An introduction to self-supervised learning

DEEP LEARNING

FALL 2023

Mohammad Kalbasi

Dr. Emad Fatemizadeh

Sharif University of Technology

Courtesy: Most of slides are adopted from CS 231 Standford and EECS 498 University of Michigan .

Recall: Supervised vs Unsupervised Learning

Supervised Learning	Unsupervised Learning
Data: (x, y) x is data, y is label	**Data**: x Just data, no labels!
Goal: Learn a *function* to map x -> y	**Goal**: Learn some underlying hidden *structure* of the data
Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.	**Examples**: Clustering, dimensionality reduction, feature learning, density estimation, etc.

Problem: Supervised Learning is Expensive!

Assume you want to label 1M images. How much will it cost?

(1,000,000 images)	(Small to medium sized dataset)
× (10 seconds/image)	(Fast annotation)
× (1/3600 hours/second)	
× ($15 / hour)	(Low wage paid to annotator)

Problem: Supervised Learning is Expensive!

Assume you want to label 1M images. How much will it cost?

(1,000,000 images)	(Small to medium sized dataset)
× (10 seconds/image)	(Fast annotation)
× (1/3600 hours/second)	
× ($15 / hour)	(Low wage paid to annotator)
= $41,667	

(Other assumptions: one annotator per image, no benefits / payroll tax / crowdsourcing fee for annotators; not accounting for time to set up tasks for annotators, etc. Real costs could easily be 3x this or more)

Problem: Supervised Learning is Expensive!

Assume you want to label **1B** images. How much will it cost?

(1,000,000,000 images)	(Large dataset)
× (10 seconds/image)	(Fast annotation)
× (1/3600 hours/second)	
× ($15 / hour)	(Low wage paid to annotator)
= **$41,666,667**	

(Other assumptions: one annotator per image, no benefits / payroll tax / crowdsourcing fee for annotators; not accounting for time to set up tasks for annotators, etc. Real costs could easily be 3x this or more)

Problem: Supervised Learning is Not How We Learn

Babies don't get supervision for everything they see!

Solution: Self-Supervised Learning

Lets build methods that learn from ”raw” data – no annotations required

Unsupervised Learning: Model isn’t told what to predict. Older terminology, not used as much today.

Self-Supervised Learning: Model is trained to predict some naturally-occurring signal in the raw data rather than human annotations.

Solution: Self-Supervised Learning

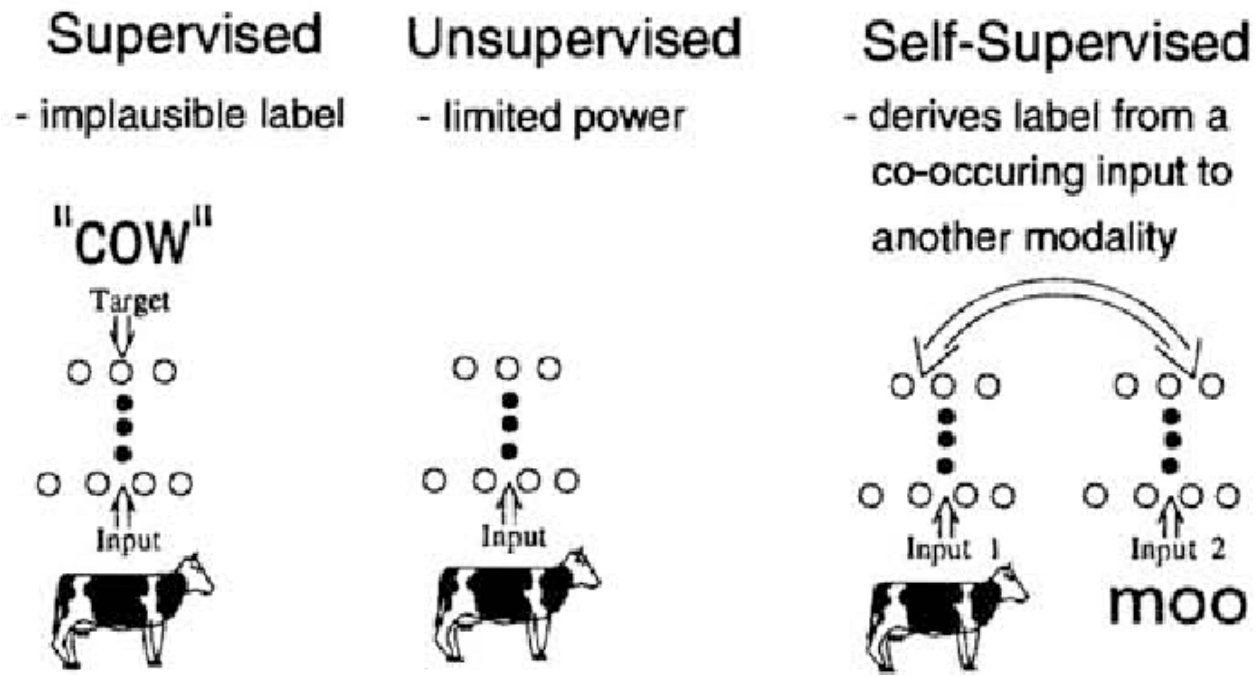


Image: Virga de Sa, 1994, Learning Classification with Unlabeled Data

Broader picture

Today's lecture

computer vision

Doersch et al., 2015

robot / reinforcement learning

Dense Object Net (Florence and Manuelli et al., 2018)

language modeling

Language Models are Few-Shot Learners

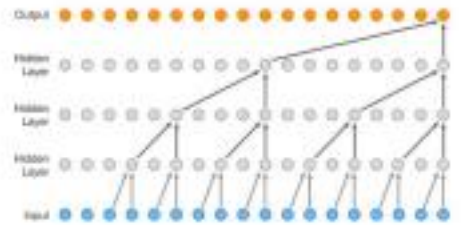
Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*
Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry
Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan
Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray
Benjamin Chess Jack Clark Christopher Berner
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei
OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

GPT3 (Brown, Mann, Ryder, Subbiah et al., 2020)

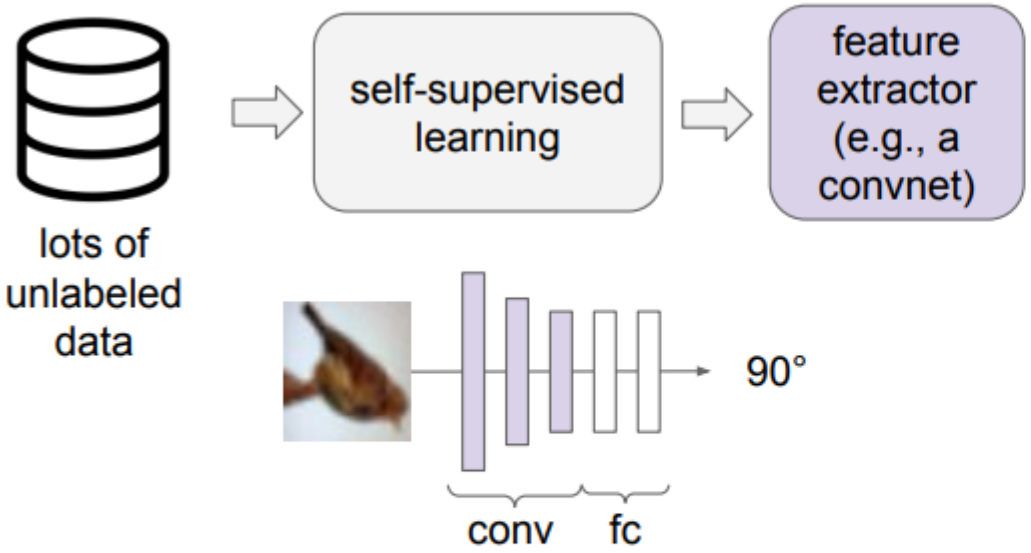
speech synthesis



Wavenet (van den Oord et al., 2016)

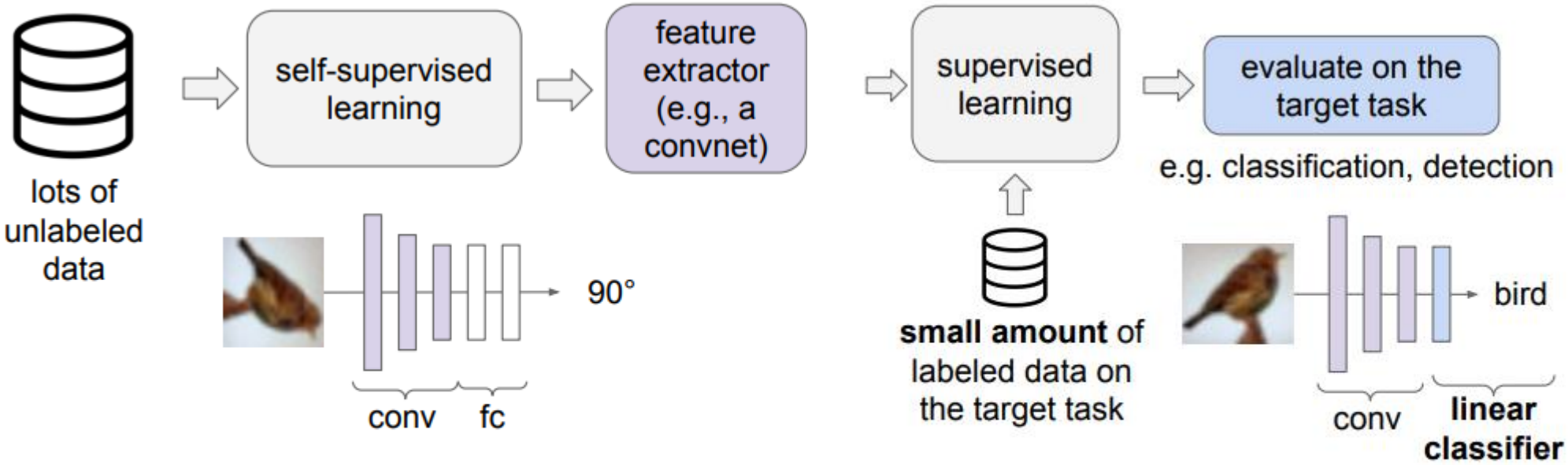
...

Pretext Task and Downstream Task



1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

Pretext Task and Downstream Task



1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

2. Attach a shallow network on the feature extractor; train the shallow network on the target task with small amount of labeled data

Main Topics

Pretext tasks from image transformations

- Rotation, inpainting, rearrangement, coloring

Contrastive representation learning

- Intuition and formulation
- Instance contrastive learning: SimCLR and MOCO

Main Topics

Pretext tasks from image transformations

- Rotation, inpainting, rearrangement, coloring

Contrastive representation learning

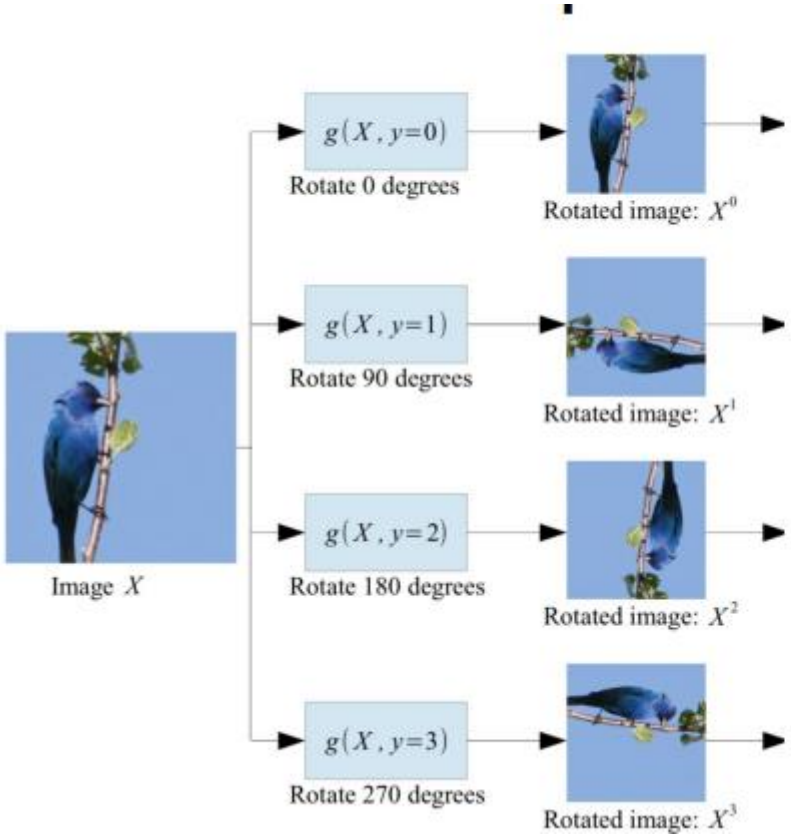
- Intuition and formulation
- Instance contrastive learning: SimCLR and MOCO

Predicting Rotation

Hypothesis: a model could recognize the correct rotation of an object only if it has the "visual commonsense" of what the object should look like unperturbed.

Image: Gidaris et al, "Unsupervised representation learning by predicting image rotations", ICLR 2018

Predicting Rotation

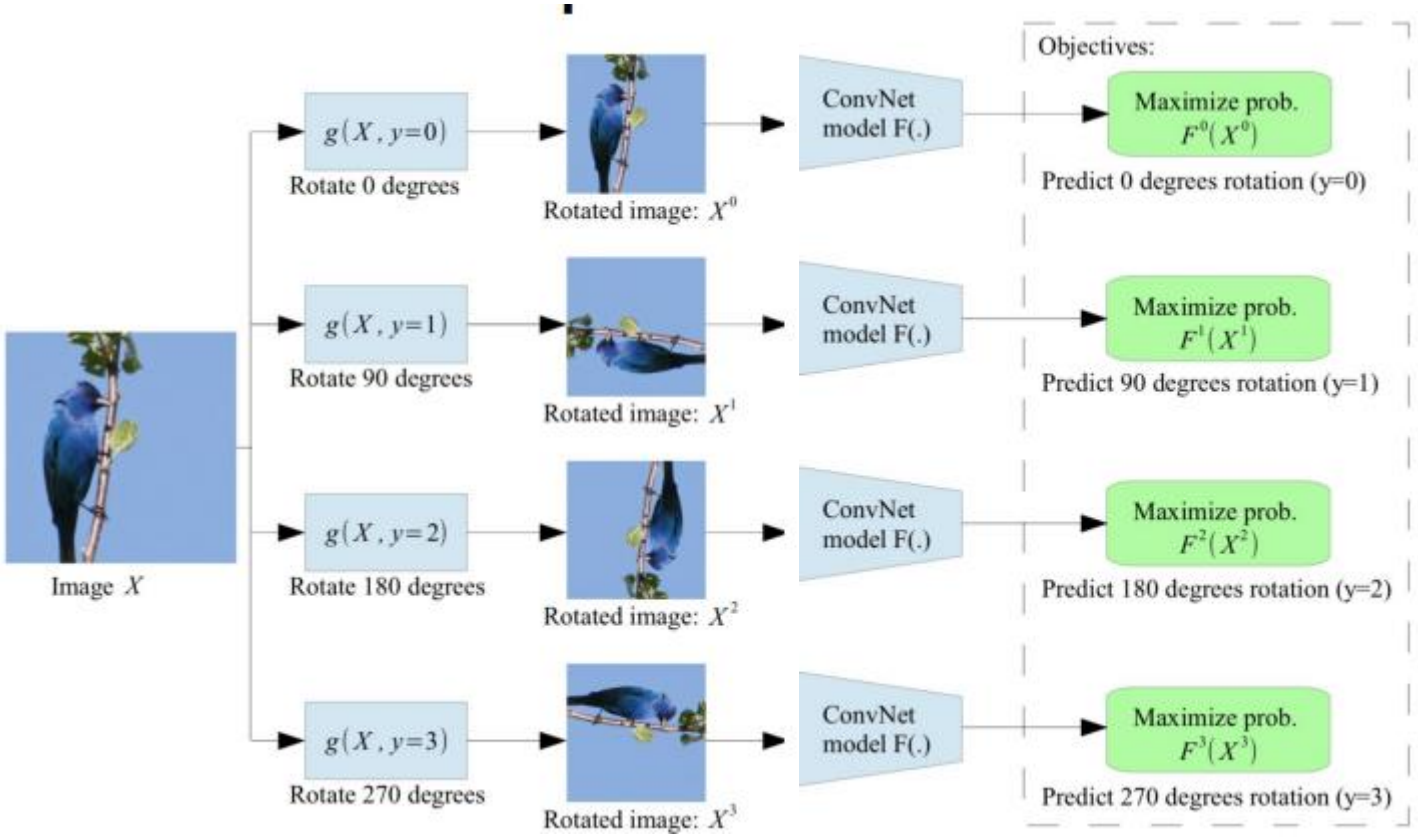


Self-supervised learning by rotating the entire input images.

