Computer Vision

Lecture 10 – Self-Supervised Learning

Kumar Bipin

BE, MS, PhD (MMMTU, IISc, IIIT-Hyderabad)

Robotics, Computer Vision, Deep Learning, Machine Learning, System Software

Agenda

11.1 Preliminaries

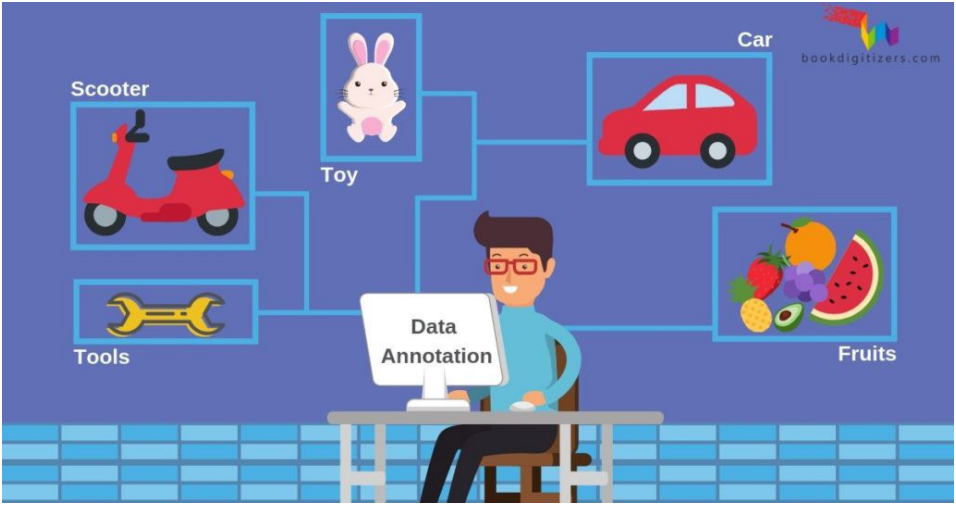
11.2 Task-specific Models

11.3 Pretext Tasks

11.4 Contrastive Learning

11.1
Preliminaries

Data Annotation



- Supervised learning requires very large amounts of annotated data!

Data Annotation

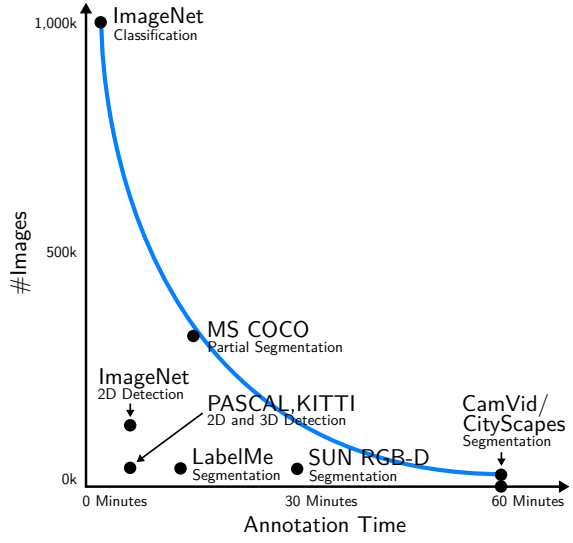


Image Classification

Types of human error:

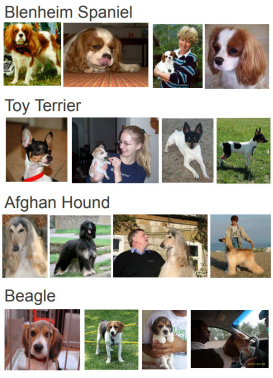
- **Fine-grained recognition.** There are more than 120 species of dogs in the dataset. We estimate that 28 (37%) of the human errors fall into this category.
- **Class unawareness.** The annotator may sometimes be unaware of the ground truth class present as a label option $\Rightarrow$ 18 (24%) of the human errors.
- **Insufficient training data.** The annotator is only presented with 13 examples of a class under every category name which is insufficient for generalization. Approximately 4 (5%) of human errors fall into this category.

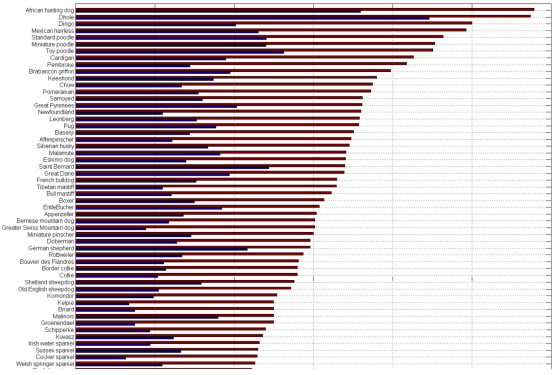
Moreover, image classification is expensive: ImageNet took **22 human years**

`http://karpathy.github.io/2014/09/02/`

`what-i-learned-from-competing-against-a-convnet-on-imagenet/`

Stanford Dog Dataset





- Subset of ImageNet: **120 dog categories**
- `http://vision.stanford.edu/aditya86/ImageNetDogs/`

Khosla, Jayadevaprakash, Yao and Fei-Fei: Novel dataset for Fine-Grained Image Categorization. Workshop on Fine-Grained Visual Categorization, CVPR, 2011.

