Z_i}$$

$$Z_i = \sum_{j=1}^{n} \exp\left(\mathbf{v}_j^T \mathbf{f}_i / \tau\right)$$

Monte Carlo approximation:

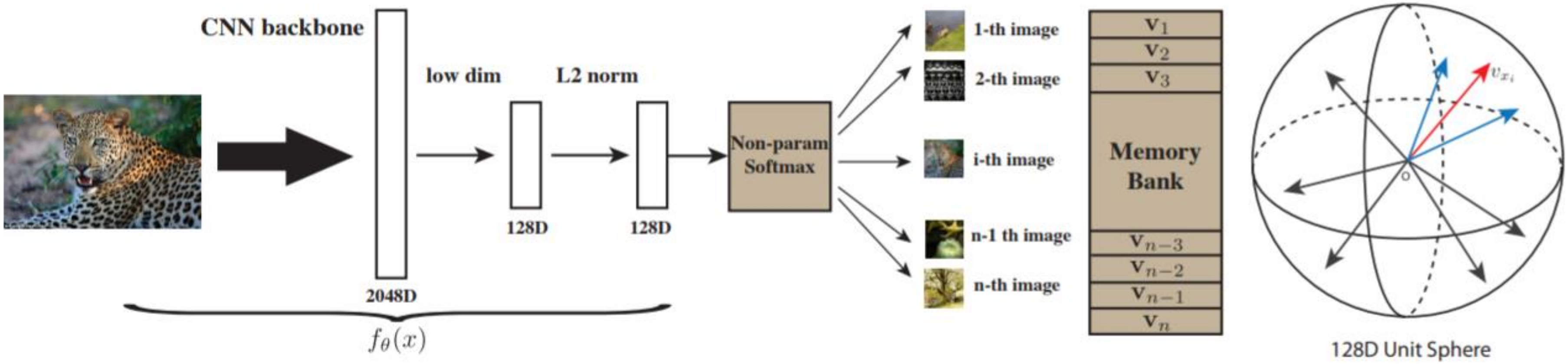
$$Z \simeq Z_i \simeq n E_j \left[\exp(\mathbf{v}_j^T \mathbf{f}_i / \tau)\right] = \frac{n}{m} \sum_{k=1}^{m} \exp(\mathbf{v}_{j_k}^T \mathbf{f}_i / \tau)$$

Prerequisite

Unsupervised Feature Learning via Non-Parametric Instance Discrimination

NCE

Memory Bank : instances dictionary



$$P(i|\mathbf{v}) = \frac{\exp(\mathbf{v}^T \mathbf{f}_i/\tau)}{Z_i}$$

Noise distribution : $P_n = 1/n$

$$h(i, \mathbf{v}) := P(D = 1|i, \mathbf{v}) = \frac{P(i|\mathbf{v})}{P(i|\mathbf{v}) + mP_n(i)}$$

Induced by posterior probability

$$J_{NCE}(\boldsymbol{\theta}) = -E_{P_d}\left[\log h(i, \mathbf{v})\right] \\ -m \cdot E_{P_n}\left[\log(1 - h(i, \mathbf{v}'))\right]$$

<u>**Consistency problem**</u>

Prerequisite

Unsupervised learning vs Self-supervised learning

In unsupervised learning, you try to find some 'structure' (clusters, densities, latent representation) in the entire while using their original form.

In self-supervised learning, you try to learn the 'dynamics' of the data at its raw level. Popular self-supervised learning, i.e image colorization uses only the gray-scale (part of the data is withheld) version and try to predict its colors.

Motivation

- 앞서 설명한 논문 : Unsupervised 방식으로 image를 embedding vector로 encoding 하도록 학습
- NLP에서 Unsupervised 방식으로 mask 처리된 단어를 embedding vector로 encoding 하도록 학습
- 왜 vision은 아직 supervised pre-training 을 많이 쓸까?

NLP vs Computer vision

- The reason may stem from differences in their respective signal spaces.

  Language tasks have discrete signal spaces(words, sub-word units, etc.) for building tokenized dictionaries.

  The raw signal of computer vision is in continuous, high-dimensional space unlike words.