The model learns to predict which rotation is applied (4-way classification)

Image: Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

Predicting Rotation

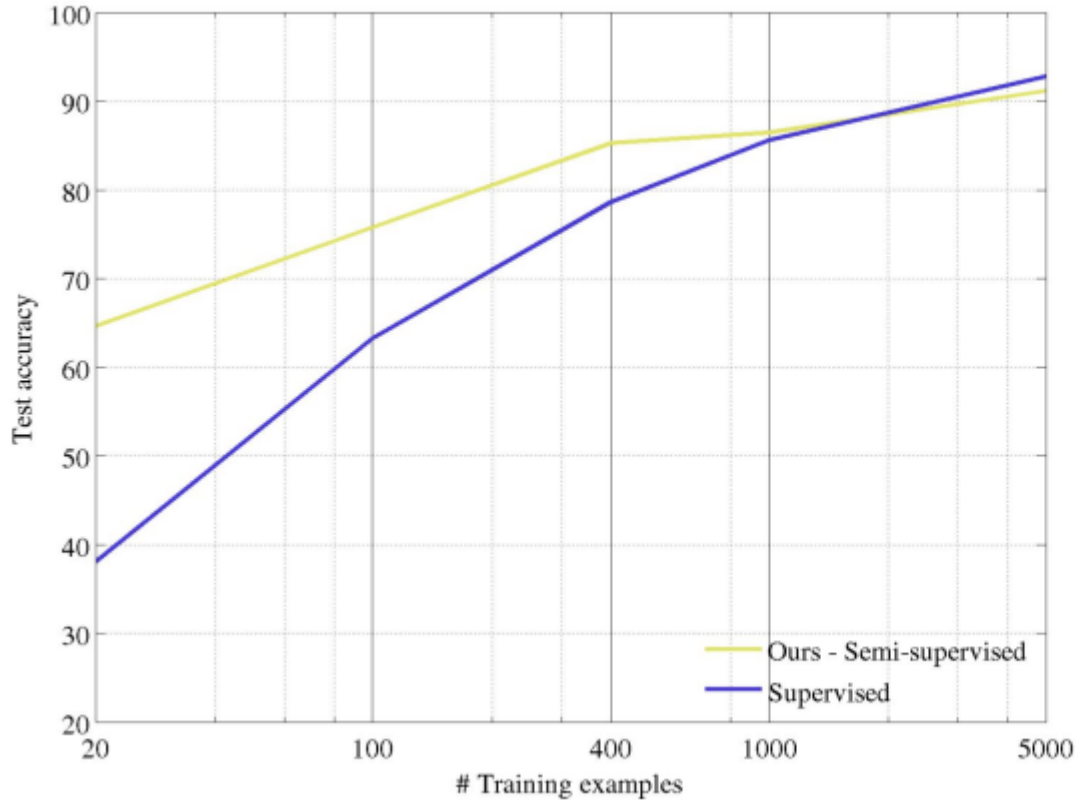


Self-supervised learning by rotating the entire input images.

The model learns to predict which rotation is applied (4-way classification)

Image: Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

Evaluation on Semi-Supervised Learning



Self-supervised learning on **CIFAR10** (entire training set).

Freeze conv1 + conv2
Learn **conv3 + linear** layers with subset of labeled CIFAR10 data (classification).

Image: Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

Transfer Learned Features to Supervised Learning

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
(Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

Pretrained with full ImageNet supervision

No pretraining

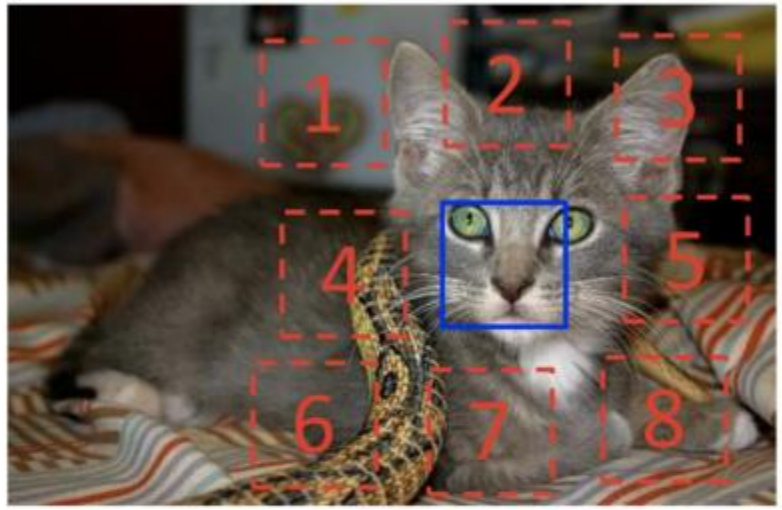
Self-supervised learning with rotation prediction

Self-supervised learning on **ImageNet** (entire training set) with AlexNet.

Finetune on labeled data from **Pascal VOC 2007**.

Image: Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

Predict Relative Patch Position



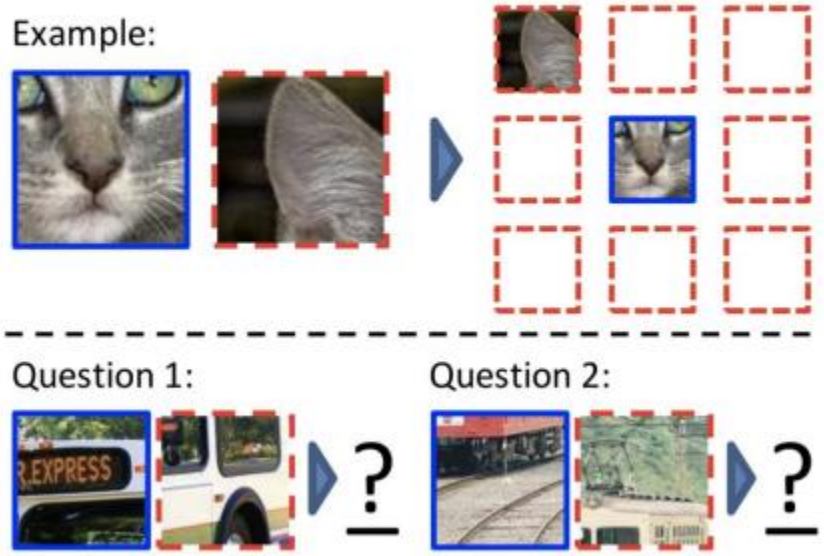
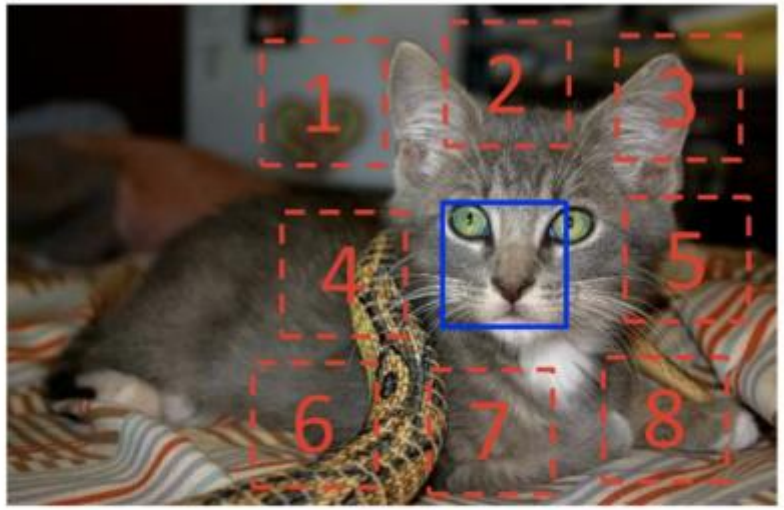


Image: Doersch et al, “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Predict Relative Patch Position



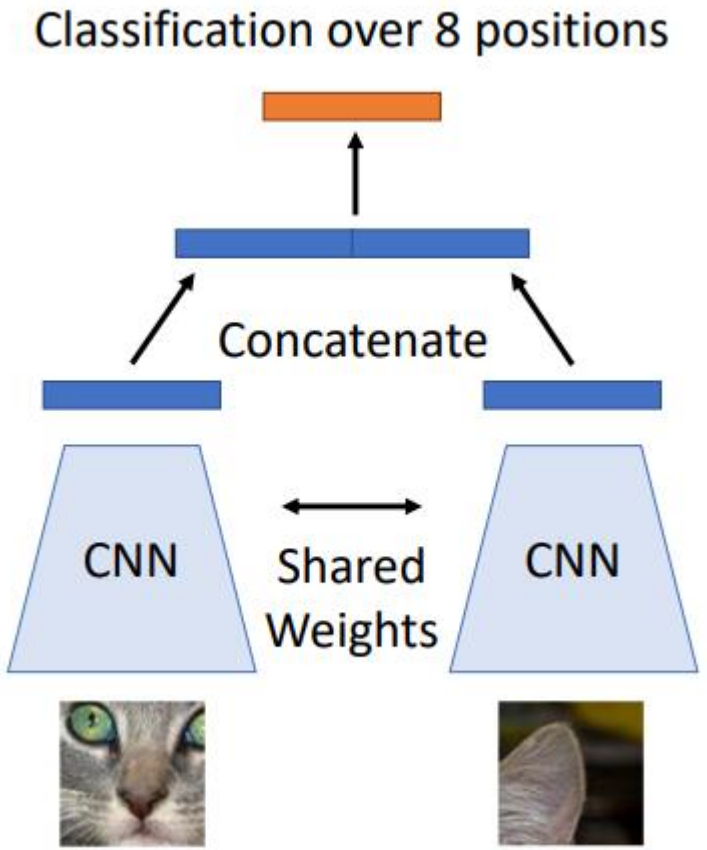


Image: Doersch et al, “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Nearest Neighbor in Feature Space

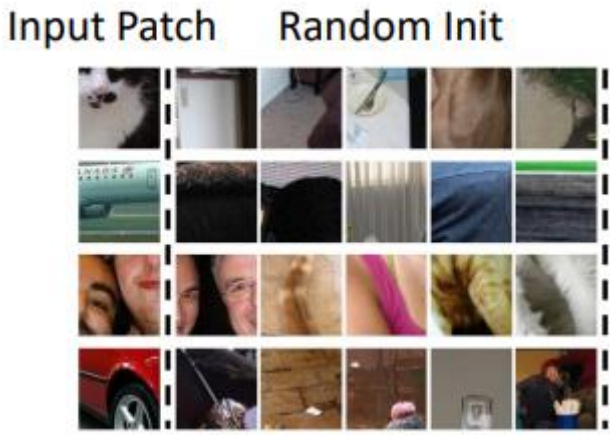


Image: Doersch et al, “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Nearest Neighbor in Feature Space

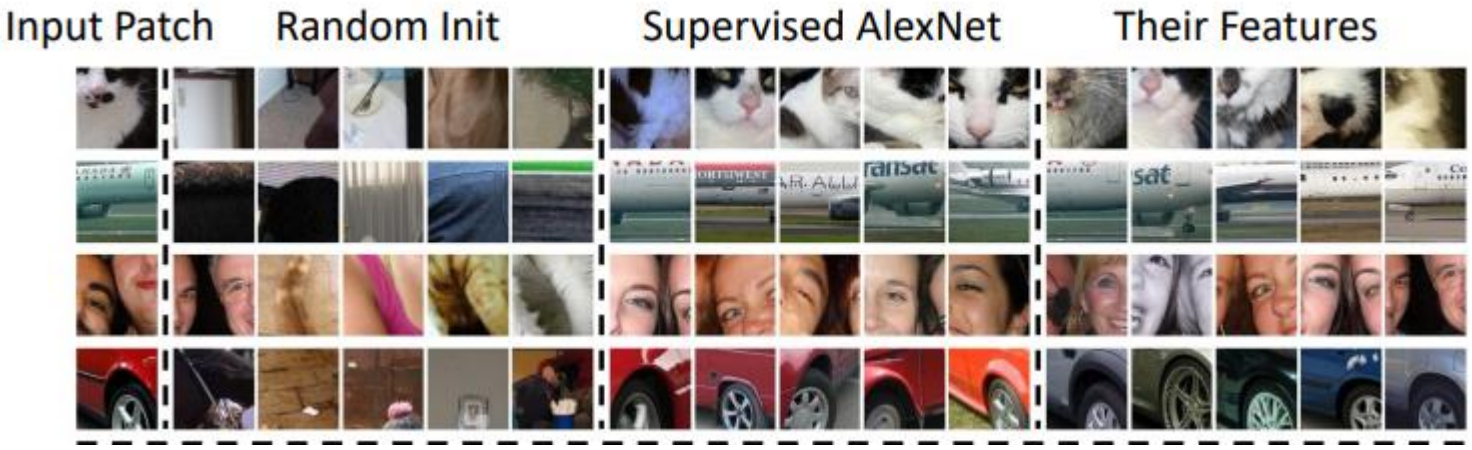


Image: Doersch et al, “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Nearest Neighbor in Feature Space

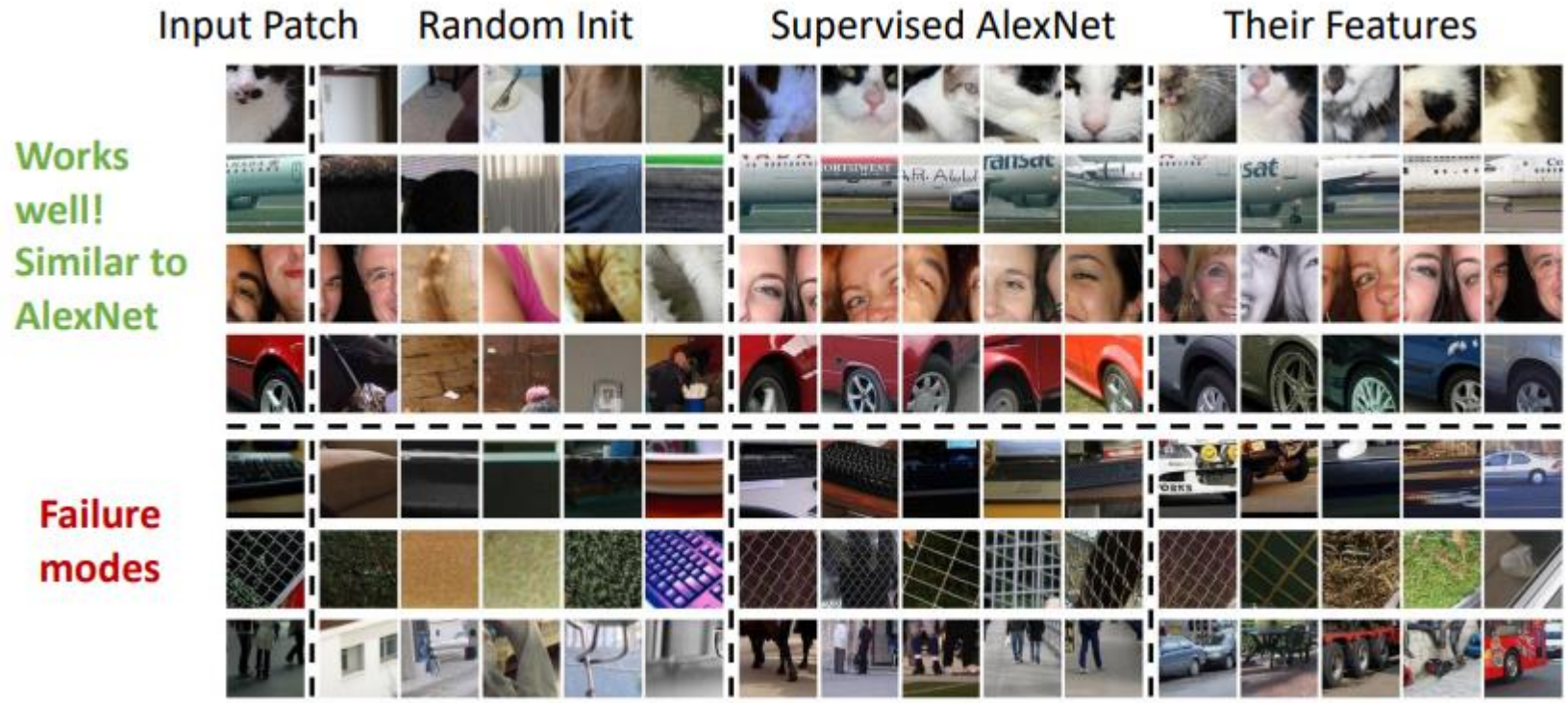


Image: Doersch et al, “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Solving Jigsaw Puzzle