Dense Semantic and Instance Annotation

- Cityscapes dataset: dense semantic and instance annotation
- Annotation time **90 minutes** per image, multiple annotators

Cordts et al.: The Cityscapes Dataset for Semantic Urban Scene Understanding. CVPR, 2016.

Stereo, Monodepth and Optical Flow

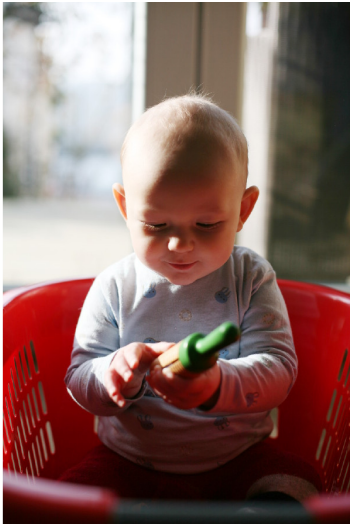
- KITTI dataset: stereo, monocular depth, optical flow, scene flow, …
- Correspondences and depth are extremely difficult to annotate for humans
- KITTI labels data using **LiDAR** and **manual object segmentation/tracking**

Geiger, Lenz and Urtasun: Are we ready for autonomous driving? The KITTI vision benchmark suite. CVPR, 2012.

Human labeling is sparse

- Provided only very few “labeled” examples, **humans generalize very well**
- Humans learn through **interaction** and **observation**

Humans learn through interaction and observation



Self-Supervision

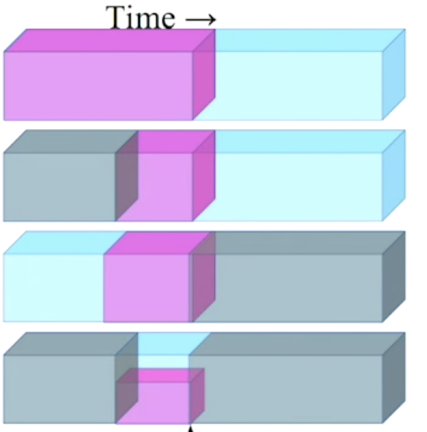
Idea of self-supervision:

- Obtain labels from raw unlabeled data itself
- Predict parts of the data from other parts

Slide credits: Yann LeCun and Ishan Misra

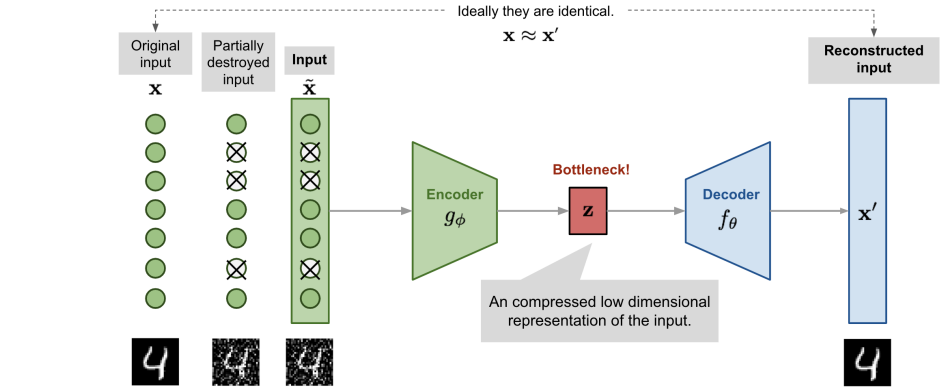
Self-Supervision

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible
- Pretend there is a part of the input you don't know and predict that.



Slide credits: Yann LeCun and Ishan Misra

Example: Denoising Autoencoder



- Example: **Denoising Autoencoder (DAE)** predicts input from corrupted version
- After training, only the encoder is kept and the decoder is thrown away

Vincent, Larochelle, Bengio and Manzagol: Extracting and composing robust features with denoising autoencoders. ICML, 2008.

Learning Problems

- **Reinforcement learning**
  - Learn model parameters using **active exploration** from sparse rewards
  - Examples: deep q learning, gradient policy, actor critique
- **Unsupervised learning**
  - Learn model parameters using **dataset without labels** $\{x_i\}_{i=1}^{N}$
  - Examples: Clustering, dimensionality reduction, generative models
- **Supervised learning**
  - Learn model parameters using **dataset of data-label pairs** $\{(x_i, y_i)\}_{i=1}^{N}$
  - Examples: Classification, regression, structured prediction
- **Self-supervised learning**
  - Learn model parameters using **dataset of data-data pairs** $\{(x_i, x_i')\}_{i=1}^{N}$
  - Examples: Self-supervised stereo/flow, contrastive learning

Learning Problems

Y. LeCun

How Much Information is the Machine Given during Learning?

