Self-Supervision

Jan 31st , 2024

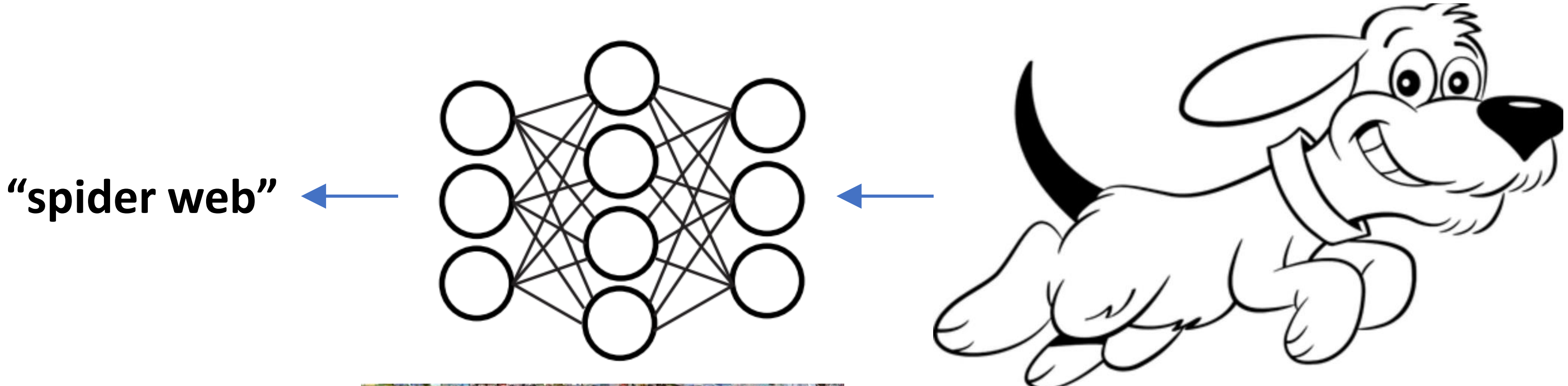
Tali Dekel

Supervised Learning

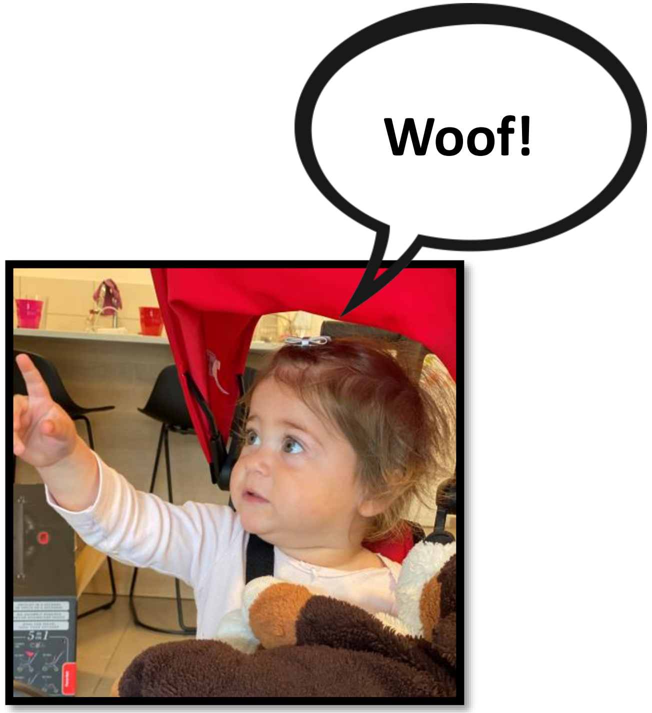
1. It’s impossible to label everything in the world

2. Not enough labeled data

3. More intelligent models wouldn’t need massive labeled data



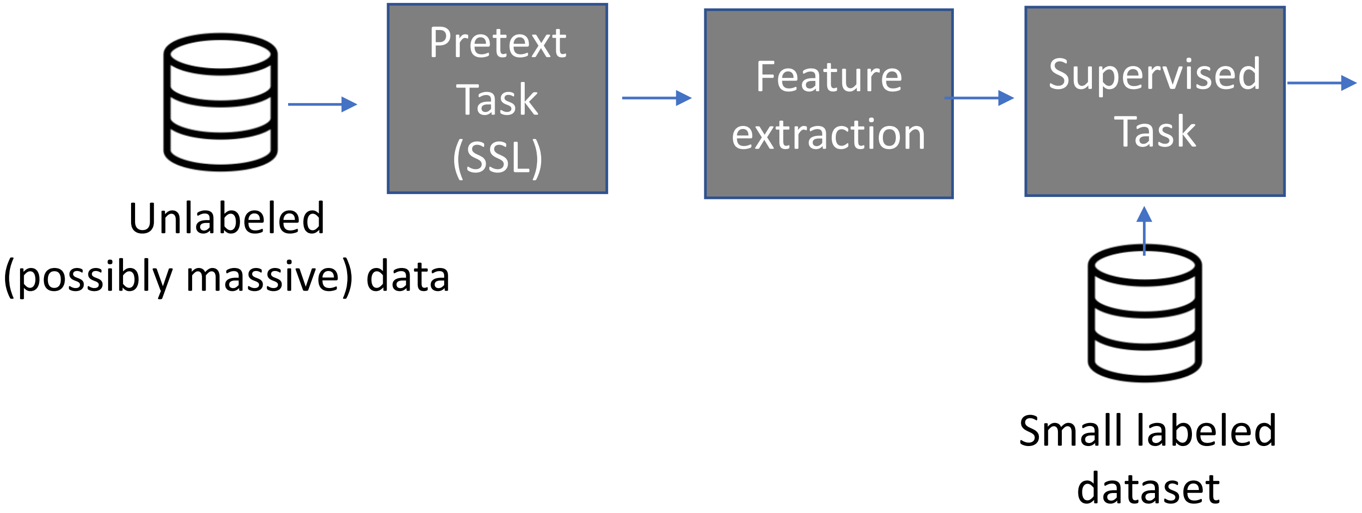
“spider web”



Self-Supervised Learning (SSL)

No human labels; supervisory signals are automatically computed from data

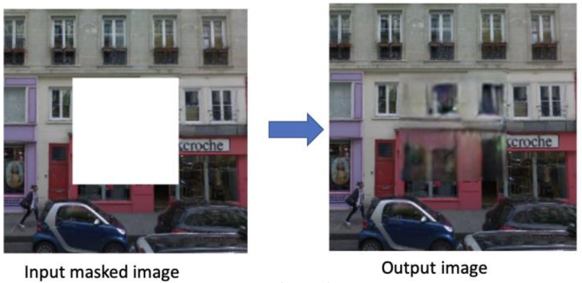
Solve a proxy, pretext task → extract learned features → finetune on a target supervised task (Transfer Learning)



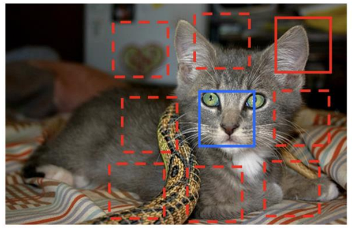
Task-Specific Models → Foundation Models

Self-Supervised Learning

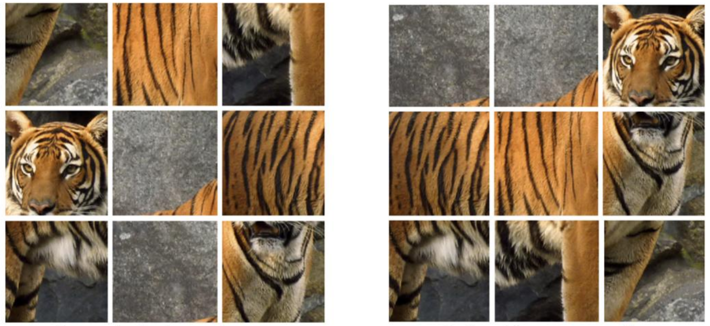
Solve a proxy, pretext task (large dataset) → extract learned features → finetune on a target supervised task (smaller dataset)



Inpainting

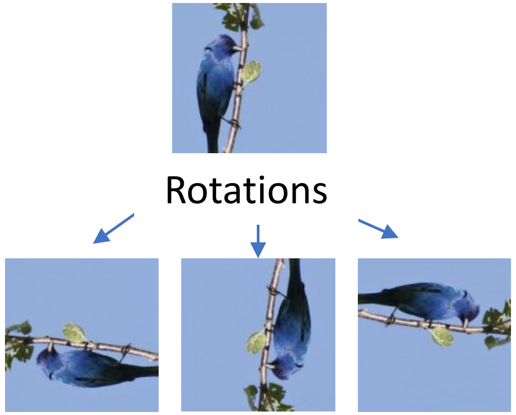


Relative patch location

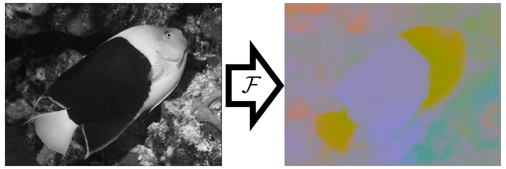


Patch jigsaw puzzel

Image context as supervision



Geometric transformations



grayscale

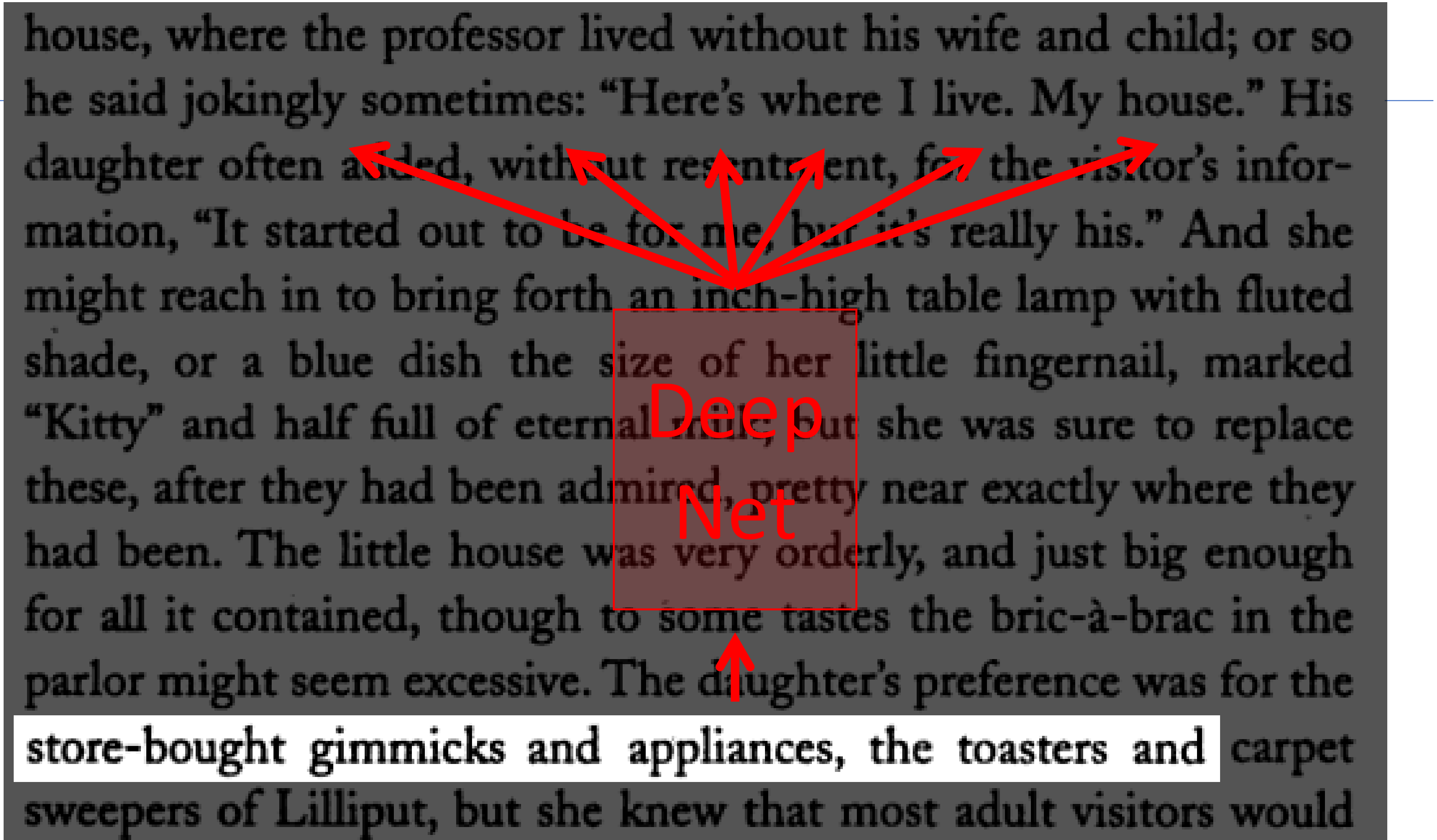
Color channles

Color transformations

slide credit: http://graphics.cs.cmu.edu/projects/deepContext/

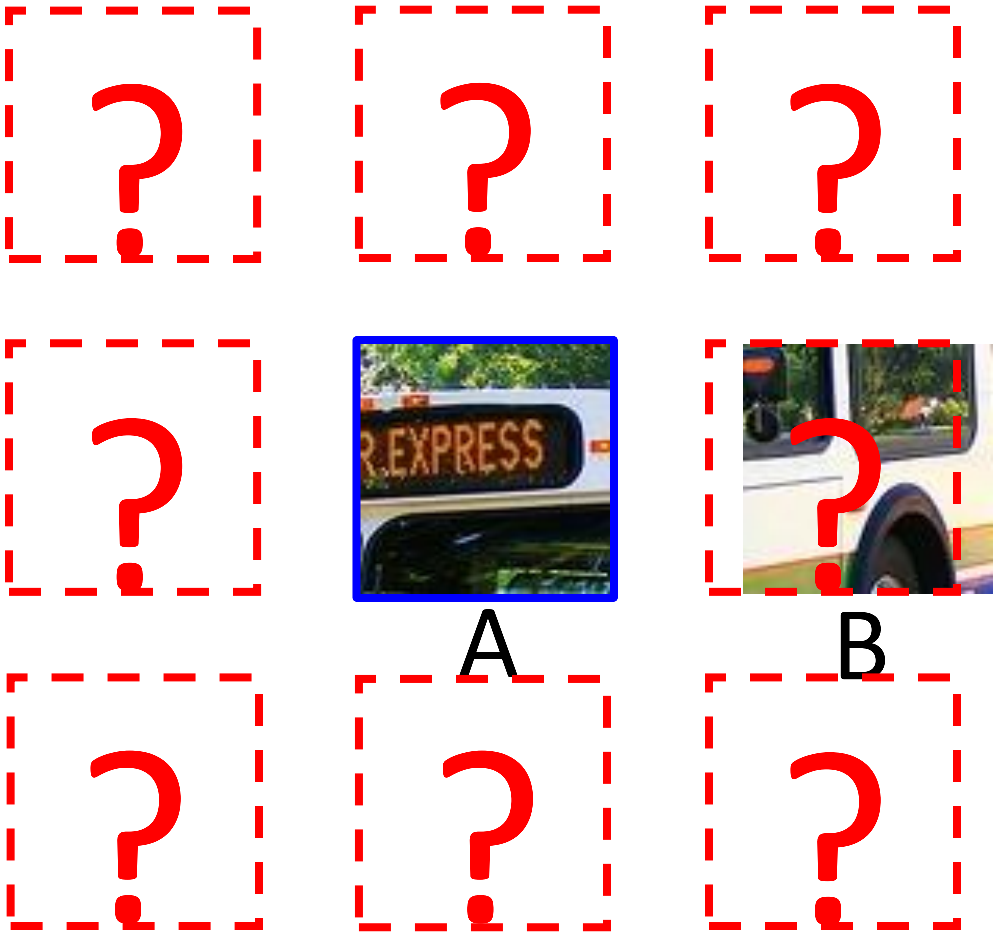
Context as Supervision

[Collobert & Weston 2008; Word2Vec by Mikolov et al. 2013]



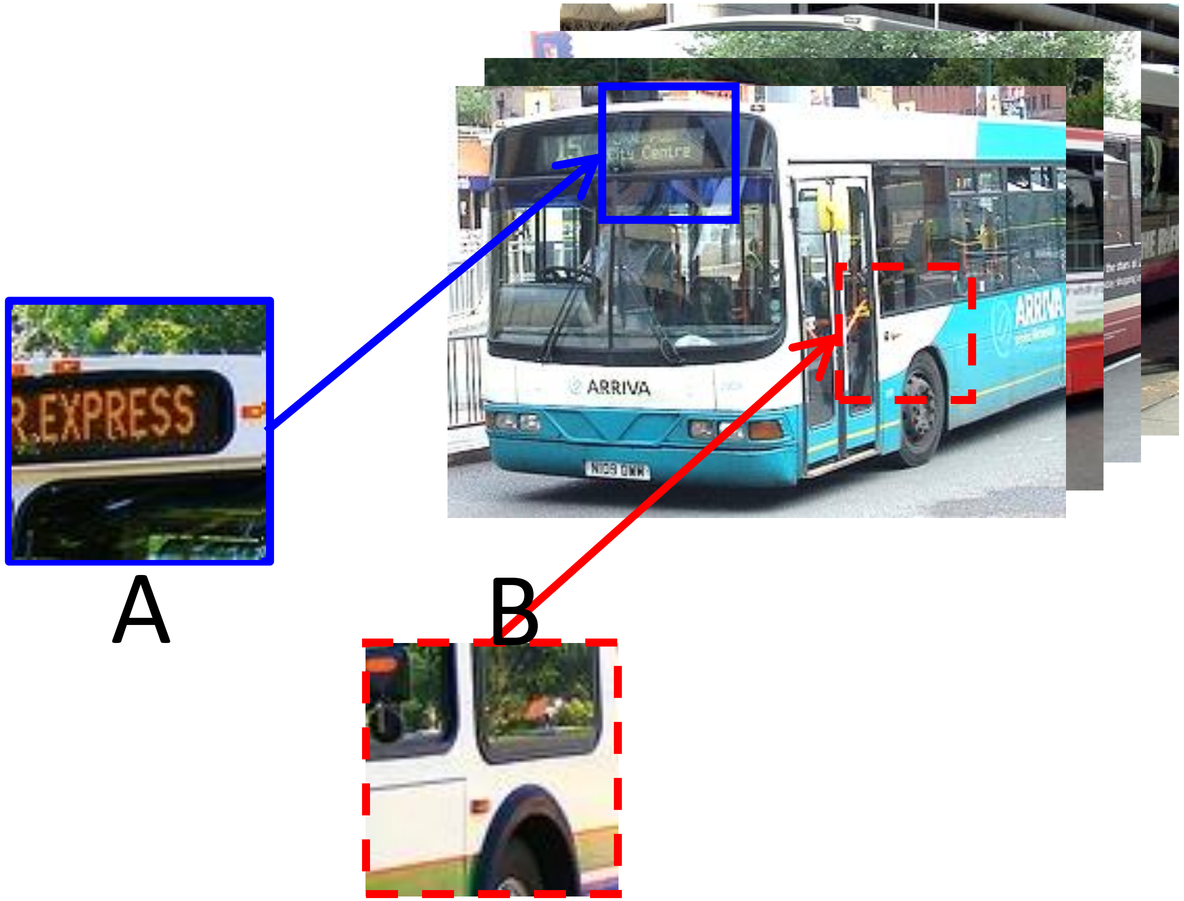
house, where the professor lived without his wife and child; or so he said jokingly sometimes: “Here’s where I live. My house.” His daughter often added, without resentment, for the visitor’s information, “It started out to be for me, but it’s really his.” And she might reach in to bring forth an inch-high table lamp with fluted shade, or a blue dish the size of her little fingernail, marked “Kitty” and half full of eternal milk, but she was sure to replace these, after they had been admired, pretty near exactly where they had been. The little house was very orderly, and just big enough for all it contained, though to some tastes the bric-à-brac in the parlor might seem excessive. The daughter’s preference was for the store-bought gimmicks and appliances, the toasters and carpet sweepers of Lilliput, but she knew that most adult visitors would

Context as Supervision: relative patch position



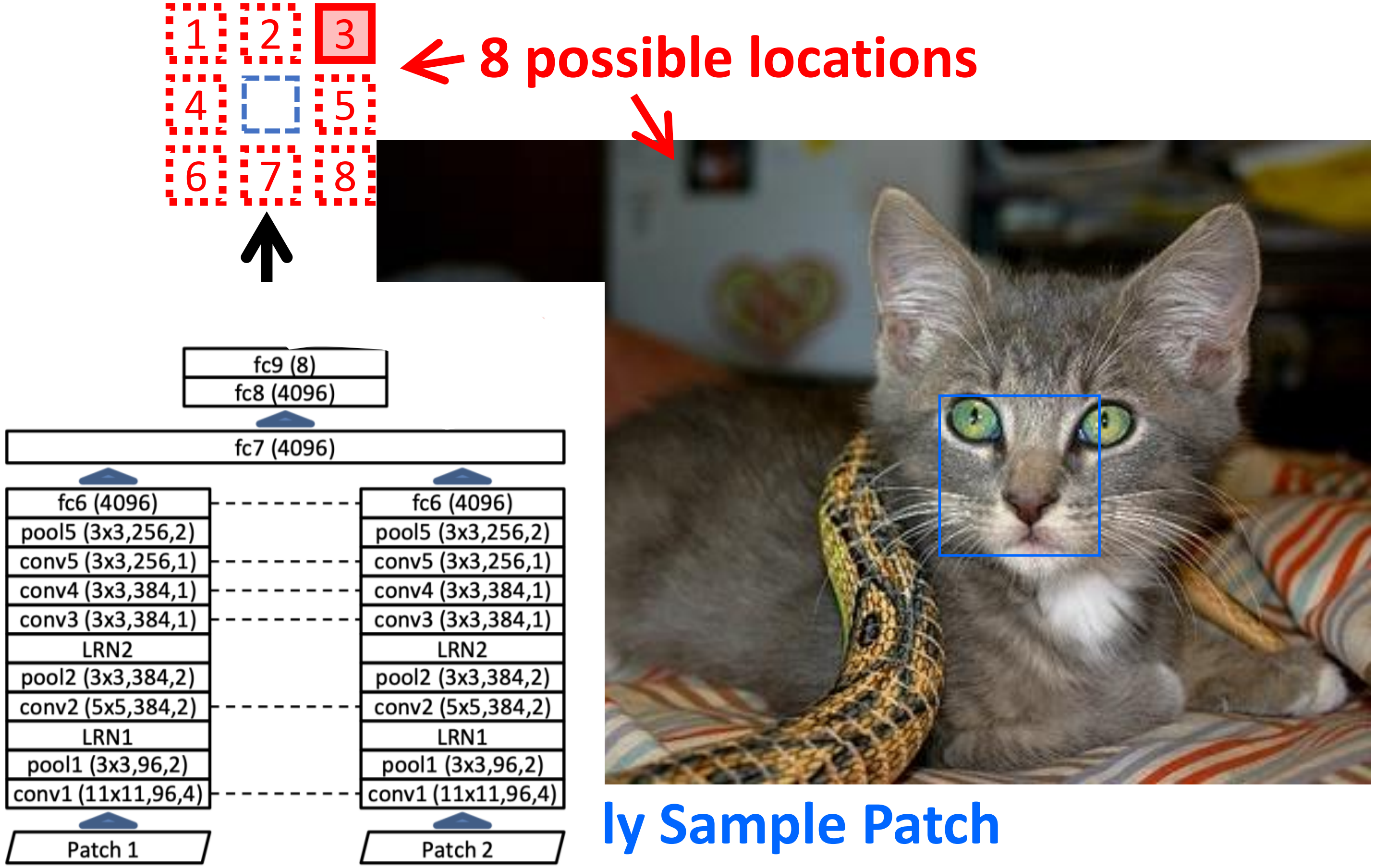
Doersch et. al, Unsupervised Visual Representation Learning by Context Prediction, ICCV 2015

Semantics from a non-semantic task



Context as Supervision: relative patch position

Avoid “cheats” (low-level “trivial solutions”) → gaps between patches + random jitter



