Unsupervised pre-training for images

Sunita Sarawagi

CS 725 Fall 2023

Reading material

- Chapter 19.2.4 in Probabilistic ML by K Murphy
- Papers

Motivation

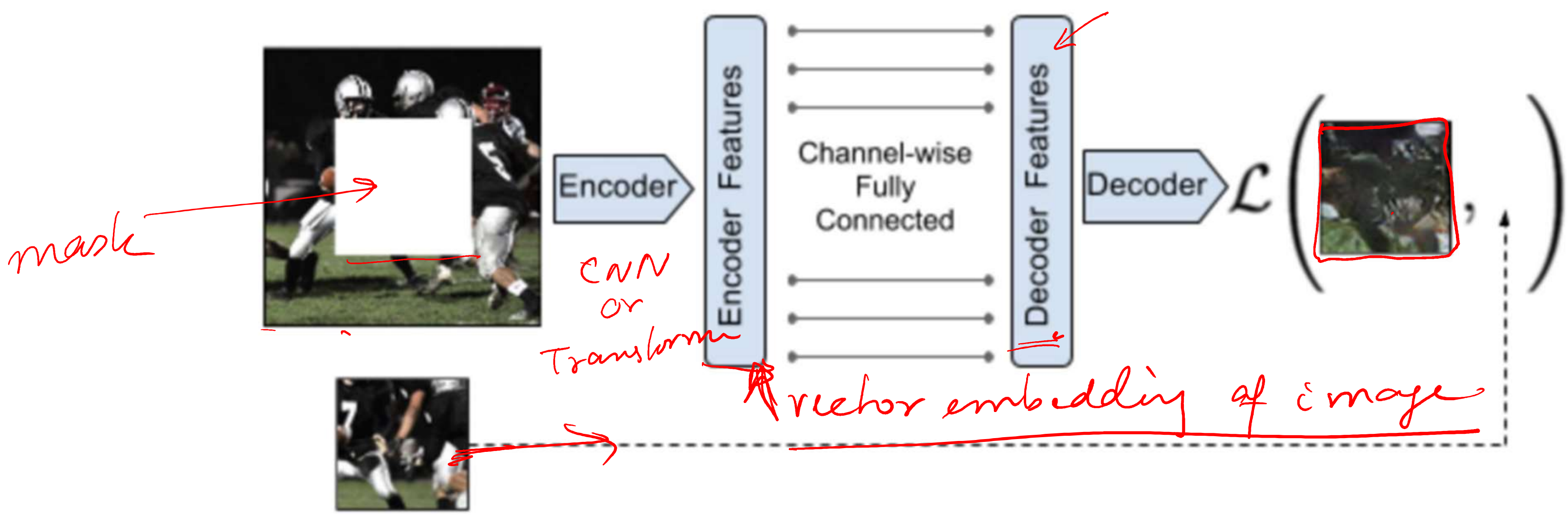
- ML models have more parameters than the amount of labelled data available for training them well
  - Labeled data: e.g.
    - Images along with labels of objects in them
    - Images with caption
- ResNET CNN with 50 layers has 23 million parameters
- Unlabelled data:
  - Large collection of images but without any labels or captions.
  - Can we harness these for pre-training a CNN for image classification or captioning?

Unsupervised → self-supervised

- Starting from unlabeled data e.g. collection of images, use a set of scripts to automatically create supervised tasks out of them.
- Example, for text data next-token prediction.
- Three types of self-supervised tasks for images:
  - Imputation
    - Mask part of image and generate that
  - Proxy or pretext tasks
    - Create image pairs and use Siamese networks to generate representations that can predict relationship between them
  - Contrastive learning
    - Like metric learning, but create similar pairs on your own.