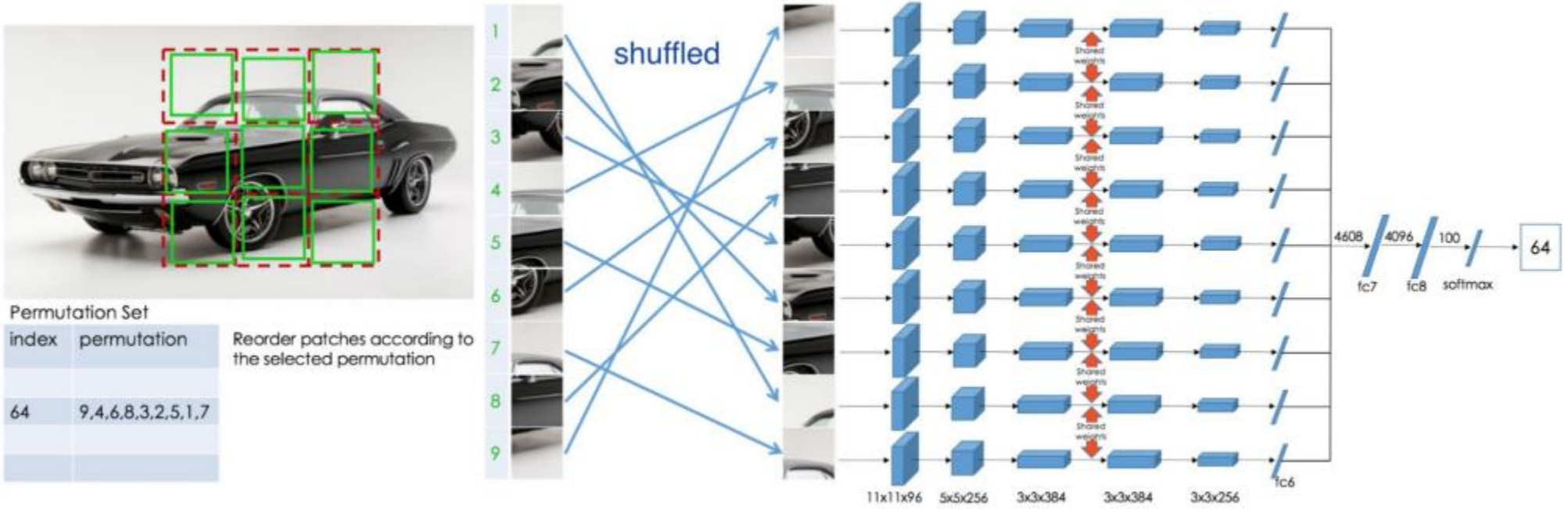


Image: Noroozi and Favoro, “Unsupervised learning of visual representations by solving jigsaw puzzles”, ECCV 2016

Solving Jigsaw Puzzle

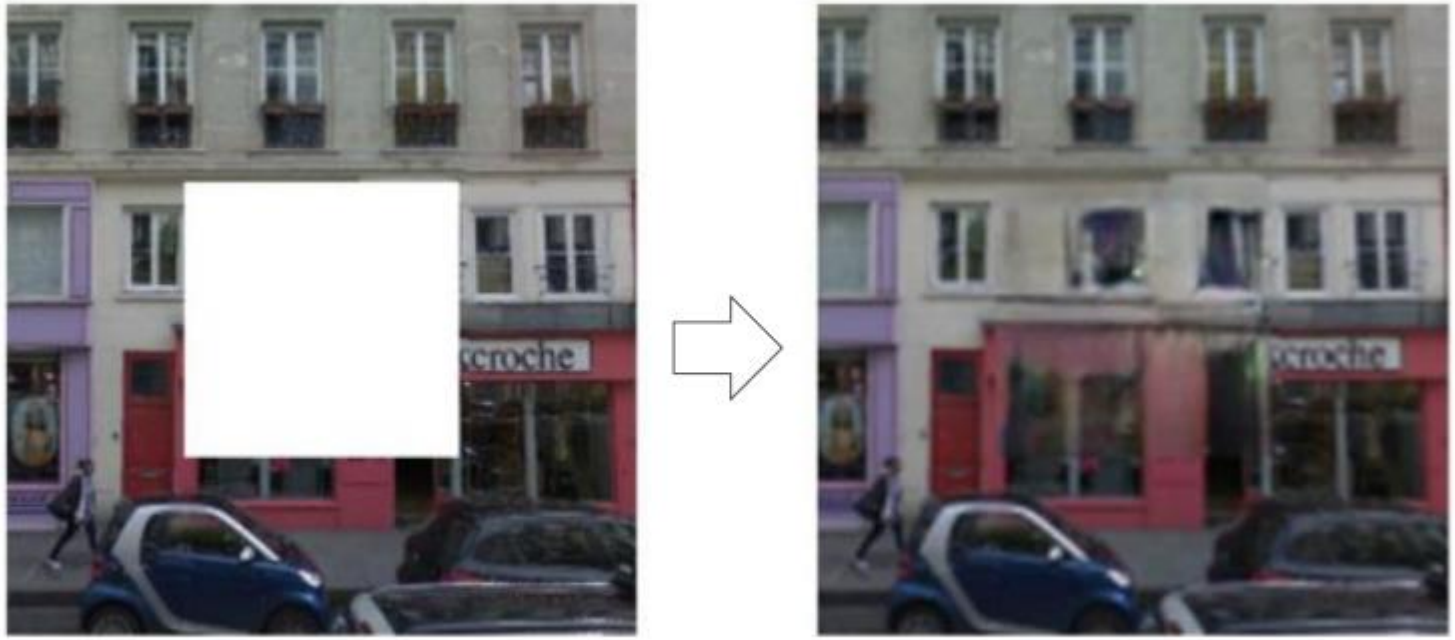
Table 1: Results on PASCAL VOC 2007 Detection and Classification. The results of the other methods are taken from Pathak *et al.* [30].

Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Wang and Gupta[39]	1 week	motion	58.4%	44.0%	-
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%	-
Pathak *et al.* [30]	14 hours	context	56.5%	44.5%	29.7%
Ours	2.5 days	context	**67.6%**	**53.2%**	**37.6%**

"Ours" is feature learned from solving image Jigsaw puzzles (Noroozi & Favaro, 2016). Doersch et al. is the method with relative patch location

Image: Noroozi and Favoro, "Unsupervised learning of visual representations by solving jigsaw puzzles", ECCV 2016

Image Inpainting



Context Encoders: Feature Learning by Inpainting (Pathak et al., 2016)

Image: Pathak et al, “Context Encoders: Feature Learning by Inpainting”

Image Inpainting

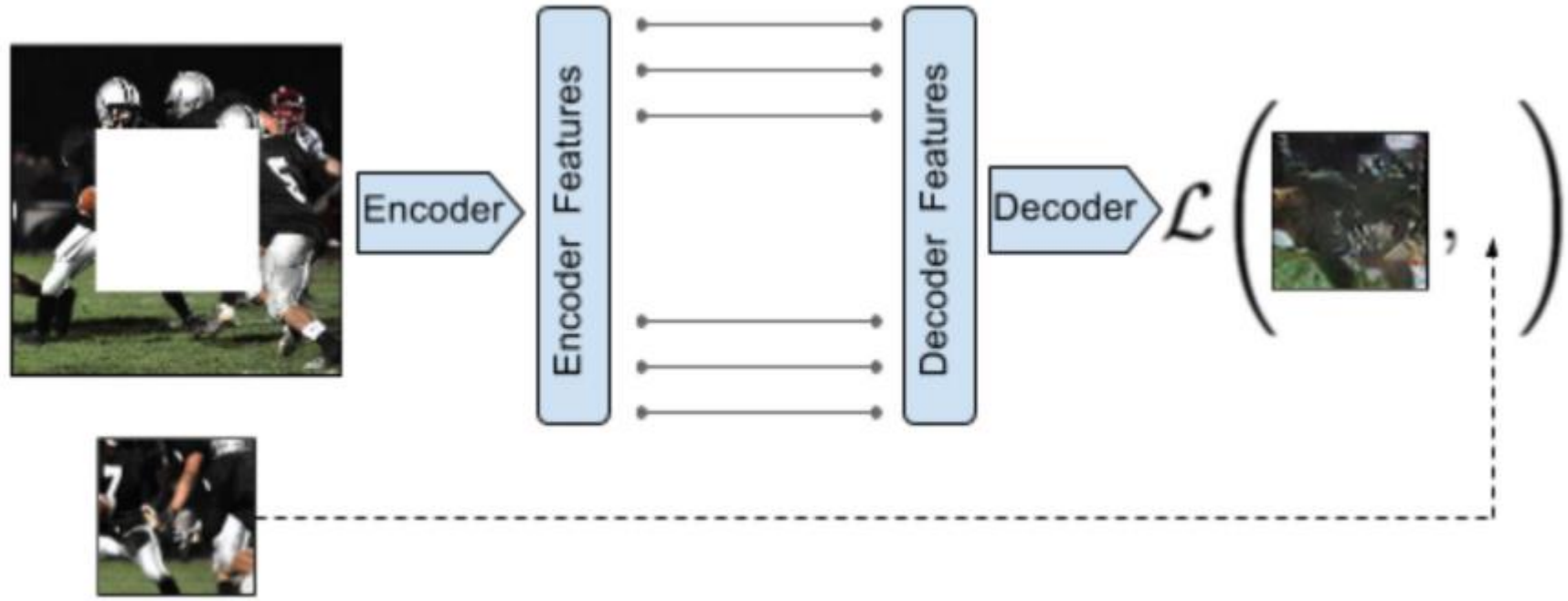


Image: Pathak et al, “Context Encoders: Feature Learning by Inpainting”

Image Inpainting

Input (context) reconstruction

Image: Pathak et al, “Context Encoders: Feature Learning by Inpainting”

Image Inpainting

Loss = reconstruction + adversarial learning

$$L(x) = L_{recon}(x) + L_{adv}(x)$$

$$L_{recon}(x) = ||M * (x - F_\theta((1 - M) * x))||_2^2$$

$$L_{adv} = \max_D \mathbb{E}[\log(D(x))] + \log(1 - D(F((1 - M) * x)))]$$

Adversarial loss between “real” images and *inpainted images*

Image: Pathak et al, “Context Encoders: Feature Learning by Inpainting”

Image Inpainting

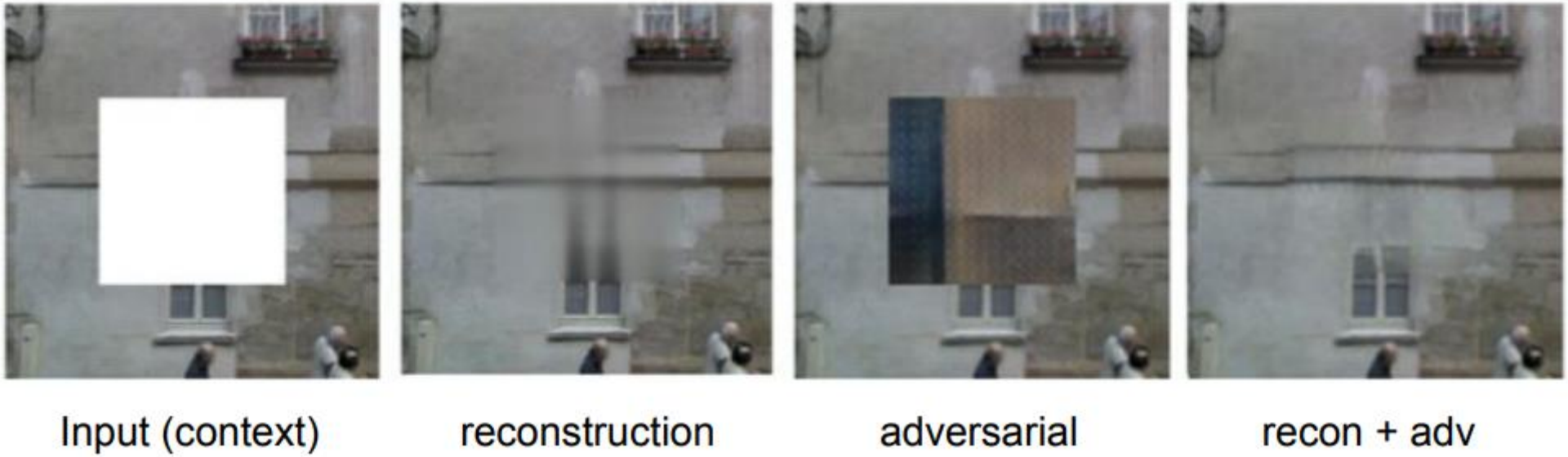


Image: Pathak et al, “Context Encoders: Feature Learning by Inpainting”

Image Inpainting

Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Wang *et al.* [39]	motion	1 week	58.7%	47.4%	-
Doersch *et al.* [7]	relative context	4 weeks	55.3%	46.6%	-
Ours	context	14 hours	56.5%	44.5%	30.0%

Self-supervised learning on ImageNet training set, transfer to classification (Pascal VOC 2007), detection (Pascal VOC 2007), and semantic segmentation (Pascal VOC 2012)

Image: Pathak et al, “Context Encoders: Feature Learning by Inpainting”

Image coloring

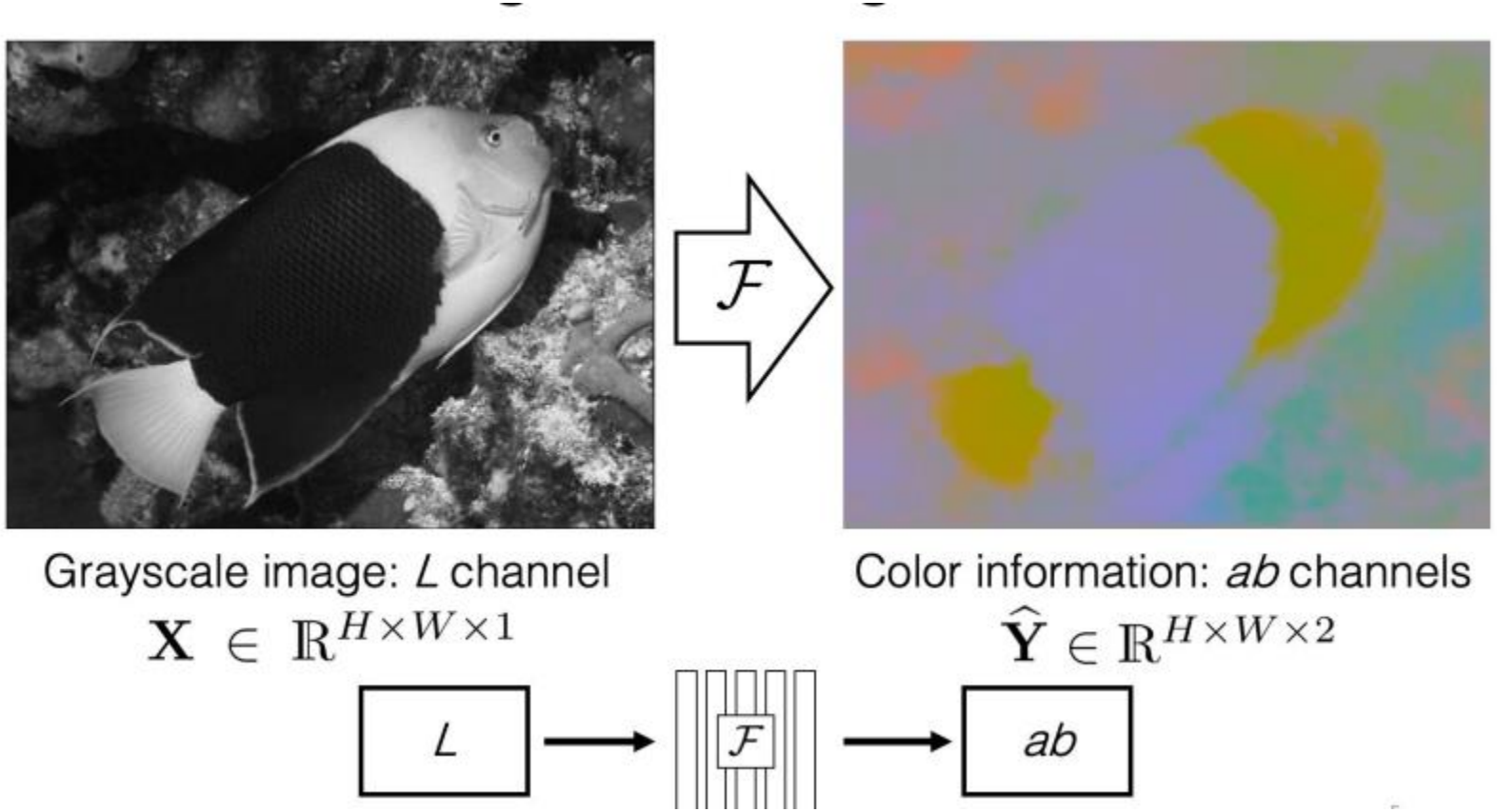


Image: Zhang et al, “Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction”,