8 possible locations

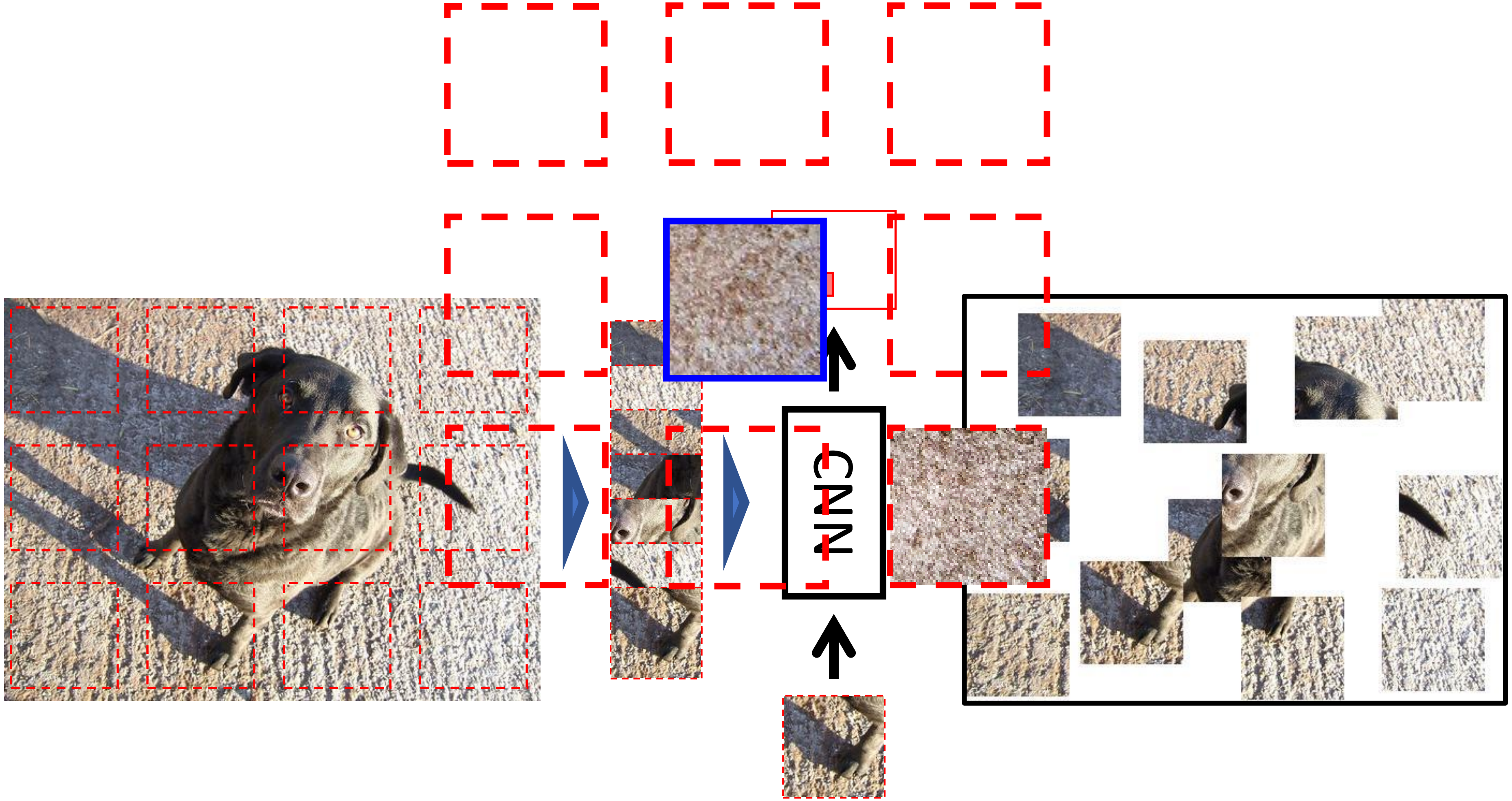
Patch Embeddings →

ly Sample Patch

Sample Second Patch

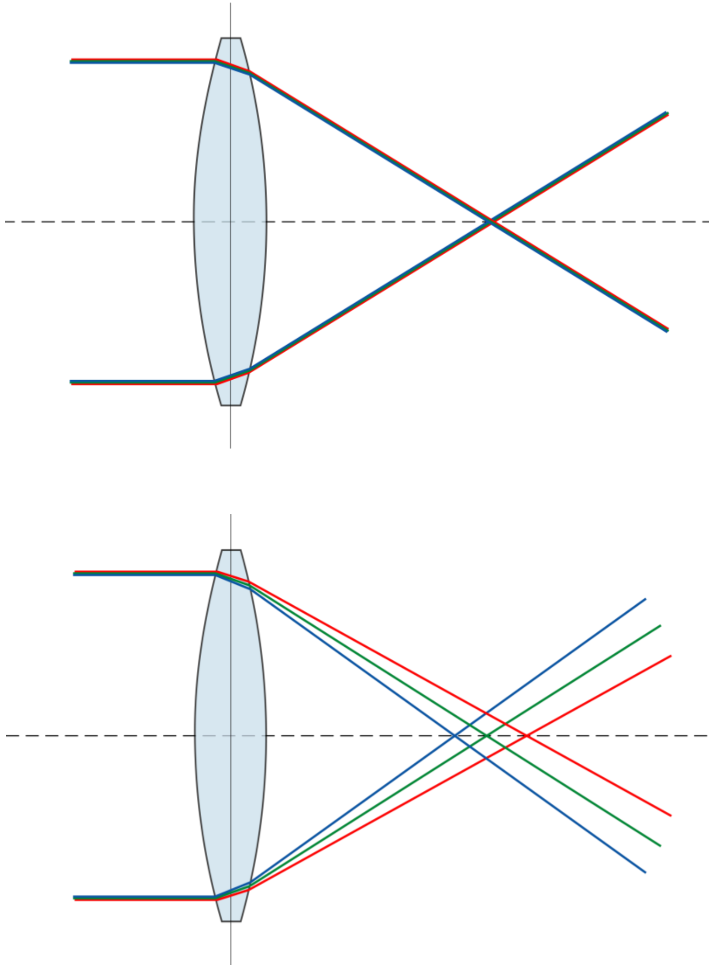
Doersch et. al, Unsupervised Visual Representation Learning by Context Prediction, ICCV 2015

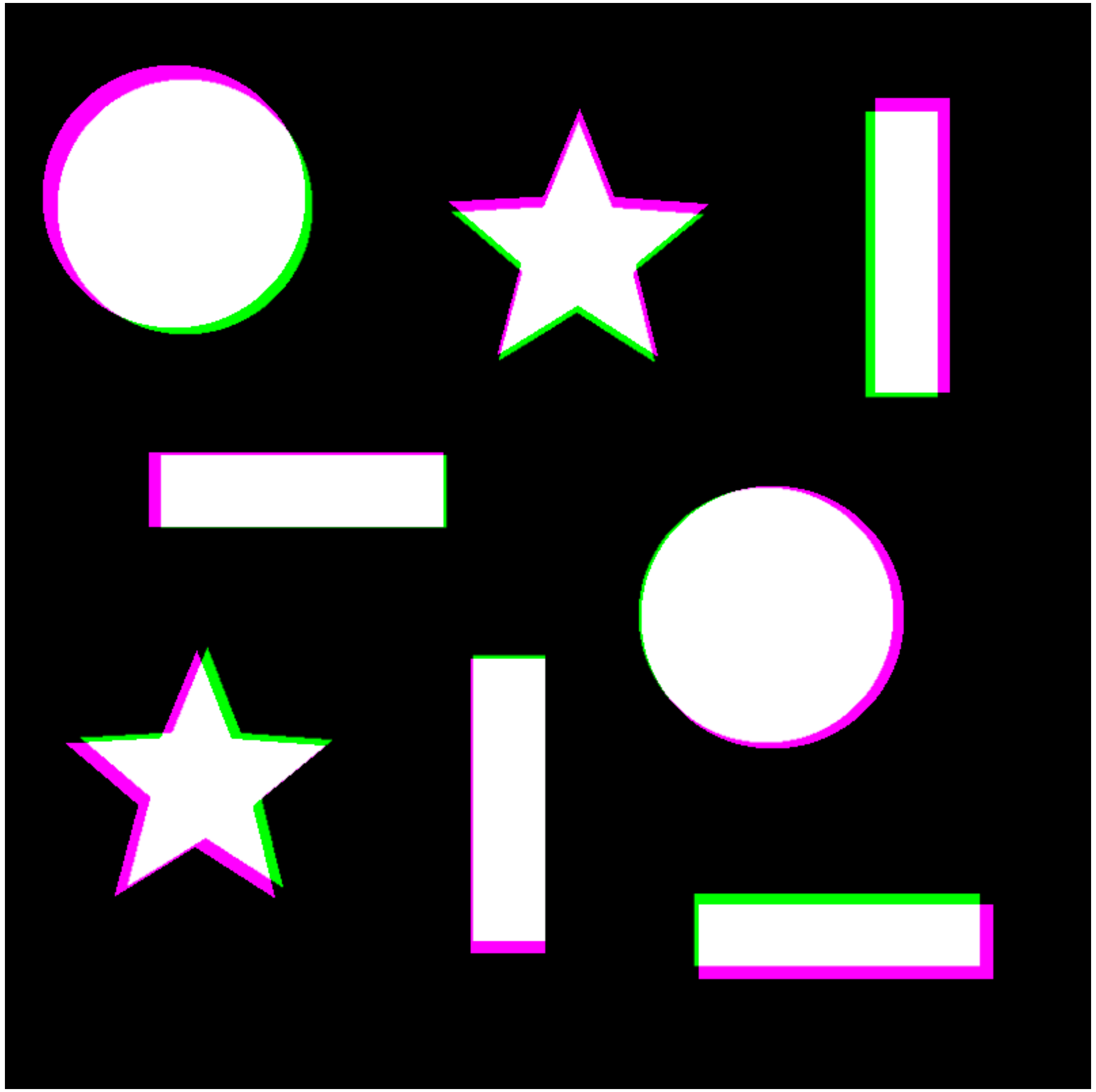
Avoid Network's "cheats"



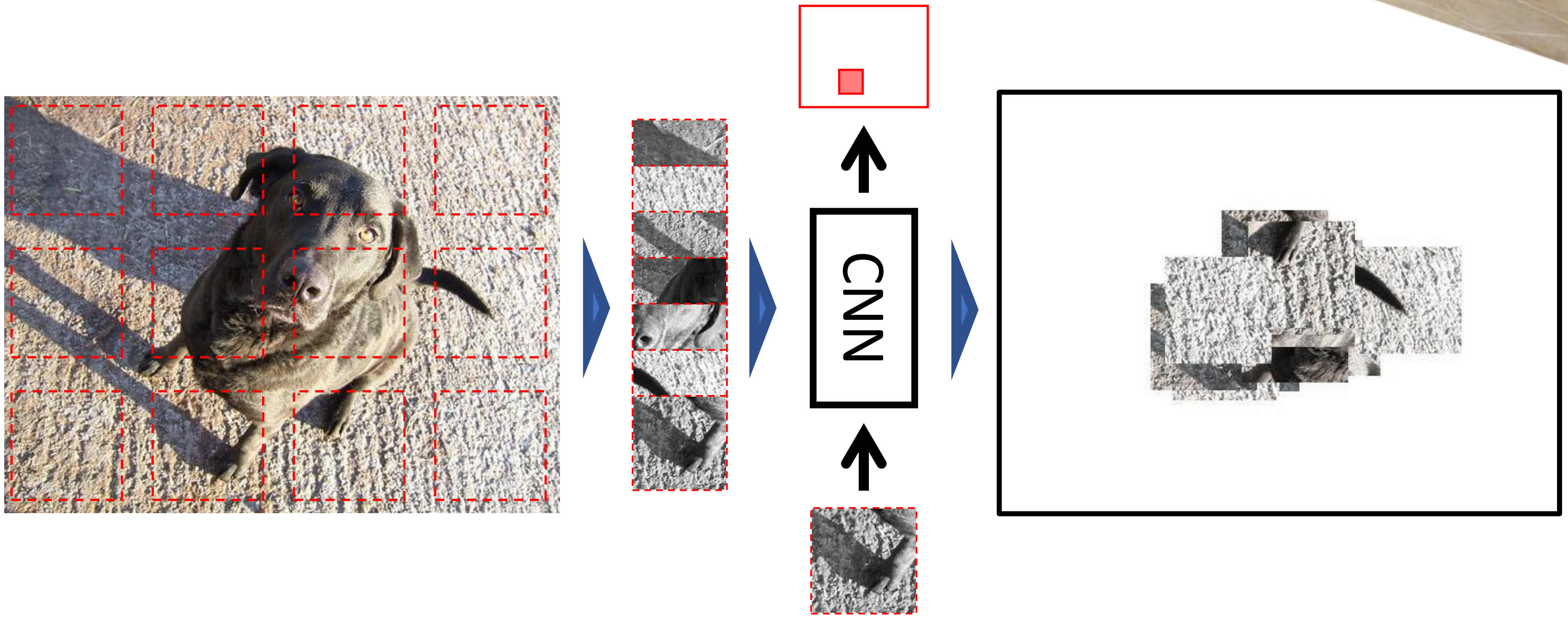
Doersch et. al, Unsupervised Visual Representation Learning by Context Prediction, ICCV 2015

Avoid Network's "cheats" (Chromatic Aberration)



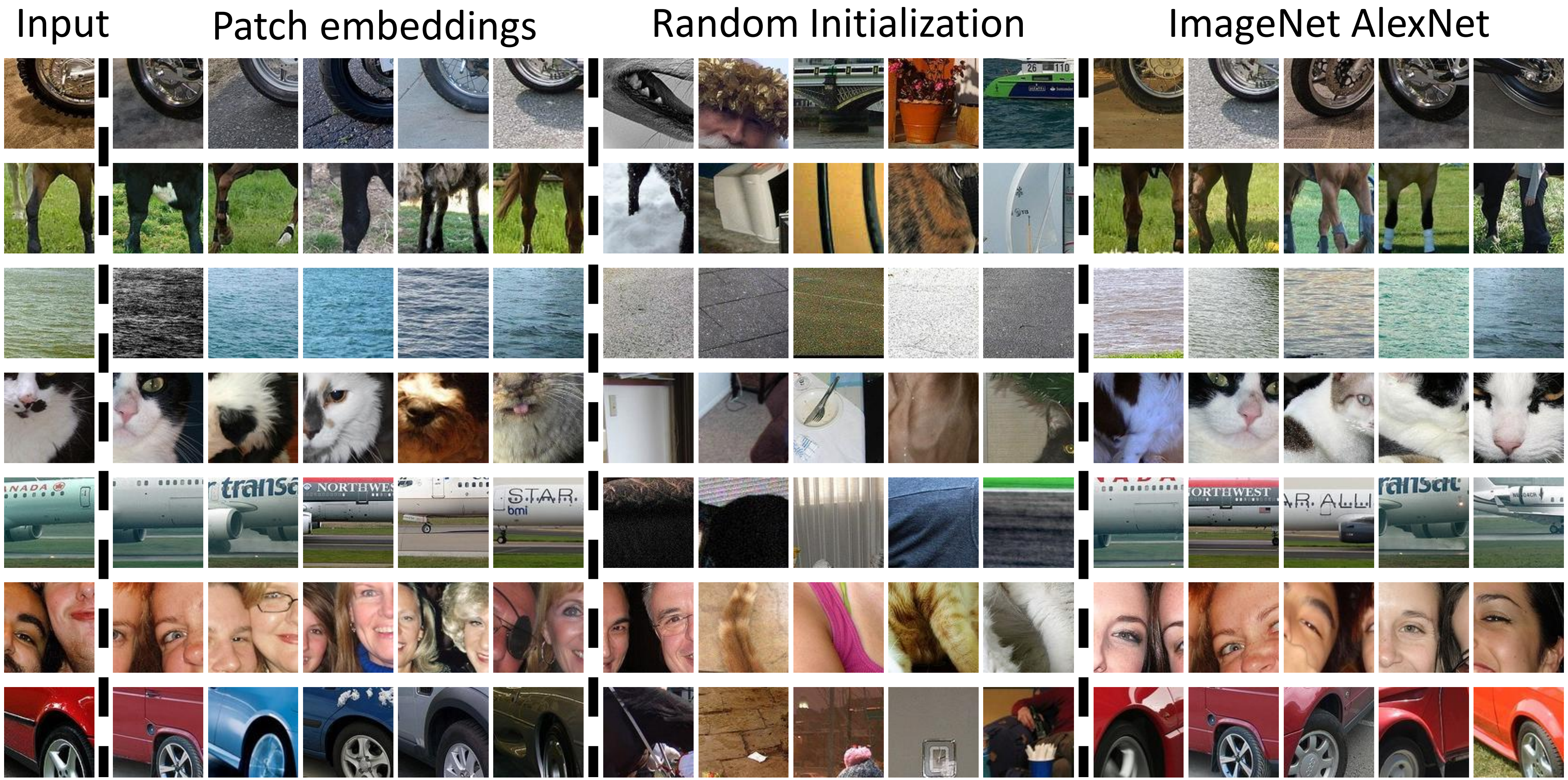


Avoid Network's "cheats"



Doersch et. al, Unsupervised Visual Representation Learning by Context Prediction, ICCV 2015

Learned Patch Embedding



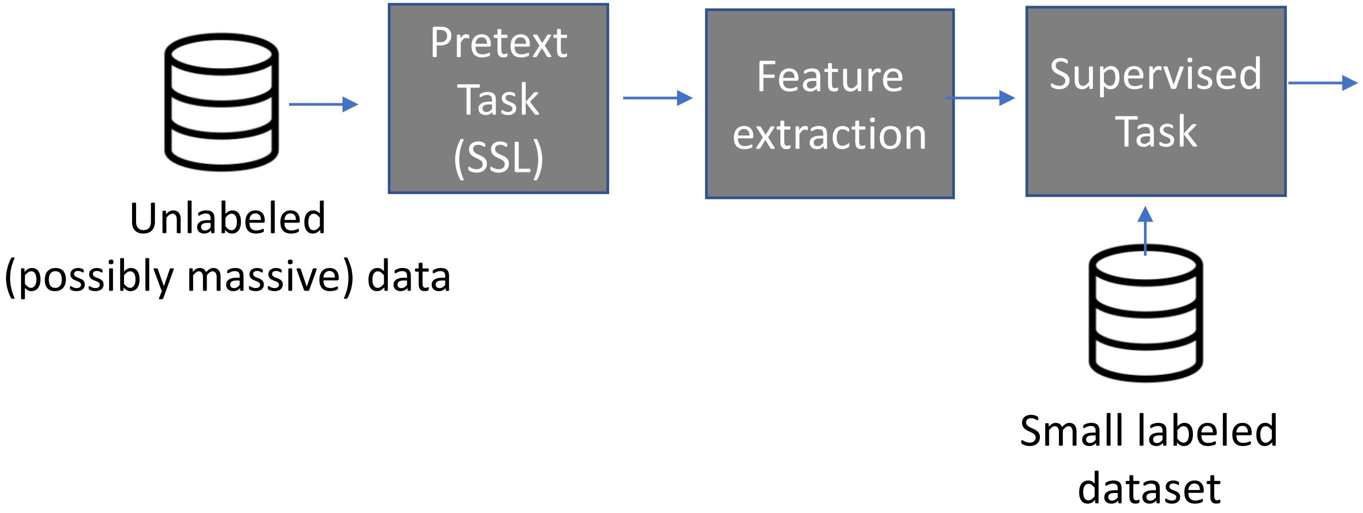
Input Patch embeddings Random Initialization ImageNet AlexNet

Doersch et. al, Unsupervised Visual Representation Learning by Context Prediction, ICCV 2015

Self-Supervised Learning (SSL)

No human labels; supervisory signals are automatically computed from data

Solve a proxy, pretext task → extract learned features → finetune on a target supervised task (Transfer Learning)

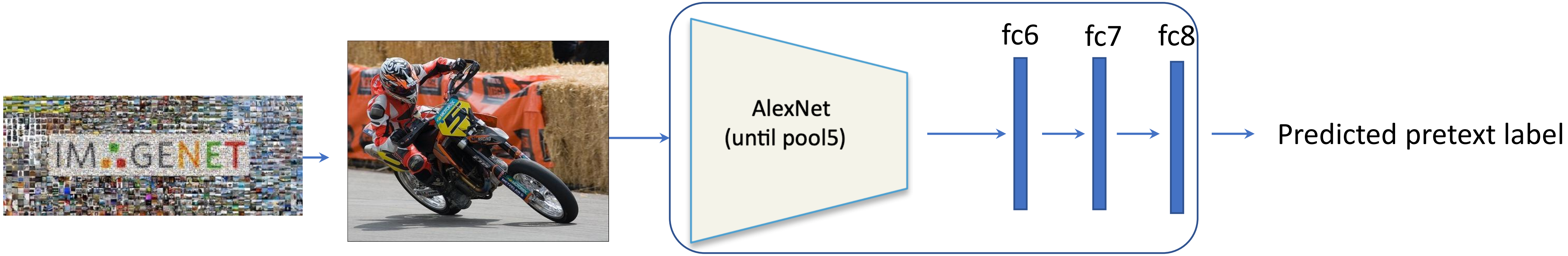


Task-Specific Models → Foundation Models

Self-supervised Transfer Learning

Pre-training on classification and detection tasks for PASCAL VOC 2007 dataset

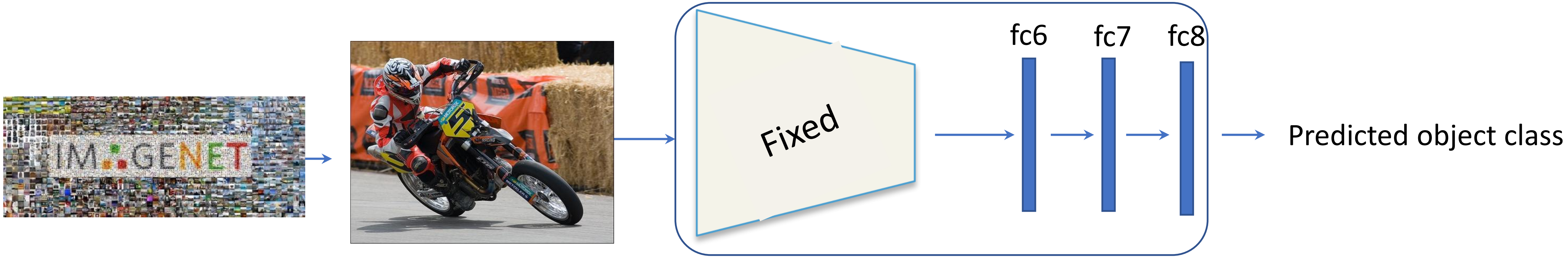
1. Pre-train on pretext task (w/o labels) on ImageNet:



Self-supervised Transfer Learning

Pre-training on classification and detection tasks for PASCAL VOC 2007 dataset

1. Pre-train on pretext task (w/o labels) on ImageNet:

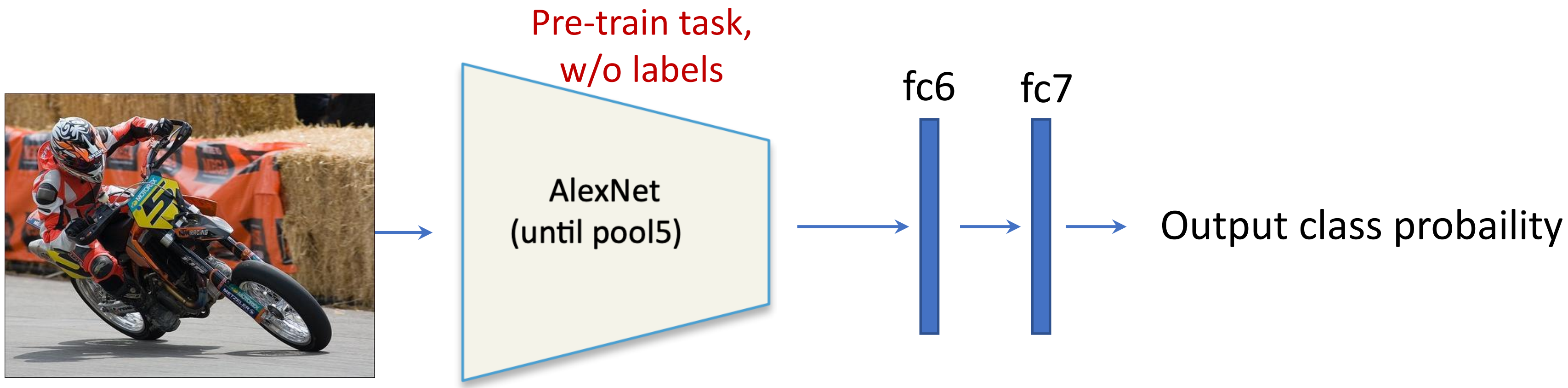


2. Train for classification on PASCAL VOC 2007
   - Fine-tune the entire model, train fully connected layers
   - Freeze Conv layers, train fully connected layers

Context as Supervision: transfer learning

Pre-training on classification and detection tasks for PASCAL VOC 2007 dataset

Classification:



Pre-train task, w/o labels

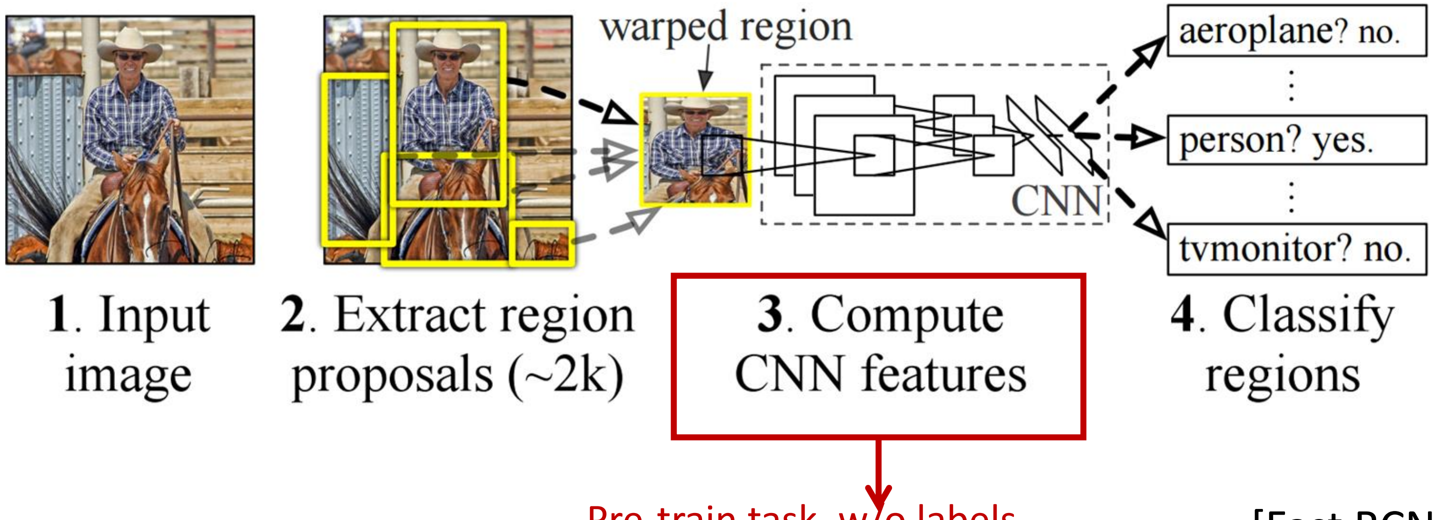
AlexNet (until pool5)

fc6

fc7

Output class probaility

Detection:

