Exploring Simple Siamese Representation Learning

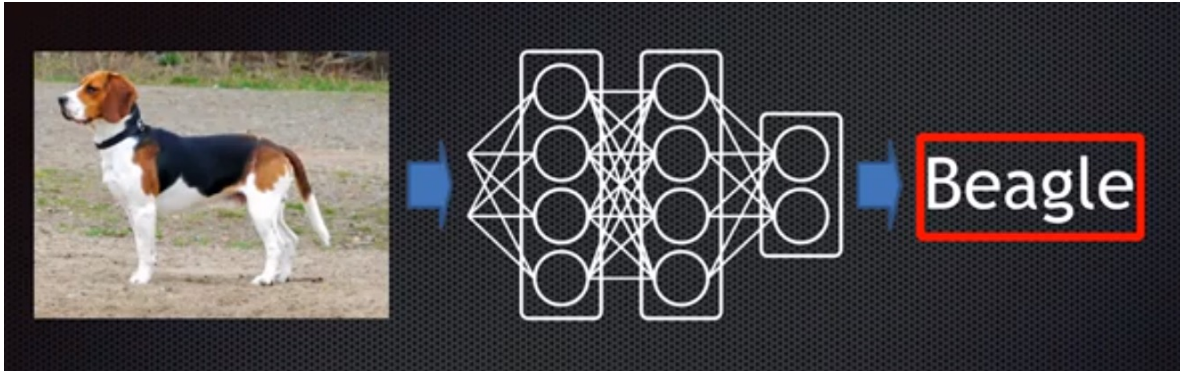
Presenter: Hanwen Jiang

09/16/2021

Motivation

Deep learning has achieved great success in many areas

- Under the supervised learning paradigm



[Image credit to Abhinav Gupta]

Motivation

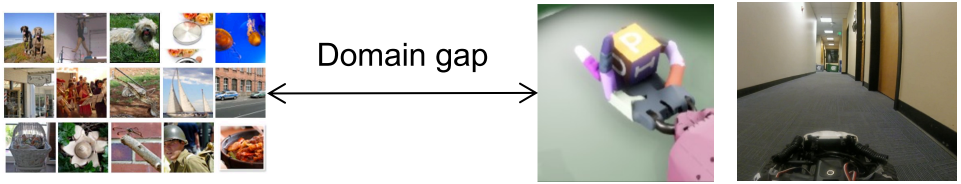
The problems of supervised learning

- Requires expensive manual labels
- Size of datasets are constrained, and learning cannot be scalable

14M images, 5 years

1B images, everyday

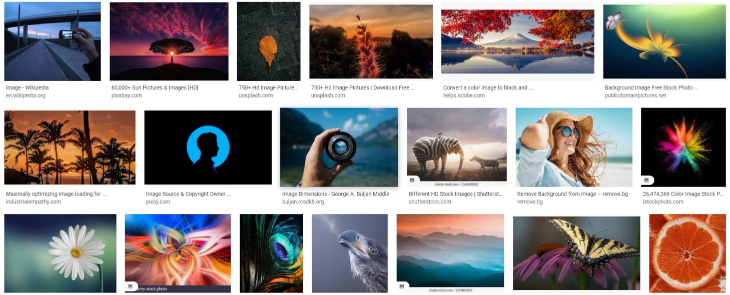
- Learning is passive and even biased, learned feature representations may not be generalizable



Motivation

What if we can learn representation without labels?

- Unconstrained and unlimited datasets



Unlabeled web-images

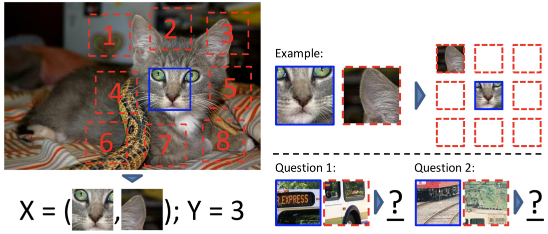
Observations during exploration

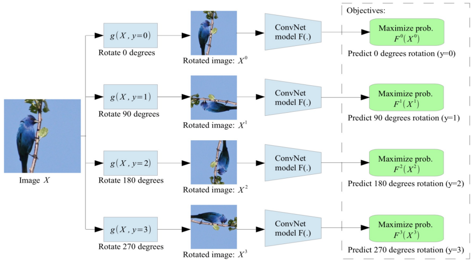
- More generalizable features (especially good for downstream robot learning tasks)
- Make it possible for active learning through perception-action loop

Self-supervised representation learning

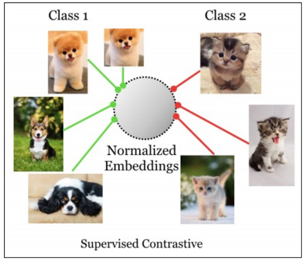
How to get self-supervision signals?