Image coloring

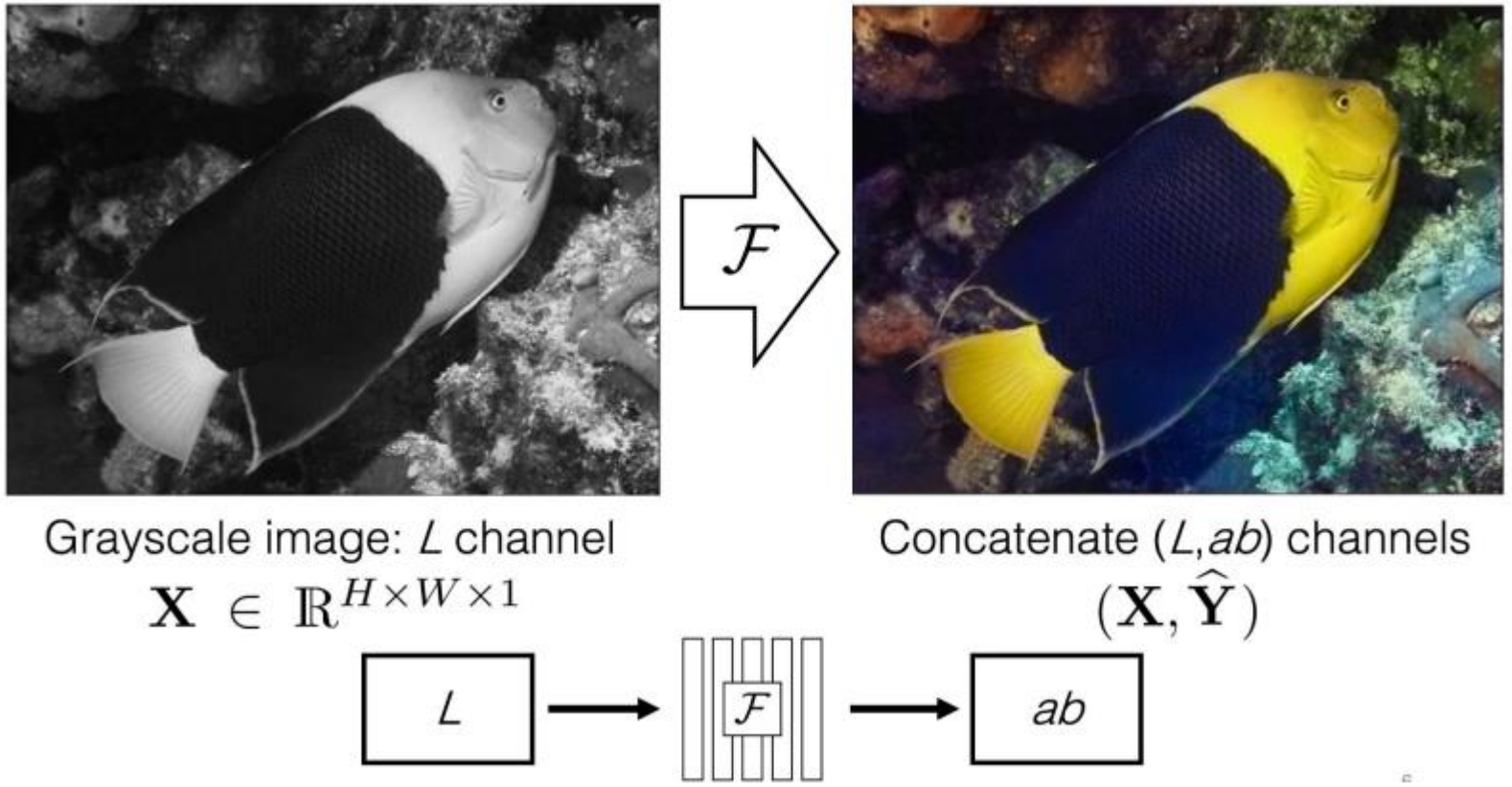


Image: Zhang et al, “Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction”,

Learning Feature from Colorization:Split-Brain Autoencoder

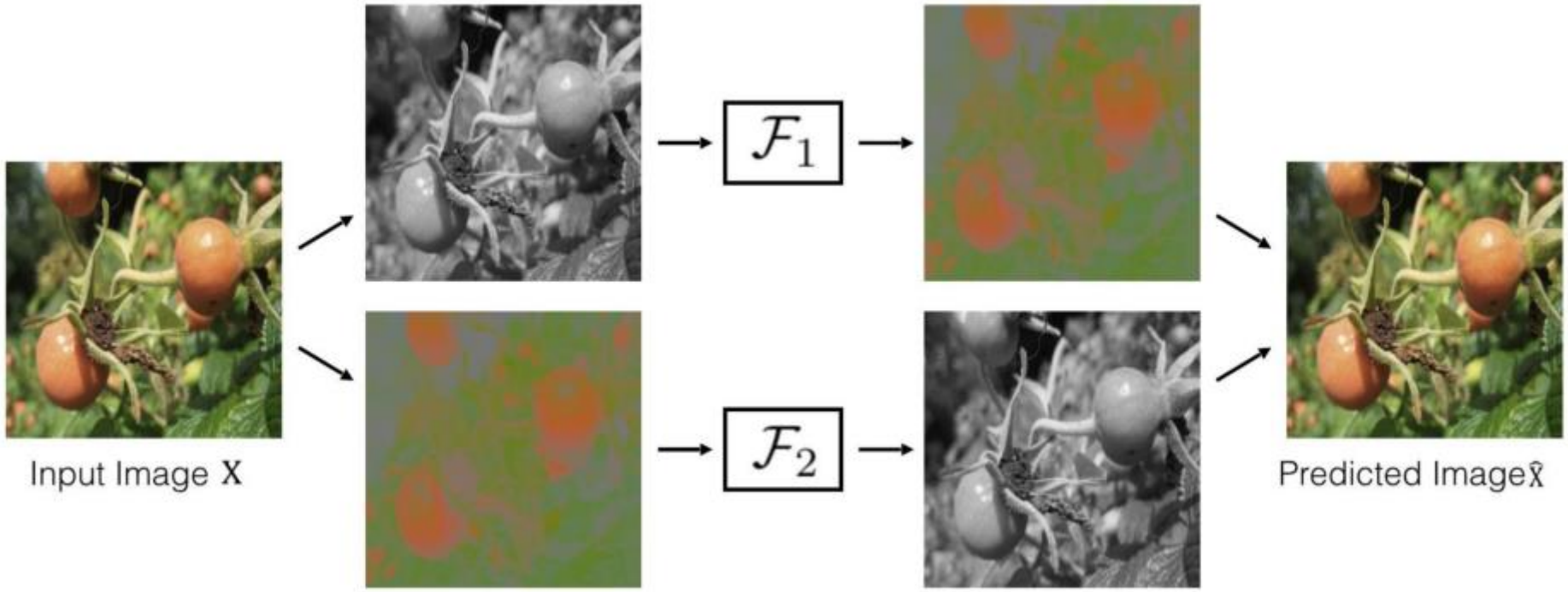


Image: Zhang et al, “Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction”,

Video Coloring

Idea: model the *temporal coherence* of colors in videos

reference frame

how should I color these frames?

t = 0

t = 1

t = 2

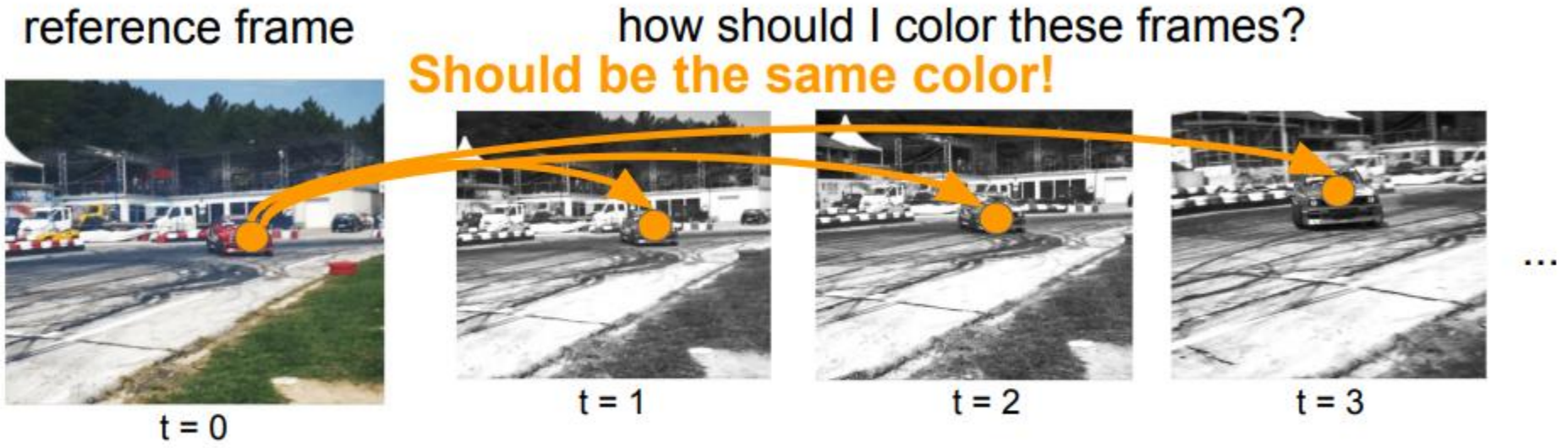
t = 3

...

Image: Vonderick et al, “Tracking Emerges by Colorizing Videos”

Video Coloring

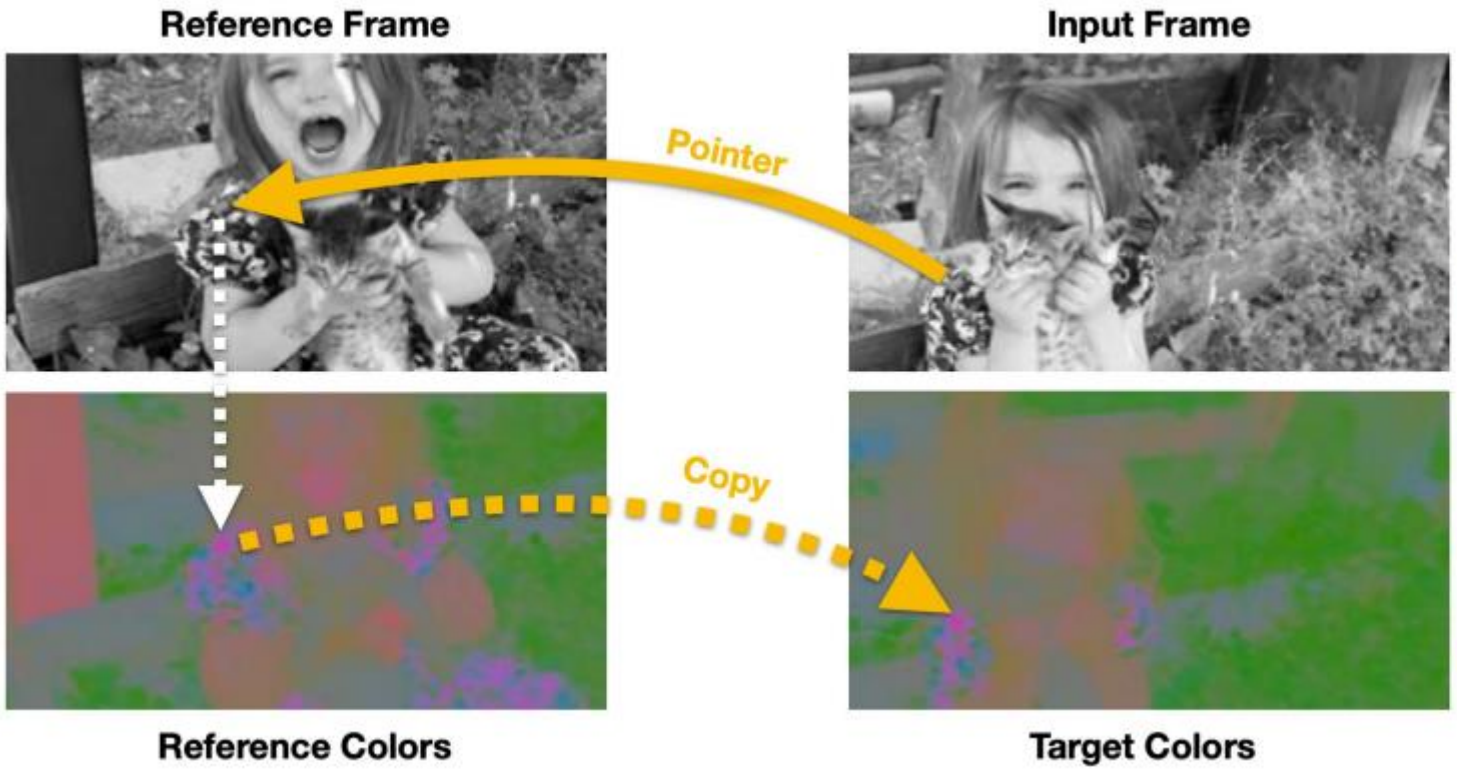
Idea: model the *temporal coherence* of colors in videos



Hypothesis: learning to color video frames should allow model to learn to track regions or objects without labels!

Image: Vonderick et al, “Tracking Emerges by Colorizing Videos”

Video Coloring



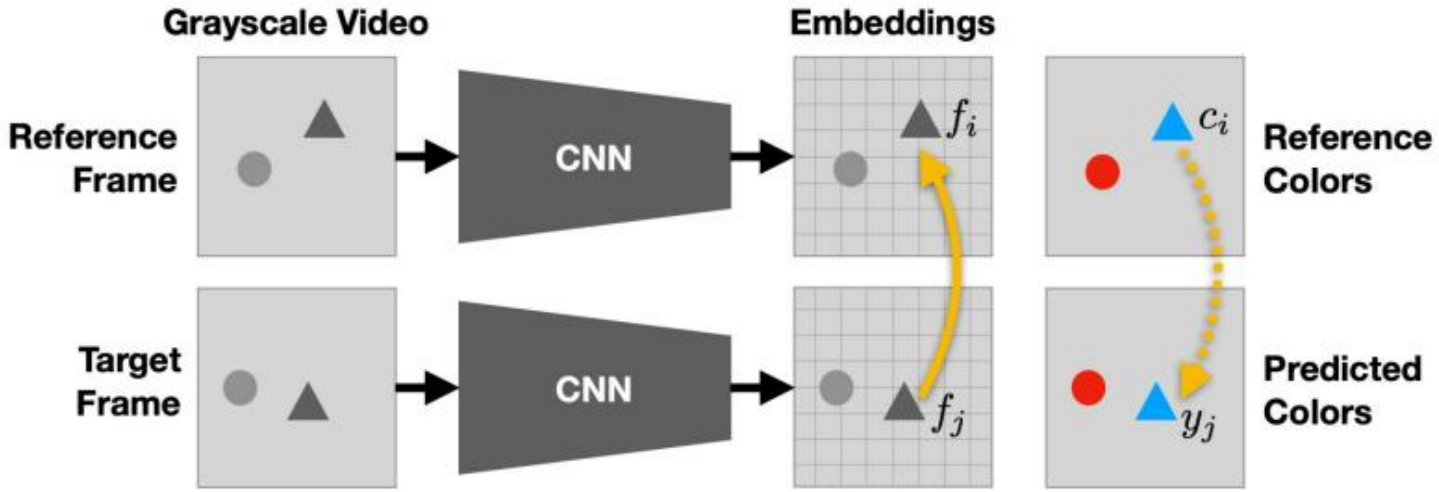
Learning objective:

Establish mappings between reference and target frames in a learned feature space.

Use the mapping as "pointers" to copy the correct color (LAB).