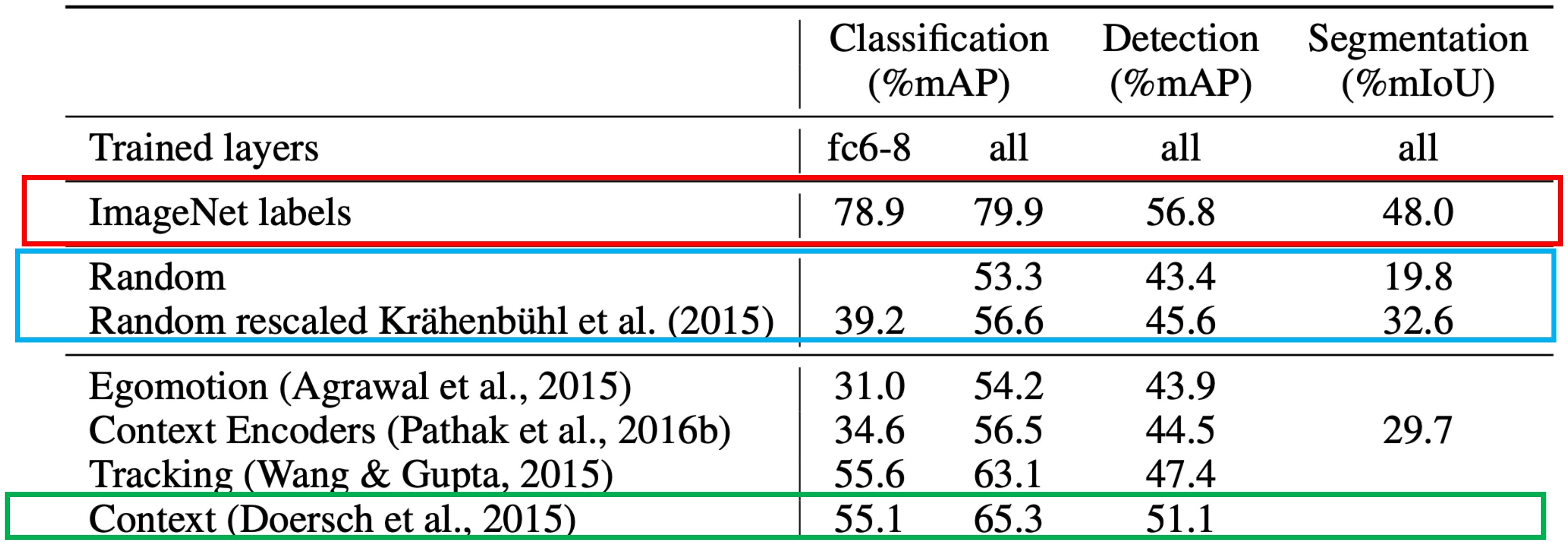


Pre-train task, w/o labels

[Fast RCNN, Girshick et al. 2014]

Self-Supervised Transfer Learning

Pre-training on classification and detection tasks for PASCAL VOC 2007 dataset



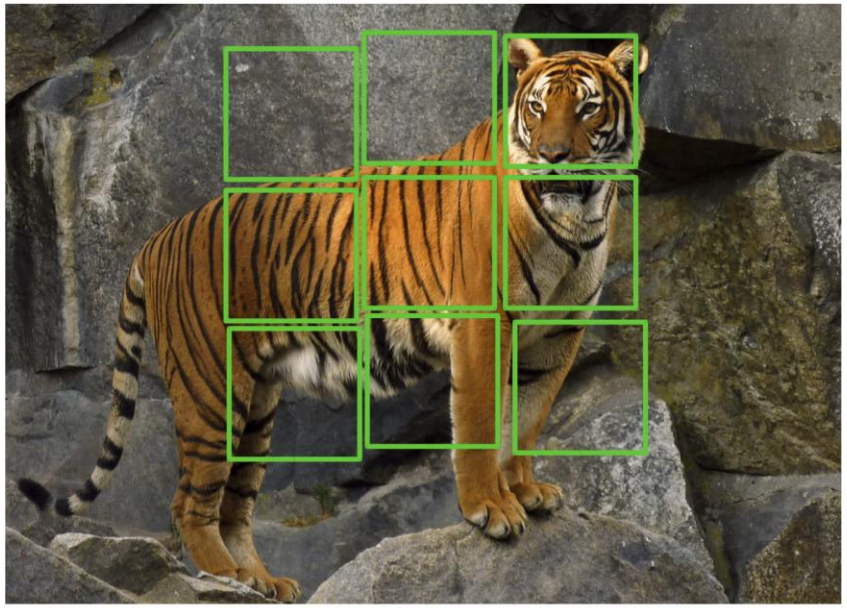
	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	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	

Supervised Pre-training on ImageNet

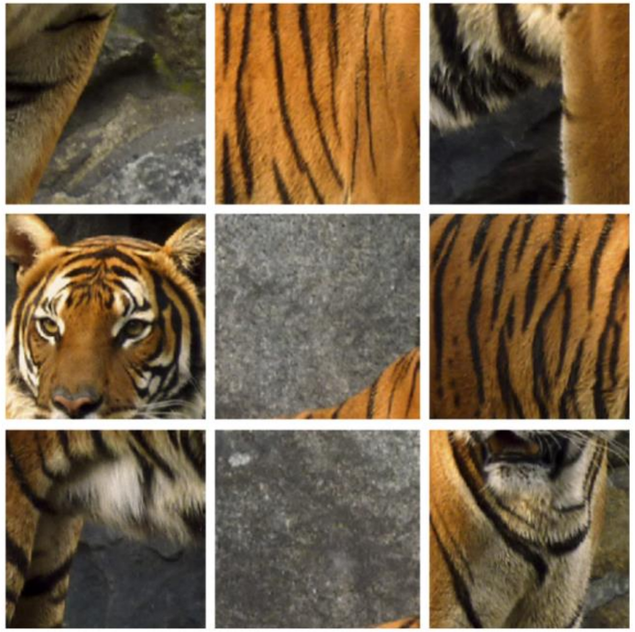
No pre-training

Pre-training with relative patch location

Context as Supervision: solving Jigsaw puzzles



Input Image



Scrambeled patches

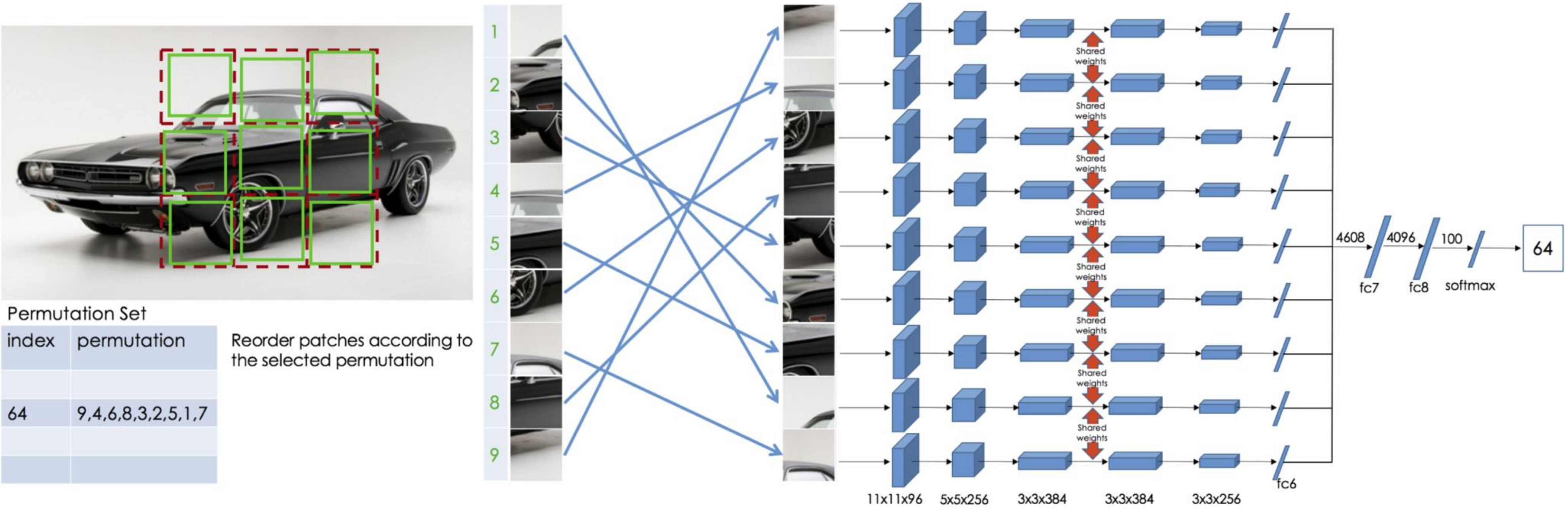
Solved image puzzle

$$9! = 362,880$$

Noroozi et. al, Unsupervised learning of visual representations by solving jigsaw puzzles, ECCV 2016

Context as Supervision: solving Jigsaw puzzles

- **Training data:** 9 tiles, shuffled by a random ordering, sampled from set of permutations
- **Output:** permutation index (1 hot vector)
- **Training loss:** cross entropy w.r.t. ground truth permutation index



Permutation Set

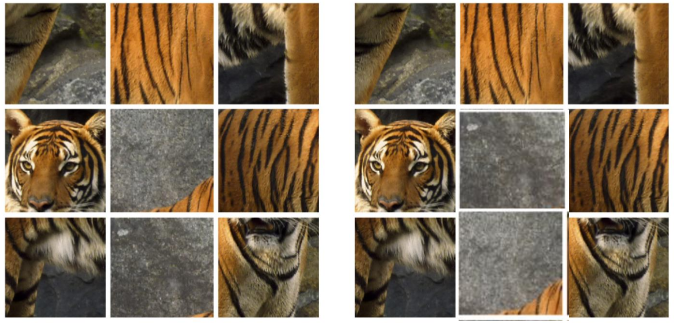
index	permutation
64	9,4,6,8,3,2,5,1,7

Reorder patches according to the selected permutation

Noroozi et. al, Unsupervised learning of visual representations by solving jigsaw puzzles, ECCV 2016

Context as Supervision: solving Jigsaw puzzles

A good self-supervised task is neither simple nor ambiguous.



The solution space is too big → select a permutation set

- Permutation set size
- Distance between permutations

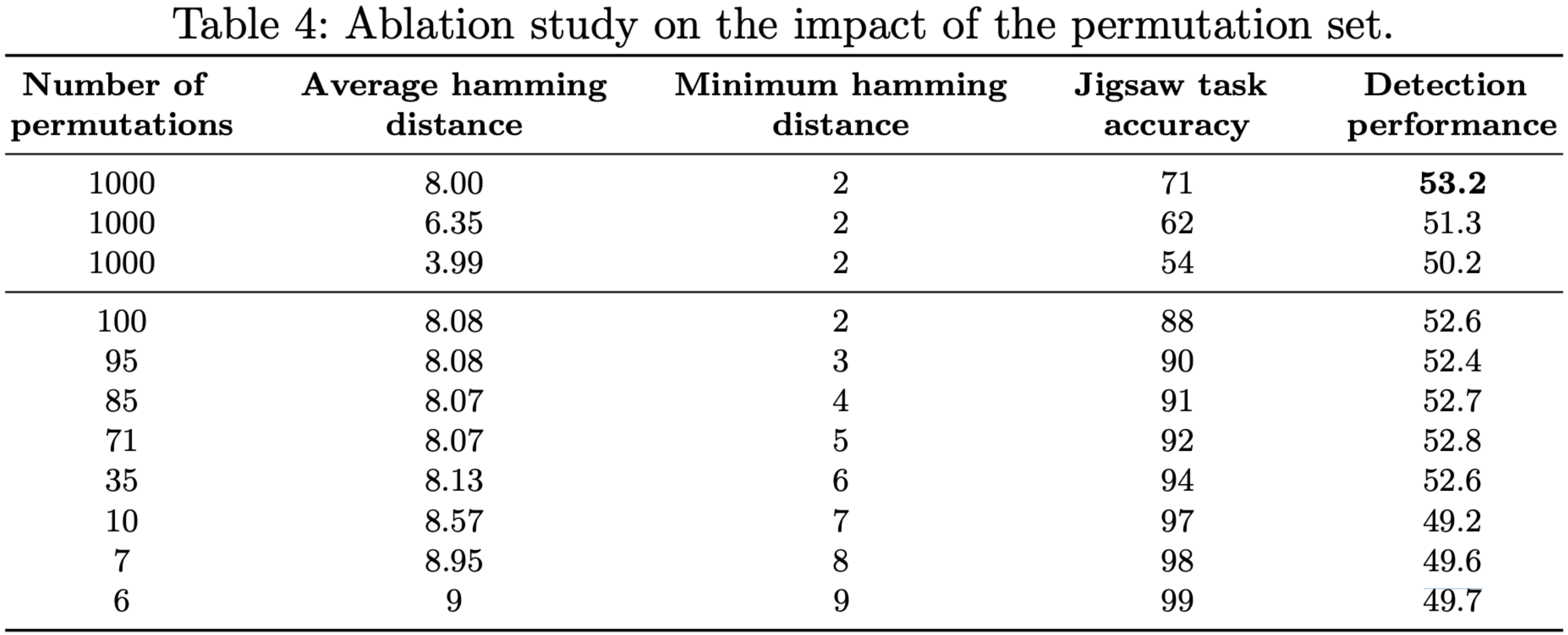


Table 4: Ablation study on the impact of the permutation set.

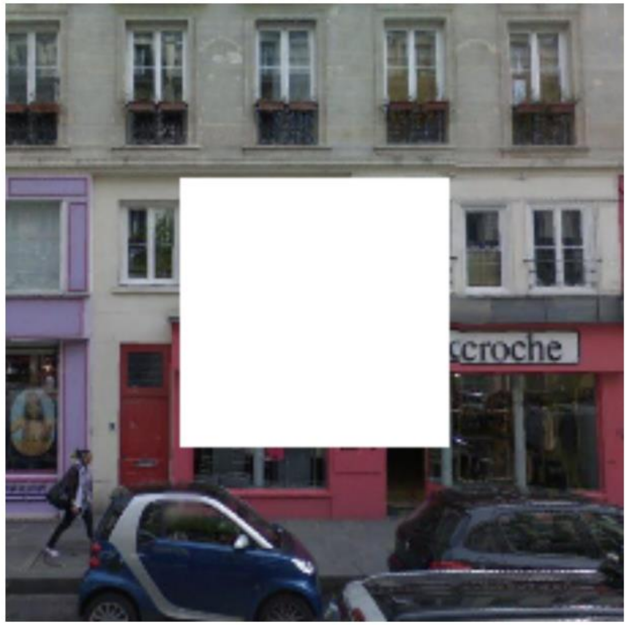
Number of permutations	Average hamming distance	Minimum hamming distance	Jigsaw task accuracy	Detection performance
1000	8.00	2	71	**53.2**
1000	6.35	2	62	51.3
1000	3.99	2	54	50.2
100	8.08	2	88	52.6
95	8.08	3	90	52.4
85	8.07	4	91	52.7
71	8.07	5	92	52.8
35	8.13	6	94	52.6
10	8.57	7	97	49.2
7	8.95	8	98	49.6
6	9	9	99	49.7

- Smaller permutation set → higher accuracy
- Smaller permutation set → Lower detection performance
- Larger distance between permutations → higher accuracy
- Larger distance between permutations → higher detection performance

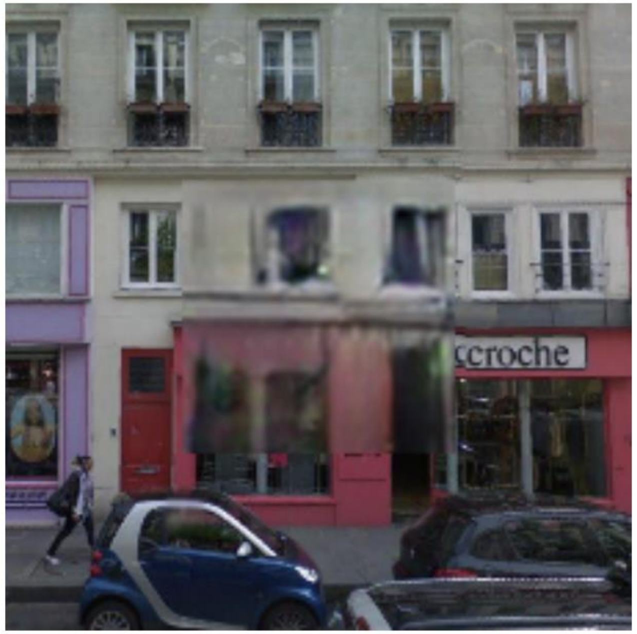
Noroozi et. al, Unsupervised learning of visual representations by solving jigsaw puzzles, ECCV 2016

Image Content as Supervision: Image Inpainting

Pretext task: fill in the missing region



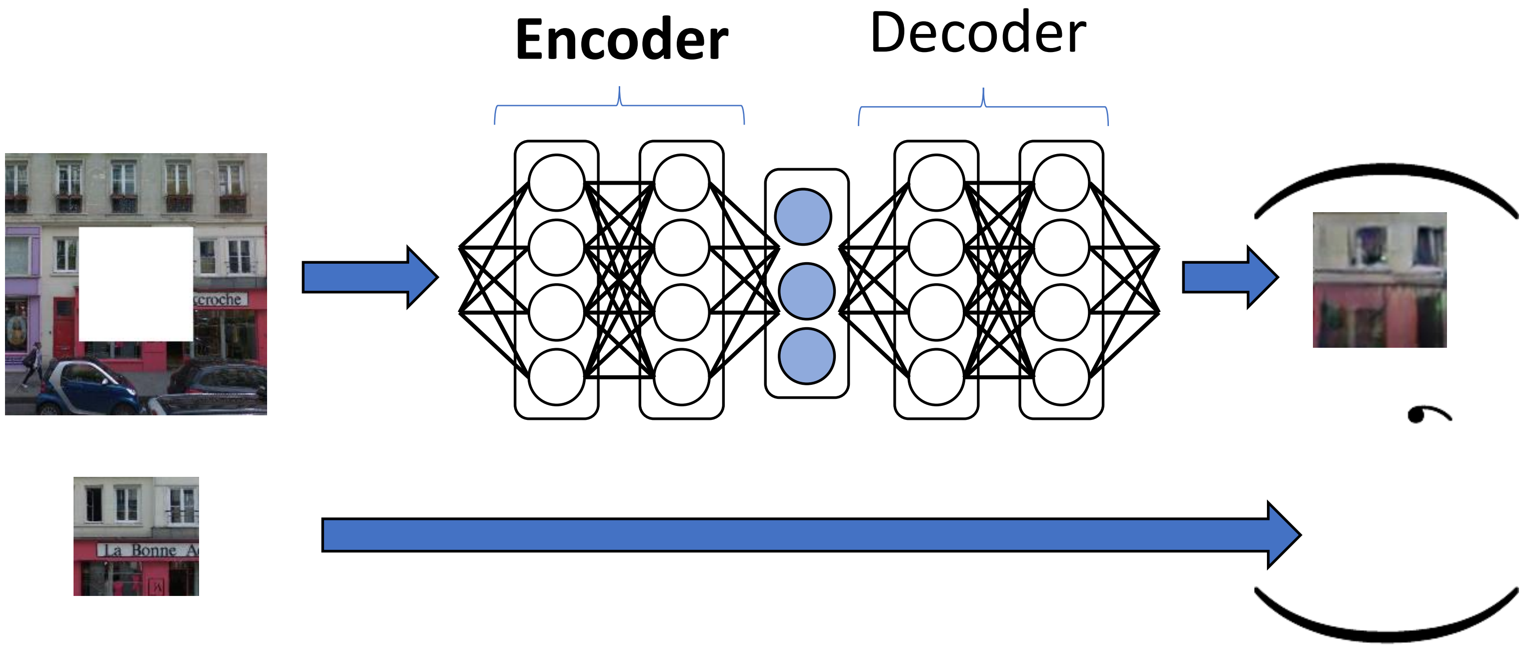
Input masked image



Output image

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

Pretext Task: Image Inpainting



$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec}$$

Reconstruction L_2 loss ensures “**correctness**’

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

Pretext Task: Image Inpainting

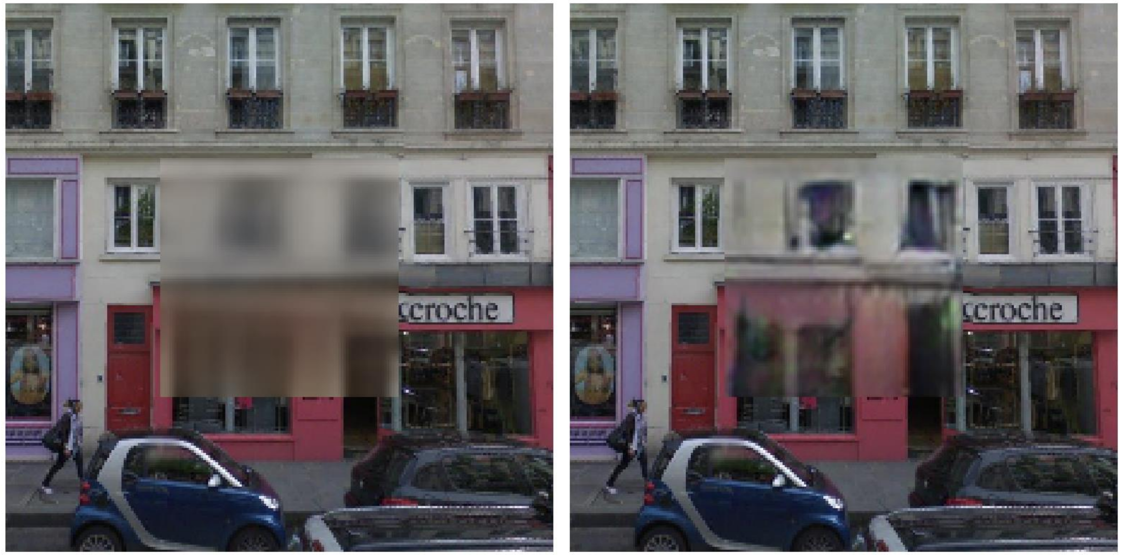
$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}.$$

Reconstruction L_2 loss ensures "**correctness**"

Adversarial Loss ensures "**realness**"

$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2,$$

$$\begin{aligned}\mathcal{L}_{adv} = \max_{D} \quad & \mathbb{E}_{x\in\mathcal{X}}[\log(D(x)) \\ & + \log(1 - D(F((1 - \hat{M}) \odot x)))],\end{aligned}$$