Motivation

- The authors hypothesize that it is desirable to build dictionaries that are :
  1. Large
  2. Consistent

- A main purpose of unsupervised learning is to pre-train representation that can be transferred to downstream tasks by fine-tuning.
- They show that in 7 downstream tasks related to detection or segment ation.
- MoCo unsupervised pre-training can surpass its ImageNet supervised c ounter part, in some cases by nontrivial margins.

Method

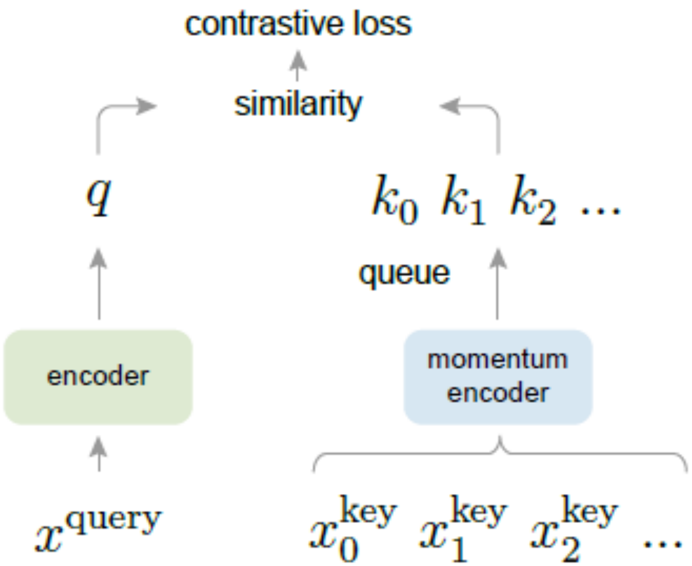
Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature
```

Initialize queue / K=4096, C=128

```
f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.