Image: Vonderick et al, "Tracking Emerges by Colorizing Videos"

Video Coloring



attention map on the reference frame

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

predicted color = weighted sum of the reference color

$$y_j = \sum_i A_{ij} c_i$$

loss between predicted color and ground truth color

$$\min_\theta \sum_j \mathcal{L}\left(y_j, c_j\right)$$

Image: Vonderick et al, “Tracking Emerges by Colorizing Videos”

Video Coloring

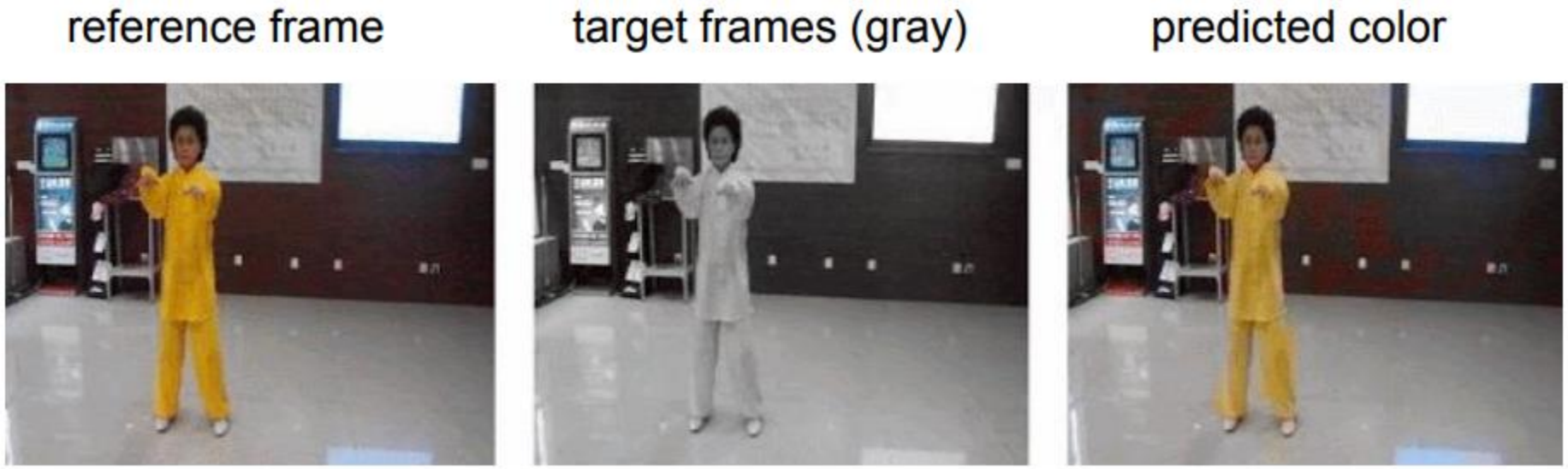


Image: Vonderick et al, “Tracking Emerges by Colorizing Videos”

Video Coloring

Propagate segmentation masks using learned attention

Image: Vonderick et al, “Tracking Emerges by Colorizing Videos”

Temporal Order Verification

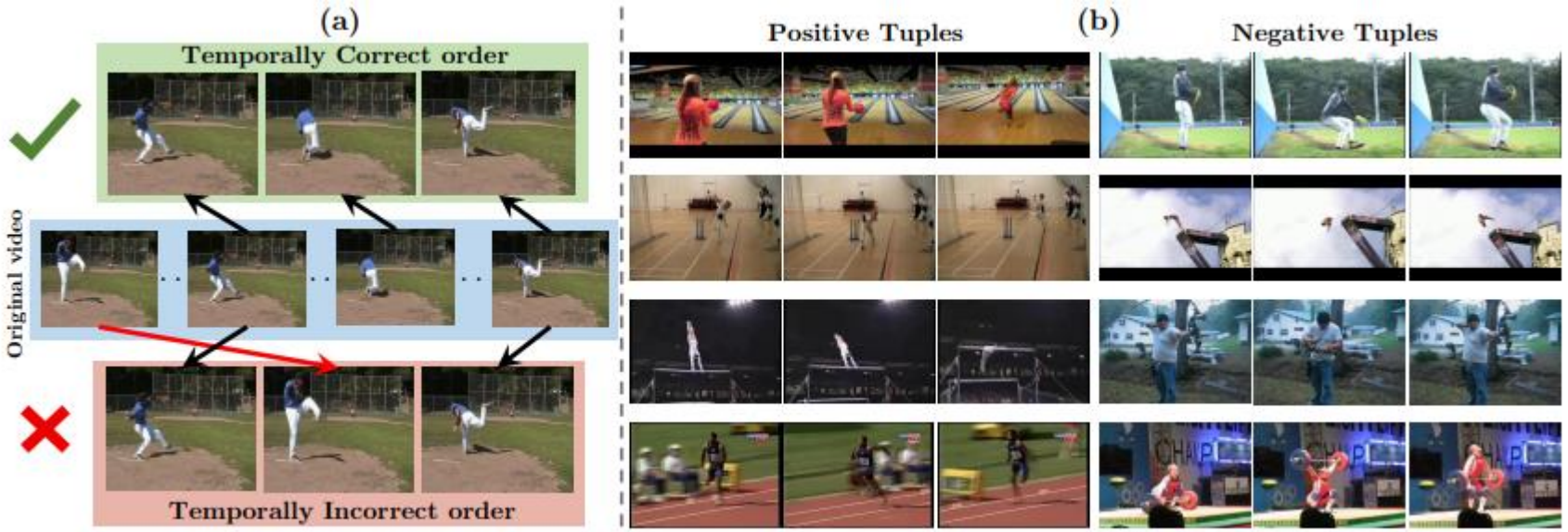


Image Misra et al, “Shuffle and Learn”

Temporal Order Verification

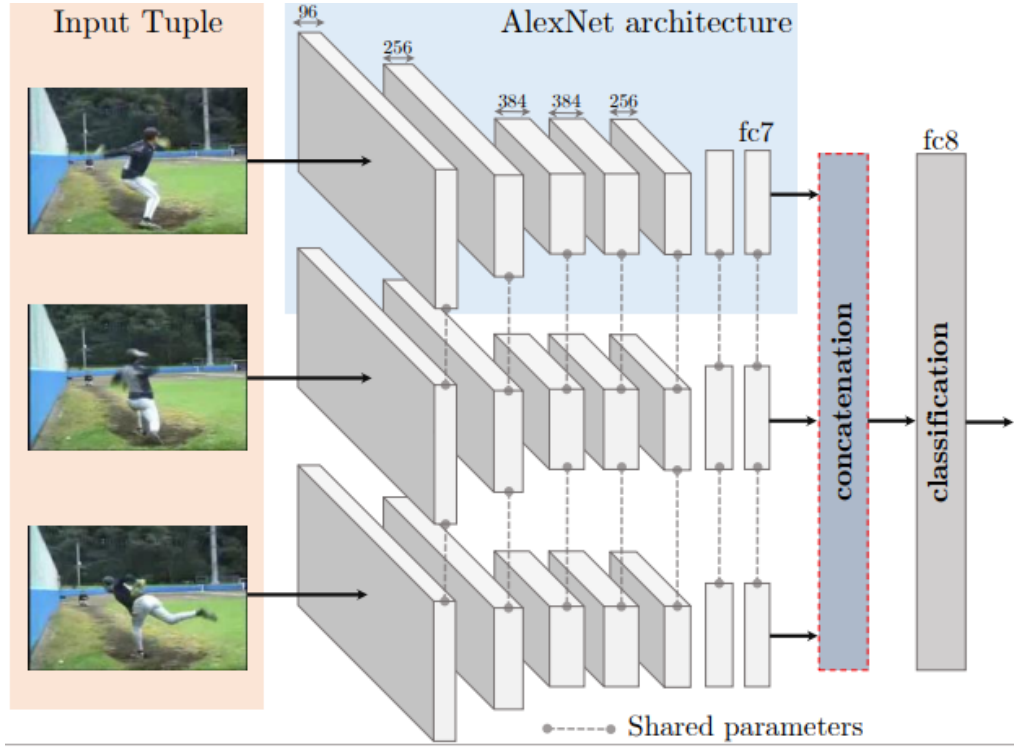


Image Misra et al, “Shuffle and Learn”

Temporal Order Verification

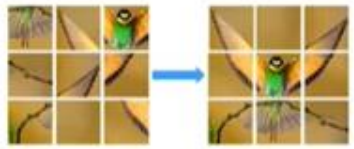
Image Misra et al, “Shuffle and Learn”

Summary: Pretext task from image transformation

- Pretext tasks focus on “visual common sense”, e.g., predict rotations, inpainting, rearrangement, and colorization.
- We don’t care about the performance of these pretext tasks, but rather how useful the learned features are for downstream tasks (classification, detection, segmentation).

Summary: Pretext task from image transformation

- Pretext tasks focus on “visual common sense”, e.g., predict rotations, inpainting, rearrangement, and colorization.
- We don’t care about the performance of these pretext tasks, but rather how useful the learned features are for downstream tasks (classification, detection, segmentation).
- Problems: 1) coming up with individual pretext tasks is tedious, and 2) the learned representations may not be general.



Pretext task

Wishing really hard

Transfer task

Summary: Pretext task from image transformation

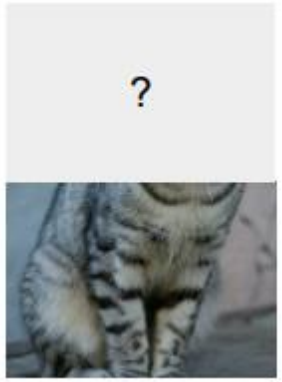
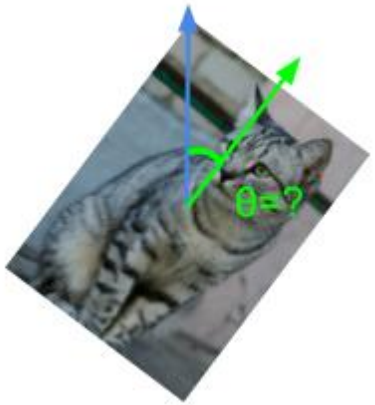


image completion



rotation prediction

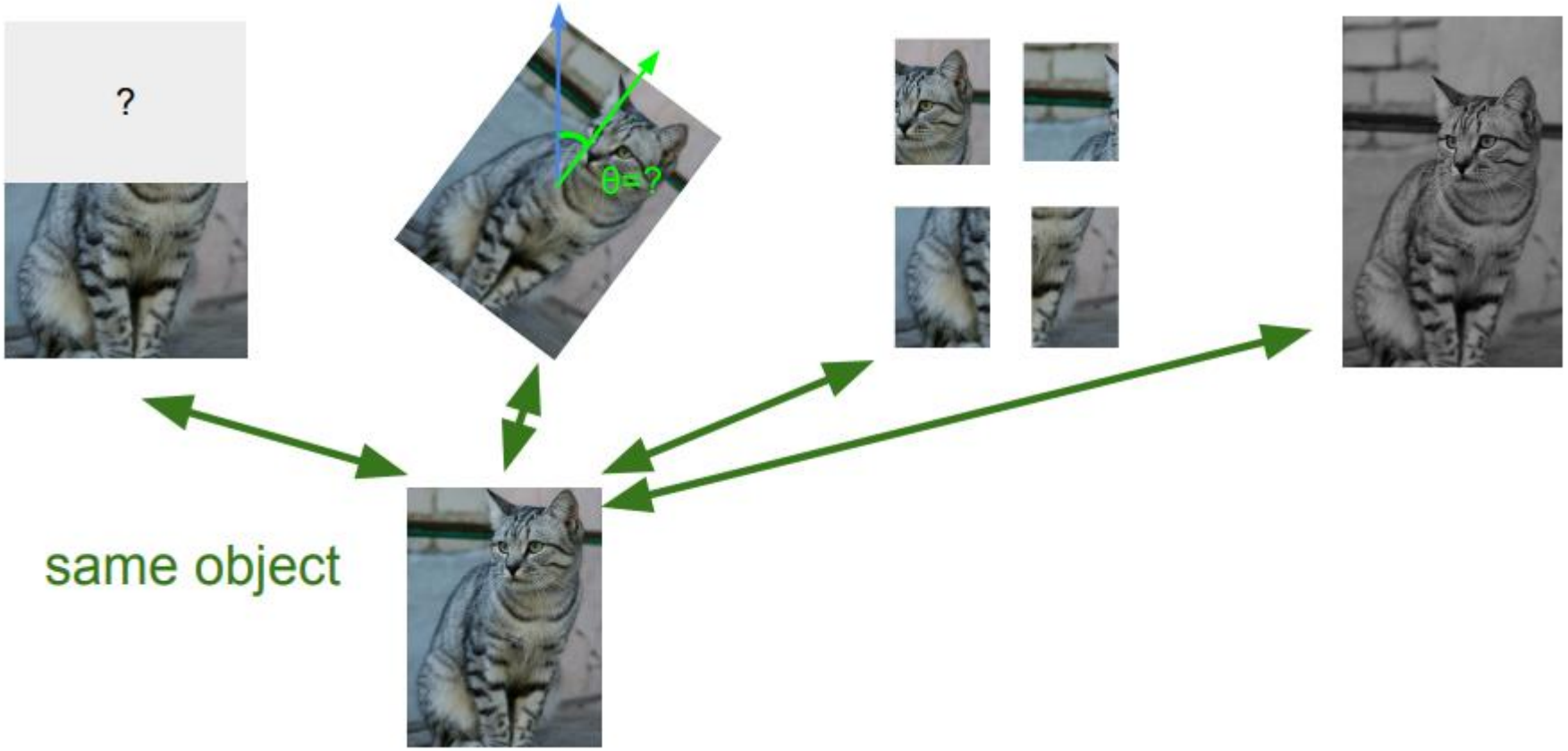
"jigsaw puzzle"

colorization

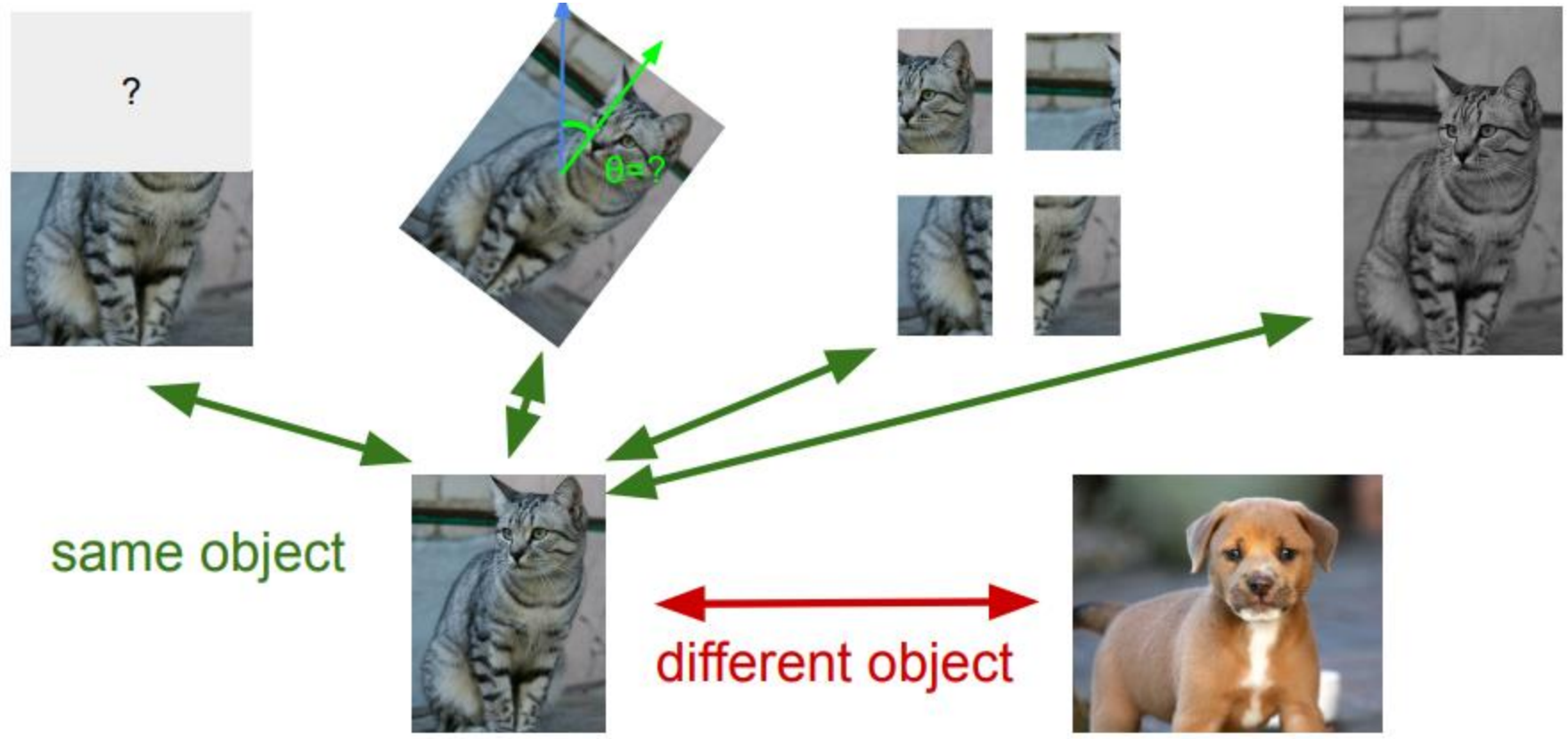
Learned representations may be tied to a specific pretext task!