- Learning via pre-text tasks: supervision comes from structure of the task





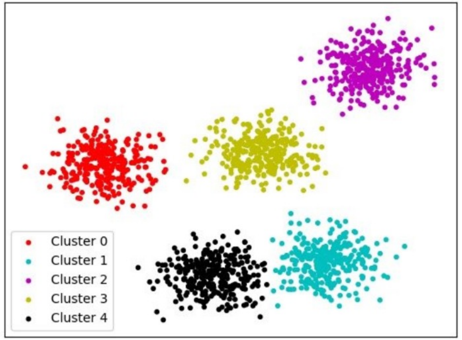
- Contrastive learning: supervision comes from structure of the data



Contrastive learning

What we learned from supervised learning?

The features of different classes are in clusters

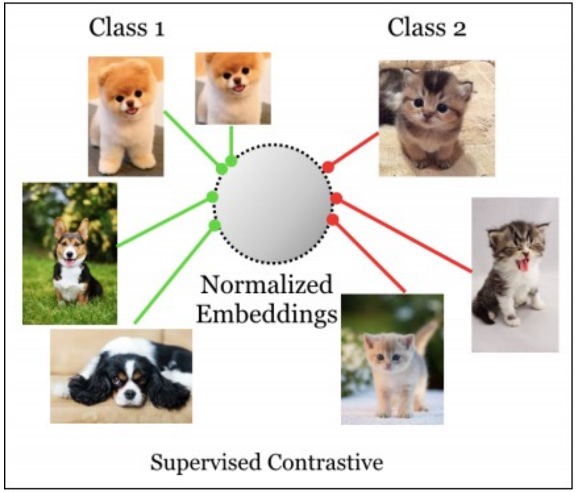


- Inter-class variance

  E.g. Features of ‘cats’ and ‘dogs’ should be far away

- Intra-class similarity

  E.g. Different dog instances have similar features



Contrastive learning

Supervision comes from structure of the data

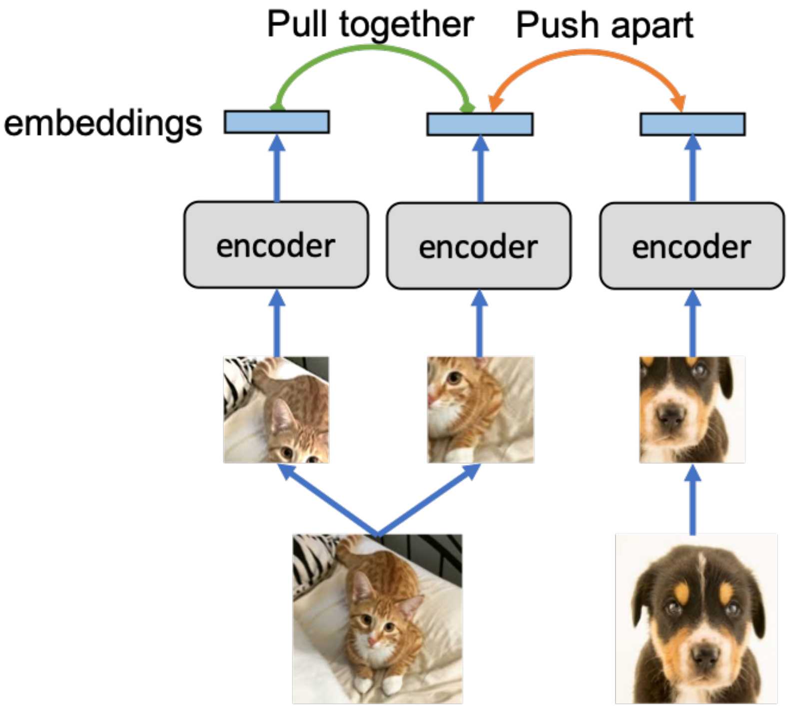
- Constructing positive and negative pairs via data augmentation