InfoNCE

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

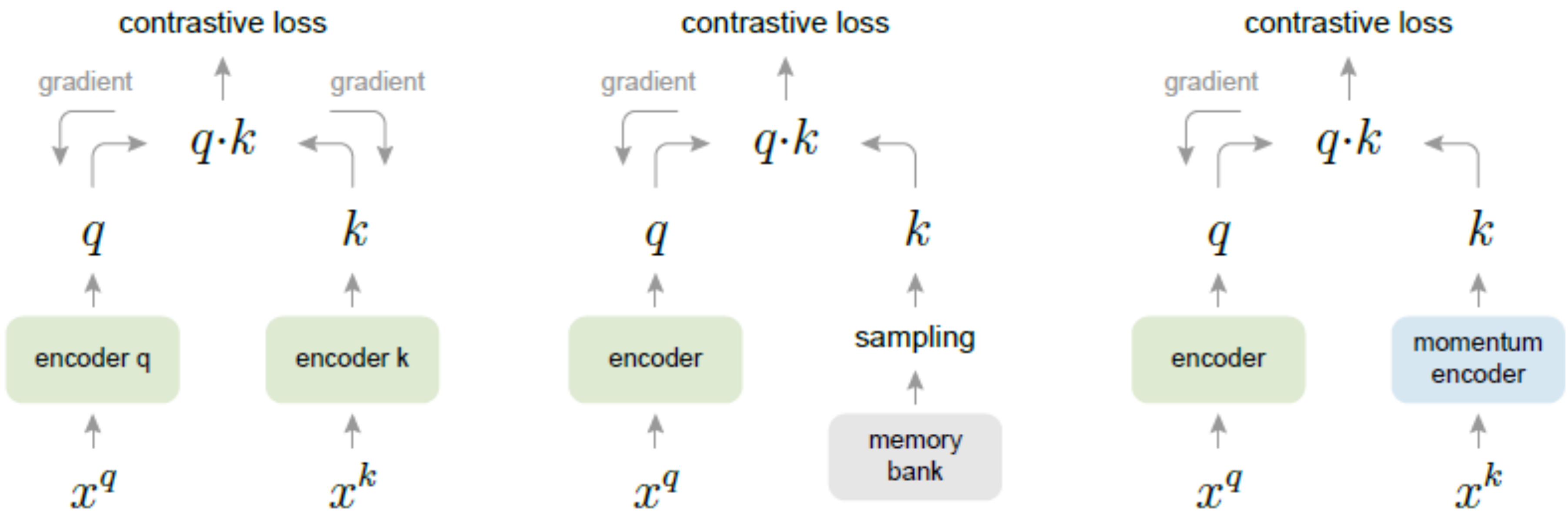
Momentum update

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}$$

aug : color jittering, horizontal flip, grayscale

Method

Comparison with existing method

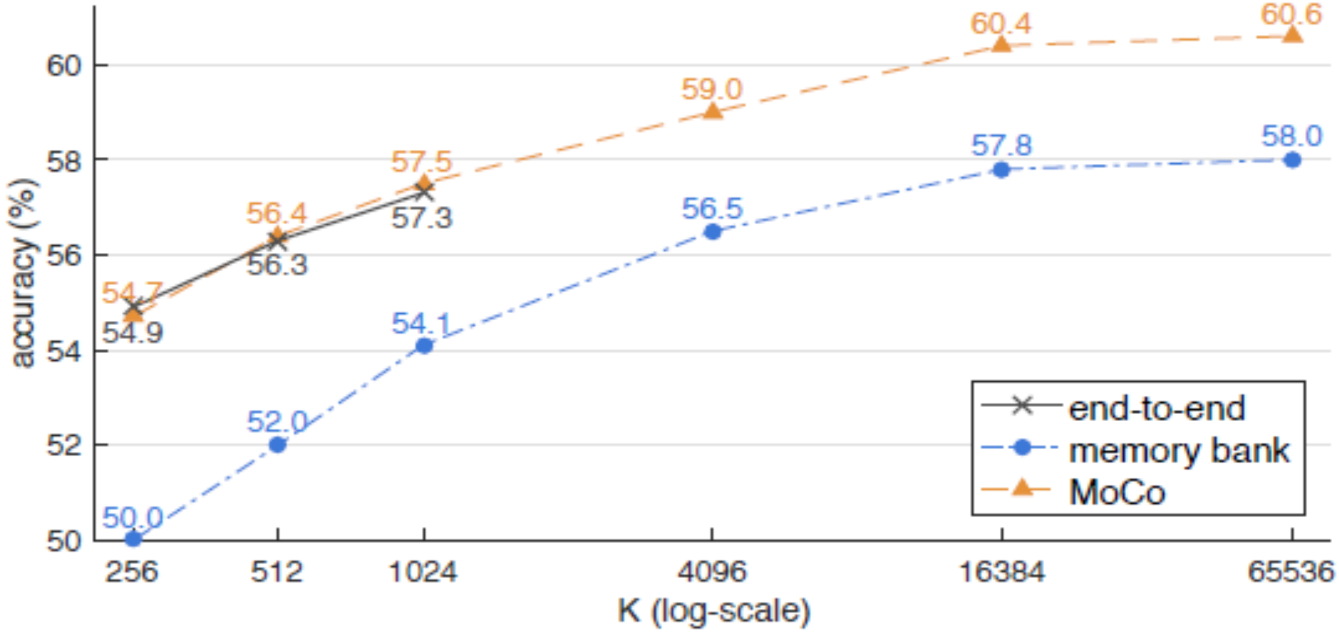


(a) end-to-end (b) memory bank (c) MoCo

Experiments

Ablation: contrastive loss mechanisms.