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

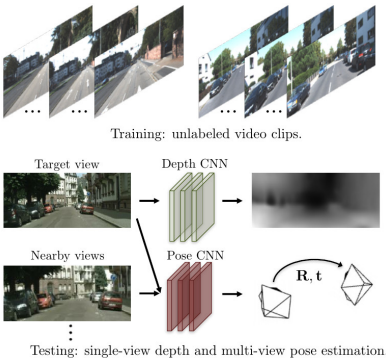
Slide Credit: Yann LeCun.

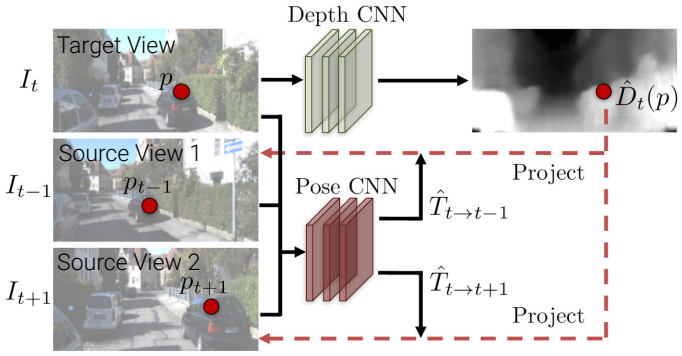
Credits

- **Ishan Misra** — Self-supervised learning in computer vision
  `https://youtu.be/8L10w1KoOU8`, `https://bit.ly/DLSP21-10L`
- **Stanford CS231n** — Convolutional Neural Networks for Visual Recognition
  `http://cs231n.stanford.edu/`
- **Y. LeCun and I. Misra** — Self-supervised learning: Dark matter of intelligence
  `https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence`
- **Lilian Weng** — Self-Supervised Representation Learning
  `https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html`

11.2
Task-specific Models

Unsupervised Learning of Depth and Ego-Motion

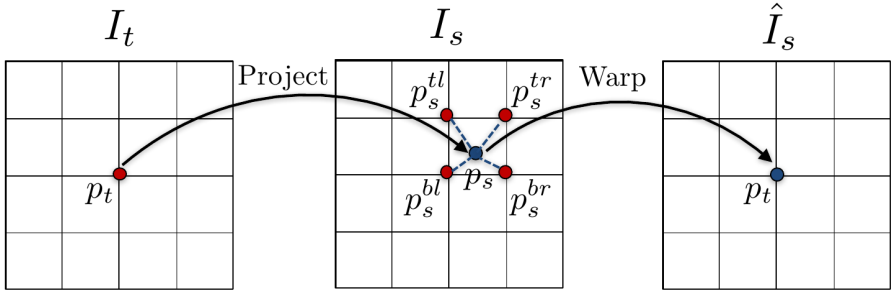




- Train CNN to **jointly predict depth and relative pose** from three video frames
- **Photoconsistency loss** btw. warped source views I_{t-1} / I_{t+1} and target view I_t

Zhou, Brown, Snavely and Lowe: Unsupervised Learning of Depth and Ego-Motion from Video. CVPR, 2017.

Unsupervised Learning of Depth and Ego-Motion



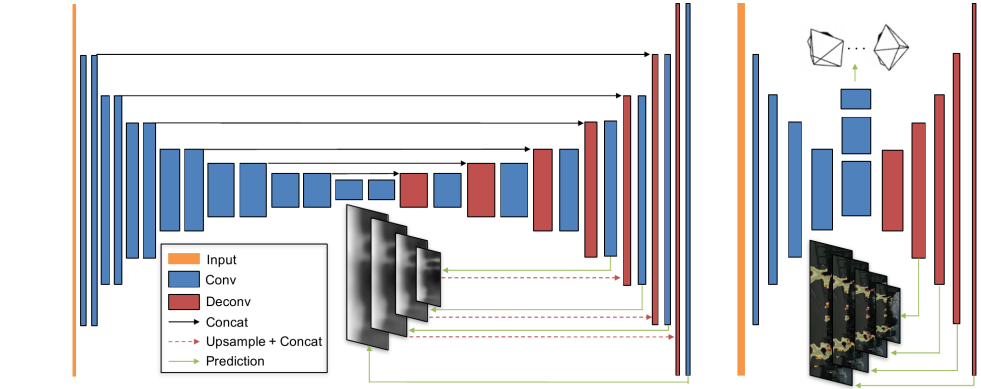
- **Project** each point of target view onto source view based on depth and pose:

$$\tilde{\mathbf{p}}_s = \mathbf{K}((\mathbf{R}\,\mathbf{D}(\bar{\mathbf{p}}_t)\,\mathbf{K}^{-1}\,\bar{\mathbf{p}}_t) + \mathbf{t})$$

- Use **bilinear interpolation** to warp the source image I_s to the target view $\Rightarrow \hat{I}_s$
- **Photoconsistency loss** between target and warped image $\mathcal{L} = \sum_p |I_t(p) - \hat{I}_s(p)|$

Zhou, Brown, Snavely and Lowe: Unsupervised Learning of Depth and Ego-Motion from Video. CVPR, 2017.