Can we come up with a more general pretext task?

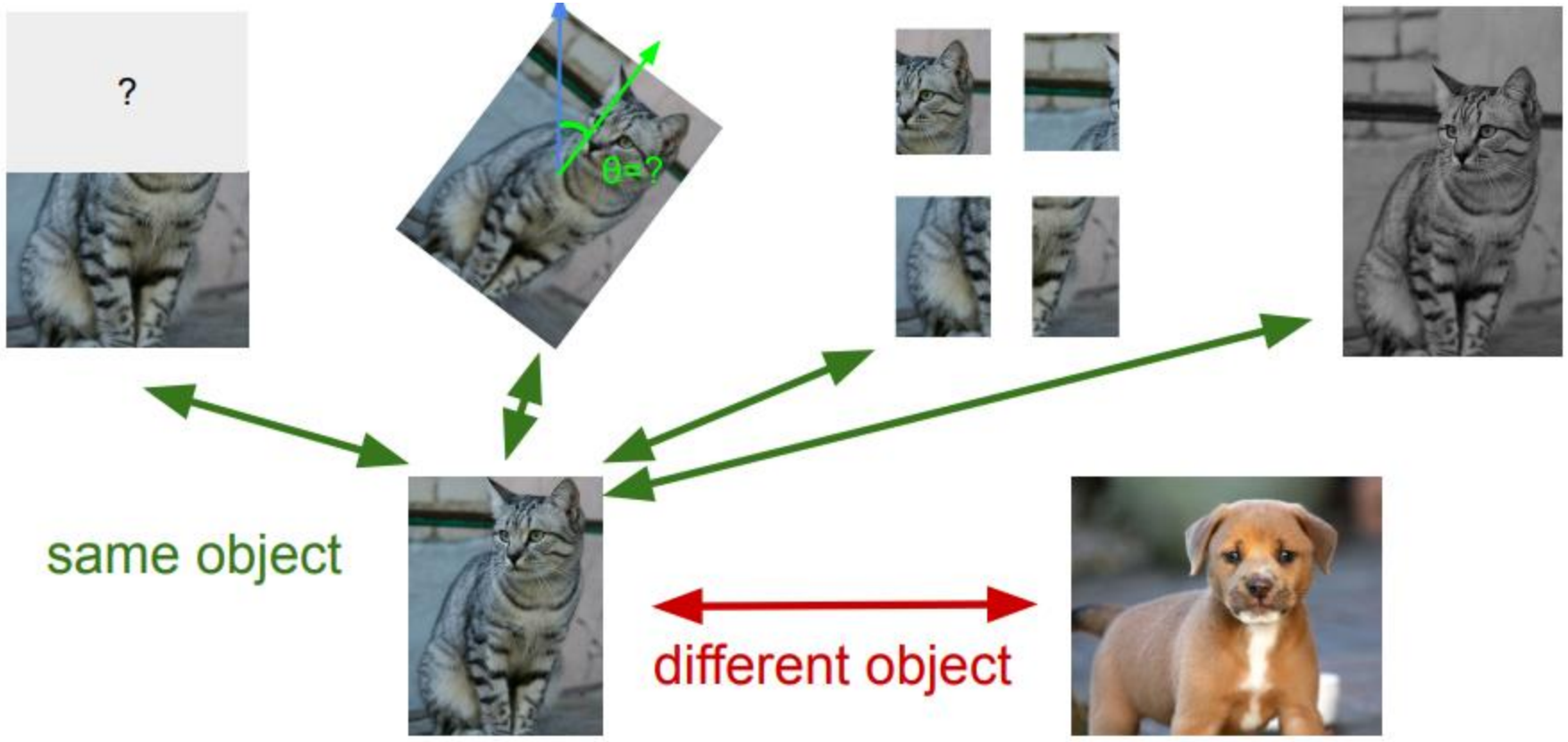
A More General Pretext task?



A More General Pretext task?



Contrastive Representation Learning



Contrastive Representation Learning

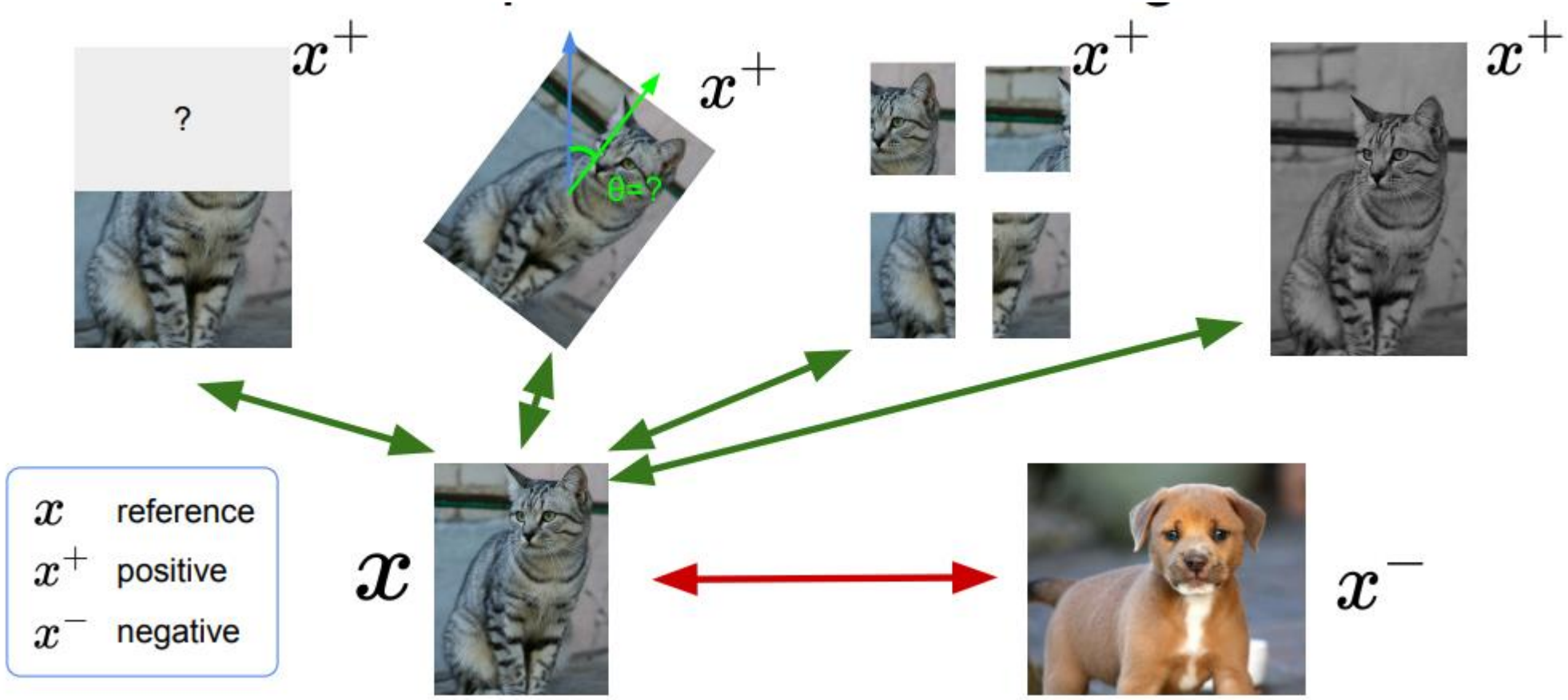
Pretext tasks from image transformations

- Rotation, inpainting, rearrangement, coloring

Contrastive representation learning

- Intuition and formulation
- Instance contrastive learning: SimCLR and MOCO

Contrastive Representation Learning



A Formulation of Contrastive Representation Learning

What we want:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

x: reference sample; x^+ positive sample; x^- negative sample

Given a chosen score function, we aim to learn an **encoder function** f that yields high score for positive pairs (x, x^+) and low scores for negative pairs (x, x^-).

A Formulation of Contrastive Representation Learning

What we want:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

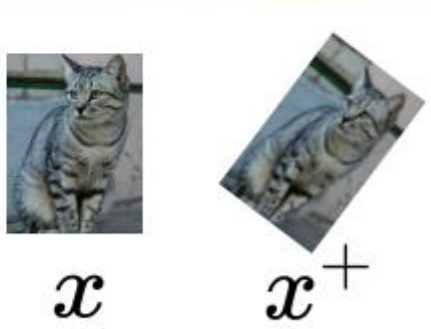
x: reference sample; x^+ positive sample; x^- negative sample

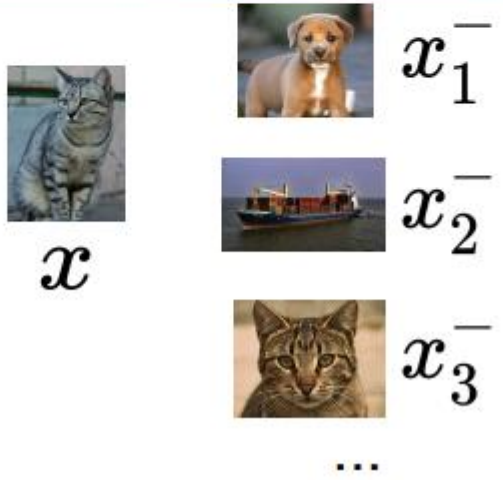
Given a chosen score function, we aim to learn an **encoder function** f that yields high score for positive pairs (x, x^+) and low scores for negative pairs (x, x^-).

A Formulation of Contrastive Representation Learning

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

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





A Formulation of Contrastive Representation Learning

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

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

score for the positive pair

score for the N-1 negative pairs

This seems familiar …

A Formulation of Contrastive Representation Learning

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

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

score for the positive pair

score for the N-1 negative pairs

This seems familiar …
Cross entropy loss for a N-way softmax classifier!
I.e., learn to find the positive sample from the N samples

SimCLR: A Simple Framework for Contrastive Learning

Cosine similarity as the score function:

$$s(u, v) = \frac{u^T v}{||u||||v||}$$

Use a projection network ***h(·)*** 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.

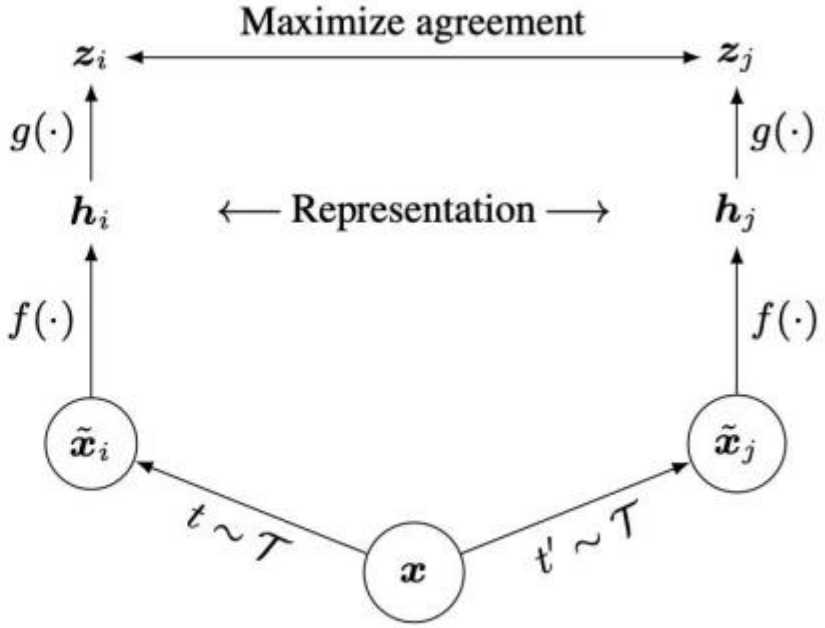


Image: Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”,

SimCLR: A Simple Framework for Contrastive Learning

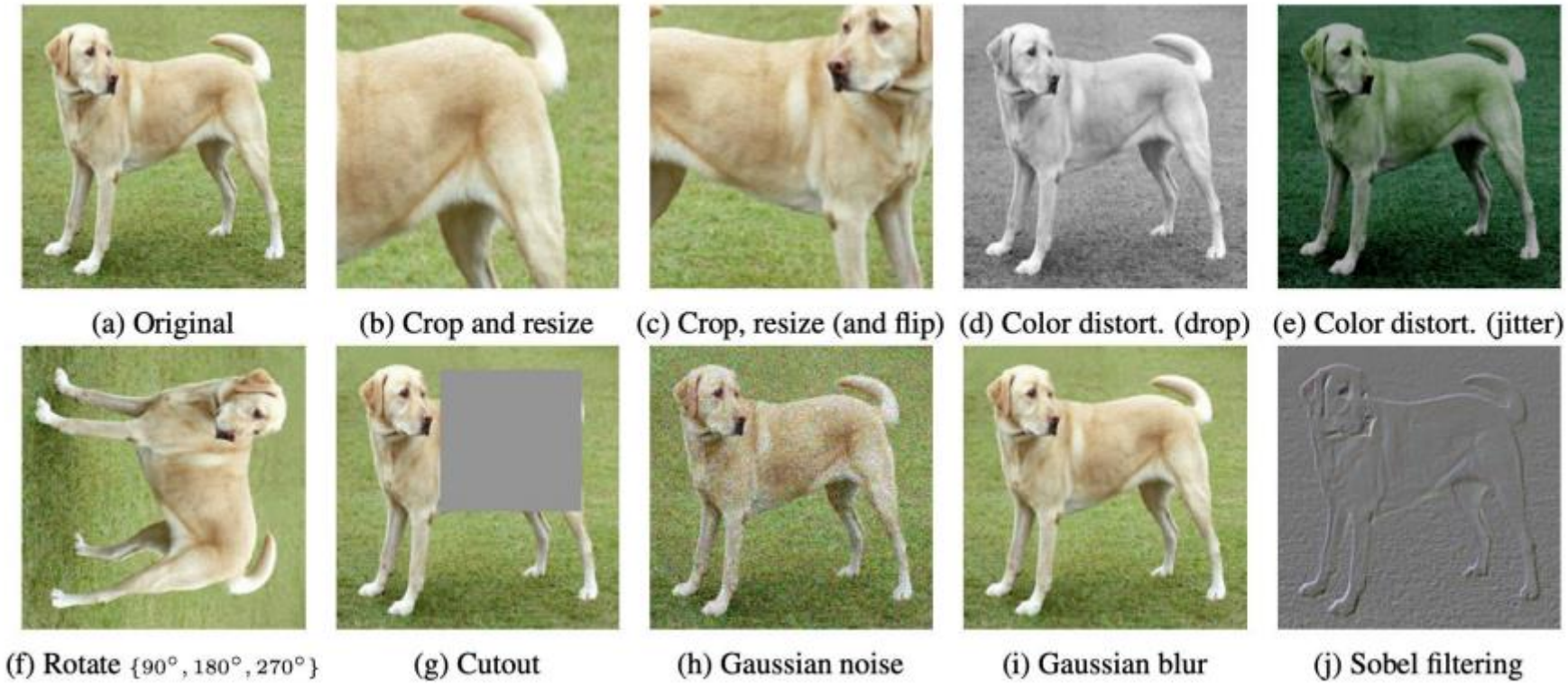


Image: Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”,

SimCLR

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 $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}, t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **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** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away