Imputation

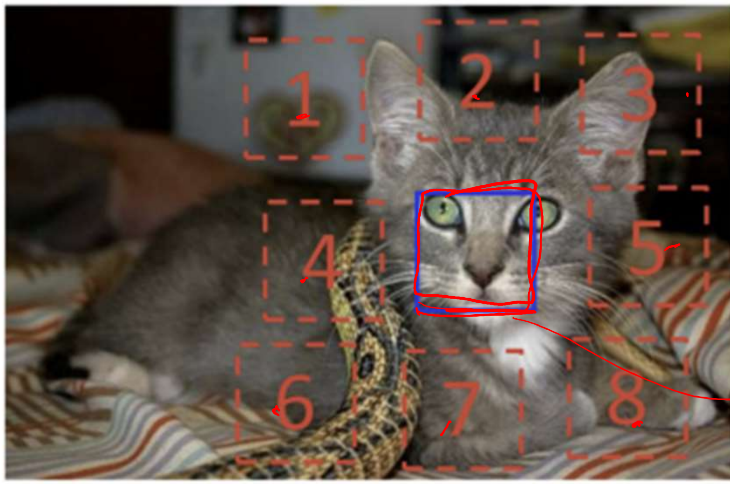
Learning to inpaint by reconstruction



Learning to reconstruct the missing pixels

Source: [Pathak et al., 2016]

Pretext task: predict relative patch locations



X = (); Y = 3

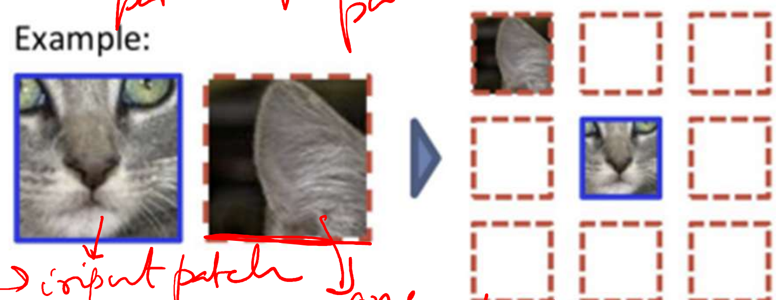
center patch

neighboring patch

orientation of neighbor

1 of 8 possible values: classification task

Example:

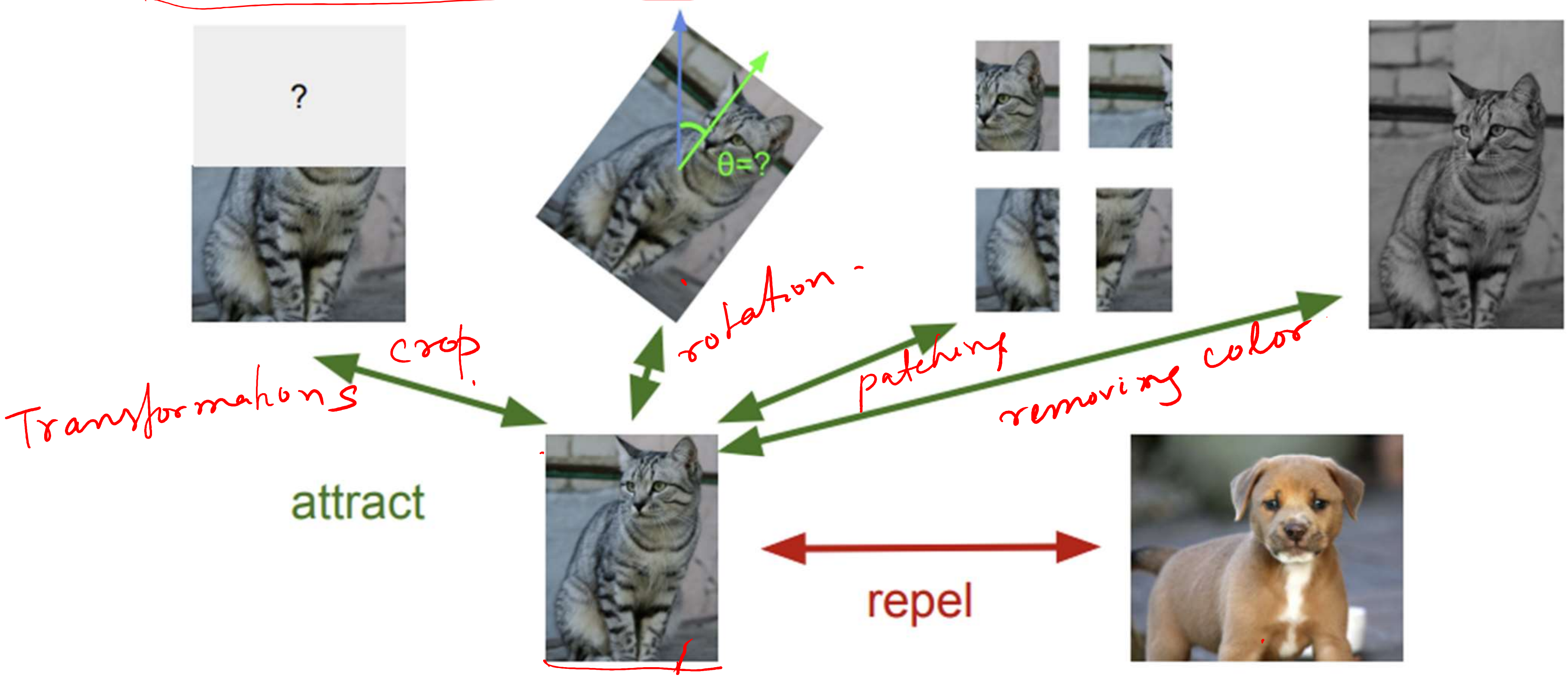


input patch

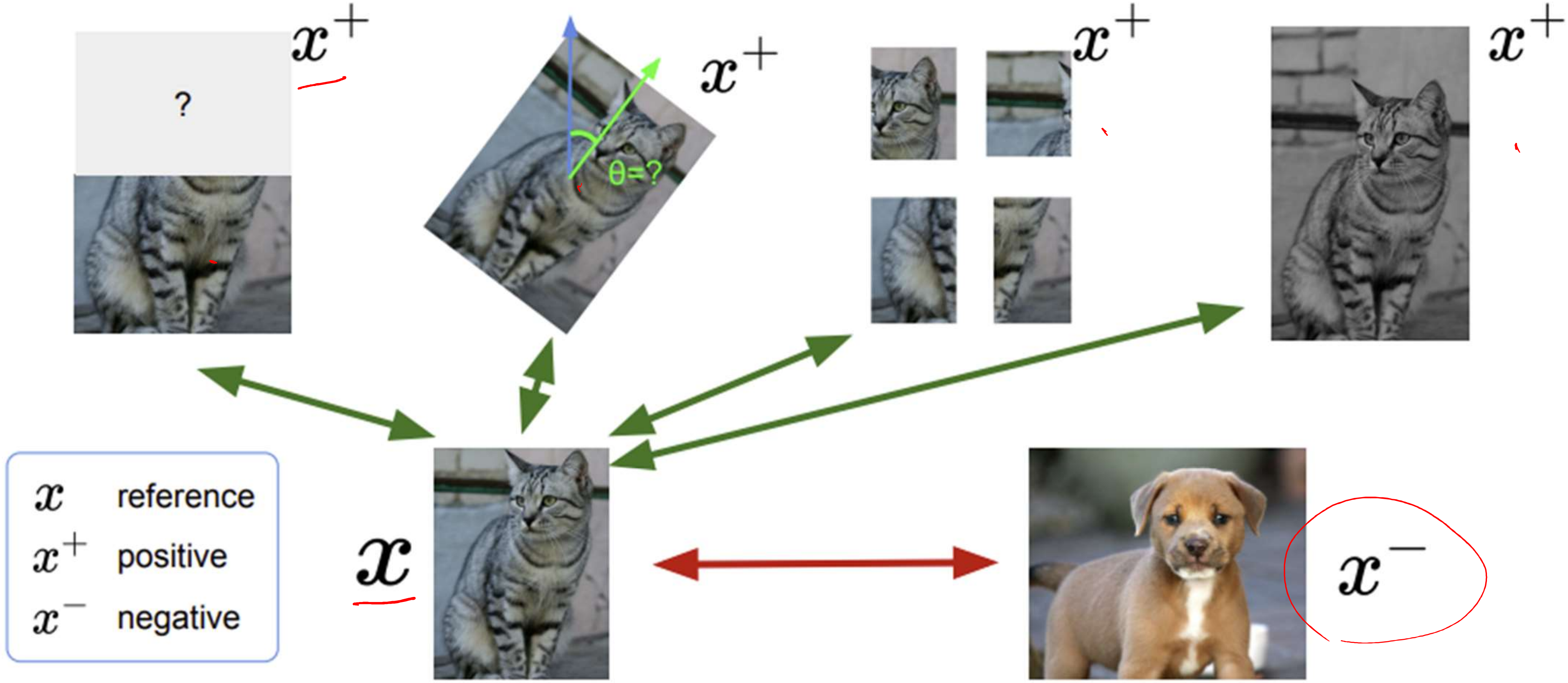
one of the eight neighboring patch.