Unsupervised Learning of Depth and Ego-Motion



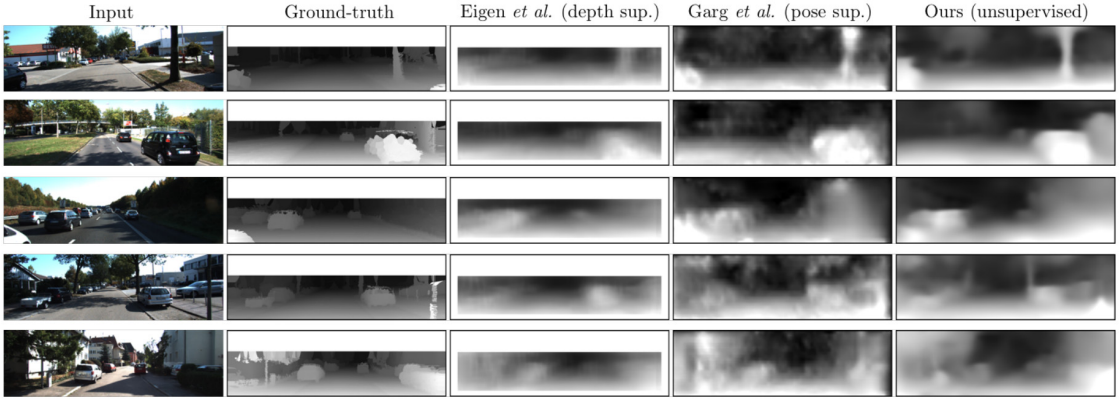
(a) Single-view depth network

(b) Pose/explainability network

- Traditional **U-Net** (DispNet) with **multi-scale** prediction/loss
- Final objective includes photoconsistency, smoothness and explainability loss

Zhou, Brown, Snavely and Lowe: Unsupervised Learning of Depth and Ego-Motion from Video. CVPR, 2017.

Unsupervised Learning of Depth and Ego-Motion



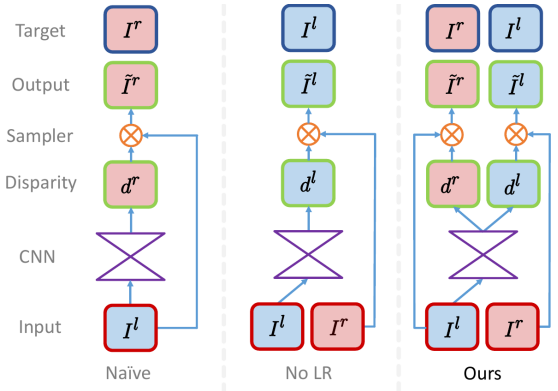
- Performance nearly on par with depth- or pose-supervised methods
- Assumes static scene $\Rightarrow$ can fail in the presence of dynamic objects

Zhou, Brown, Snavely and Lowe: Unsupervised Learning of Depth and Ego-Motion from Video. CVPR, 2017.

Unsupervised Monocular Depth Estimation from Stereo

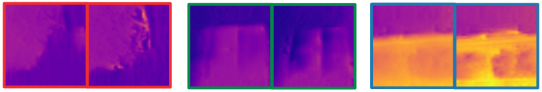
Monodepth from Stereo Supervision:

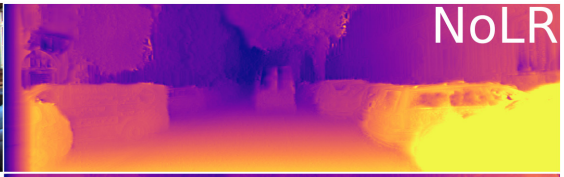
- With naive sampling the CNN produces a disparity map aligned with the target instead of the input
- No LR corrects for this, but suffers from artifacts
- The proposed approach uses the left image to produce disparities for both images, improving quality by enforcing mutual consistency

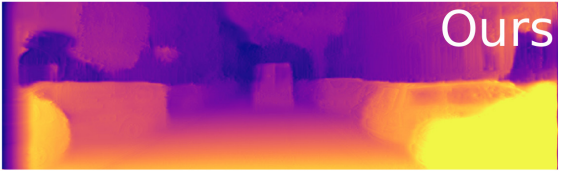


Godard, Aodha and Brostow: Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR, 2017.