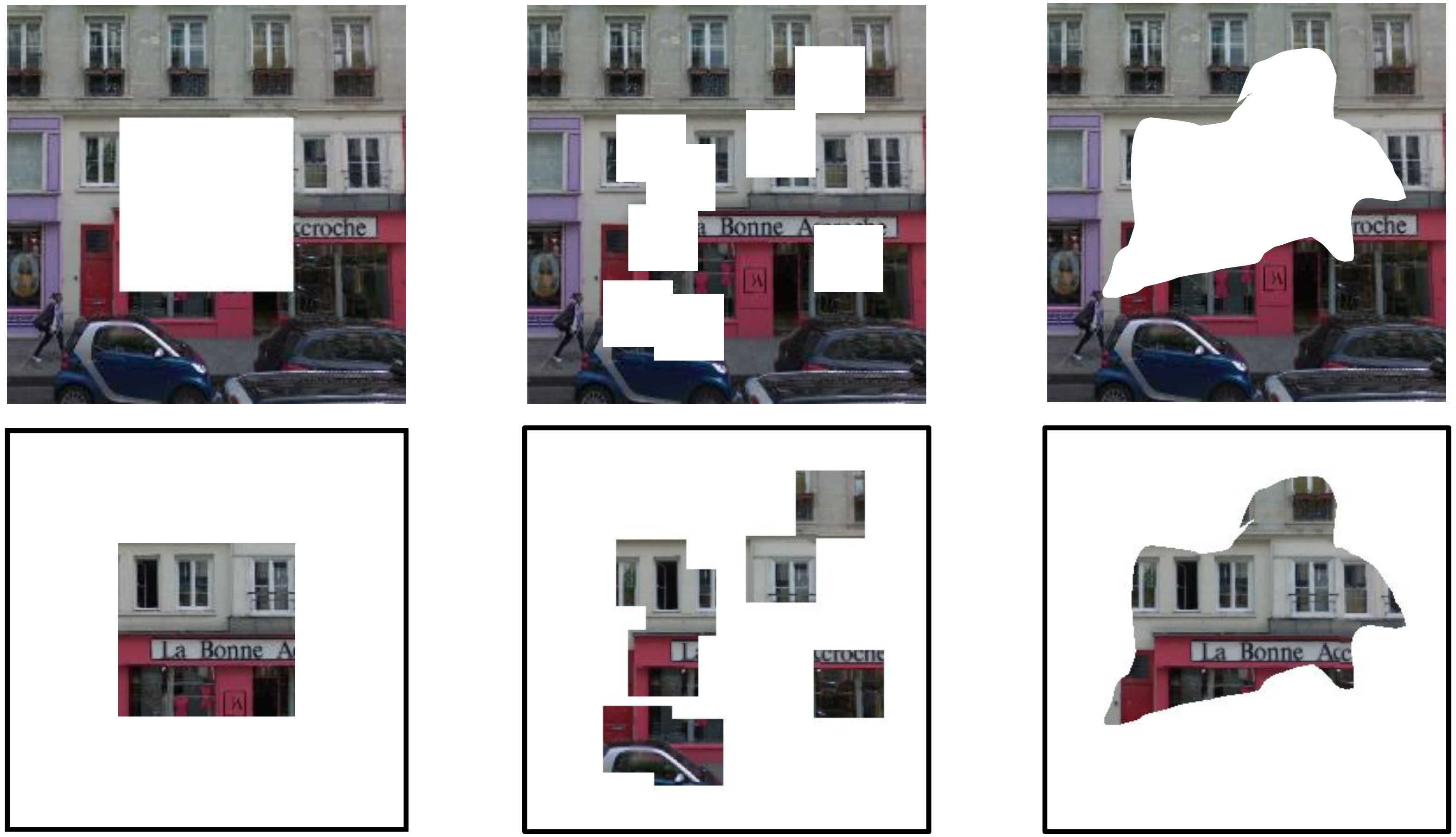


(c) Context Encoder ($L2$ loss)

(d) Context Encoder ($L2$ + Adversarial loss)

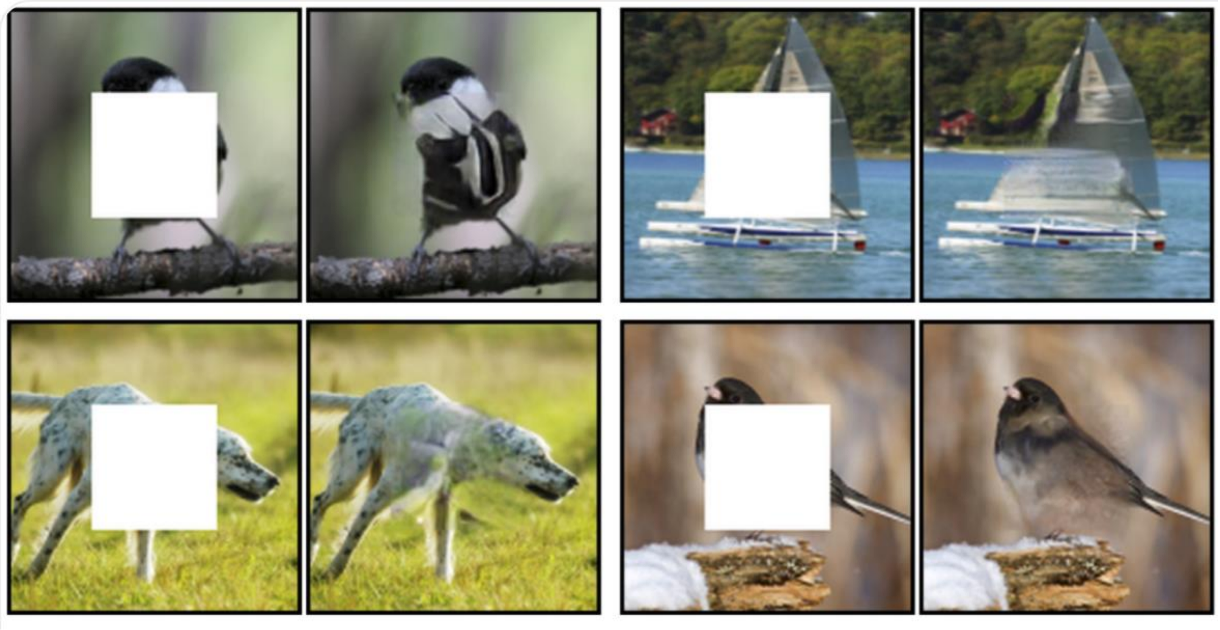
Pathak et al., Context Encoders: Feature Learning by Inpainting, 2016

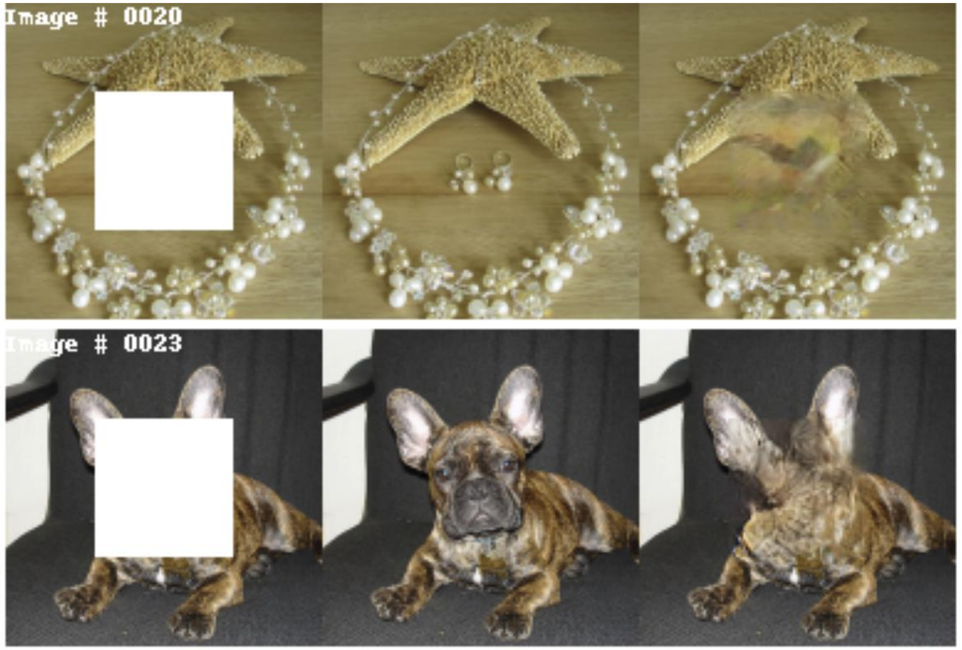
Again… dealing with network's "cheats"



(a) Center Region (b) Random Blocks (c) Random Shapes

Pretext Task: Image Inpainting





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

Context as Supervision: transfer learning

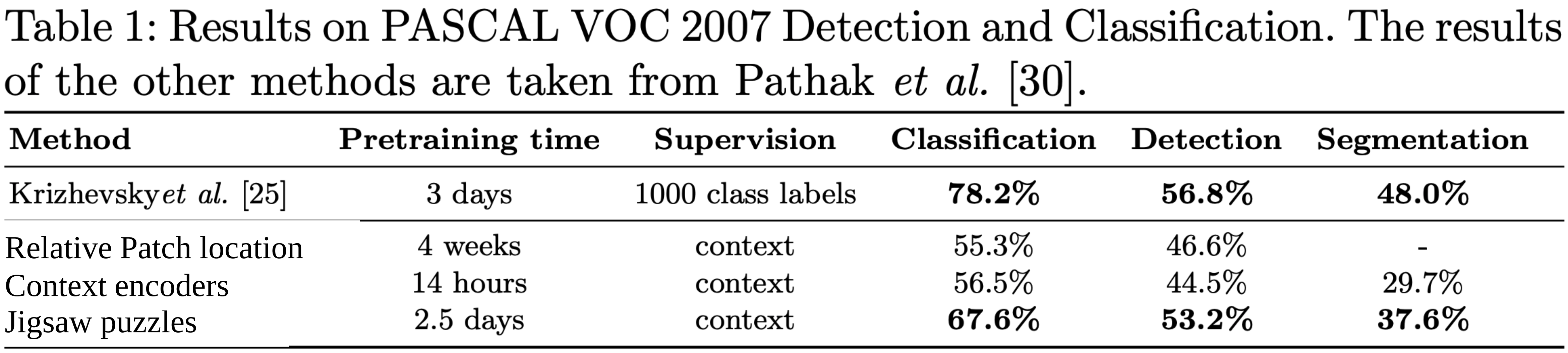
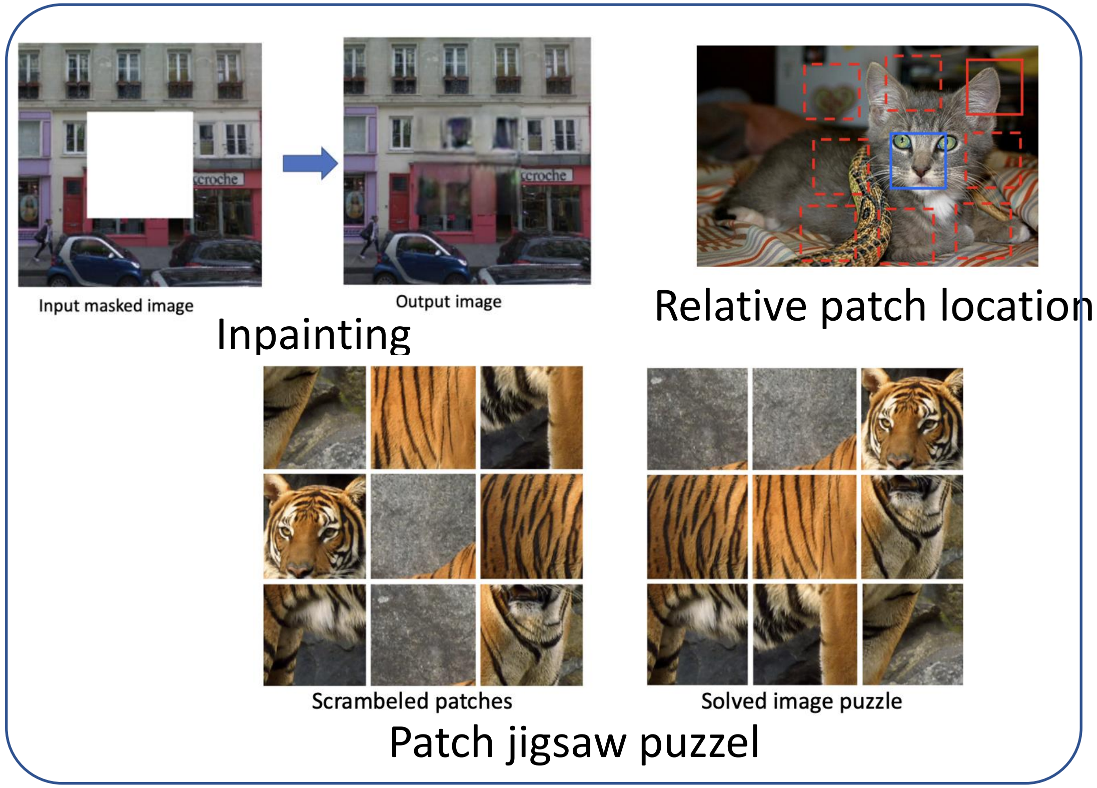


Table 1: Results on PASCAL VOC 2007 Detection and Classification. The results of the other methods are taken from Pathak *et al.* [30].

Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Relative Patch location	4 weeks	context	55.3%	46.6%	-
Context encoders	14 hours	context	56.5%	44.5%	29.7%
Jigsaw puzzles	2.5 days	context	**67.6%**	**53.2%**	**37.6%**

Self-Supervised Learning

Solve a proxy, pretext task (large dataset) → extract learned features → finetune on a target supervised task (smaller dataset)



Input masked image

Output image

Inpainting

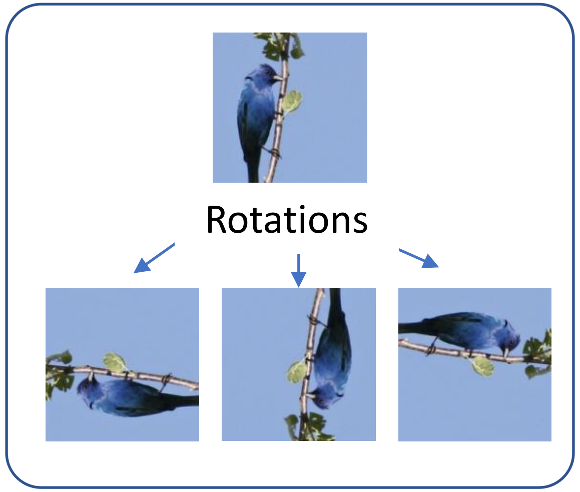
Relative patch location

Scrambeled patches

Solved image puzzle

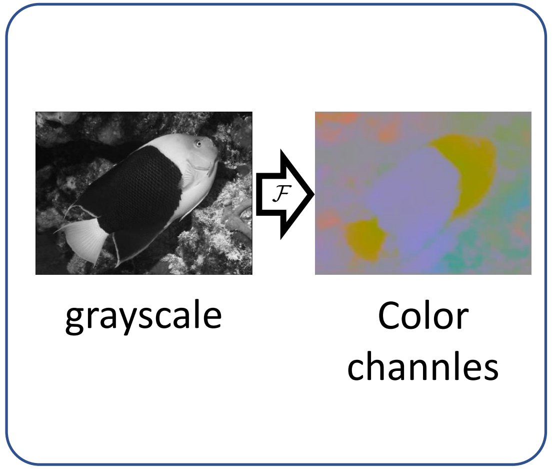
Patch jigsaw puzzel

Image context as supervision



Rotations

Geometric transformations



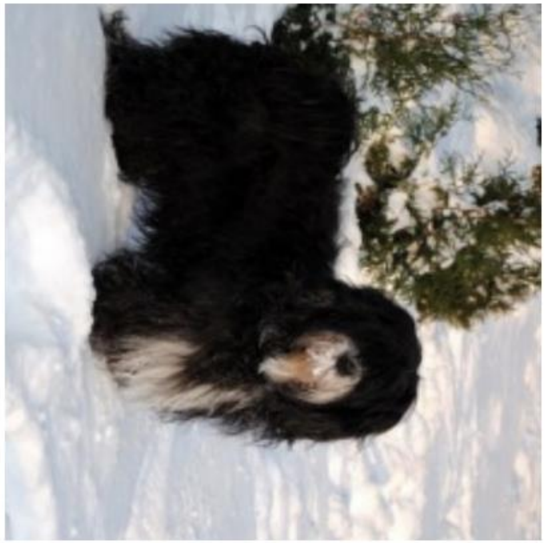
grayscale

Color channles

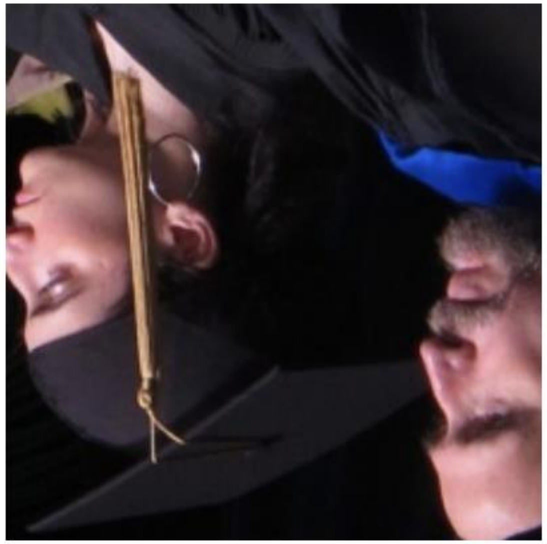
Color transformations

Pretext task: predicting image rotations

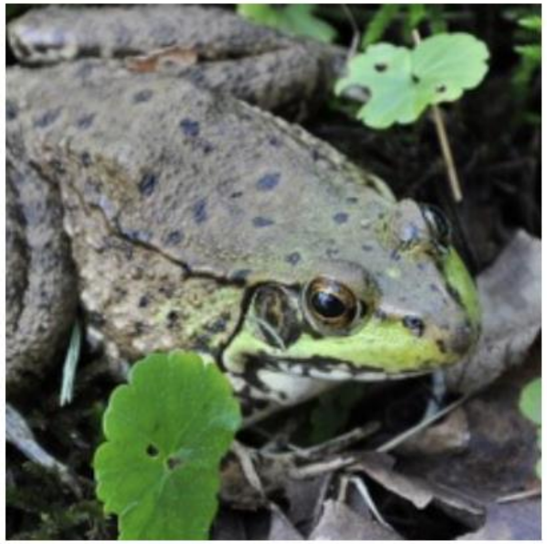
To recognize rotations, the model has to learn concepts of the objects



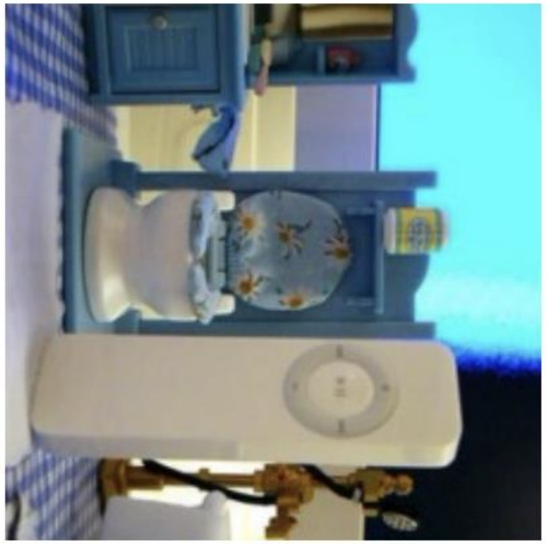
270° rotation



180° rotation



0° rotation

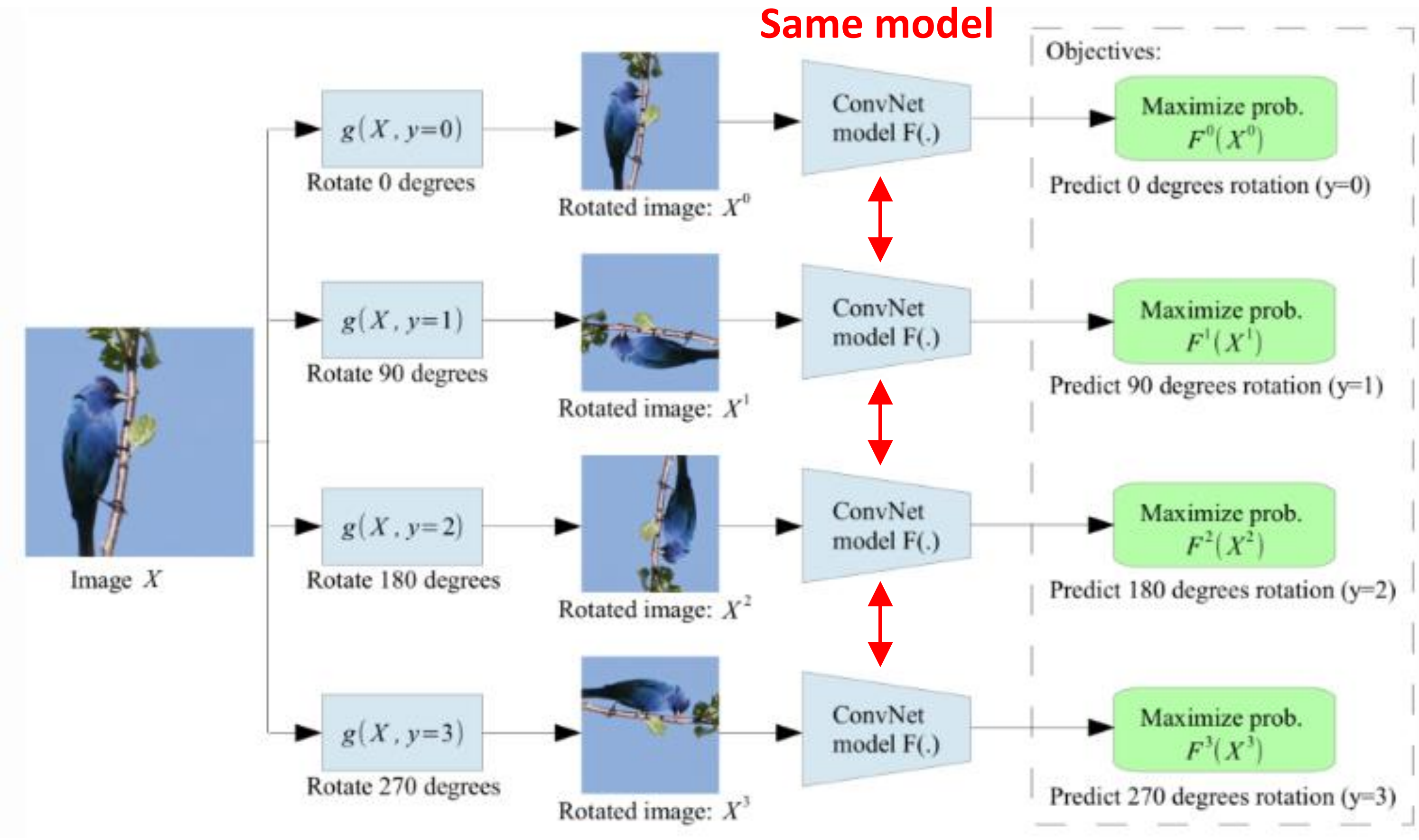


270° rotation

Gidaris et. al, Unsupervised Representation Learning by Predicting Image Rotations, 2018

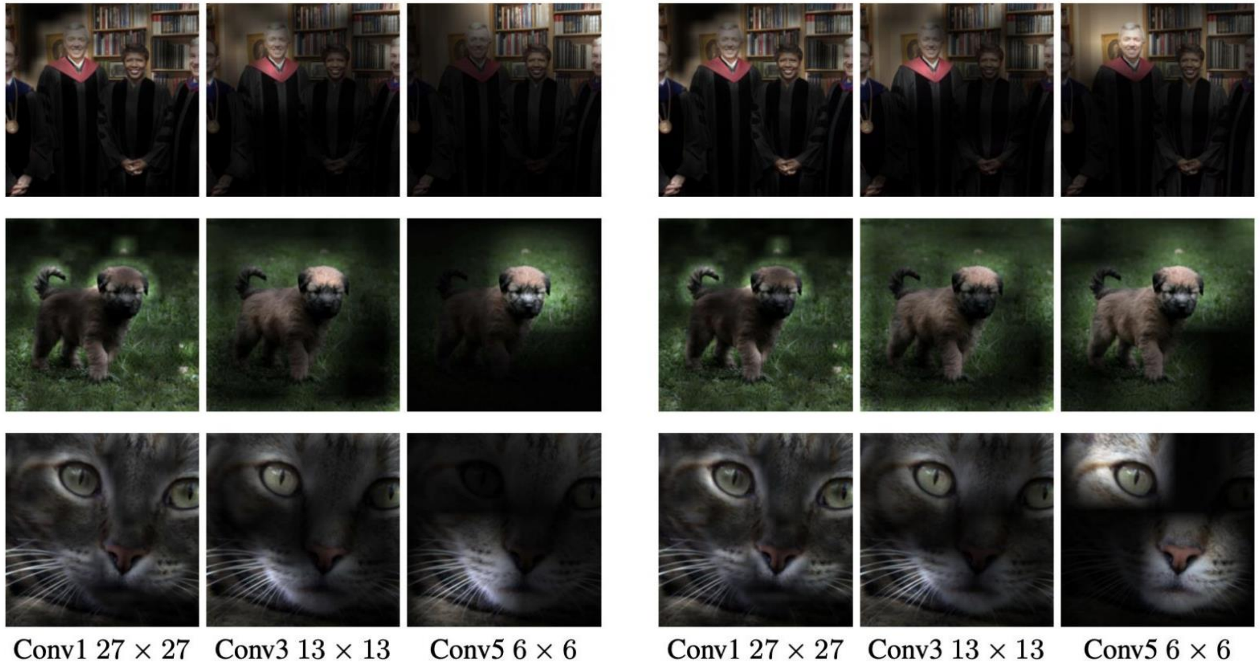
Pretext task: predicting image rotations

- **Training data:** images rotated by: 0°, 90°, 180°, and 270° (via flip and transpose operations)
- **Task:** predict which rotation is applied; 4-way classification task
- **Training loss:** assign a “label” to each rotation; apply cross entropy loss w.r.t. ground truth