(Image source: Doersch et al., 2015)

Contrastive Representation Learning



Contrastive Representation Learning



A formulation of contrastive learning

Loss function given 1 positive sample and N - 1 negative samples:

similarity function

representation or embedding of input

input

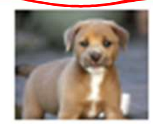
$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\exp(s(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))} \right]$$

x

x^+

$y = 1$

x



x_1^-

x_2^-

x_3^-

...

SimCLR: A Simple Framework for Contrastive Learning

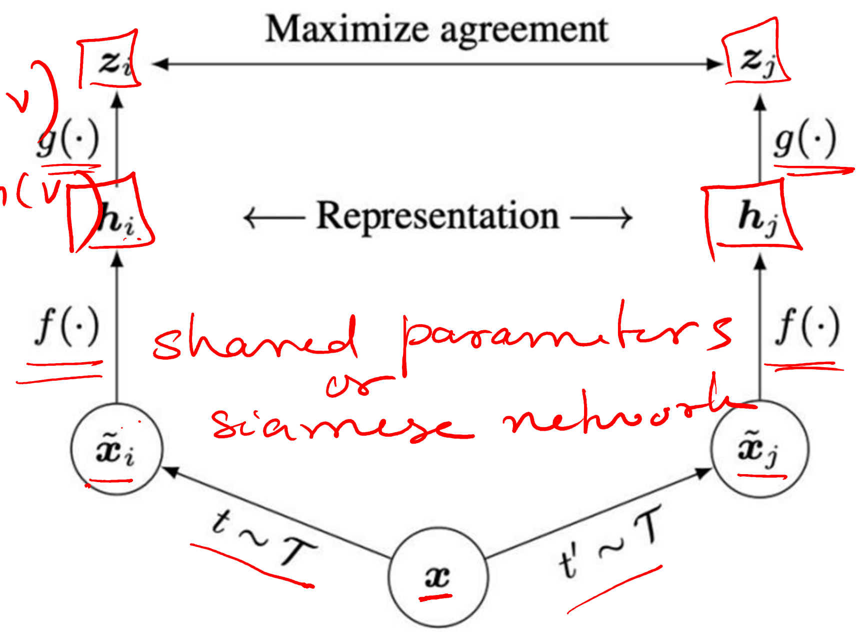
Cosine similarity as the score function:

$$s(u,v) = \frac{u^T v}{||u||||v||} = \frac{\text{dot product}(u,v)}{\text{norm}(u)\,\text{norm}(v)}$$

Use a projection network ***g(·)*** to project features to a space where contrastive learning is applied