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}$$

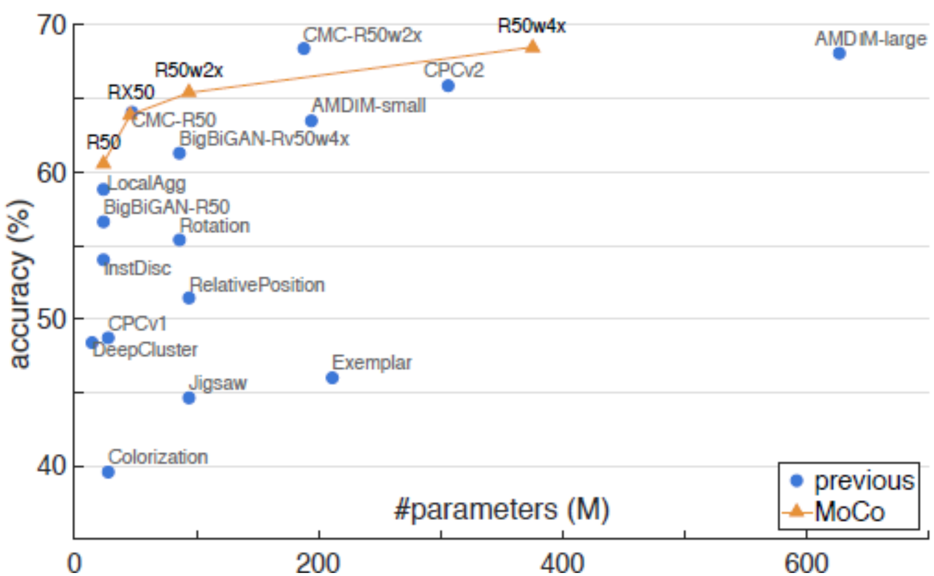


Ablation: momentum. The table below shows ResNet-50 accuracy with different MoCo momentum values (m in Eqn.(2)) used in pre-training ($K = 4096$ here) :

momentum m	0	0.9	0.99	0.999	0.9999
accuracy (%)	*fail*	55.2	57.8	59.0	58.9

Experiments

Comparison with previous results under the linear classification on ImageNet.



method	architecture	#params (M)	accuracy (%)
Exemplar [15]	R50w3×	211	46.0 [36]
RelativePosition [11]	R50w2×	94	51.4 [36]
Jigsaw [43]	R50w2×	94	44.6 [36]
Rotation [17]	Rv50w4×	86	55.4 [36]
Colorization [62]	R101*	28	39.6 [12]
DeepCluster [3]	VGG [51]	15	48.4 [4]
BigBiGAN [14]	R50	24	56.6
	Rv50w4×	86	61.3
methods based on contrastive learning follow:			
InstDisc [59]	R50	24	54.0
LocalAgg [64]	R50	24	58.8
CPC v1 [44]	R101*	28	48.7
CPC v2 [33]	$R170^*_{wider}$	303	65.9
CMC [54]	$R50_{L+ab}$	47	$64.1^\dagger$
	$R50w2\times_{L+ab}$	188	$68.4^\dagger$
AMDIM [2]	$AMDIM_{small}$	194	$63.5^\dagger$
	$AMDIM_{large}$	626	$68.1^\dagger$
MoCo	R50	24	60.6
	RX50	46	63.9
	R50w2×	94	65.4
	R50w4×	375	**68.6**

Experiments

PASCAL VOC Object Detection

Ablation : backbones

pre-train	AP_{50}	AP	AP_{75}
random init.	58.0	32.8	32.5
super. IN-1M	81.5	53.6	58.9
MoCo IN-1M	81.1 (−0.4)	53.8 (+0.2)	58.6 (−0.3)
MoCo IG-1B	81.6 (+0.1)	54.8 (+1.2)	60.3 (+1.4)

(a) Faster R-CNN, R50-**dilated-C5**

pre-train	AP_{50}	AP	AP_{75}
random init.	52.5	28.1	26.2
super. IN-1M	80.8	52.0	56.5
MoCo IN-1M	81.4 (+0.6)	55.2 (+3.2)	61.2 (+4.7)
MoCo IG-1B	82.1 (+1.3)	56.2 (+4.2)	62.3 (+5.8)

(b) Faster R-CNN, R50-C4

Ablation : contrastive loss mechanisms

	R50-dilated-C5			R50-C4		
pre-train	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}
end-to-end	77.8	50.1	53.8	79.7	53.0	57.9
memory bank	79.6	51.9	56.3	80.3	53.9	58.9
MoCo	**81.1**	**53.8**	**58.6**	**81.4**	**55.2**	**61.2**

Ablation : Comparison with previous results

	AP_{50}					AP	AP_{75}	
pre-train	RelPos, by [12]	Multi-task [12]	Jigsaw, by [24]	LocalAgg [64]	**MoCo**	**MoCo**	Multi-task [12]	**MoCo**
super. IN-1M	74.2	74.2	70.5	74.6	74.4	42.4	44.3	42.7
unsup. IN-1M	66.8 (−7.4)	70.5 (−3.7)	61.4 (−9.1)	69.1 (−5.5)	74.9 (+0.5)	46.6 (+4.2)	43.9 (−0.4)	50.1 (+7.4)
unsup. IN-14M	-	-	69.2 (−1.3)	-	75.2 (+0.8)	46.9 (+4.5)	-	50.2 (+7.5)
unsup. YFCC-100M	-	-	66.6 (−3.9)	-	74.7 (+0.3)	45.9 (+3.5)	-	49.0 (+6.3)
unsup. IG-1B	-	-	-	-	75.6 (+1.2)	47.6 (+5.2)	-	51.7 (+9.0)