- Inter-class variance (Uniformity)

  Learned from pushing negative pairs far away

- Intra-class similarity (Alignment)

  Learned from pulling positive pairs together



Contrastive learning

Problem formulation

Target: $d(f(x), f(x^+)) \ll d(f(x), f(x^-))$

or

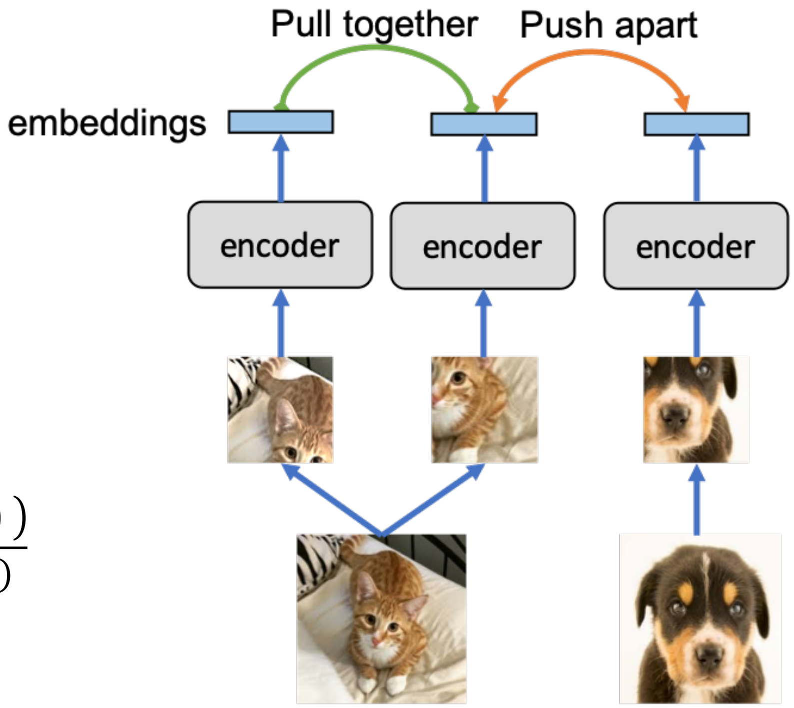
$s(f(x), f(x^+)) \gg s(f(x), f(x^-))$

Learn with infoNCE loss

$$z = f_\theta(x)$$

$$s(z_i, z_j) = \frac{z_i^T z_j}{\left\|z_i\right\|_2 \left\|z_j\right\|_2}$$

$$L = -\frac{log\left(\exp\left(s(z, z^+)\right)\right)}{\Sigma_{j=0}^{N} \exp(s(z, z_j^-))}$$



Contrastive learning

Biggest problem of CL: Model collapse to a sub-optimal

I. e. All samples are encoded to a same representation

$$L = -\frac{log\left(\exp\left(s(z, z^{+})\right)\right)}{\Sigma_{j=0}^{N}\exp(s(z, z_j^{-}))}$$

Solutions:

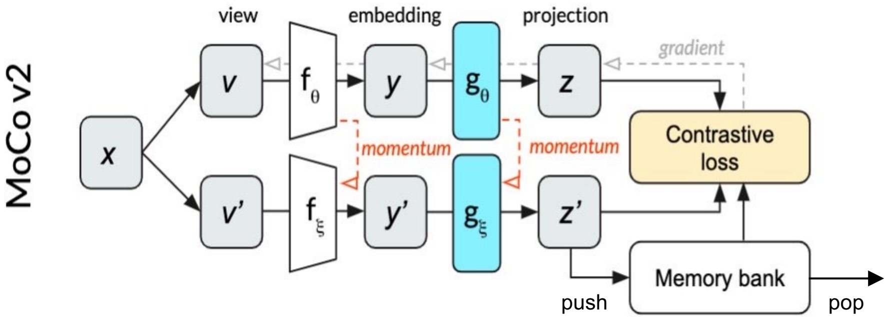
- Adding more 'contrastive' (negative pairs)
- Learning without any negative pairs

Related works

Adding more 'contrastive': use larger number of negative samples

MoCo: Use memory bank (A queue contains tons of negative sample features)