Generate a positive pair by sampling data augmentation functions

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

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

Mini Batch Training

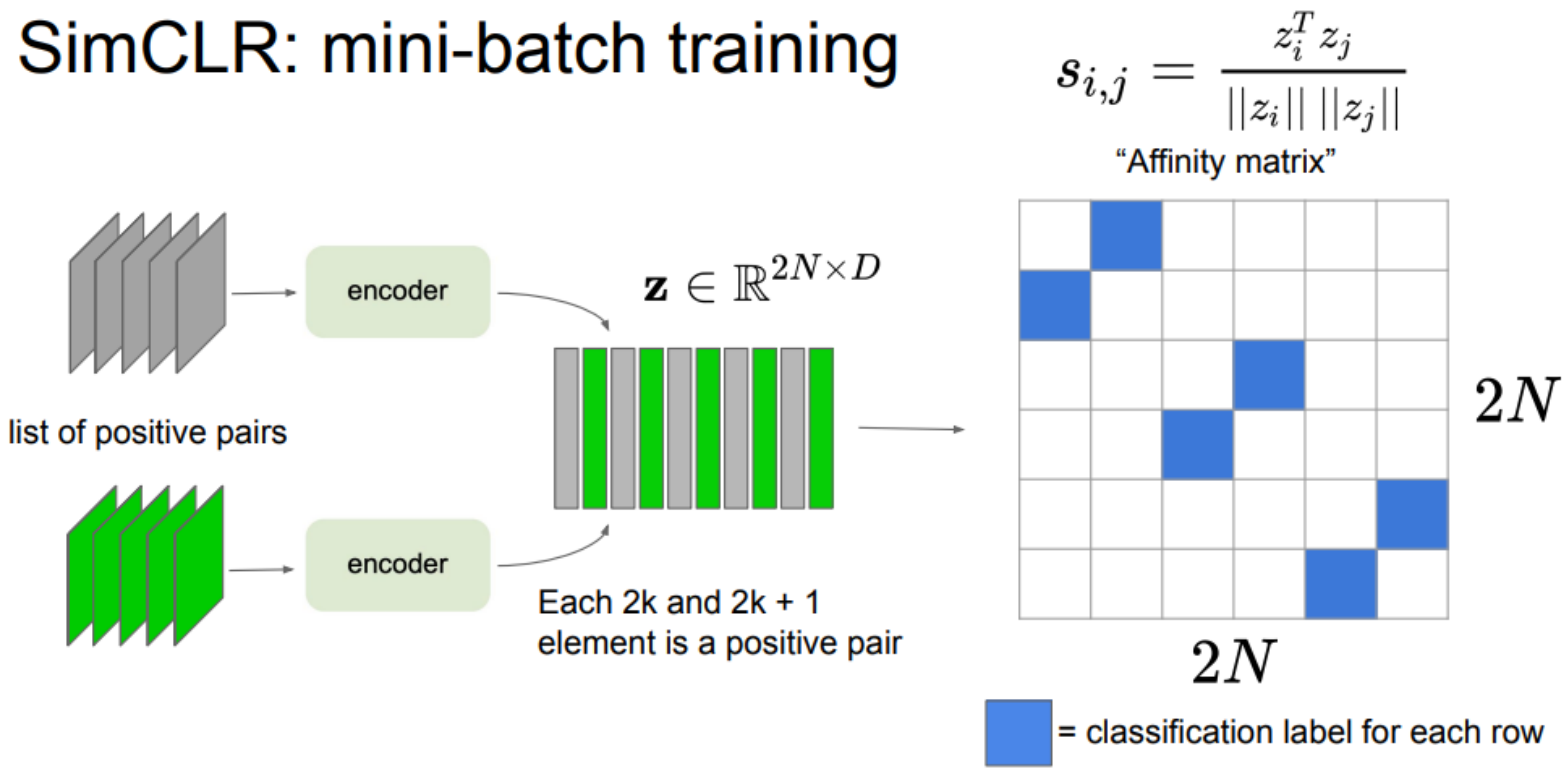
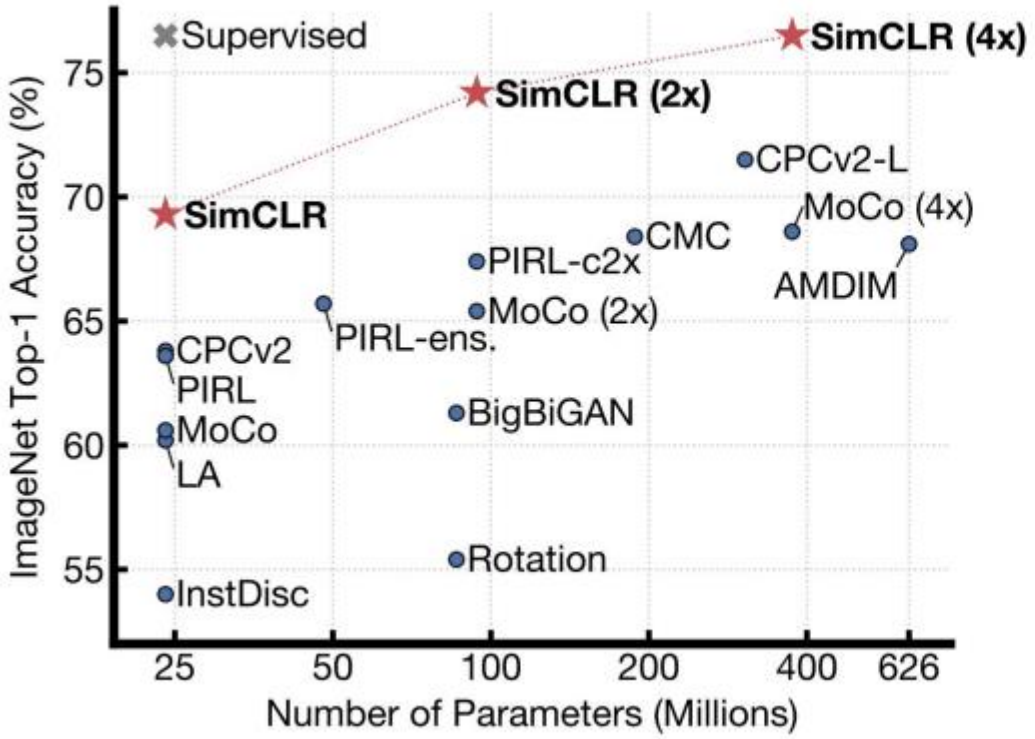


Image: Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations",

Training Linear Classifier on Extracted Features



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

Freeze feature encoder, train a linear classifier on top with labeled data.

Image: Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”,

Semi-Supervised Learning

Method	Architecture	Label fraction 1%	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**

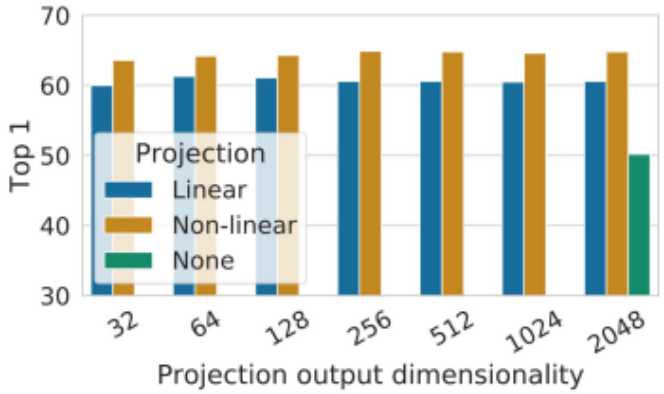
Table 7. ImageNet accuracy of models trained with few labels.

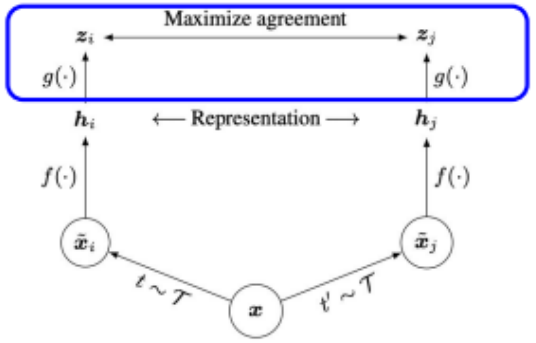
Train feature encoder on **ImageNet** (entire training set) using SimCLR.

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

Image: Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”,

Semi-Supervised Learning





Linear / non-linear projection heads improve representation learning.

Image: Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”,

Effect of Batch Size

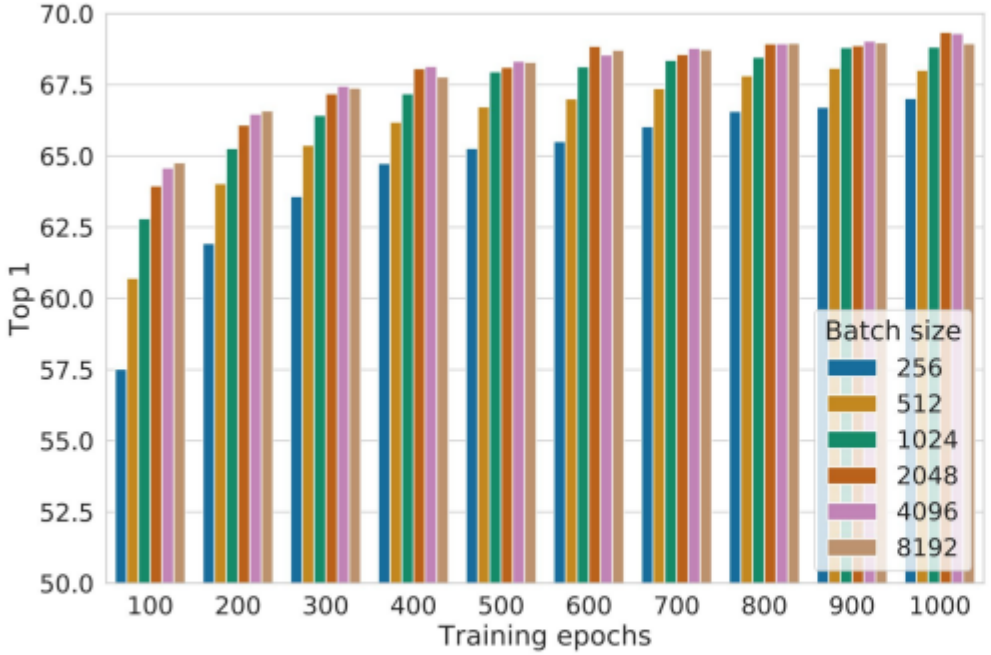


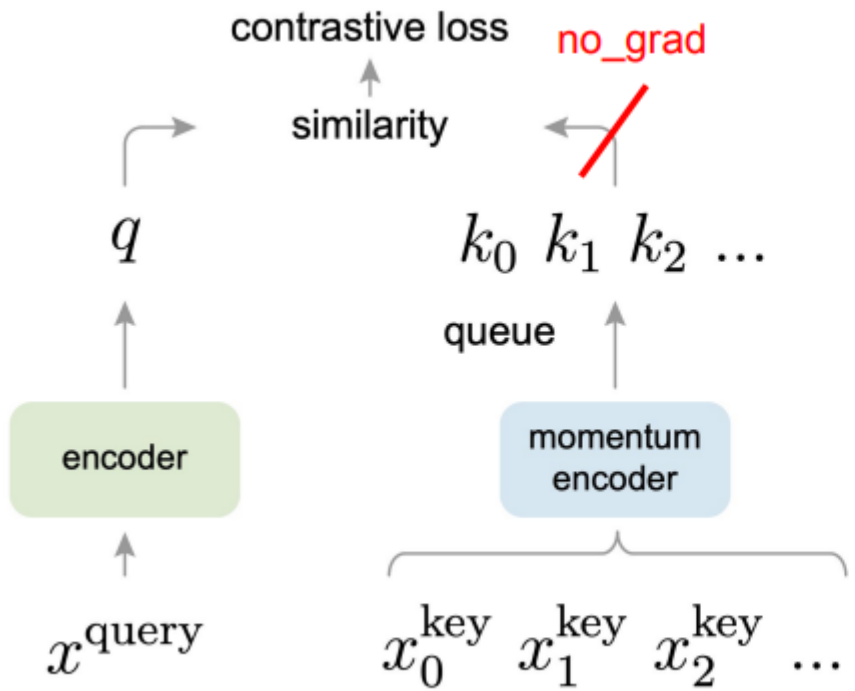
Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

Large training batch size is crucial for SimCLR!

Large batch size causes large memory footprint during backpropagation: requires distributed training on TPUs (ImageNet experiments)

Image: Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”,

Momentom Contrastive Learning (MoCo)



Key differences to SimCLR:

- Keep a running queue of keys (negative samples).
- Compute gradients and update the encoder only through the queries.
- Decouple min-batch size with the number of keys: can support a large number of negative samples.
- The key encoder is slowly progressing through the momentum update rules:

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

Image: Chen et al, “Momentum Contrast for Unsupervised Visual Representation Learning”,

MoCo

Generate a positive pair by sampling data augmentation functions

No gradient through the positive sample

Use the running queue of keys as the negative samples

InfoNCE loss

Update f_k through momentum

Update the FIFO negative sample queue

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

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.

MoCo V2

Improved Baselines with Momentum Contrastive Learning

Xinlei Chen Haoqi Fan Ross Girshick Kaiming He
Facebook AI Research (FAIR)

A hybrid of ideas from SimCLR and MoCo:

- **From SimCLR**: non-linear projection head and strong data augmentation.
- **From MoCo**: momentum-updated queues that allow training on a large number of negative samples (no TPU required!).

Image: Chen et al, “Improved Baselines with Momentum Contrastive Learning”,

MoCo V2

case	unsup. pre-train MLP	aug+	cos	epochs	batch	ImageNet acc.
MoCo v1 [6]				200	256	60.6
SimCLR [2]	✓	✓	✓	200	256	61.9
SimCLR [2]	✓	✓	✓	200	8192	66.6
MoCo v2	✓	✓	✓	200	256	**67.5**
*results of **longer** unsupervised training follow:*						
SimCLR [2]	✓	✓	✓	1000	4096	69.3
MoCo v2	✓	✓	✓	800	256	**71.1**

Table 2. **MoCo *vs.* SimCLR**: ImageNet linear classifier accuracy (**ResNet-50, 1-crop 224×224**), trained on features from unsupervised pre-training. "aug+" in SimCLR includes blur and stronger color distortion. SimCLR ablations are from Fig. 9 in [2] (we thank the authors for providing the numerical results).

Key takeaways:

- Non-linear projection head and strong data augmentation are crucial for contrastive learning.
- Decoupling mini-batch size with negative sample size allows MoCo-V2 to outperform SimCLR with smaller batch size (256 vs. 8192).

Image: Chen et al, "Improved Baselines with Momentum Contrastive Learning",

MoCo V2

mechanism	batch	memory / GPU	time / 200-ep.
MoCo	256	**5.0G**	**53 hrs**
end-to-end	256	7.4G	65 hrs
end-to-end	4096	93.0G†	n/a

Table 3. **Memory and time cost** in 8 V100 16G GPUs, implemented in PyTorch. †: based on our estimation.

Key takeaways:

- Non-linear projection head and strong data augmentation are crucial for contrastive learning.
- Decoupling mini-batch size with negative sample size allows MoCo-V2 to outperform SimCLR with smaller batch size (256 vs. 8192).
- … all with much smaller memory footprint! ("end-to-end" means SimCLR here)

Image: Chen et al, "Improved Baselines with Momentum Contrastive Learning",

Contrastive learning Between Image and Natural Language

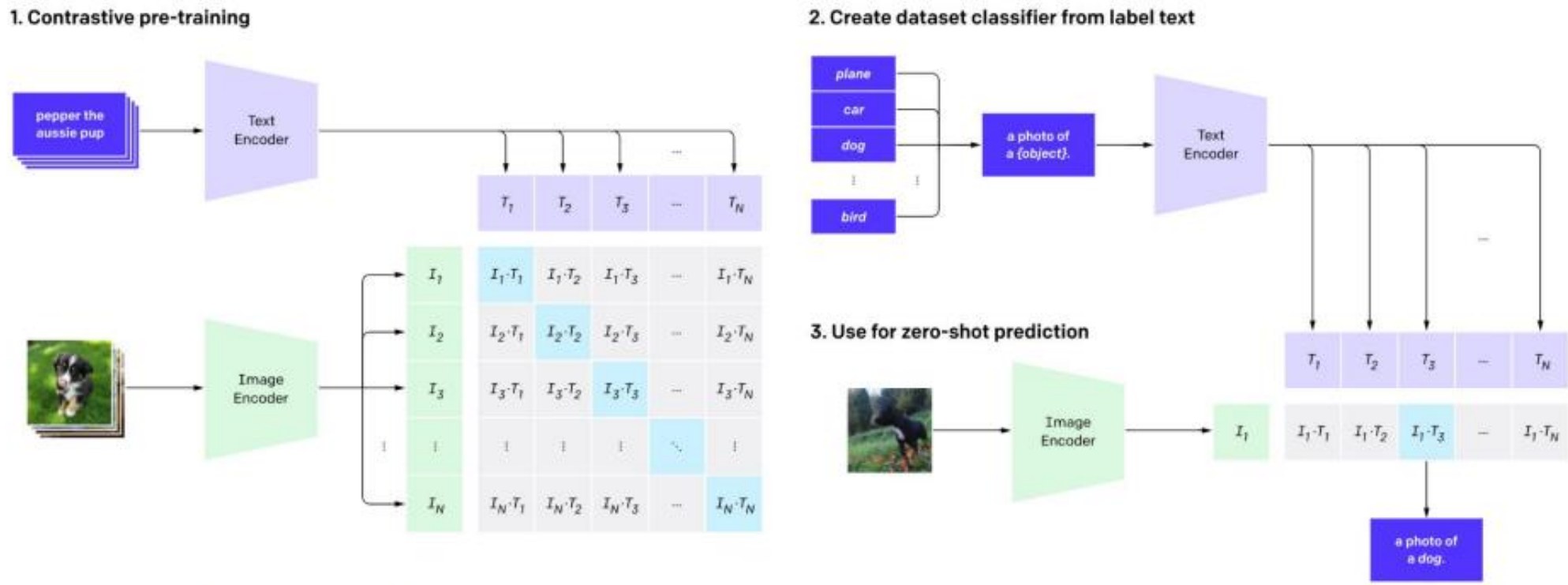


Image: Radford et al, “Learning Transferable Visual Models form Natural Language Supervision”,