Experiments

COCO Object Detection and Segmentation

pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	31.0	49.5	33.2	28.5	46.8	30.4
super. IN-1M	38.9	59.6	42.7	35.4	56.5	38.1
MoCo IN-1M	38.5 (−0.4)	58.9 (−0.7)	42.0 (−0.7)	35.1 (−0.3)	55.9 (−0.6)	37.7 (−0.4)
MoCo IG-1B	38.9 (0.0)	59.4 (−0.2)	42.3 (−0.4)	35.4 (0.0)	56.5 (0.0)	37.9 (−0.2)

(a) Mask R-CNN, R50-**FPN**, **1**× schedule

pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	36.7	56.7	40.0	33.7	53.8	35.9
super. IN-1M	40.6	61.3	44.4	36.8	58.1	39.5
MoCo IN-1M	40.8 (+0.2)	61.6 (+0.3)	44.7 (+0.3)	36.9 (+0.1)	58.4 (+0.3)	39.7 (+0.2)
MoCo IG-1B	41.1 (+0.5)	61.8 (+0.5)	45.1 (+0.7)	37.4 (+0.6)	59.1 (+1.0)	40.2 (+0.7)

(b) Mask R-CNN, R50-**FPN**, **2**× schedule

pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	26.4	44.0	27.8	29.3	46.9	30.8
super. IN-1M	38.2	58.2	41.2	33.3	54.7	35.2
MoCo IN-1M	38.5 (+0.3)	58.3 (+0.1)	41.6 (+0.4)	33.6 (+0.3)	54.8 (+0.1)	35.6 (+0.4)
MoCo IG-1B	39.1 (+0.9)	58.7 (+0.5)	42.2 (+1.0)	34.1 (+0.8)	55.4 (+0.7)	36.4 (+1.2)

(c) Mask R-CNN, R50-**C4**, **1**× schedule

pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	35.6	54.6	38.2	31.4	51.5	33.5
super. IN-1M	40.0	59.9	43.1	34.7	56.5	36.9
MoCo IN-1M	40.7 (+0.7)	60.5 (+0.6)	44.1 (+1.0)	35.4 (+0.7)	57.3 (+0.8)	37.6 (+0.7)
MoCo IG-1B	41.1 (+1.1)	60.7 (+0.8)	44.8 (+1.7)	35.6 (+0.9)	57.4 (+0.9)	38.1 (+1.2)

(d) Mask R-CNN, R50-**C4**, **2**× schedule

Experiments

More Downstream Tasks

pre-train	COCO keypoint detection AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
random init.	65.9	86.5	71.7
super. IN-1M	65.8	86.9	71.9
MoCo IN-1M	66.8 (+1.0)	87.4 (+0.5)	72.5 (+0.6)
MoCo IG-1B	66.9 (+1.1)	87.8 (+0.9)	73.0 (+1.1)

pre-train	COCO dense pose estimation AP^{dp}	AP^{dp}_{50}	AP^{dp}_{75}
random init.	39.4	78.5	35.1
super. IN-1M	48.3	85.6	50.6
MoCo IN-1M	50.1 (+1.8)	86.8 (+1.2)	53.9 (+3.3)
MoCo IG-1B	50.6 (+2.3)	87.0 (+1.4)	54.3 (+3.7)

pre-train	LVIS v0.5 instance segmentation AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	22.5	34.8	23.8
super. IN-1M†	24.4	37.8	25.8
MoCo IN-1M	24.1 (−0.3)	37.4 (−0.4)	25.5 (−0.3)
MoCo IG-1B	24.9 (+0.5)	38.2 (+0.4)	26.4 (+0.6)

pre-train	Cityscapes instance seg.		Semantic seg. (mIoU)	
	AP^{mk}	AP^{mk}_{50}	Cityscapes	VOC
random init.	25.4	51.1	65.3	39.5
super. IN-1M	32.9	59.6	74.6	74.4
MoCo IN-1M	32.3 (−0.6)	59.3 (−0.3)	75.3 (+0.7)	72.5 (−1.9)
MoCo IG-1B	32.9 (0.0)	60.3 (+0.7)	75.5 (+0.9)	73.6 (−0.8)