Contrast with each negative sample in the bank



$$f_\epsilon = m * f_\theta + (1 - m) * f_\epsilon$$

Momentum encoder is designed for a continuous update of memory bank

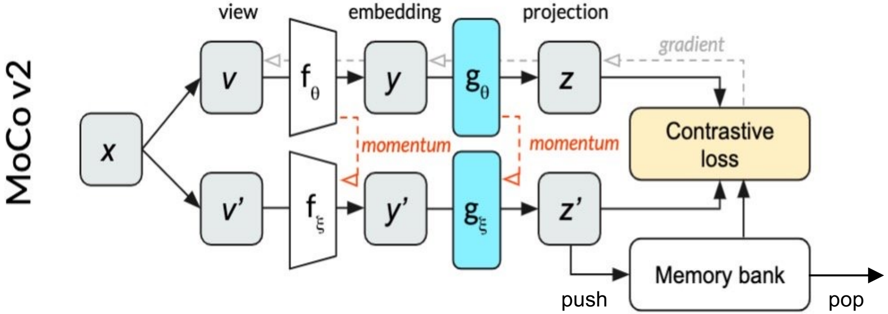
- Embedding space of negative samples in bank are changing continuously

Related works

Adding more 'contrastive': use larger number of negative samples

MoCo: Use memory bank (A queue contains tons of negative sample features)

Contrast with each negative sample in the bank



$$f_\epsilon = m * f_\theta + (1 - m) * f_\epsilon$$

Stop gradient: The compute graph of previous negative samples in the bank is lost

Related works

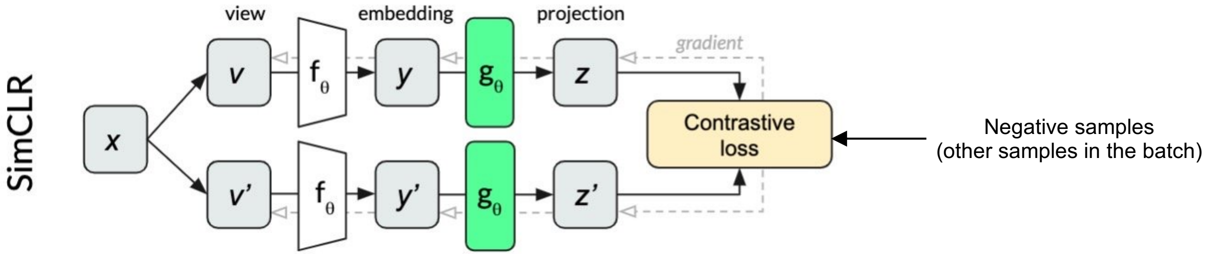
Adding more 'contrastive': use larger number of negative samples

SimCLR: Use very large batchsize on TPU, contrastive with each other in the batch

A brute-force method, but have contributions on:

(1) exploring the data augmentations

(2) using Projector to get rid of augmentation-related information

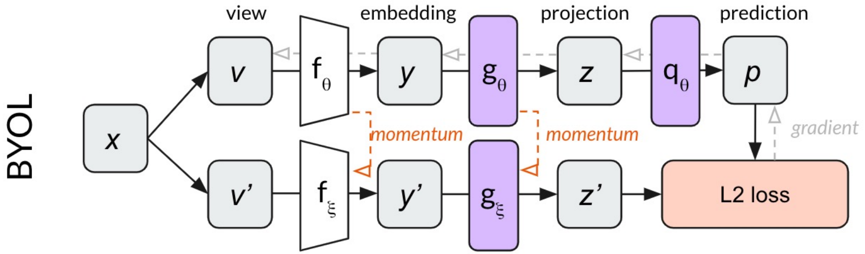


Related works

Learning without any negative samples

BYOL: Reason for collapse: enforcing the similarity between z and z^+ with infoNCE $L = -\frac{log\left(\exp\left(s(z,z^+)\right)\right)}{\Sigma_{j=0}^{N}\exp(s(z,z_j^-))}$

Solution: Add a predictor to predict the z^+ (target feature) from $\mathrm{p} = \mathrm{q}_\theta(\mathrm{z})$



Reason for **stop-gradient** of the momentum encoder is different from MoCo!