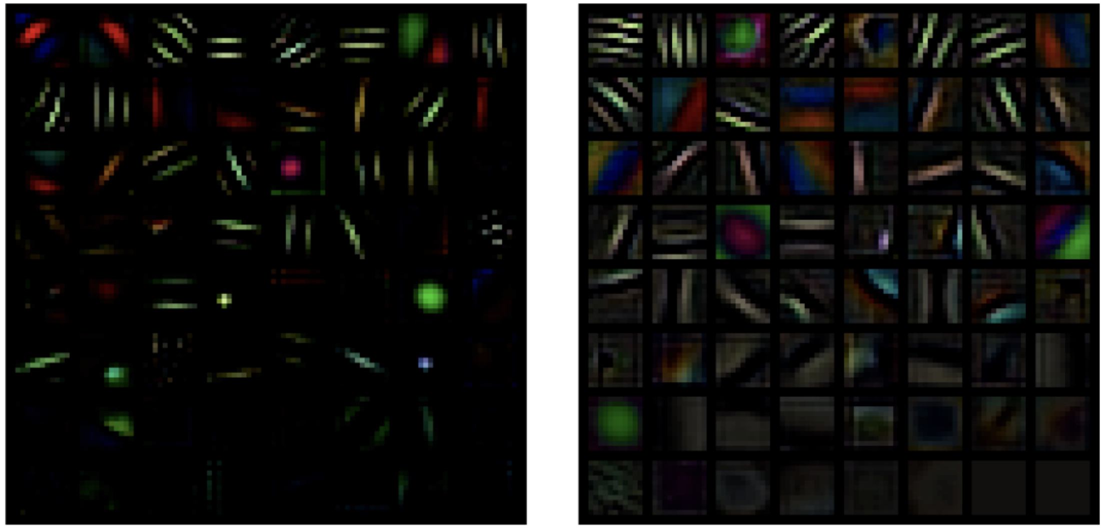
Gidaris et. al, Unsupervised Representation Learning by Predicting Image Rotations, 2018

Predicting image rotations vs. supervised classification



(a) **Attention maps of supervised model**

(b) **Attention maps of our self-supervised model**



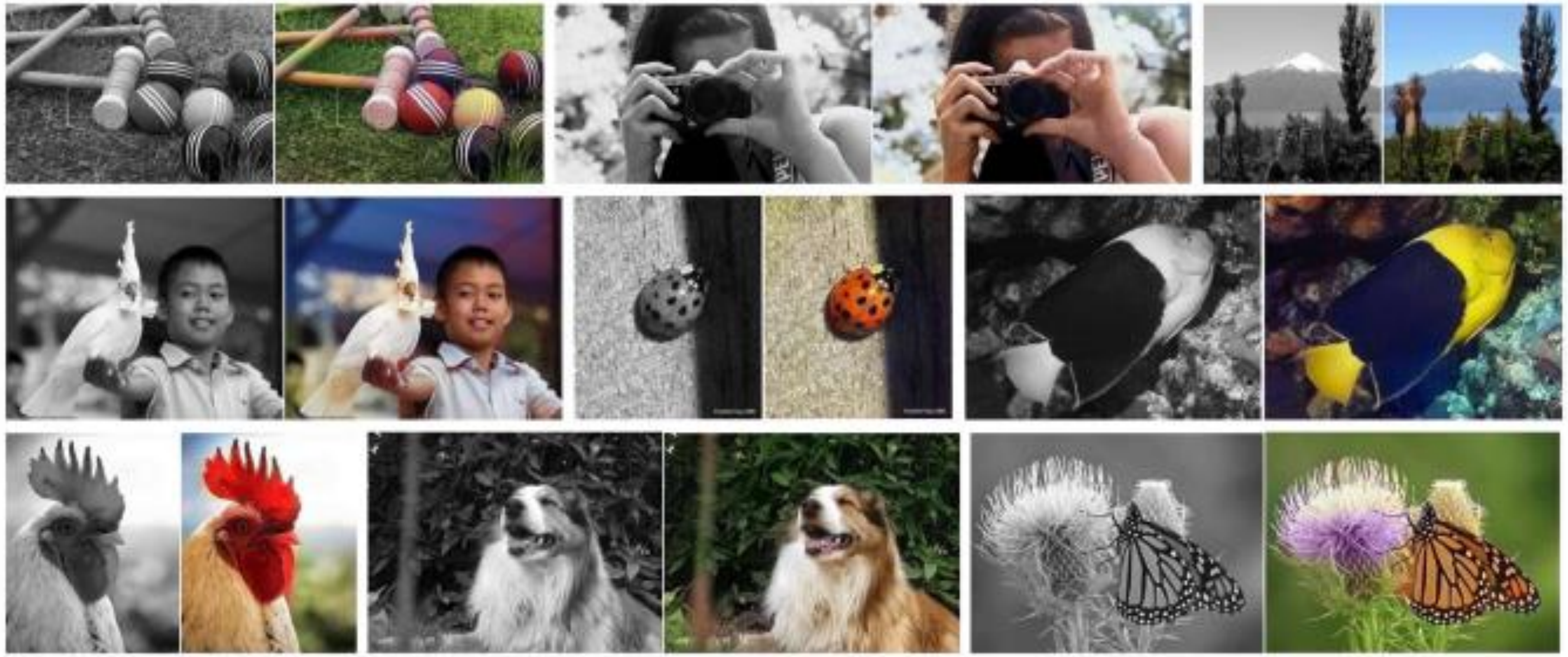
(a) **Supervised**

(b) **Self-supervised to recognize rotations**

Gidaris et. al, Unsupervised Representation Learning by Predicting Image Rotations, 2018

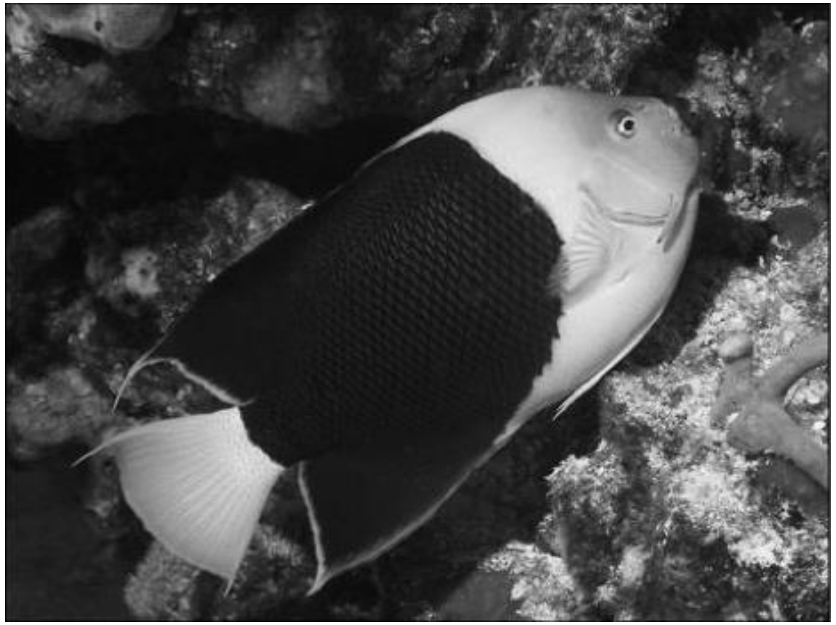
Pretext task: colorization

- **Training data:** grayscale images (and their ground truth color images)
- **Task:** generate a plausible color image

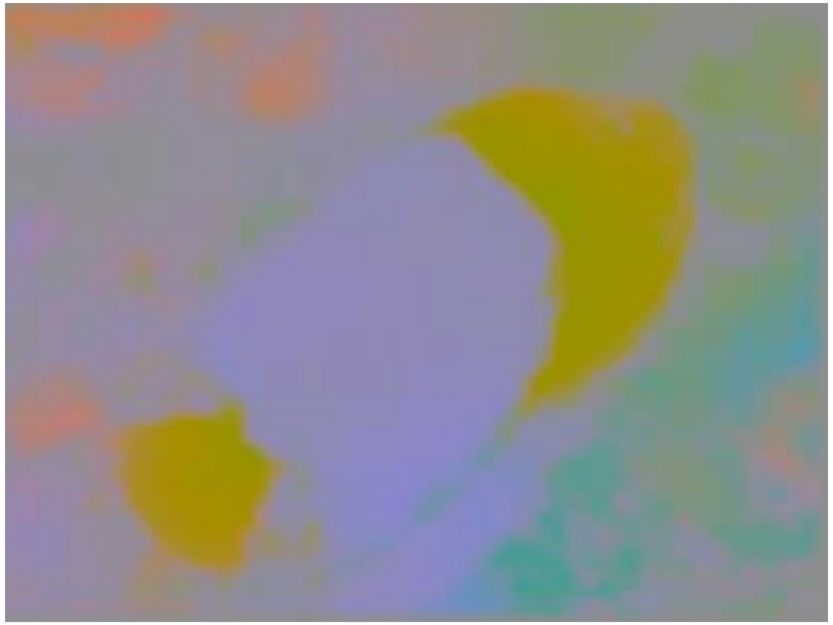


Zhang et. al, Colorful Image Colorization, 2016

Pretext task: colorization



$\xrightarrow{\mathcal{F}}$

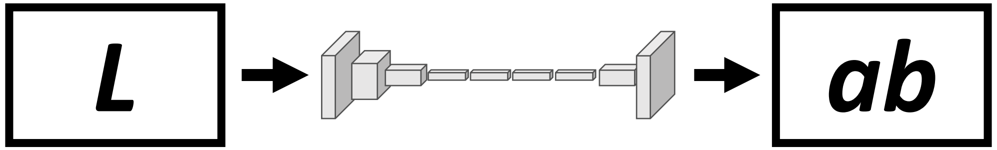


Grayscale image: *L* channel

$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Color information: *ab* channels

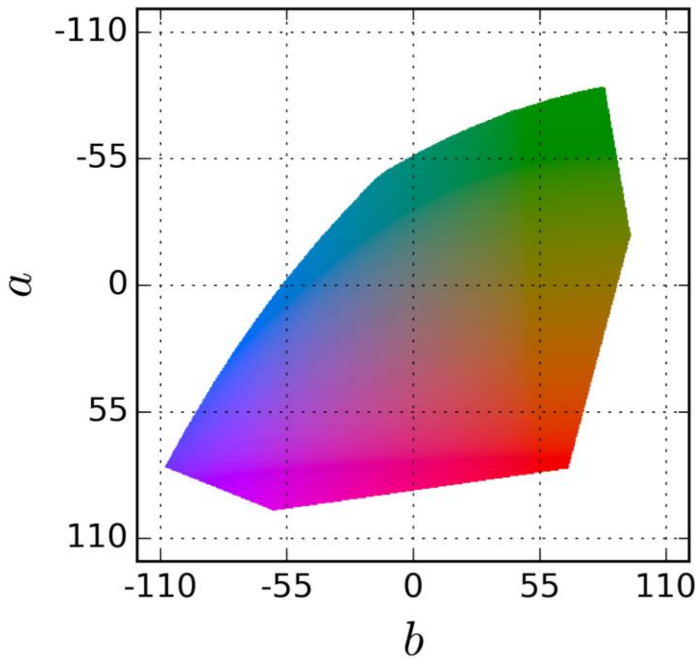
$\widehat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$



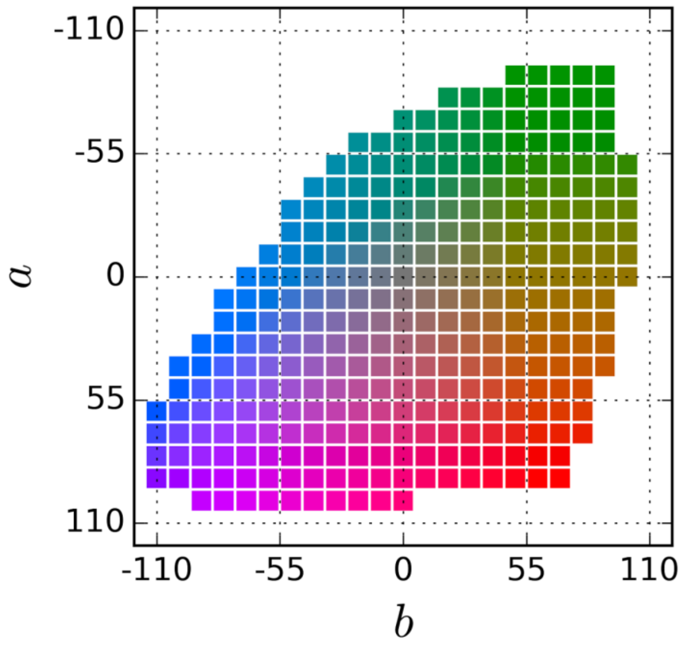
Loss Function

- Regression with L2 loss inadequate

$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w}\|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

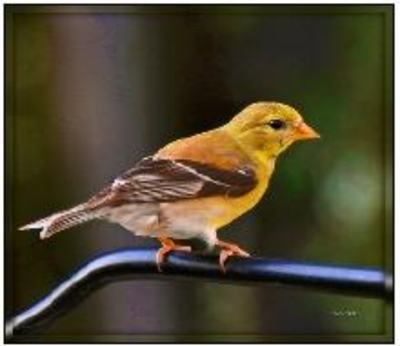


Colors in *ab* space (continuous)

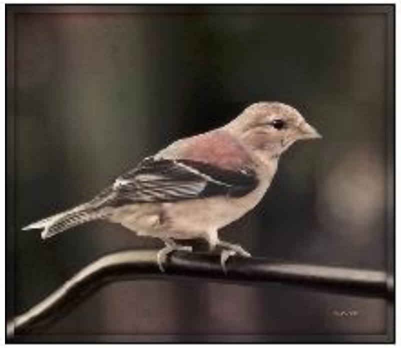


Colors in *ab* space (discrete)

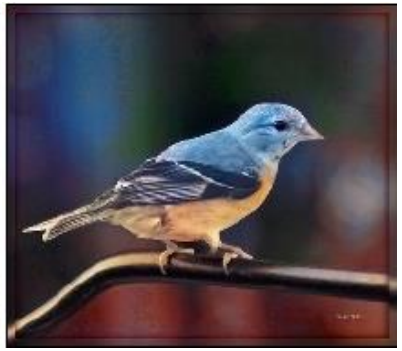
Ground Truth



L2 Regression

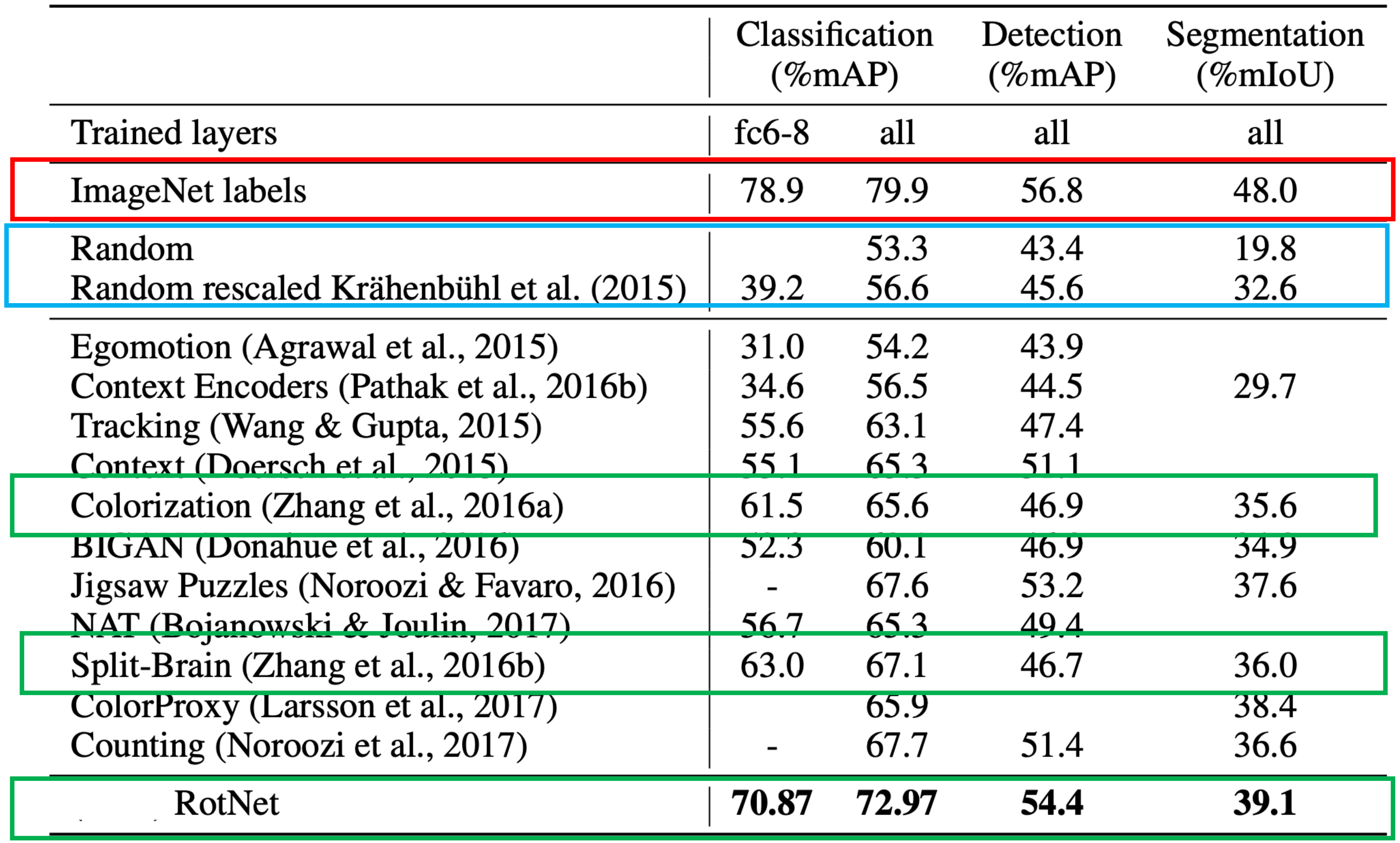


Multimodal class loss



Transfer Learning

Pre-training on classification and detection tasks for PASCAL VOC 2007 dataset



	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
RotNet	**70.87**	**72.97**	**54.4**	**39.1**

Supervised Pre-training on ImageNet

No pre-training

Colorization

Colorization

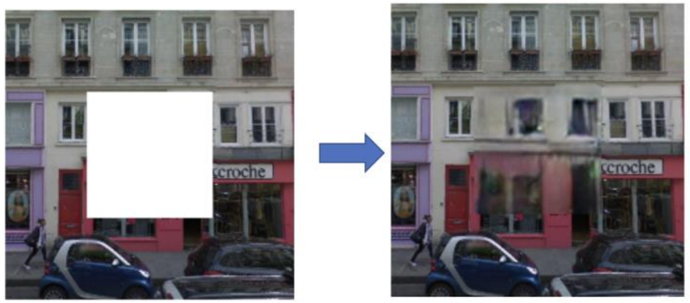
Pre-training with rotation prediction

Gidaris et. al, Unsupervised Representation Learning by Predicting Image Rotations, 2018

Self-Supervised Learning via Specific Pretext Task

Learned representations are task specific!

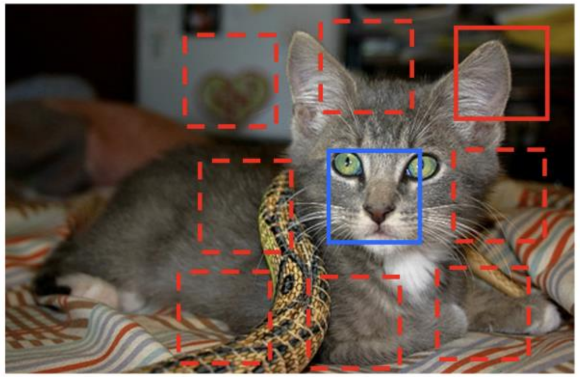
Can we define a more general pretext task?



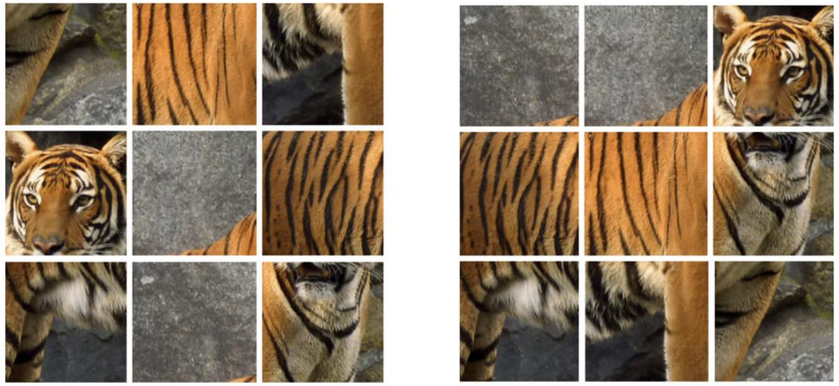
Input masked image

Output image

Inpainting



Relative patch location



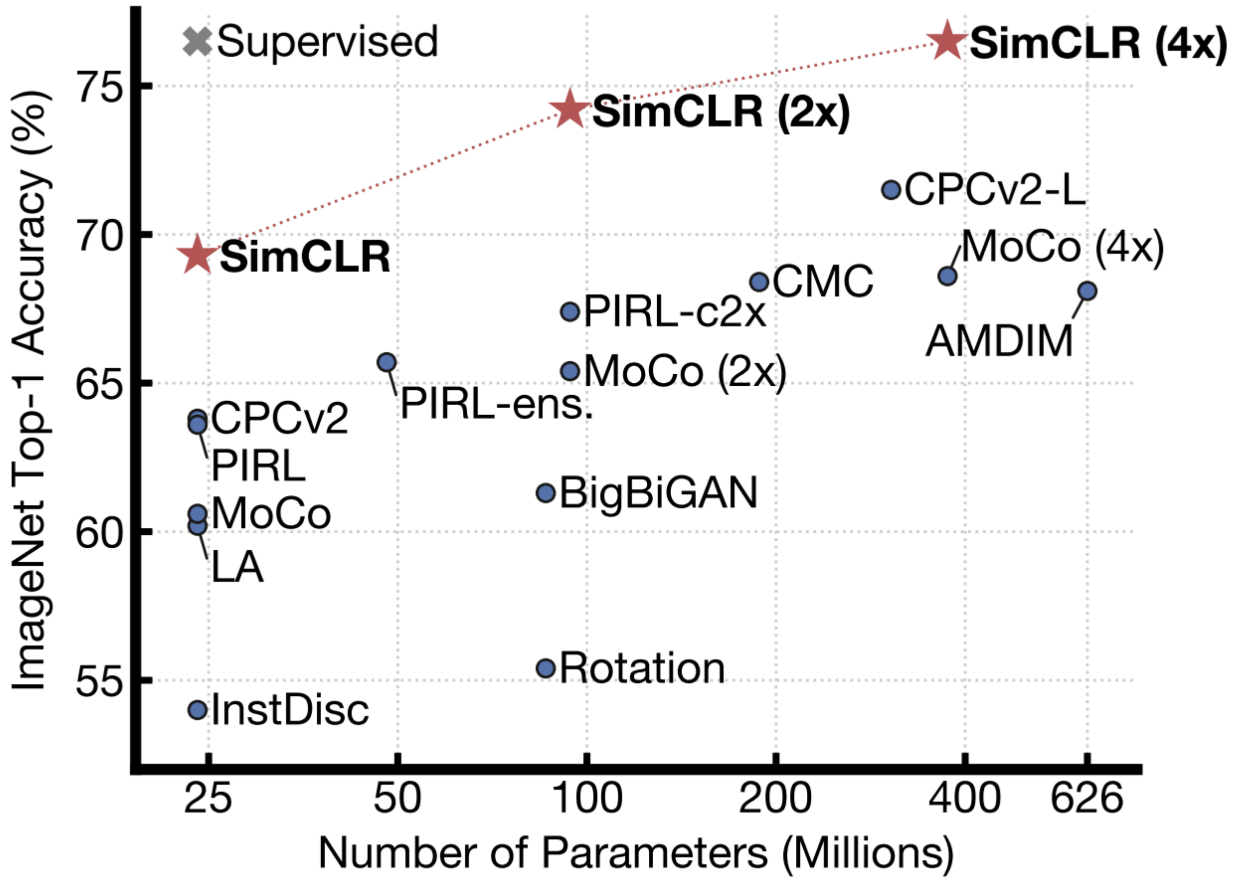
Scrambeled patches

Solved image puzzle

Patch jigsaw puzzel

SimCLR

a Simple framework for Contrastive Learning of Representations



- Train feature encoder on ImageNet using SimCLR
- Freeze feature encoder
- Train a linear classifier on top with labeled data

Chen et. al., A Simple Framework for Contrastive Learning of Visual Representations, 2020, **14600 citations**