Unsupervised Monocular Depth Estimation from Stereo



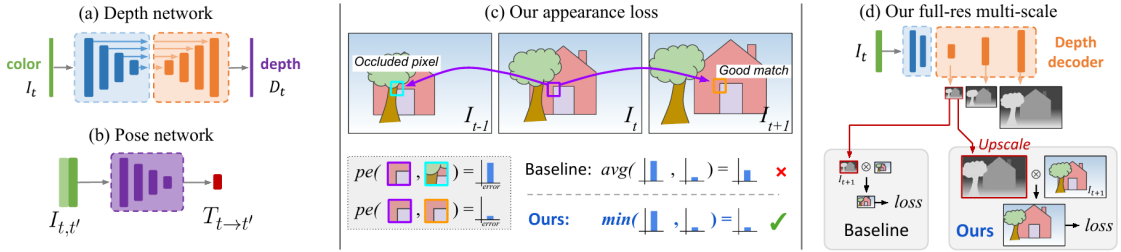




- Losses: photoconsistency, disparity smoothness and left-right consistency
- Here: comparison with and without **left-right consistency loss**

Godard, Aodha and Brostow: Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR, 2017.

Digging Into Self-Supervised Monocular Depth Estimation

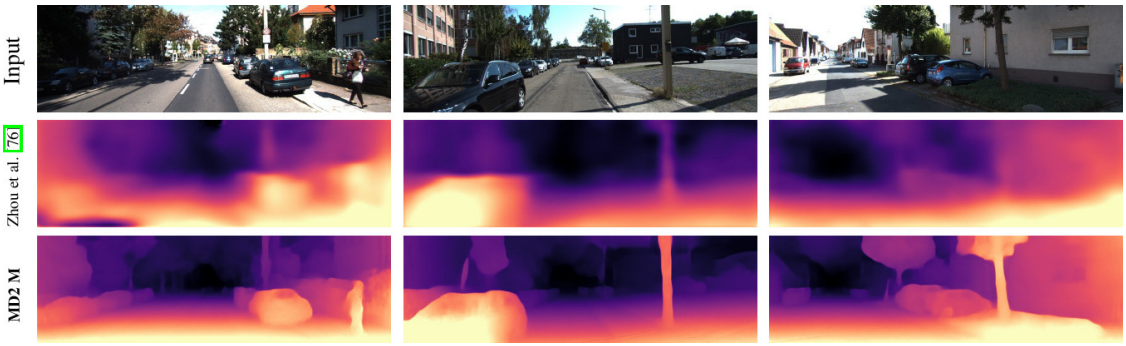


Monodepth2:

- **Per-pixel minimum** photoconsistency loss to handle occlusions
- Computation of all losses at the input **resolution** to reduce texture-copy artifacts
- **Auto-masking** loss to ignore images in which the camera does not move

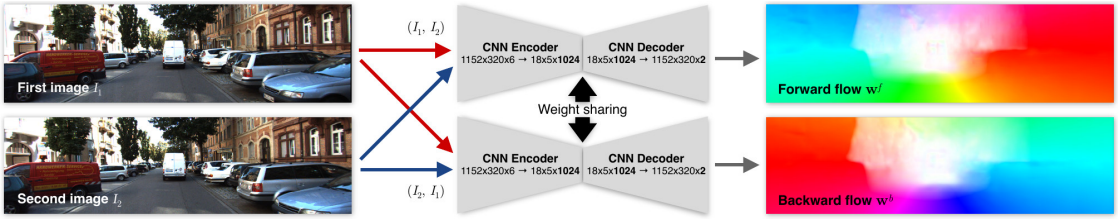
Godard, Aodha, Firman and Brostow: Digging Into Self-Supervised Monocular Depth Estimation. ICCV, 2019.

Digging Into Self-Supervised Monocular Depth Estimation



- Monodepth2 leads to significantly sharper depth boundaries and **more details**
- `https://www.youtube.com/watch?v=sIN1Tp3wIbQ`

Unsupervised Learning of Optical Flow

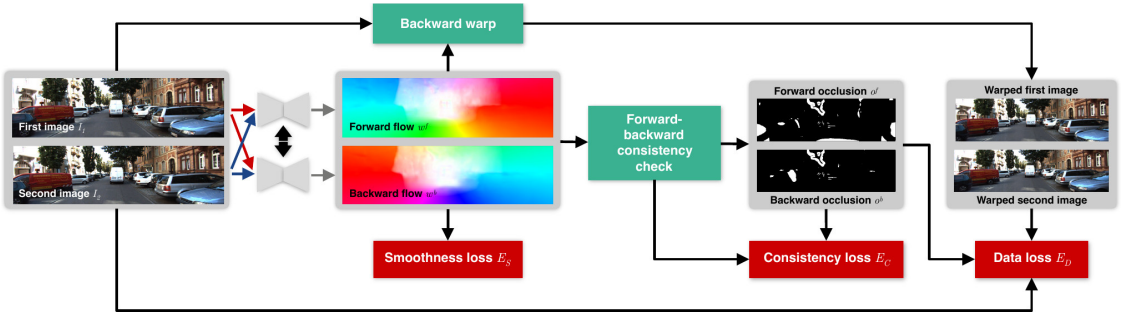


Bidirectional Training:

- Share weights between both directions to train a **universal network** for optical flow that predicts accurate flow in either direction (forward and backward)
- As network architecture, **FlowNetC** [Dosovitskiy et al., 2015] is used

Meister, Hur and Roth: UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss. AAAI, 2018.