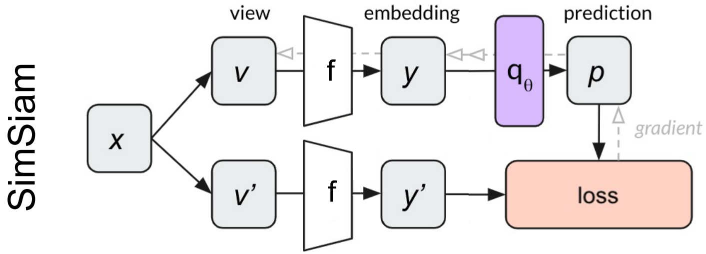
- Without negative samples, BYOL doesn't suffer from the gradient lost problem
- It is a special design in BYOL

SimSiam

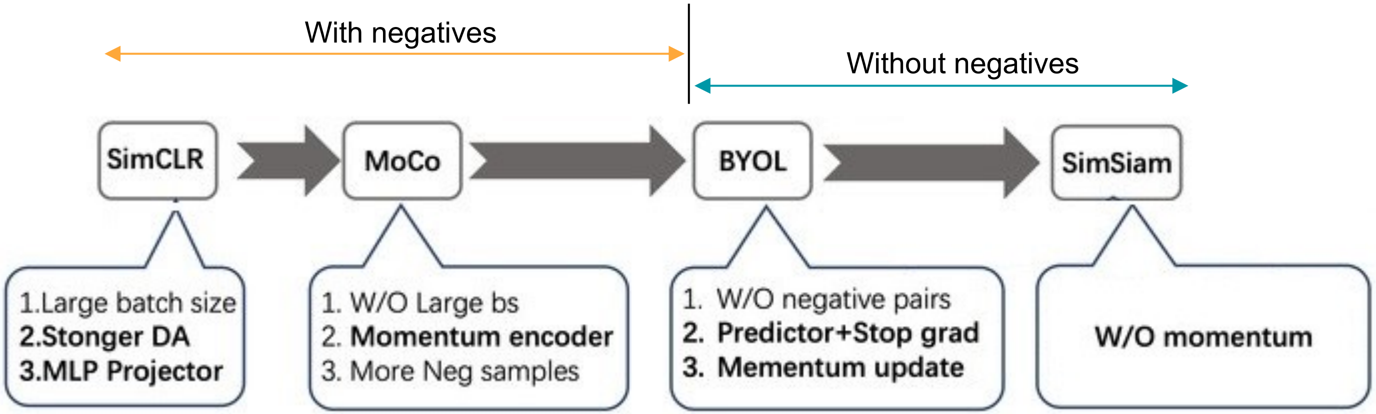
Learning without any negative samples

SimSiam ≈ BYOL without Momentum encoder

- The authors found the stop gradient is the key of preventing collapse
- Also, add symmetric learning



Comparison



With negatives
Without negatives
SimCLR
MoCo
BYOL
SimSiam
1.Large batch size
2.Stonger DA
3.MLP Projector
1. W/O Large bs
2. Momentum encoder
3. More Neg samples
1. W/O negative pairs
2. Predictor+Stop grad
3. Mementum update
W/O momentum

Experimental Setup

Self-supervised pre-training on ImageNet

Downstream tasks and datasets:

- Image classification on ImageNet
- Object detection on VOC 07 and COCO (tranfer ability)

Experimental Results

ImageNet classification

- **Linear classification:** Freeze the trained encoder (Res50) via SSL, add a linear layer

method	batch size	negative pairs	momentum encoder	100 ep	200 ep	400 ep	800 ep
SimCLR (repro.+)	4096	✓		66.5	68.3	69.8	70.4
MoCo v2 (repro.+)	**256**	✓	✓	67.4	69.9	71.0	72.2
BYOL (repro.)	4096		✓	66.5	**70.6**	**73.2**	**74.3**
SwAV (repro.+)	4096			66.5	69.1	70.7	71.8
SimSiam	**256**			**68.1**	70.0	70.8	71.3

Table 4. **Comparisons on ImageNet linear classification**. All are based on **ResNet-50** pre-trained with **two 224×224 views**. Evaluation is on a single crop. All competitors are from our reproduction, and "+" denotes *improved* reproduction *vs*. original papers (see supplement).