SimCLR

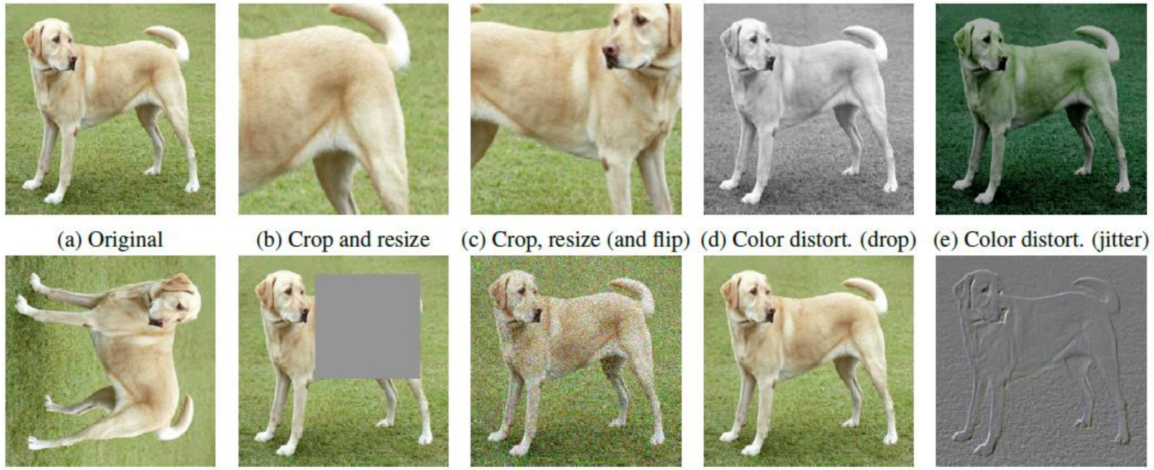
(a) Original

x

Same object

Not same object

set of augmentation applied on the original image



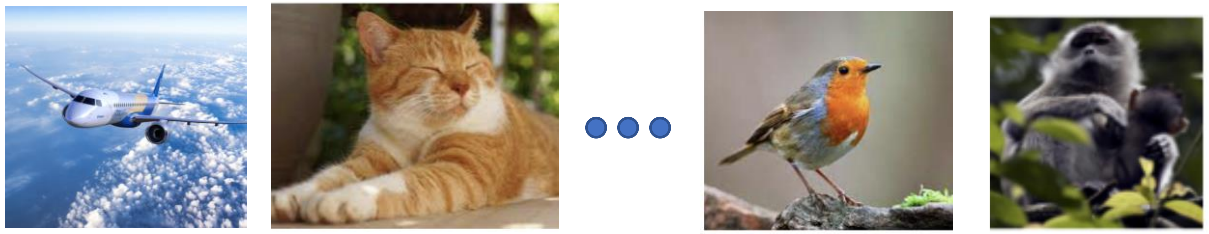
(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

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

x^+

Positive example

Random set of other images

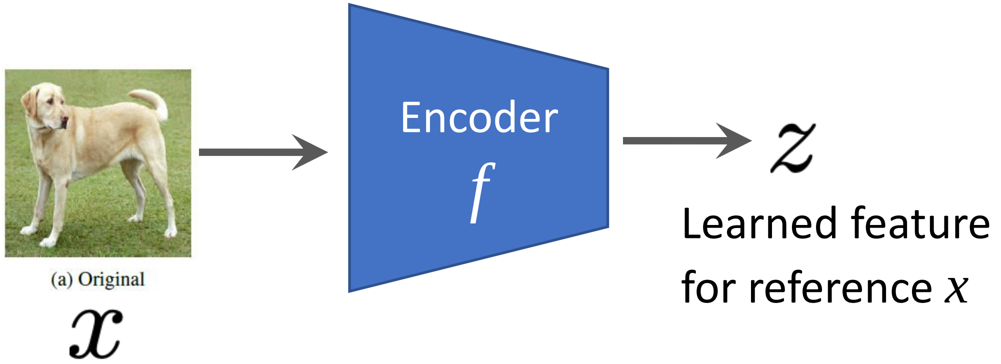


x^-

Negative example

Chen et. al., A Simple Framework for Contrastive Learning of Visual Representations, 2020

SimCLR



Learn an encoder function *f* such that:

$$sim(z, z_i^+) >> sim(z, z_j^-)$$

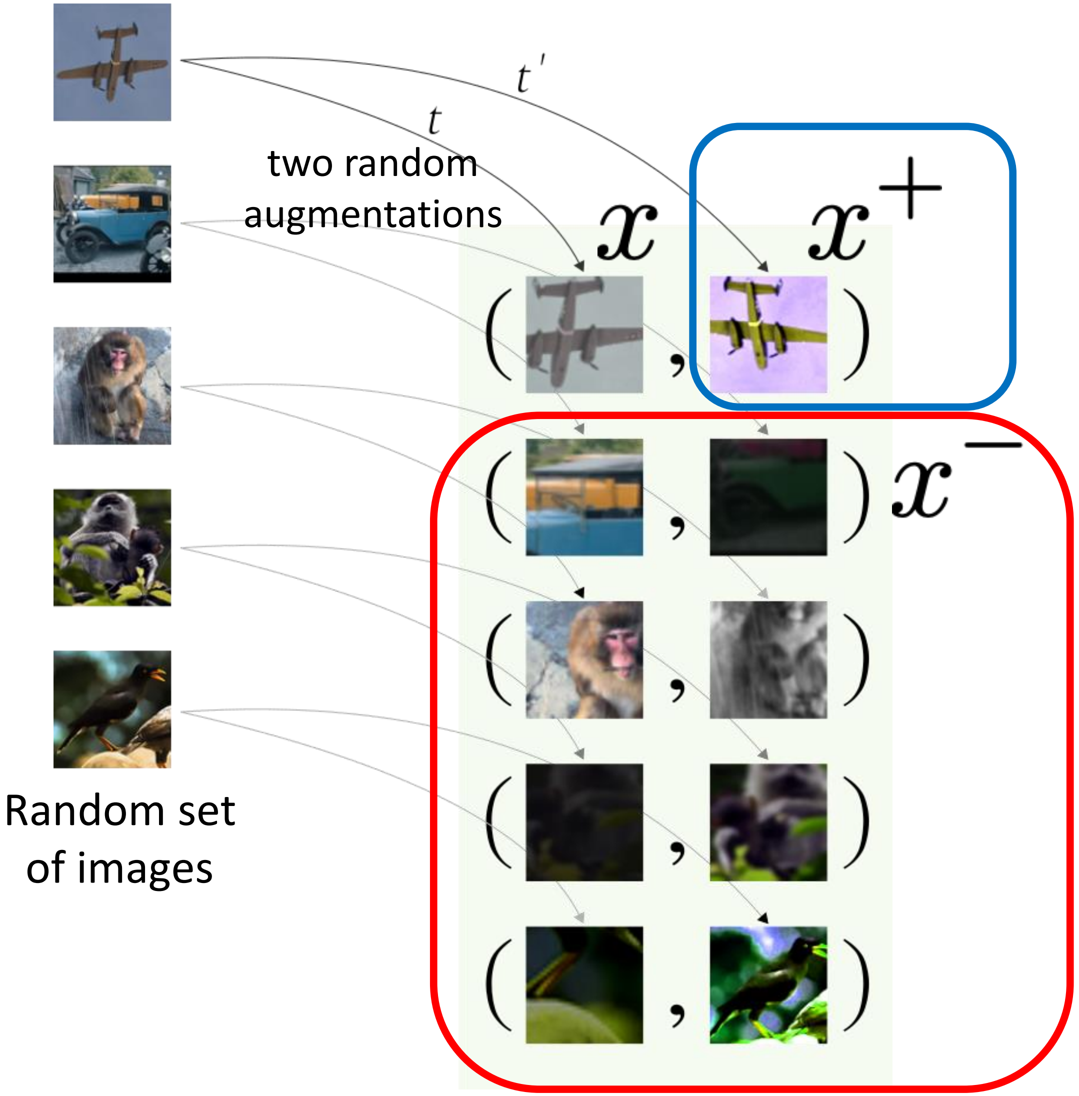
$f(x_i^+) = z_i^+$ Learned feature for positive example

$f(x_j^-) = z_j^-$ Learned feature for negative example

Chen et. al., A Simple Framework for Contrastive Learning of Visual Representations, 2020

SimCLR: working with mini-batches

For each example x, we take 1 positive example and 2(N-1) negative examples:



Source: https://sthalles.github.io/simple-self-supervised-learning/

Training Loss: Contrastive Learning formulation

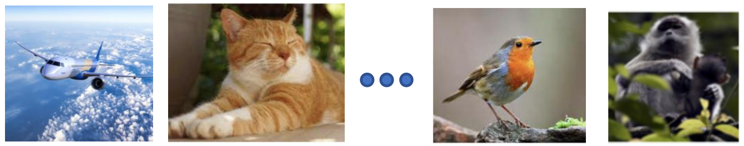
For each example *x*, we take 1 positive example and 2(N-1) negative examples:

(a) Original

x

x^+

Positive example



x^-

Negative examples

$$\mathcal{L} = -E_X \left[\log \frac{\exp\left(sim(z, z_i^+)\right)}{\exp\left(sim(z, z_i^+)\right) + \sum_{j=1}^{N-1} \exp\left(sim(z, z_j^-)\right)} \right]$$

$f(x) = z$ Learned feature for the refernce

$f(x_i^+) = z_i^+$ Learned feature for positive example

$f(x_j^-) = z_j^-$ Learned feature for negative example

Chen et. al., A Simple Framework for Contrastive Learning of Visual Representations, 2020

Training Loss: Contrastive Learning formulation

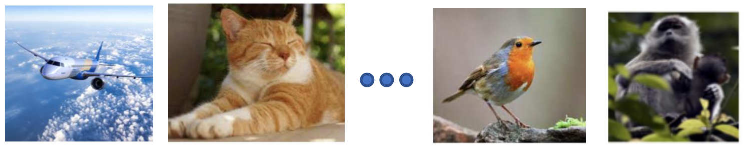
For each example x, we take 1 positive example and 2(N-1) negative examples:

(a) Original

x

x^+

Positive example



x^-

Negative examples

$$\mathcal{L} = -E_X \left[\log \frac{\exp\left(sim(z, z_i^+)\right)}{\exp\left(sim(z, z_i^+)\right) + \sum_{j=1}^{N-1} \exp\left(sim(z, z_j^-)\right)} \right]$$

Score for positive pair

Scores for all negative pairs

Cross entropy loss for N-way softmax classifier
(“classes” are the positive and negative examples)

$f(x) = z$ Learned feature for the refernce

$f(x_i^+) = z_i^+$ Learned feature for positive example

$f(x_j^-) = z_j^-$ Learned feature for negative example

Chen et. al., A Simple Framework for Contrastive Learning of Visual Representations, 2020

Training Loss: Contrastive Learning formulation

For each example x, we take 1 positive example and 2(N-1) negative examples:

$$\mathcal{L} = -E_X\left[\log \frac{\exp\left(sim(z, z_i^+)\right)}{\underbrace{\exp\left(sim(z, z_i^+)\right)}_{\text{Score for positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp\left(sim(z, z_j^-)\right)}_{\text{Scores for all negative pairs}}}\right]$$

Commonly used loss in **Contrastive Learning**, also known as:

- Noise-Contrastive Estimation (NCE) loss
- InfoNCE loss
- Contrastive cross-entropy loss

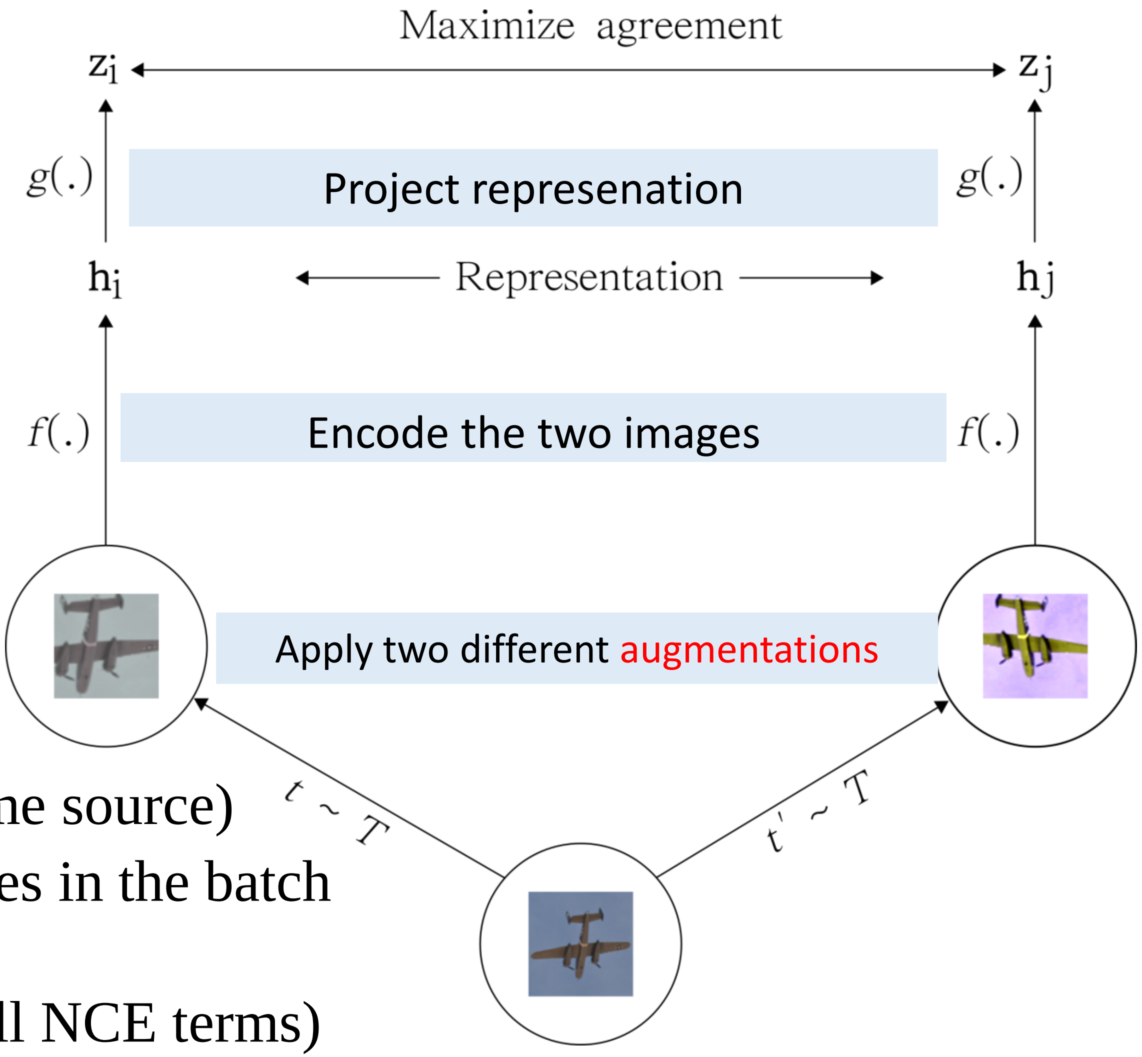
$$\mathrm{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right) = \frac{\boldsymbol{z}_i^T \boldsymbol{z}_j}{\|\boldsymbol{z}_i\| \|\boldsymbol{z}_i\|}$$

Cosine similarity between the features

Sohn, Improved Deep Metric Learning with Multi-class N-pair Loss Objective, 2016

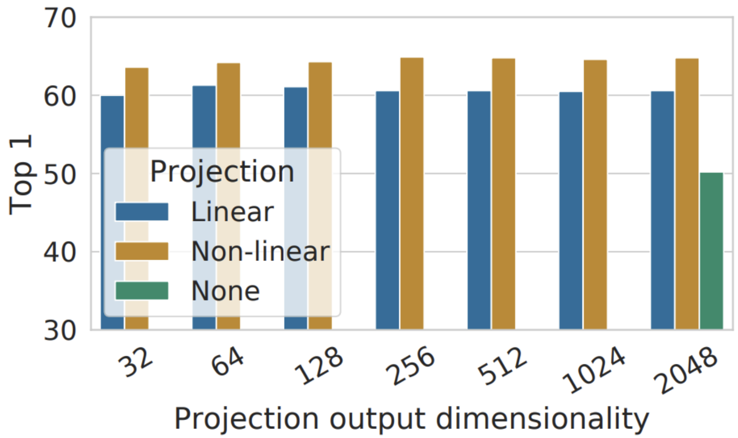
SimCLR Framework

```
Repeat:
    Randomly sample a N size mini batch
    for each sample x do:
        (1) Apply two augmentations $t, t'$ on $\boldsymbol{x}$:
            $\tilde{\boldsymbol{x}}_i = t(\boldsymbol{x})$ and $\tilde{\boldsymbol{x}}_j = t'(\boldsymbol{x})$
        (2) Compute latent representation:
            $\boldsymbol{h}_i = f(\tilde{\boldsymbol{x}}_i)$ and $\boldsymbol{h}_j = f(\tilde{\boldsymbol{x}}_j)$
        (3) Project using projection head g:
            $\boldsymbol{z}_i = g(\boldsymbol{h}_i)$ and $\boldsymbol{z}_j = g(\boldsymbol{h}_j)$
    end for
    Positive example: $z_i$ and $z_j$ (augmentations of the same source)
    Negative examples: all other 2(N-1) augmented images in the batch
    Compute the NCE loss for all positive pairs
    Update $g$ and $f$ to minimize the total loss (sum over all NCE terms)
return encoder network $f(\cdot)$, and throw away $g(\cdot)$
```



SimCLR Design Choices

- Projection head improves the learned representation for downstream tasks:



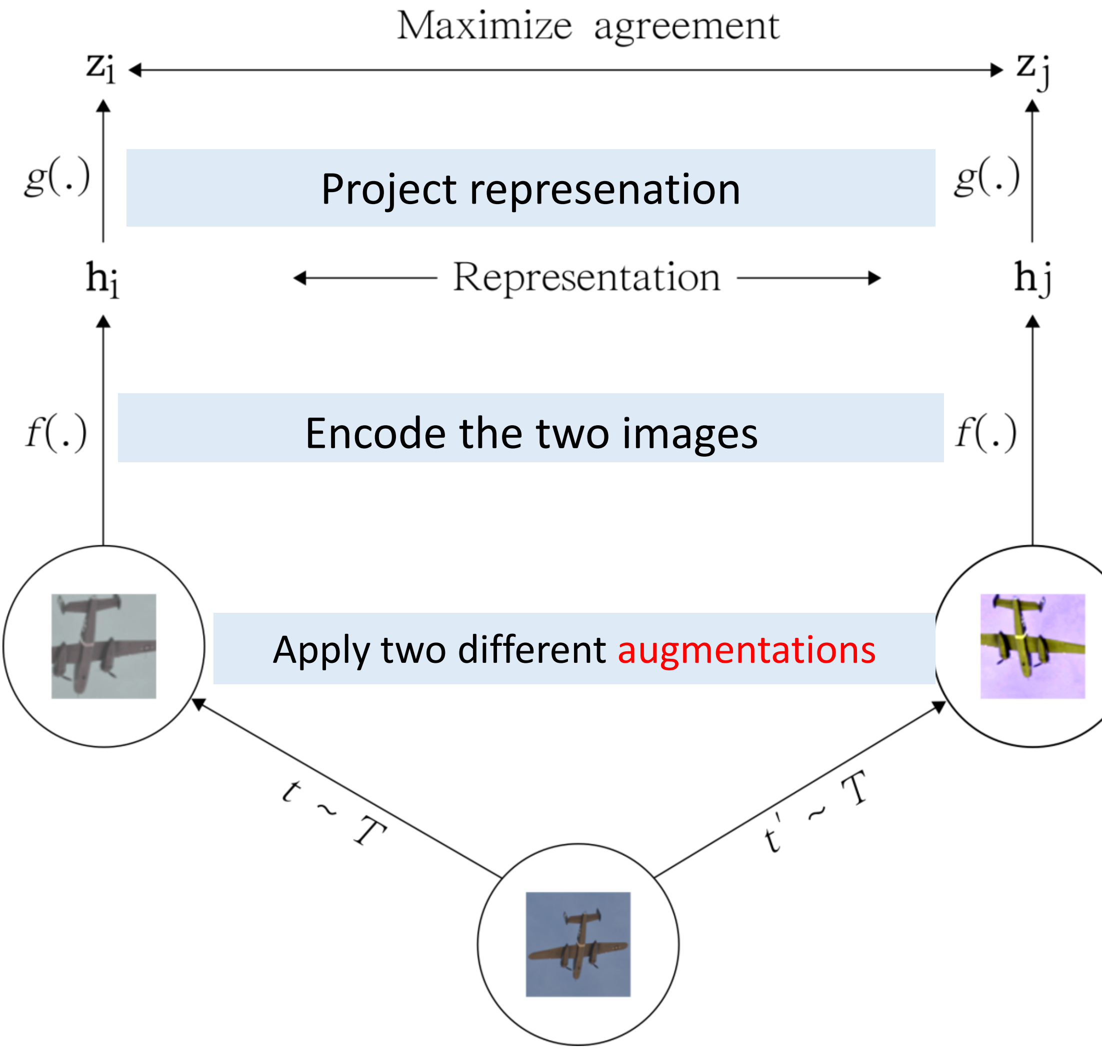
- Large training batch size is crucial

  Large memory; requires distributed training on TPUs

  He et. al, **Momentum Contrast for Unsupervised Visual Representation Learning (MoCo), CVPR 2020**

  - Decouples batch size and number of negative samples
  - Running queue of negative examples