Masked Auto Encoder

A ~~new~~ old method dethrones contrastive learning? Denoising Autoencoder with Vision Transformer

Image: He et al, “Masked Autoencoders are Scalable Vision Learners”,

Masked Auto Encoder

A ~~new~~ old method dethrones contrastive learning? Denoising Autoencoder with Vision Transformer

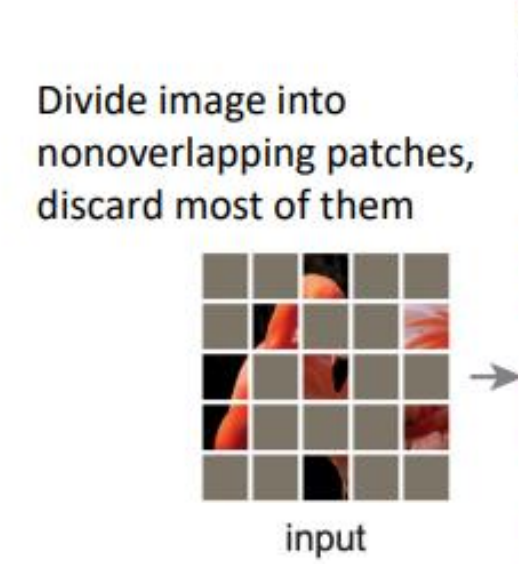


Image: He et al, “Masked Autoencoders are Scalable Vision Learners”,

Masked Auto Encoder

A ~~new~~ old method dethrones contrastive learning? Denoising Autoencoder with Vision Transformer

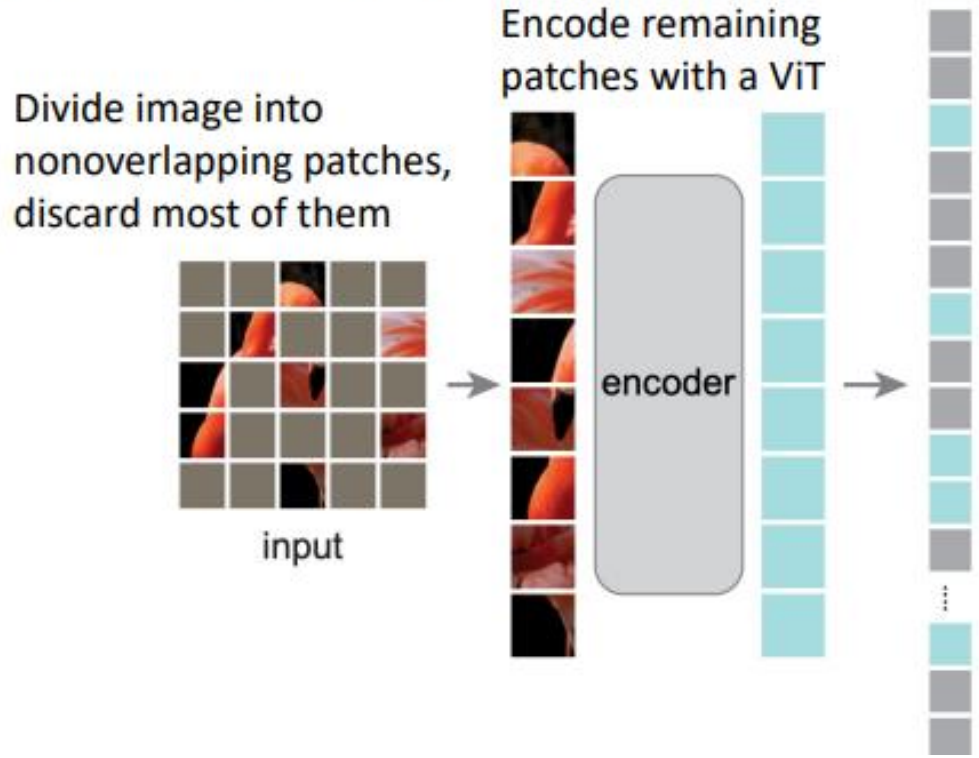


Image: He et al, “Masked Autoencoders are Scalable Vision Learners”,

Masked Auto Encoder

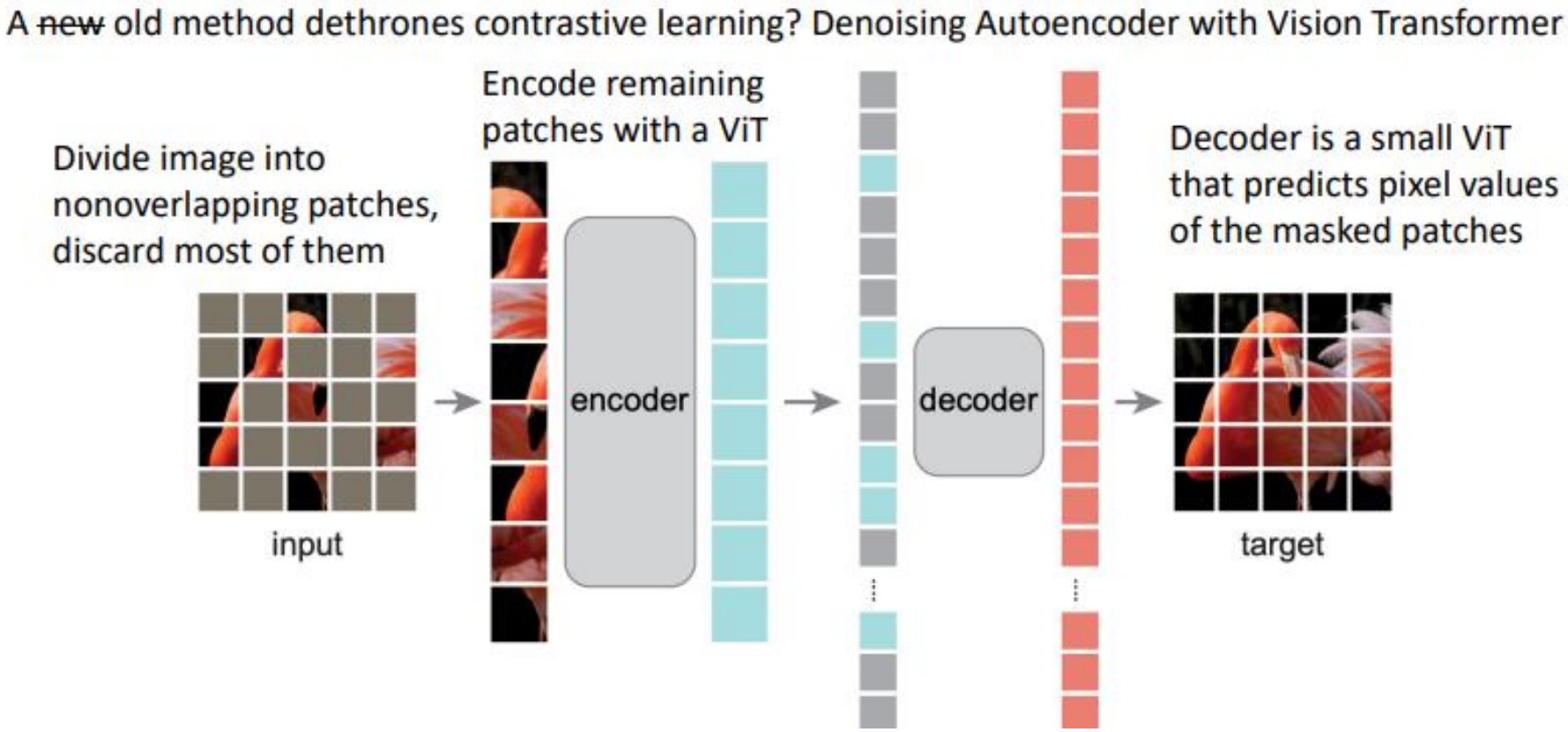


Image: He et al, “Masked Autoencoders are Scalable Vision Learners”,

Masked Auto Encoder

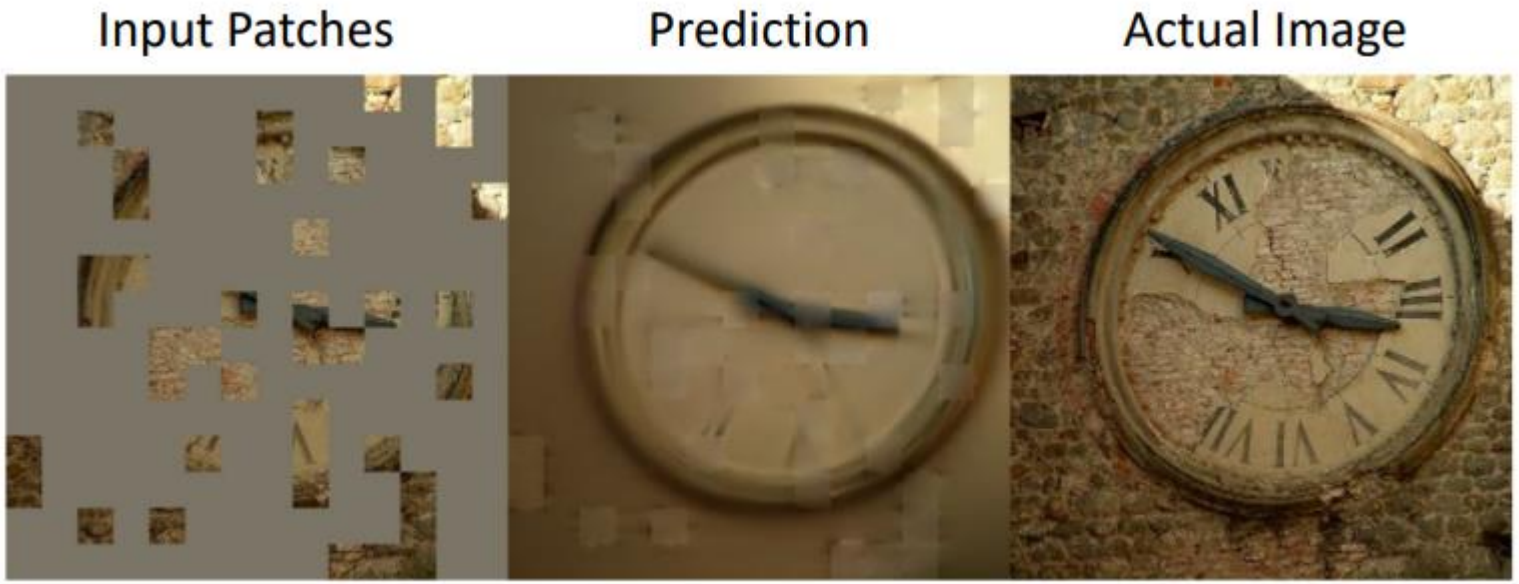


Image: He et al, “Masked Autoencoders are Scalable Vision Learners”,

Masked Auto Encoder

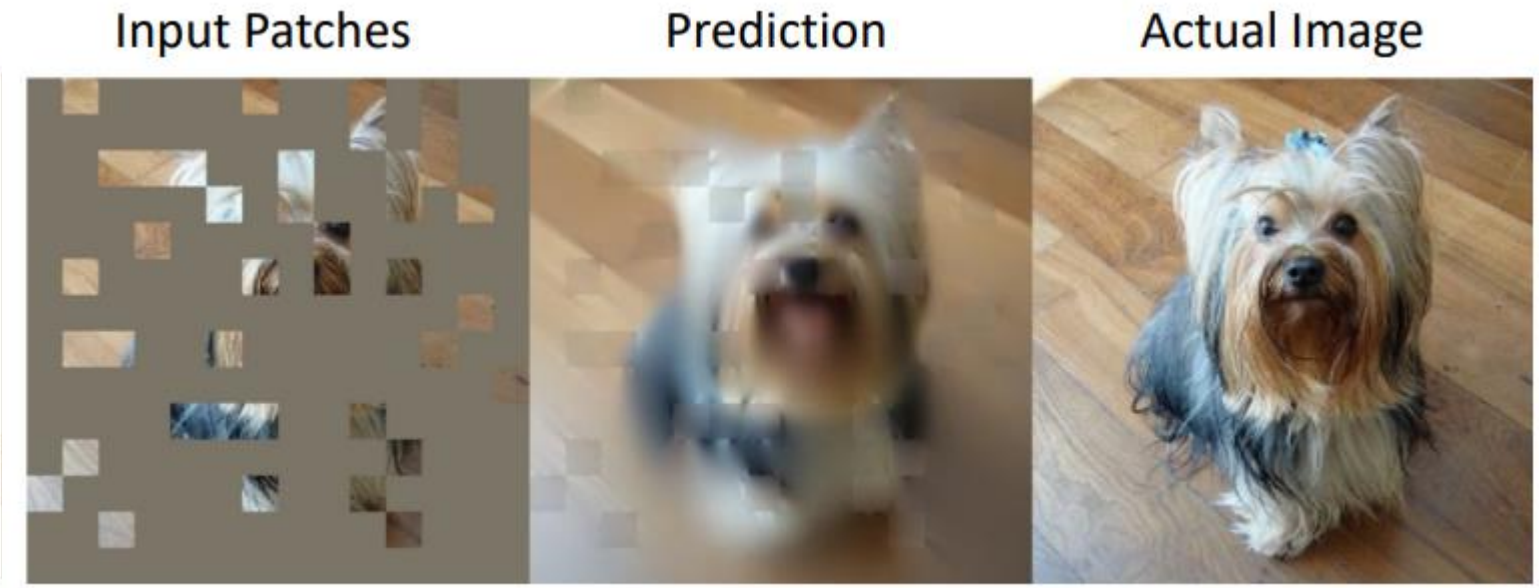


Image: He et al, “Masked Autoencoders are Scalable Vision Learners”,

DINO

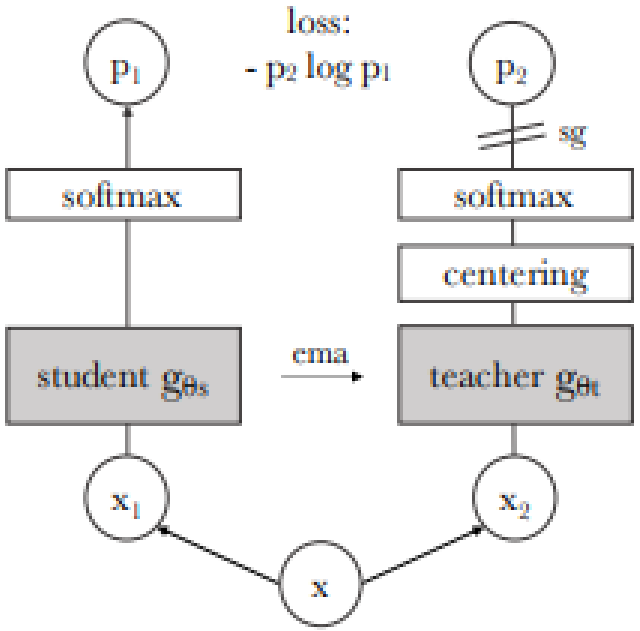


Image: Caron et al, “Emerging Properties in Self-Supervised Vision Transformers”,

DINO

Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

Image: Caron et al, “Emerging Properties in Self-Supervised Vision Transformers”,

DINO

Figure 1: **Self-attention from a Vision Transformer with 8×8 patches trained with no supervision.** We look at the self-attention of the [CLS] token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations.

Image: Caron et al, “Emerging Properties in Self-Supervised Vision Transformers”,

Summary

- Self-Supervised Learning (SSL) aims to scale up to larger datasets without human annotation
- First train for a pretext task, then transfer to downstream tasks
- Many pretext tasks: context prediction, jigsaw, colorization, clustering, rotation
- SSL has been wildly successful for language
- Intense research on SSL in vision; current best are contrastive, masked autoencoding

Recommended Reading

Bootstrap your own latent: A new approach to self-supervised Learning:

https://arxiv.org/abs/2006.07733

DINOv2: Learning Robust Visual Features without Supervision:

https://arxiv.org/abs/2304.07193

Barlow Twins: Self-Supervised Learning via Redundancy Reduction

https://arxiv.org/abs/2103.03230

Deep Clustering for Unsupervised Learning of Visual Features:

https://arxiv.org/abs/1807.05520

Recommended Reading

Unsupervised Learning of Visual Features by Contrasting Cluster Assignments:

https://arxiv.org/abs/2006.09882

Understanding Dimensional Collapse in Contrastive Self-supervised Learning:

https://arxiv.org/abs/2110.09348

Self-supervised learning: The dark matter of intelligence:

https://ai.meta.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

Thanks for Your Time

Any Questions?