- Simple design, good performance
- 100 epoch is good enough
- Momentum encoder benefits performance

Experimental Results

Object Detection

- Transfer learning: initiate encoder with pre-trained weights, and finetune

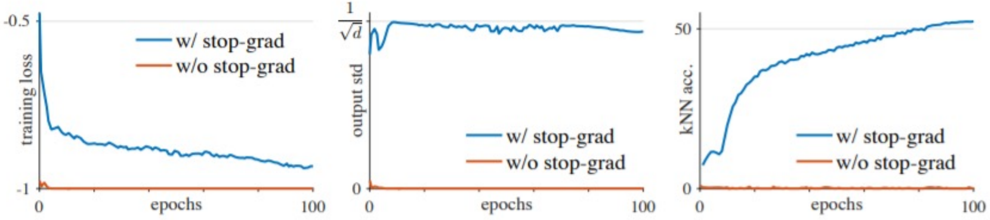
pre-train	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance seg.		
	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
scratch	35.9	16.8	13.0	60.2	33.8	33.1	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet supervised	74.4	42.4	42.7	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR (repro.+)	75.9	46.8	50.1	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2 (repro.+)	**77.1**	**48.5**	**52.5**	**82.3**	**57.0**	**63.3**	**58.8**	**39.2**	**42.5**	**55.5**	**34.3**	**36.6**
BYOL (repro.)	**77.1**	47.0	49.9	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SwAV (repro.+)	75.5	46.5	49.6	81.5	55.4	61.4	57.6	37.6	40.3	54.2	33.1	35.1
SimSiam, base	75.5	47.0	50.2	**82.0**	56.4	62.8	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam, optimal	**77.3**	**48.5**	**52.5**	**82.4**	**57.0**	**63.7**	**59.3**	**39.2**	**42.1**	**56.0**	**34.4**	**36.7**

Table 5. **Transfer Learning**. All unsupervised methods are based on 200-epoch pre-training in ImageNet. *VOC 07 detection*: Faster R-CNN [32] fine-tuned in VOC 2007 trainval, evaluated in VOC 2007 test; *VOC 07+12 detection*: Faster R-CNN fine-tuned in VOC 2007 trainval + 2012 train, evaluated in VOC 2007 test; *COCO detection* and *COCO instance segmentation*: Mask R-CNN [18] (1× schedule) fine-tuned in COCO 2017 train, evaluated in COCO 2017 val. All Faster/Mask R-CNN models are with the C4-backbone [13]. All VOC results are the average over 5 trials. **Bold entries** are within 0.5 below the best.

- Learned representations transfer well!

Experimental Results

Abaltion: stop gradient and symmetric training



	acc. (%)
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

Figure 2. **SimSiam with *vs*. without stop-gradient**. **Left plot**: training loss. Without stop-gradient it degenerates immediately. **Middle plot**: the per-channel std of the ℓ_2-normalized output, plotted as the averaged std over all channels. **Right plot**: validation accuracy of a kNN classifier [36] as a monitor of progress. **Table**: ImageNet linear evaluation ("w/ stop-grad" is mean±std over 5 trials).

	sym.	asym.	asym. 2×
acc. (%)	68.1	64.8	67.3

- Stop gradient is the key for preventing collapse
- Symmetric training can boost performance

Critique / Limitations / Open Issues

SimSiam is a simple but effective contrastive learning method

- Contribution: Find the key for model collapse, and simplify the designs
- Kaiming's Philosophy: Only simple designs can capture the essence, and transfer well

However, the method cannot be explained in a thermotical way

- In the paper, their hypothesis is that, SimSiam is doing Expectation-Maximization (EM)

Moreover, can the model transfer to other downstream tasks? Especially for robot learning

- Data augmentation make the model get many invariance, e.g. rotation invariance
- This may hurt when you transfer it as a pose estimation backbone