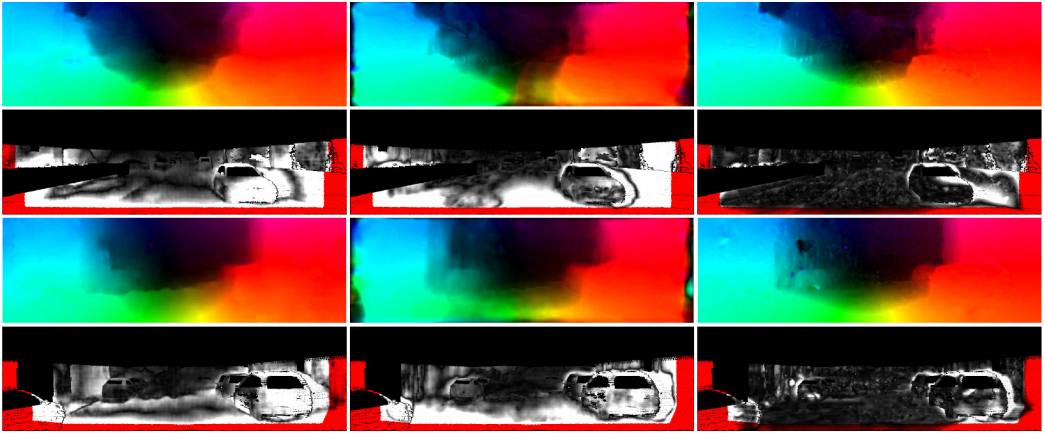
Unsupervised Learning of Optical Flow



- The **data loss** compares flow-warped images to the original images
- The **consistency loss** ensures forward/backward consistency
- The data and consistency loss are masked based on the estimated **occlusion**

Meister, Hur and Roth: UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss. AAAI, 2018.

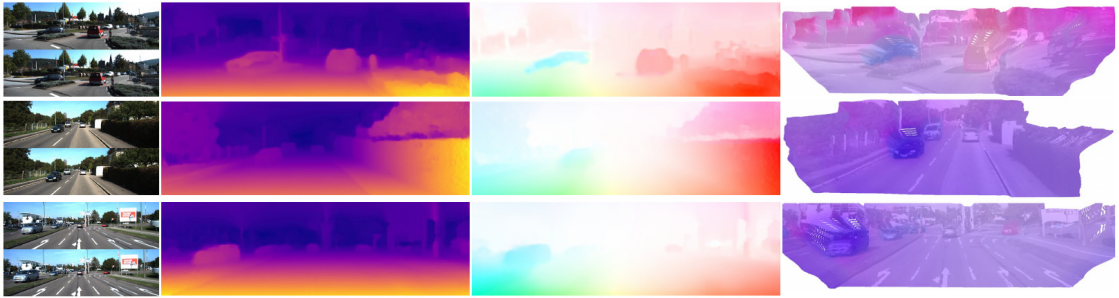
Unsupervised Learning of Optical Flow



- Left-to-right: Supervised FlowNetS, UnsupFlowNet, UnFlow (this work)
- Top row: estimated flow field; Bottom row: flow error (white=large)

Meister, Hur and Roth: UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss. AAAI, 2018.

Self-Supervised Monocular Scene Flow Estimation



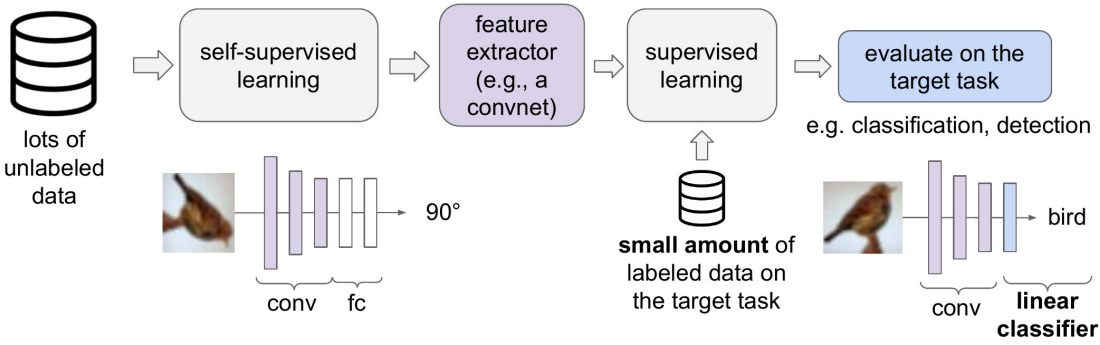
(a) Input images (b) Monocular depth (c) Optical flow (d) 3D visualization of scene flow

- Combining monocular depth and optical flow $\Rightarrow$ **monocular sceneflow**
- Supervised using stereo videos with smoothness and photoconsistency losses

Hur and Roth: Self-Supervised Monocular Scene Flow Estimation. CVPR, 2020.