    MoCo-V2, MoCo-V3…



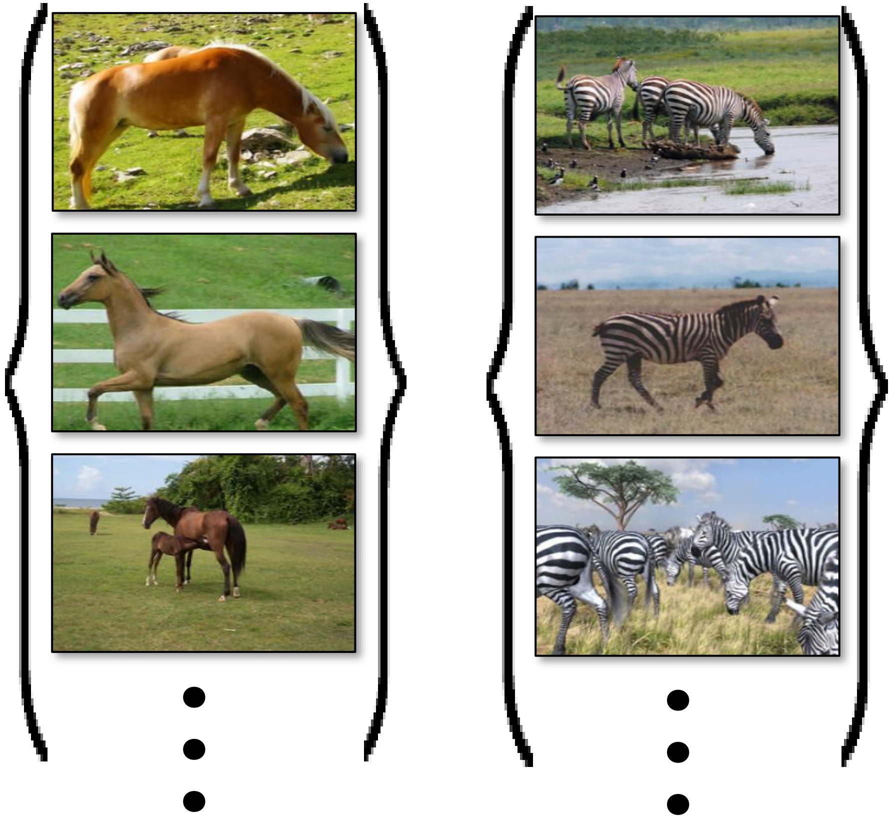
Project represenation

Representation

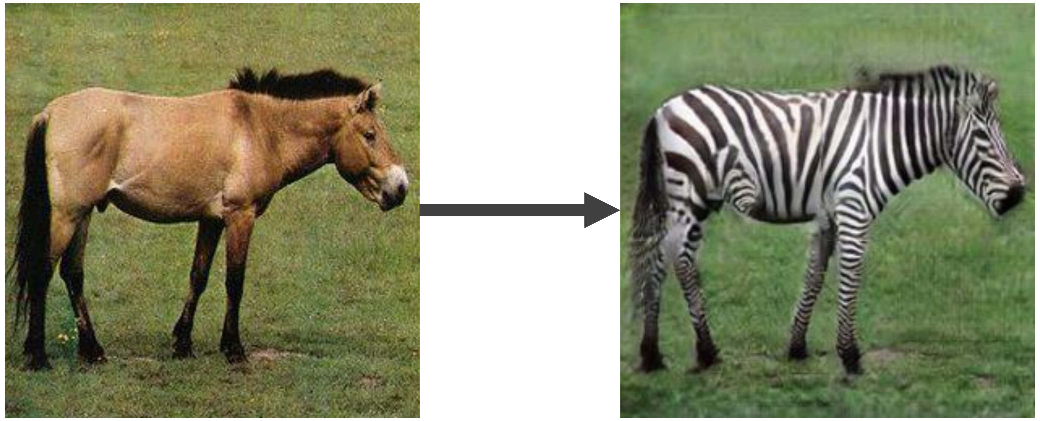
Encode the two images

Apply two different augmentations

Unpaired Image-to-Image Translation



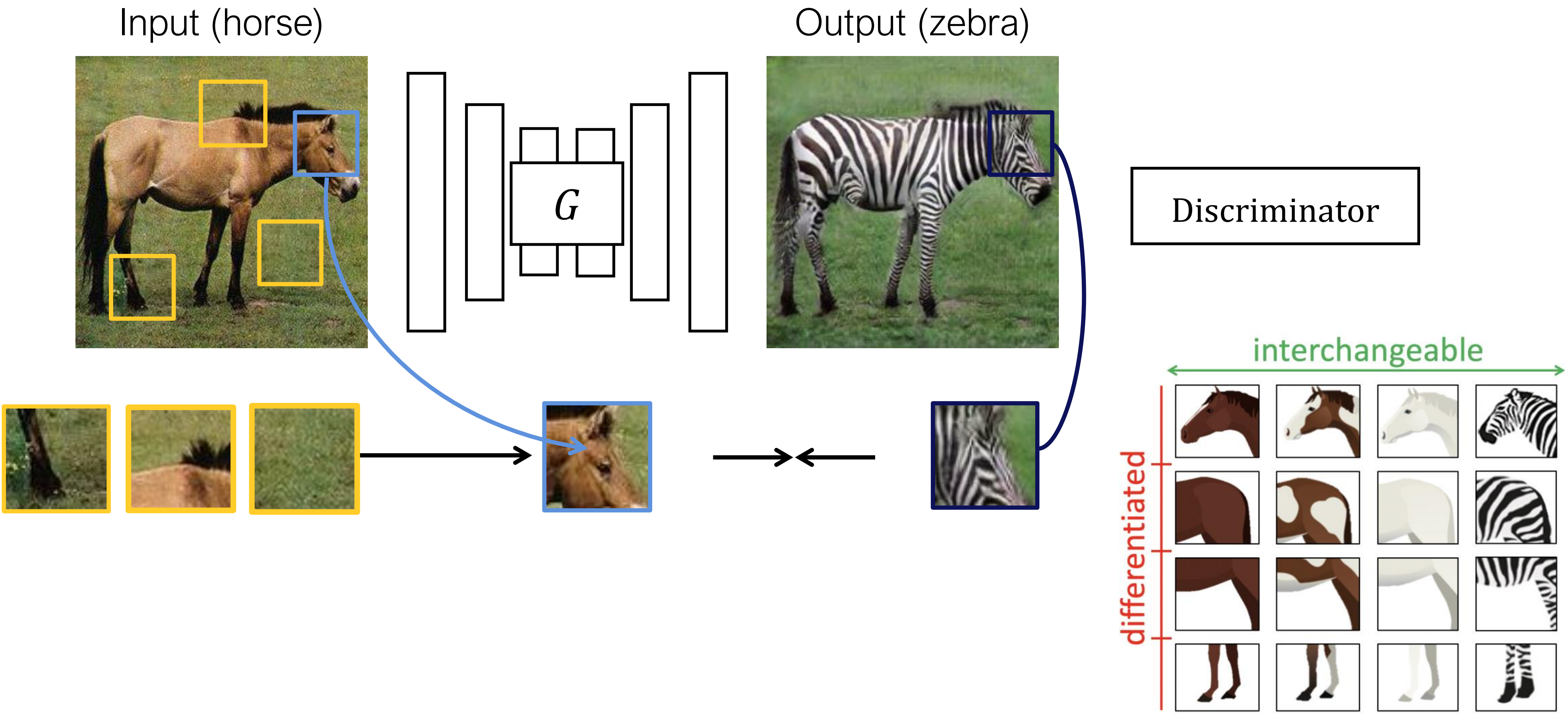
Training Set



Test-time behavior

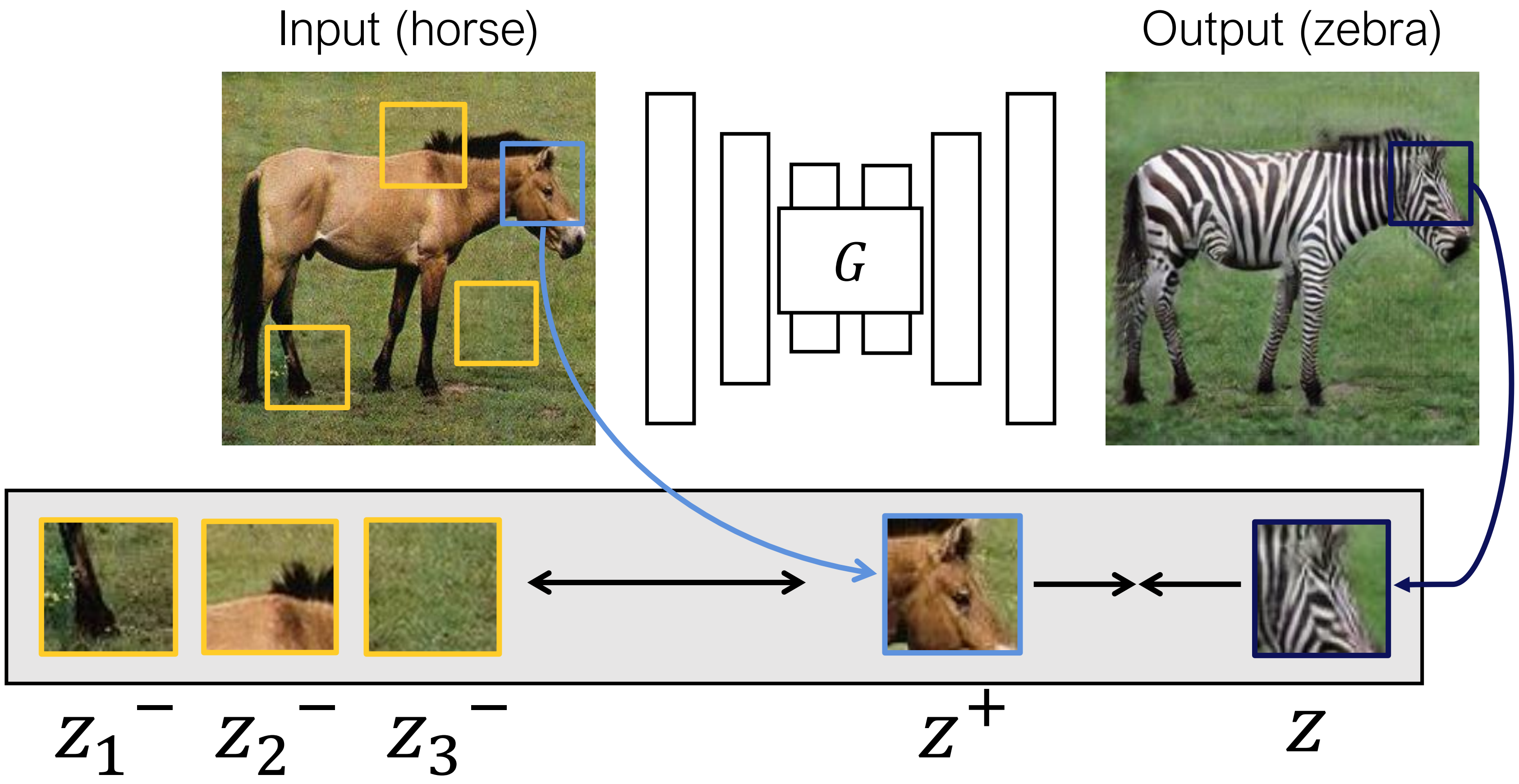
Park et. al., Contrastive Learning for Unpaired Image-to-Image Translation, ECCV 2020

Unpaired Image-to-Image Translation via Contrastive Learning



Park et. al., Contrastive Learning for Unpaired Image-to-Image Translation, ECCV 2020

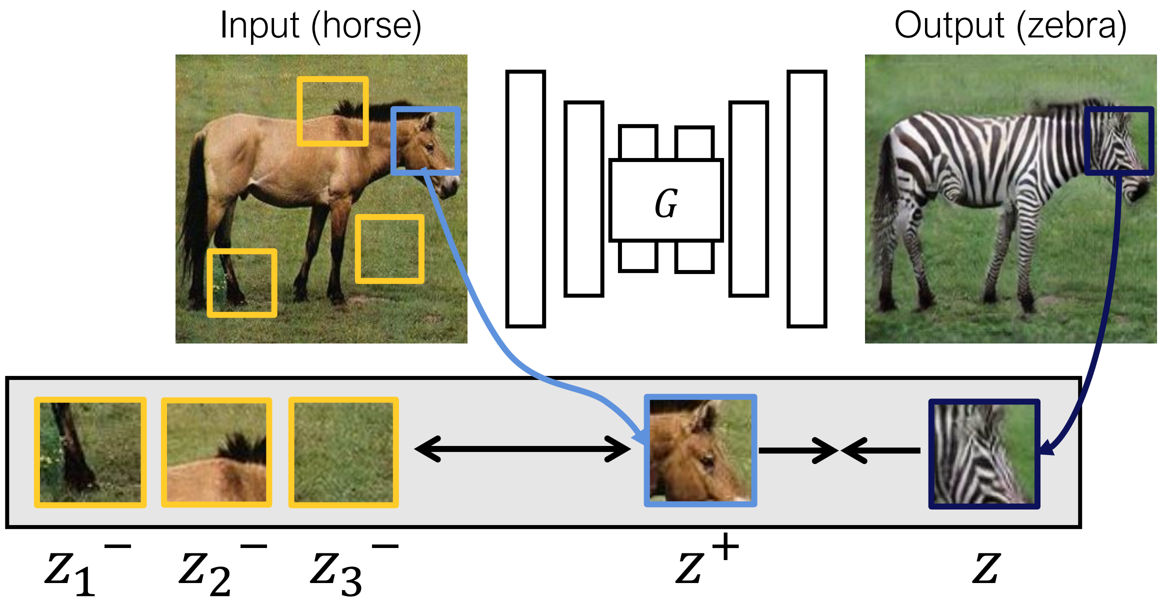
Unpaired Image-to-Image Translation via Contrastive Learning



Corresponding patches should have **high similarity**

Park et. al., Contrastive Learning for Unpaired Image-to-Image Translation, ECCV 2020

Patch-based Contrastive Loss



$$\text{softmax}\begin{pmatrix} \uparrow z \cdot z^+ / \tau \\ \downarrow z \cdot z_1^- / \tau \\ \downarrow z \cdot z_2^- / \tau \\ \vdots \\ \downarrow z \cdot z_N^- / \tau \end{pmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

softmax (cosine similarities $/\tau$)

τ=0.07

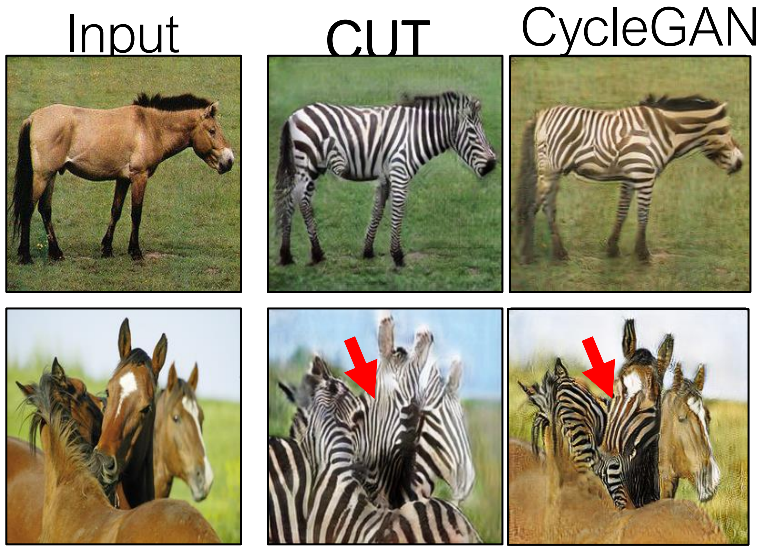
- Use the same InfoNCE loss as in MoCo and SimCLR

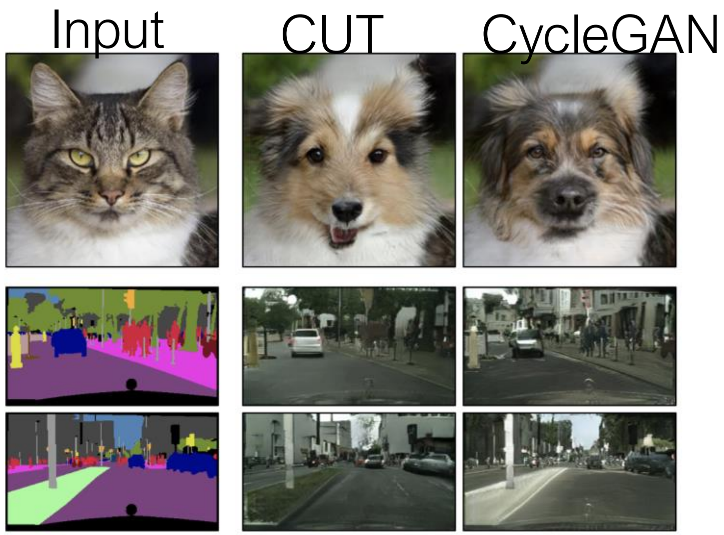
To produce positive pairs:

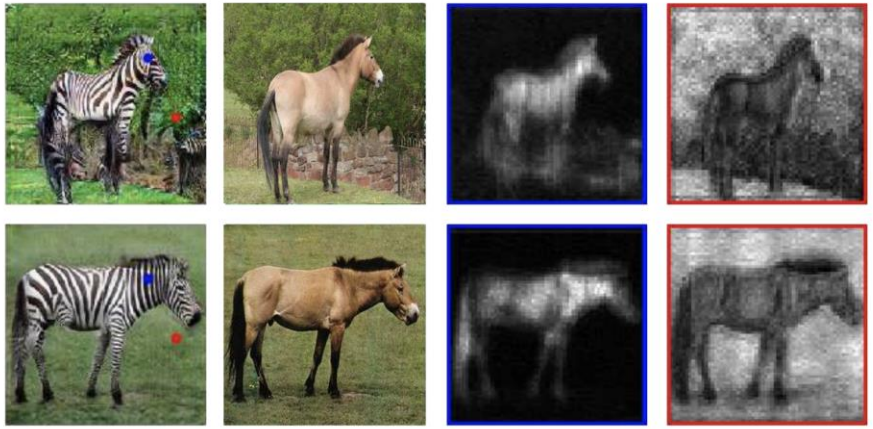
- Handcrafted data augmentation (MoCo, SimCLR, etc.) vs. Input and synthesized images

MoCo: He et al., CVPR20, SimCLR: Chen et al., ICML20

Patch-based Contrastive Loss







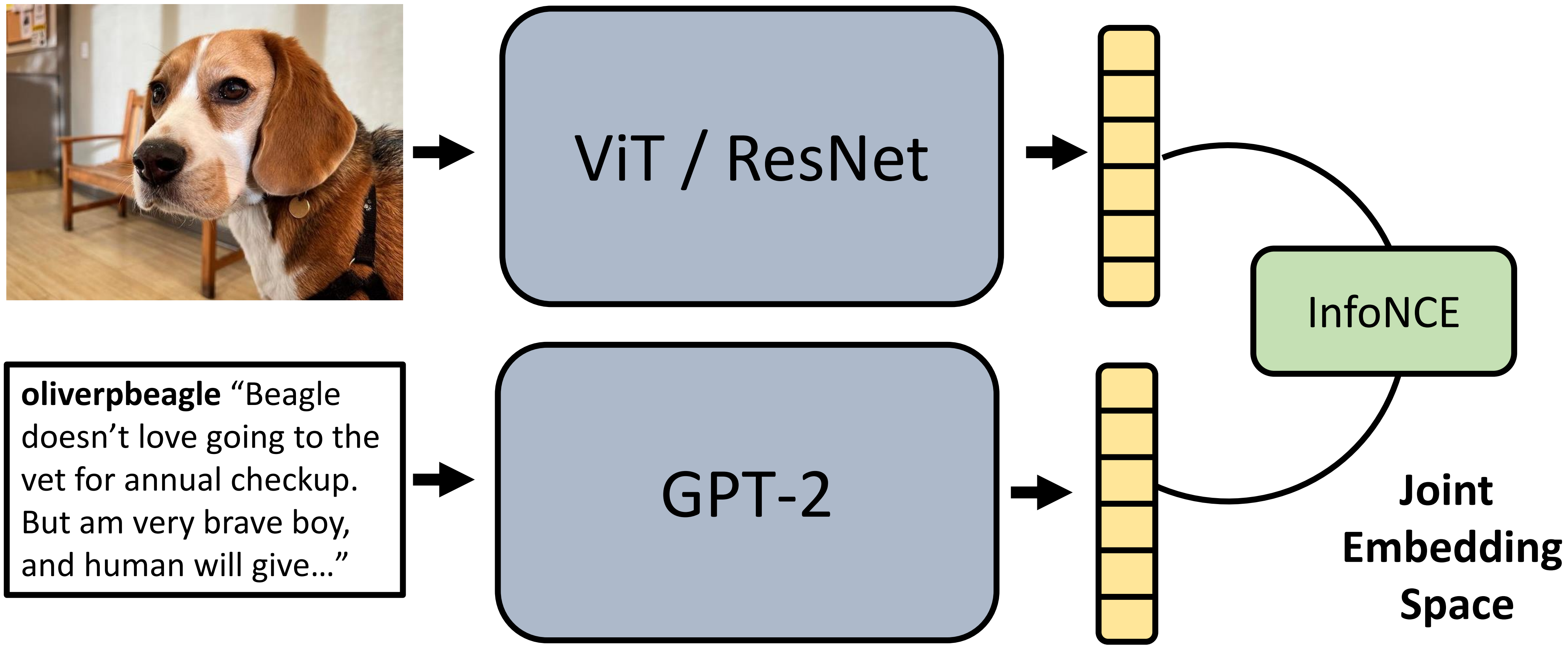
(a) Translated $\hat{y}$ & query points

(b) Input image x

(c) Learned similarity from query points to input image x

Park et. al., Contrastive Learning for Unpaired Image-to-Image Translation, ECCV 2020

CLIP – Connecting Images and Text (Open-AI)



oliverpbeagle “Beagle doesn’t love going to the vet for annual checkup. But am very brave boy, and human will give…”

ViT / ResNet

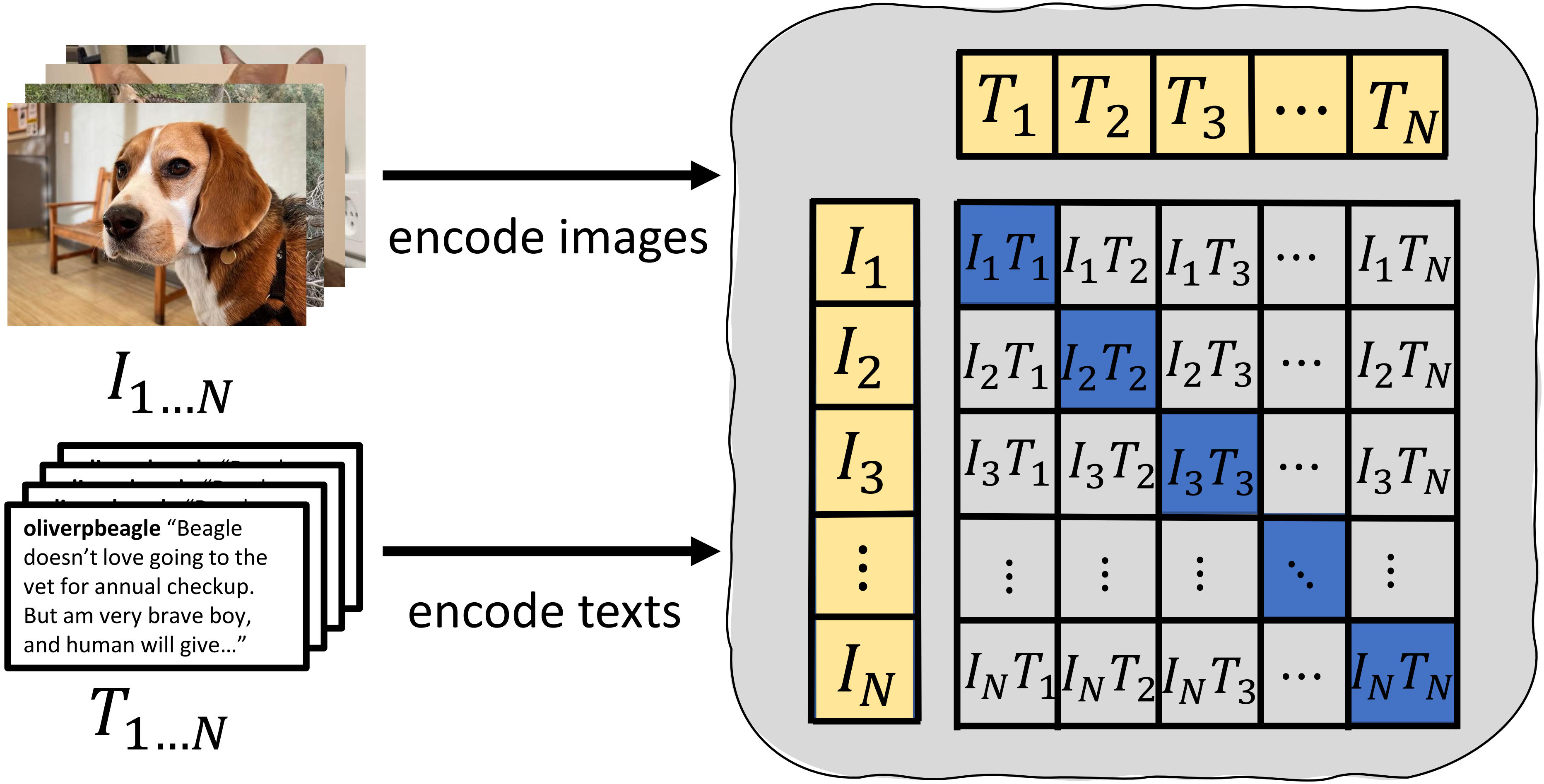
GPT-2

InfoNCE

Joint Embedding Space

Radford et. al, Learning Transferable Visual Models From Natural Language Supervision, ArXiv’21
slide credit: Shir Amir

CLIP – Connecting Images and Text (Open-AI)



$I_{1\ldots N}$

encode images

oliverpbeagle “Beagle doesn’t love going to the vet for annual checkup. But am very brave boy, and human will give…”

$T_{1\ldots N}$

encode texts

$$\begin{array}{c|ccccc} & T_1 & T_2 & T_3 & \cdots & T_N \\ \hline I_1 & I_1T_1 & I_1T_2 & I_1T_3 & \cdots & I_1T_N \\ I_2 & I_2T_1 & I_2T_2 & I_2T_3 & \cdots & I_2T_N \\ I_3 & I_3T_1 & I_3T_2 & I_3T_3 & \cdots & I_3T_N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ I_N & I_NT_1 & I_NT_2 & I_NT_3 & \cdots & I_NT_N \end{array}$$

joint embedding space

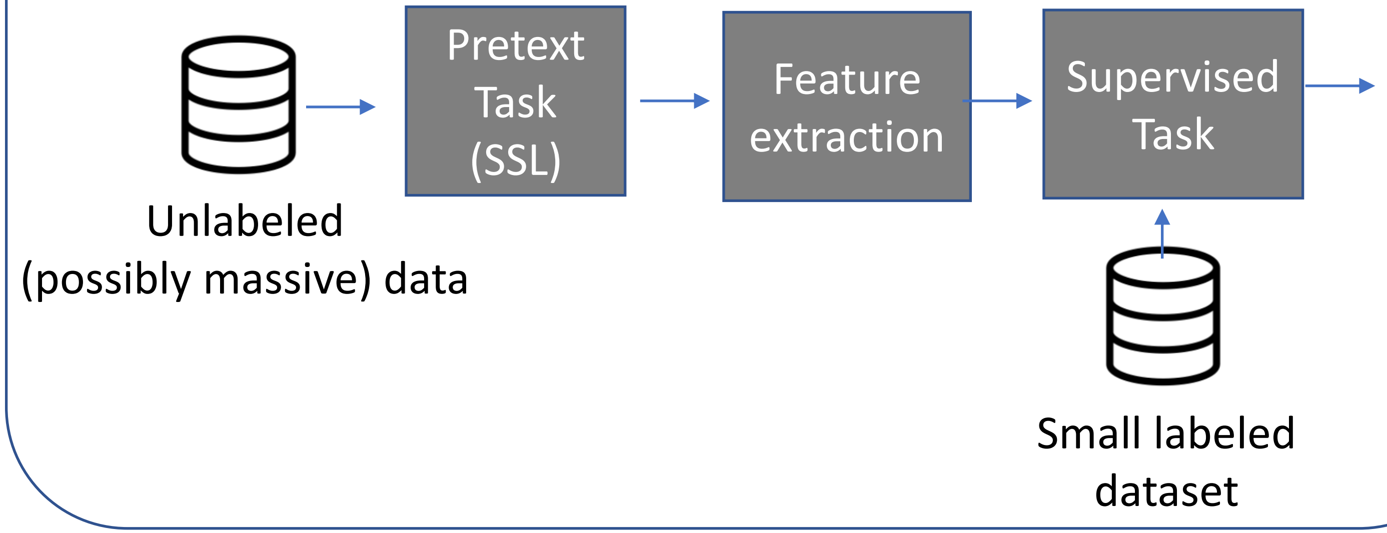
Radford et. al, Learning Transferable Visual Models From Natural Language Supervision, ArXiv’21
slide credit: Shir Amir

Self-Supervised Learning (SSL)

No human labels; supervisory signals are automatically computed from data

In-direct self-supervised methods

Solve a proxy, pretext task → extract learned features → finetune on a target supervised task (Transfer Learning)



Direct self-supervised methods

Train directly for the task in hand:

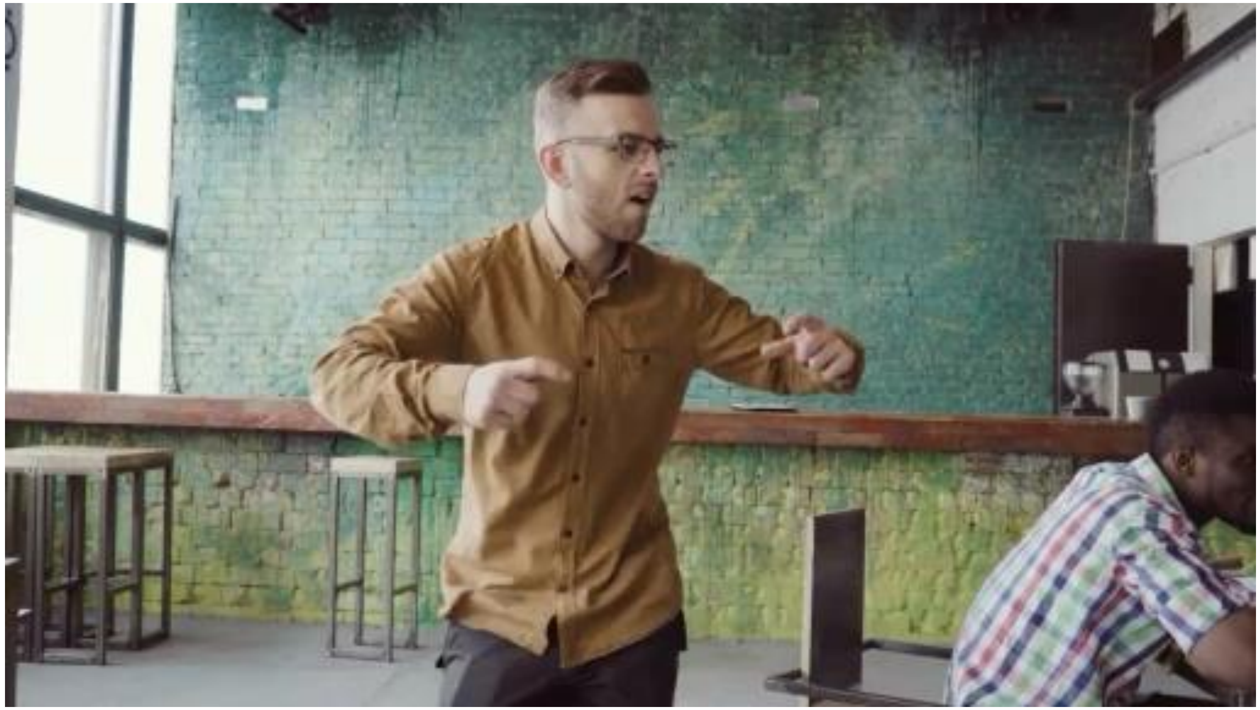
Examples you've seen:

- Generative models
- ZSSR
- Cocourrance of signals (e.g., captions and images)

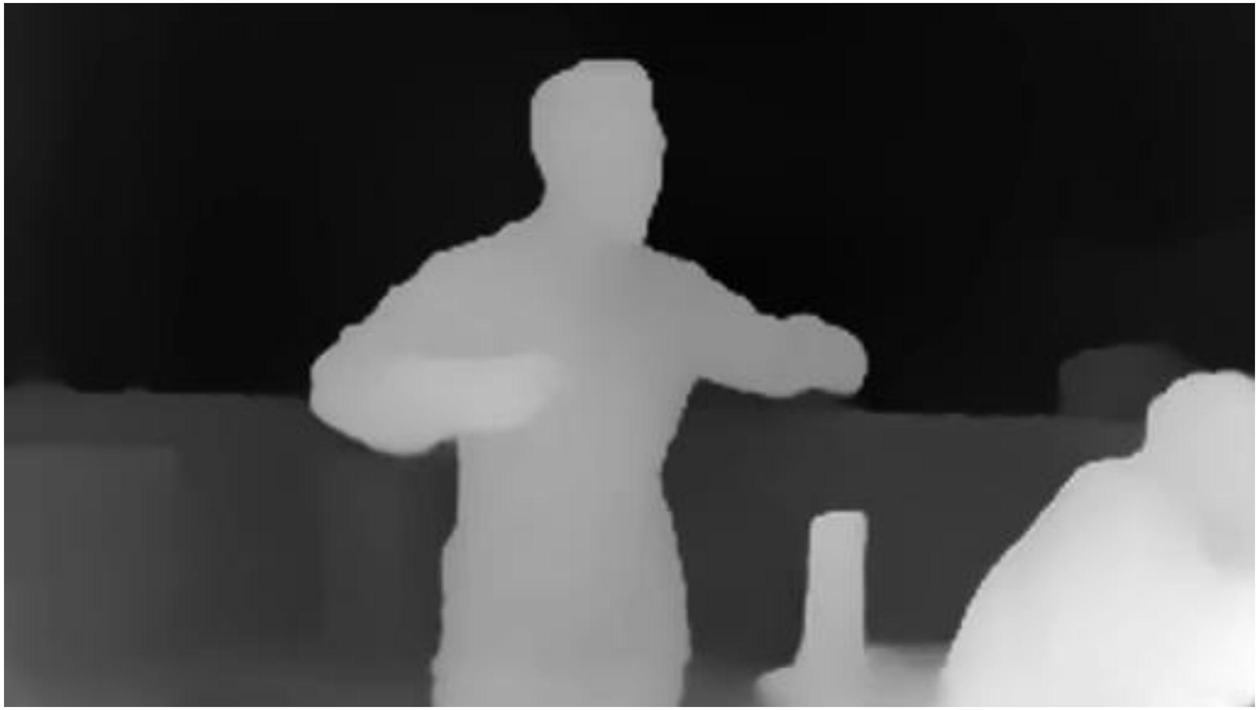
More advanced signals:

- Apply computer vision methdologies to extract supervion

Goal: Predict depth when both camera and people are moving



Input



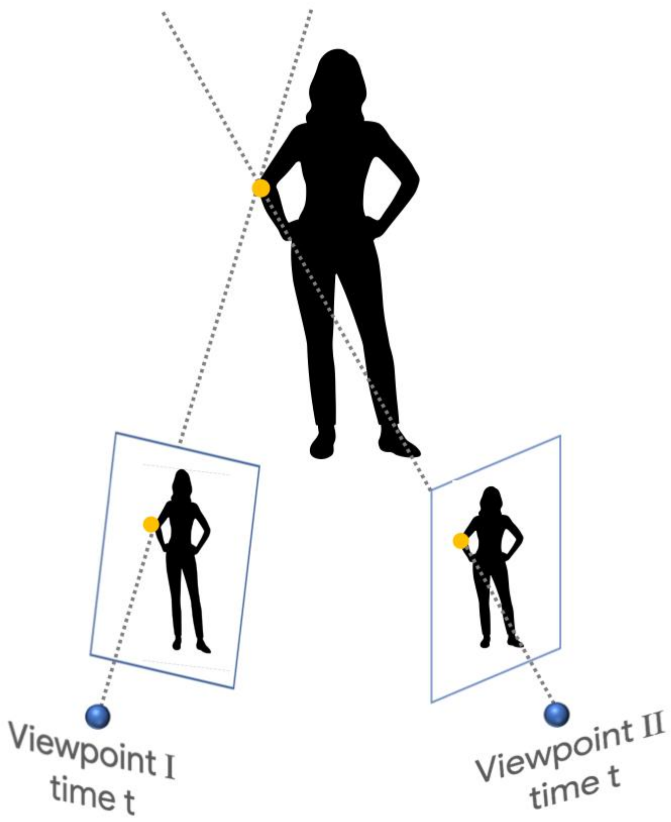
Our depth predictions*

Joint work with: Zhengqi Li, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, Bill Freeman

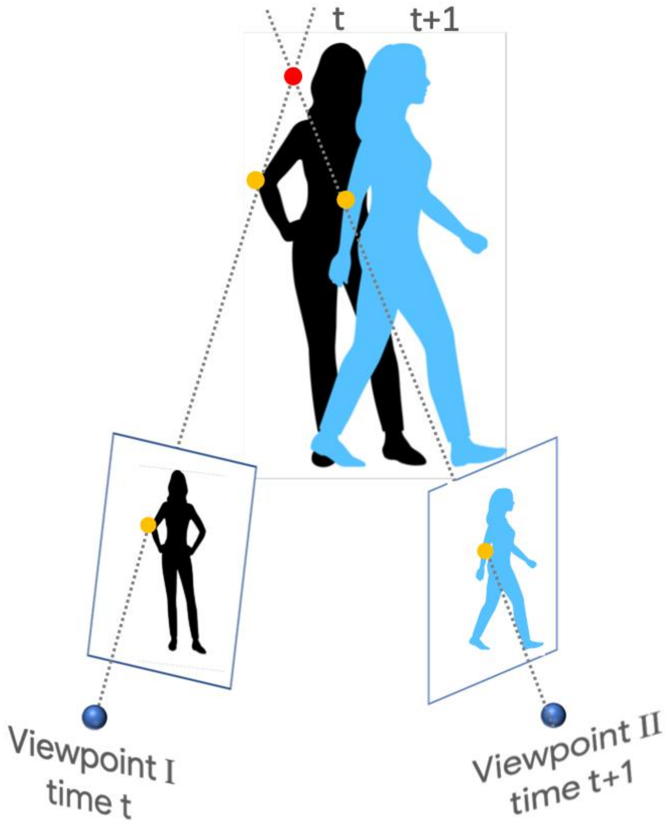
"Learning the Depths of Moving People by Watching Frozen People", CVPR'19 , **Best Paper Honorable Mention**

Challenge: geometric constraints do not hold

Traditional Stereo



Our use case



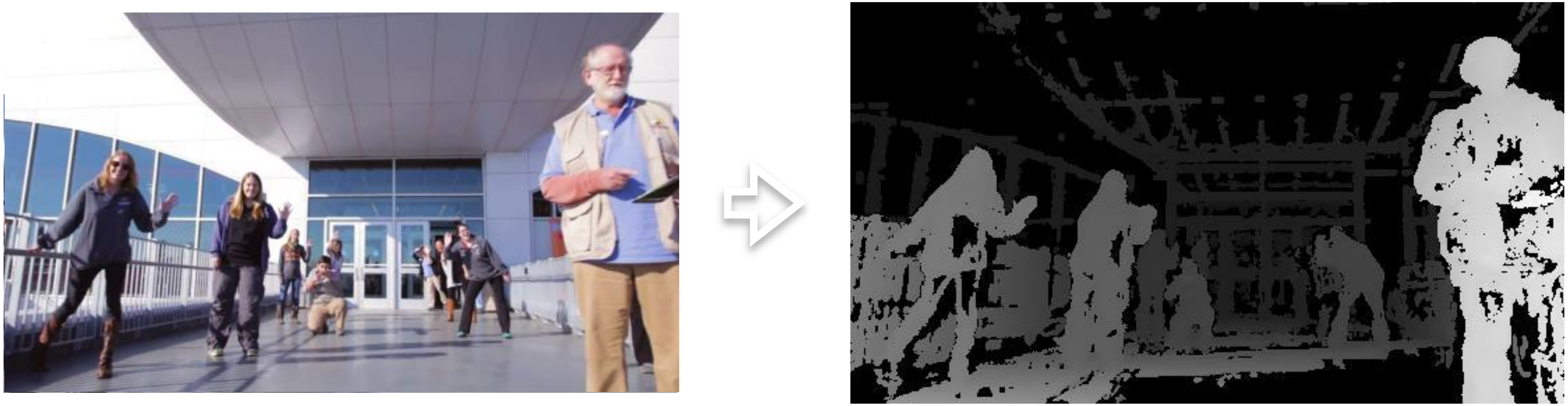
Approach:
Learn the depths of moving people by watching frozen people

MannequinChallenge Dataset:

- 2000 YouTube Videos
- People frozen while camera is moving
- Diverse scenes, natural human poses

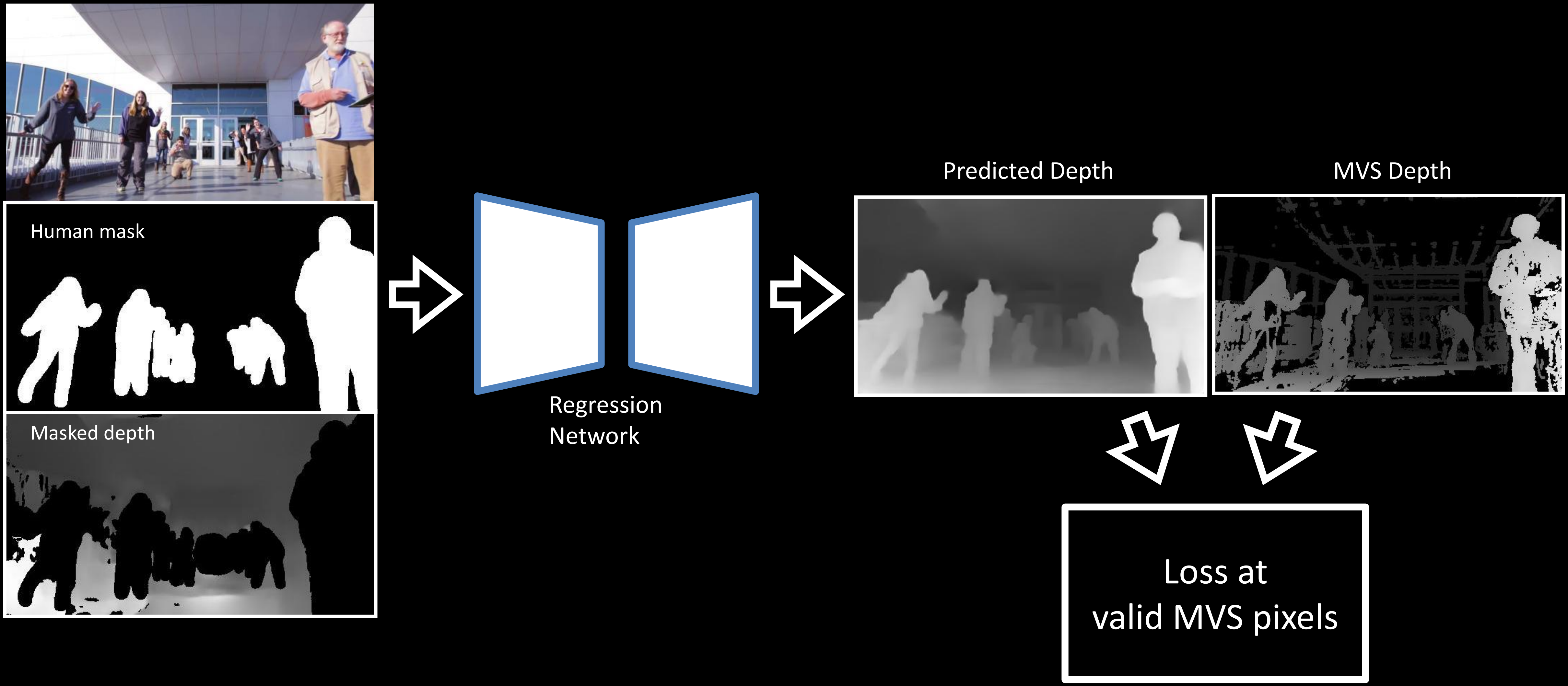
WAIC

MannequinChallenge Training Data



“Ground truth” depth from SfM + Multi View Stereo (MVS)

Training Setup



Human mask

Masked depth

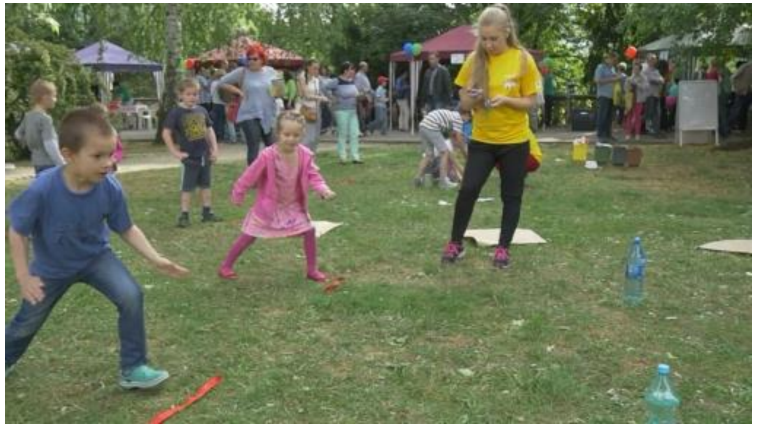
Regression Network

Predicted Depth

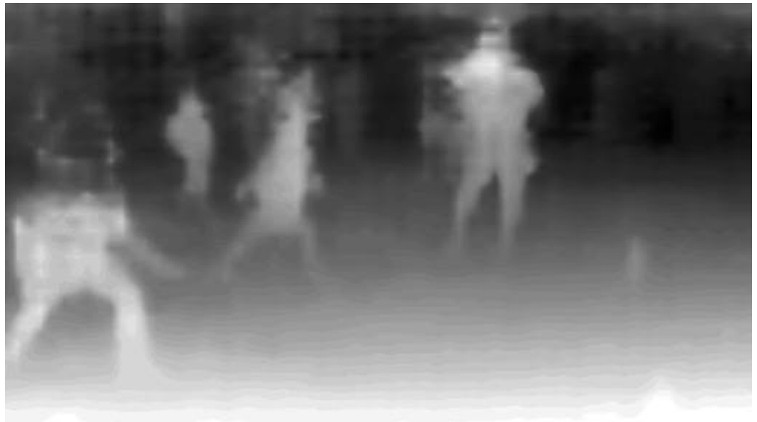
MVS Depth

Loss at valid MVS pixels

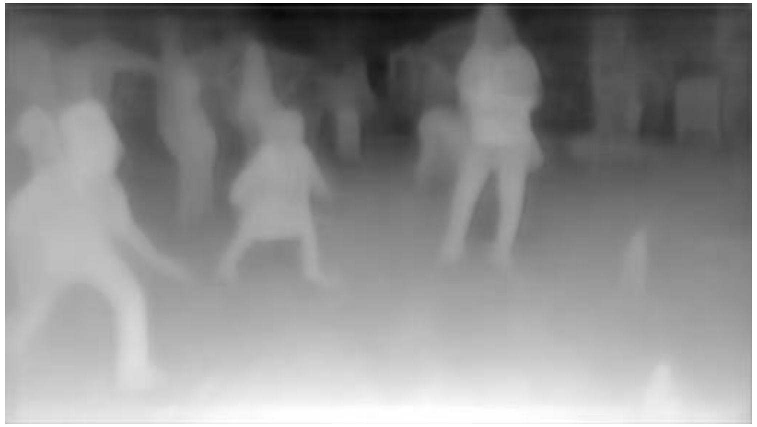
Results and Comparison for Moving People



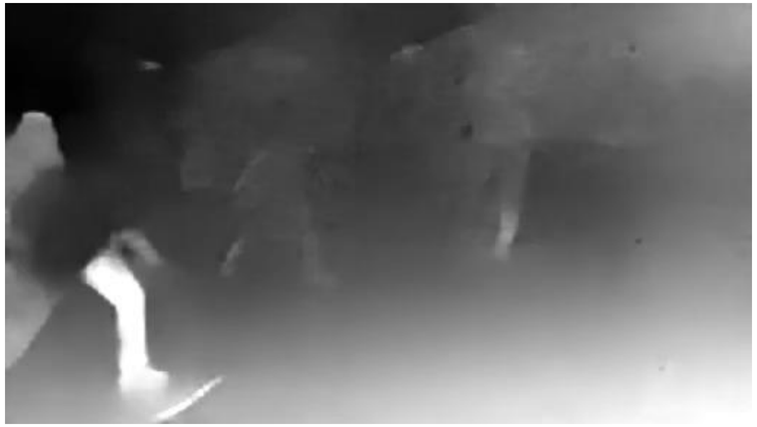
Input sequence



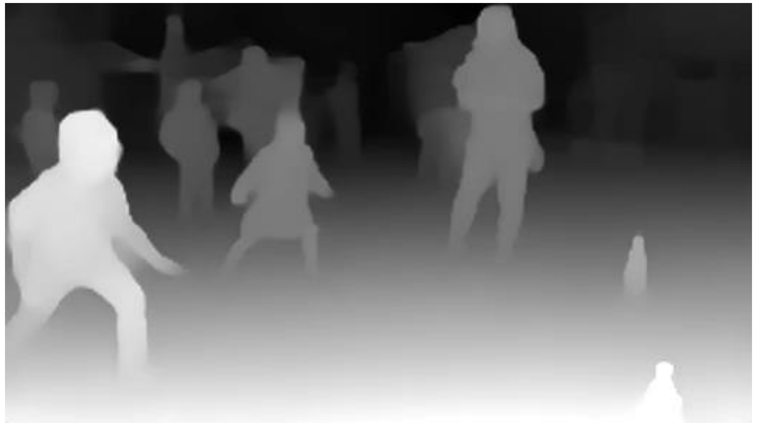
DORN (monocular)



Chen *et al*. (monocular)



DeMoN (stereo)

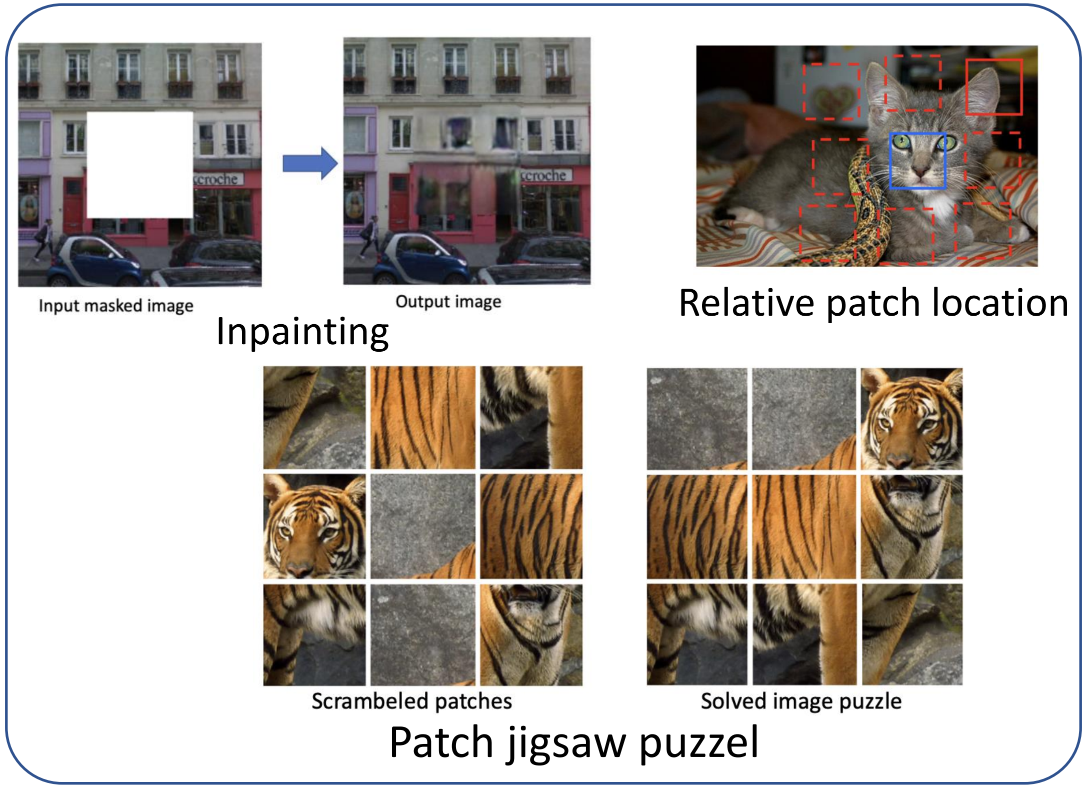


Ours

Comparison to recent SOTA learning based depth prediction methods

Self-Supervised Learning

Solve a proxy, pretext task (large dataset) → extract learned features → finetune on a target supervised task (smaller dataset)



Input masked image

Output image

Inpainting

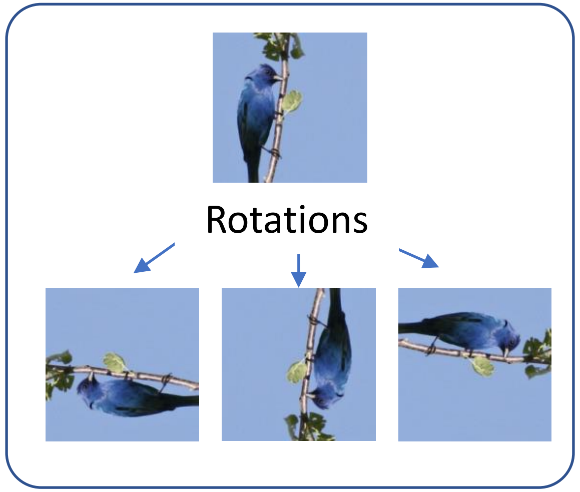
Relative patch location

Scrambeled patches

Solved image puzzle

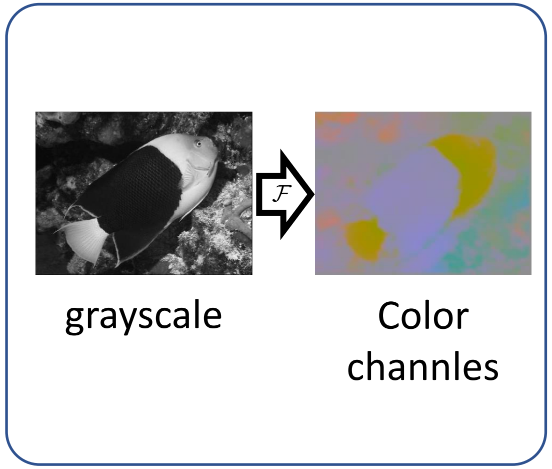
Patch jigsaw puzzel

Image context as supervision



Rotations

Geometric transformations



grayscale

Color channles

Color transformations

Next class:

“More self-supervision”