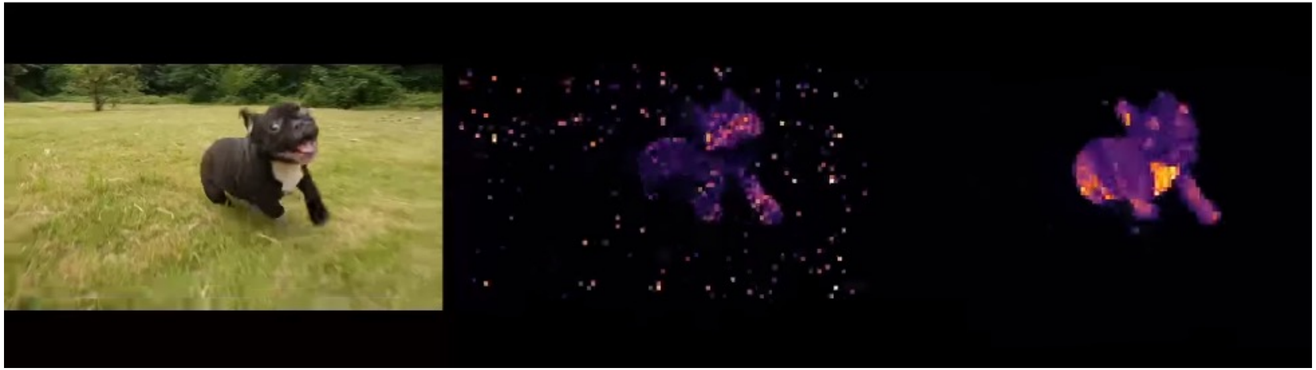
Future Work for Paper / Reading

Three trends in self-supervised learning

1. Exploring the transformer architecture for self-supervised learning

Supervised ViT

Unsupervised ViT



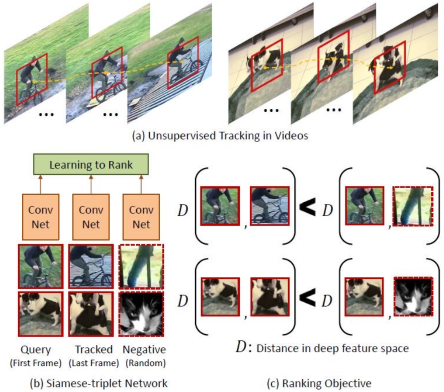
- Performance is better
- Good properties emerge

[1] Caron et al., Emerging Properties in Self-Supervised Vision Transformers, 2021
[2] Chen et al., An Empirical Study of Training Self-supervised Vision Transformers, 2021

Future Work for Paper / Reading

Three trends in self-supervised learning

2. Exploring spatial-temporal information

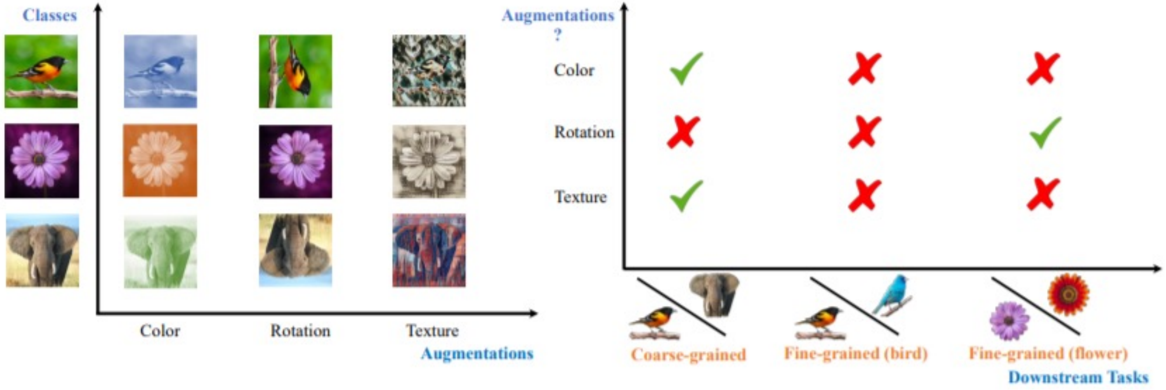


[1] Wang et al., Unsupervised Learning on Visual Representations using Videos, 2016
[2] Feichtenhofer et al., A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning, 2021
[3] Qian et al., Spatial-temporal Contrastive Video Representation Learning, 2020

Future Work for Paper / Reading

Three trends in self-supervised learning

3. Exploring the invariance (data augmentation) and its influence on downstream tasks



Xiao et al., What Should Not Be Contrastive in Contrastive Learning, 2021

Summary

SimSiam

- Target: Explore the reason for model collapse
- Key insight: stop gradient of one side of the Siamese network
- Momentum encoder is not the key for preventing collapse
- Also validate many other designs, e.g. momentum encoder, predictor
- Limitation: theoretically hard to understand