11.3
Pretext Tasks

Pretext Task

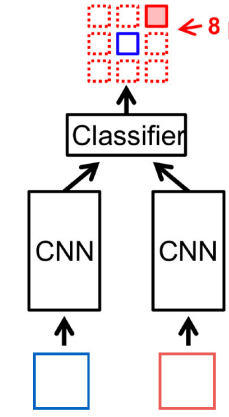


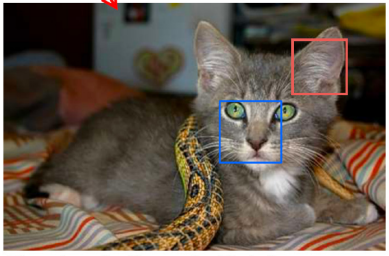
- So far: Self-supervised models tailored towards specific tasks
- Now: Learn more general neural representations using self-supervision
- Define an auxiliary task (e.g., rotation) for **pre-training** with **lots of unlabeled data**
- Then, remove last layers, train **smaller network** on **fewer data** of **target task**

Pretext Task

From Wikipedia:
"A **pretext** (adj: pretextual) is an excuse to do something or say something that is not accurate. **Pretexts** may be based on a half-truth or developed in the context of a misleading fabrication. **Pretexts** have been used to conceal the true purpose or rationale behind actions and words."

Visual Representation Learning by Context Prediction





Randomly Sample Patch
Sample Second Patch

Input: Two patches
Output: 8-way classification

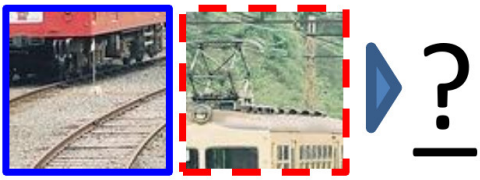
- Goal of context prediction: **predict relative position of patches** (discrete set)
- Hope that this task requires the model to learn to recognize objects and their parts

Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

Visual Representation Learning by Context Prediction

Question 1:

Question 2:

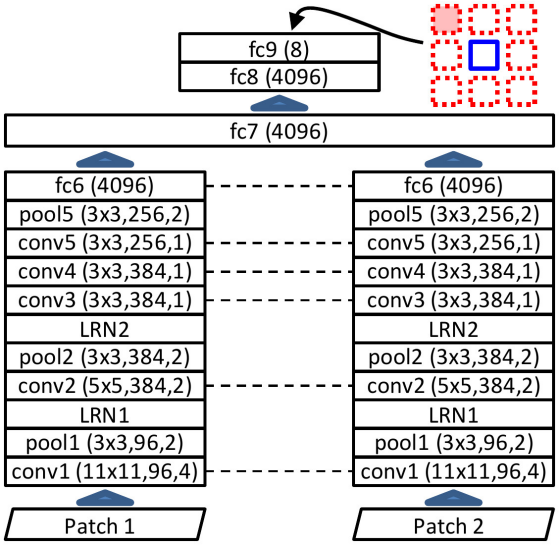


Let's play a game!

- Can you identify where the red patch is located relative to the blue one?

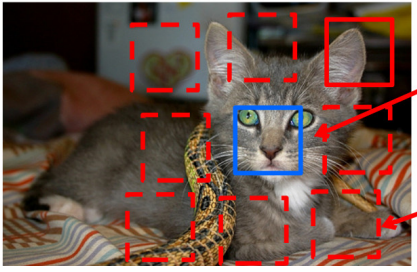
Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

Visual Representation Learning by Context Prediction



Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

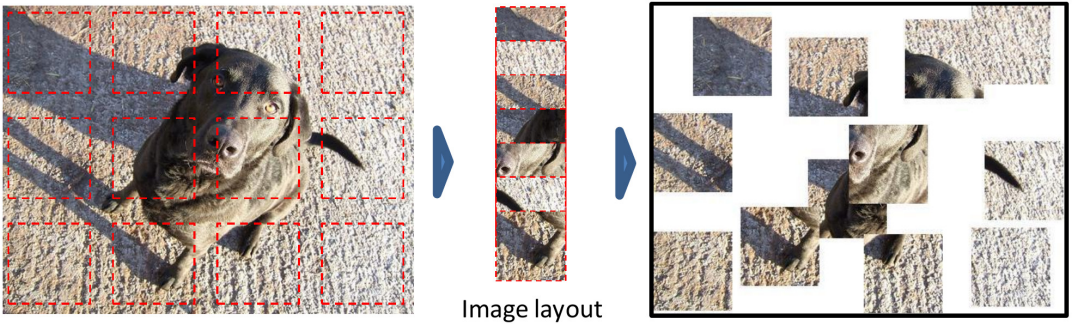
Visual Representation Learning by Context Prediction