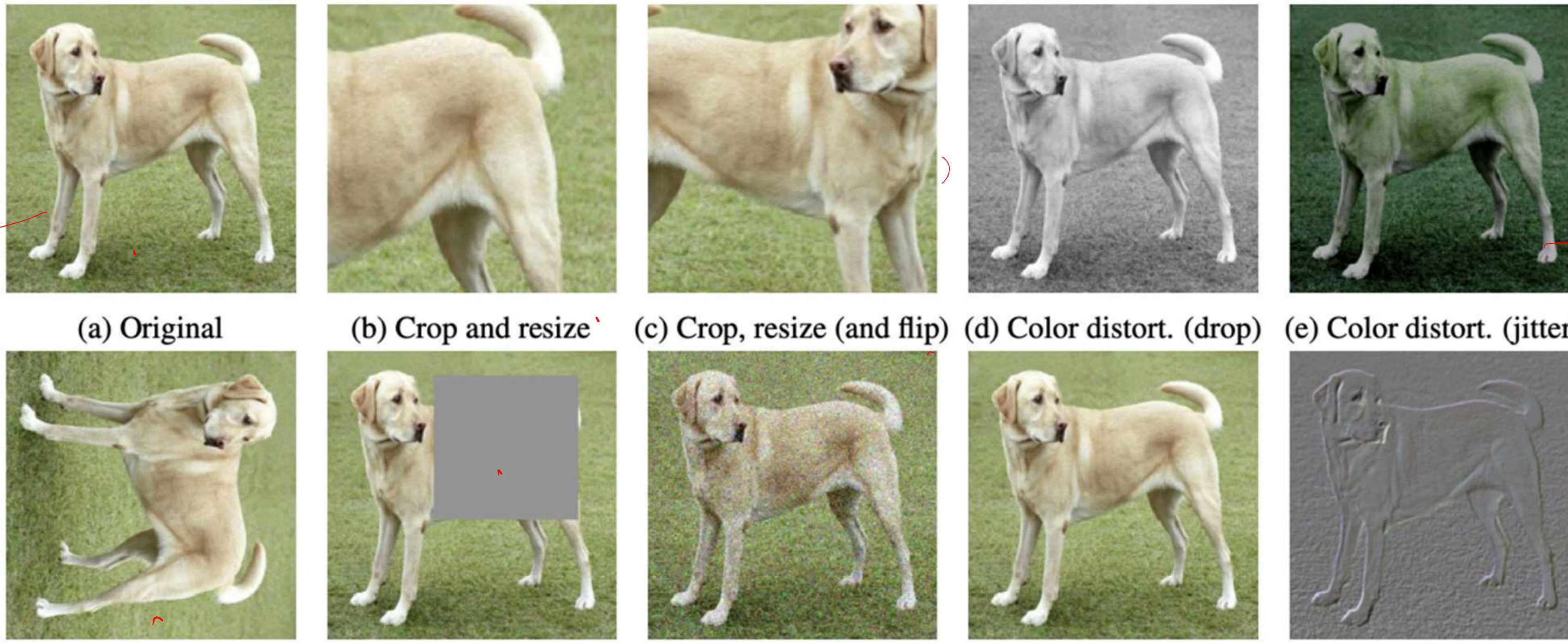
Generate positive samples through data augmentation:

- random cropping, random color distortion, and random blur.



Source: Chen et al., 2020

SimCLR: generating positive samples from data augmentation



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Source: [Chen et al., 2020]

SimCLR

Generate a positive pair by sampling data augmentation functions

Iterate through and use each of the 2N sample as reference, compute average loss

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^N$ **do**
  for all $k \in \{1, \dots, N\}$ **do**
    $\boxed{\text{draw two augmentation functions } t \sim \mathcal{T}, t' \sim \mathcal{T}}$
    # the first augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \dots, 2N\}$ and $j \in \{1, \dots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i,j)$ **as** $\boxed{\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

*We use a slightly different formulation in the assignment. You should follow the assignment instructions.

InfoNCE loss:
Use all non-positive samples in the batch as x^-

Source: Chen et al., 2020

Semi-supervised learning on SimCLR features

Method	Architecture	Label fraction 1%	Label fraction 10%
		Top 5	
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Train feature encoder on **ImageNet** (entire training set) using SimCLR.

Finetune the encoder with 1% / 10% of labeled data on ImageNet.

Source: Chen et al., 2020

Self-supervised multi-modal pre-training

- Learning to jointly encode image and text

https://icml.cc/media/icml-2021/Slides/9193.pdf