- Care has to be taken to **avoid trivial shortcuts** (e.g., edge continuity)

Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

Visual Representation Learning by Context Prediction



Initial layout, with sampled patches in red

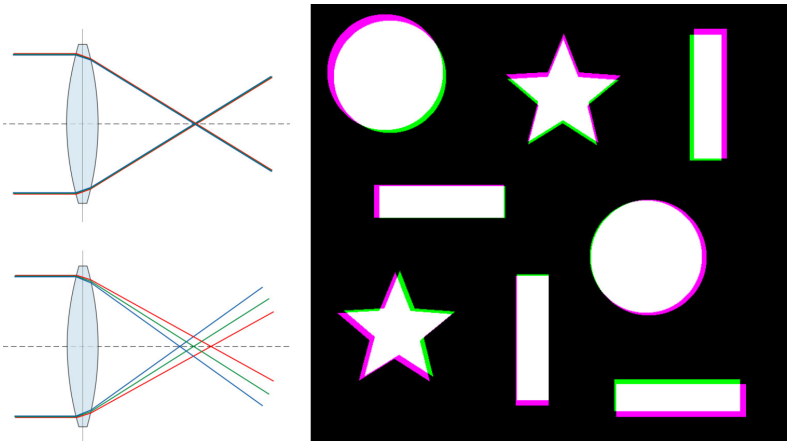
Image layout is discarded

We can recover image layout automatically

- A network can predict the **absolute image location** of randomly sampled patches
- In this case, the relative location can be inferred easily. Why is this happening?

Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

Visual Representation Learning by Context Prediction

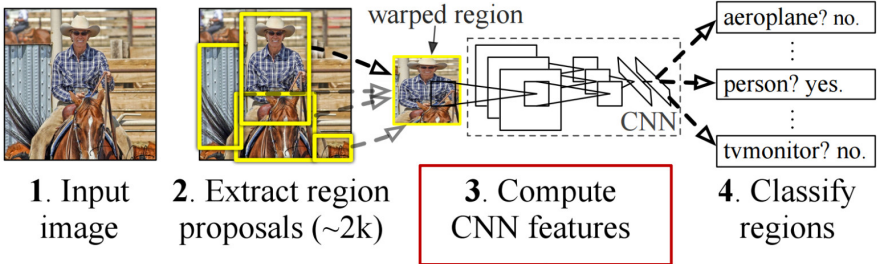


- **Aberration:** Color channels shift with respect to the image location
- Solution: Random dropping of color channels or projection towards gray

Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

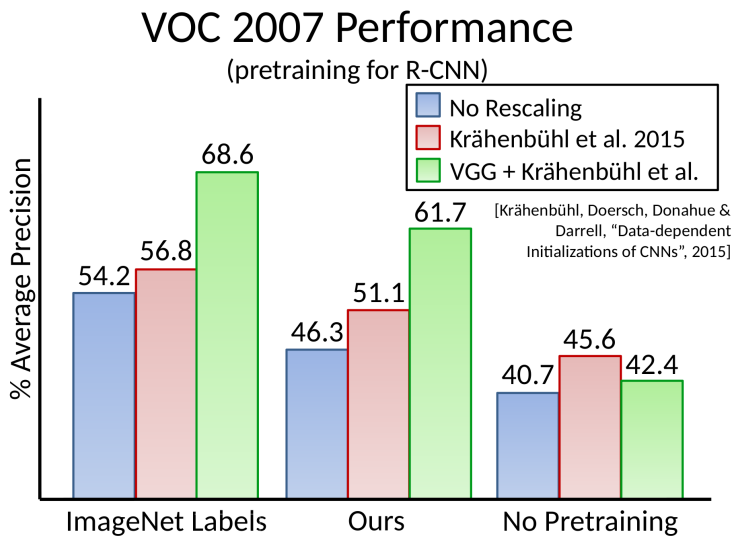
Visual Representation Learning by Context Prediction

Pre-Training for R-CNN



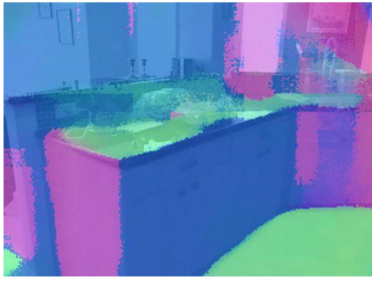
Pre-train on relative-position task, w/o labels

Visual Representation Learning by Context Prediction



Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

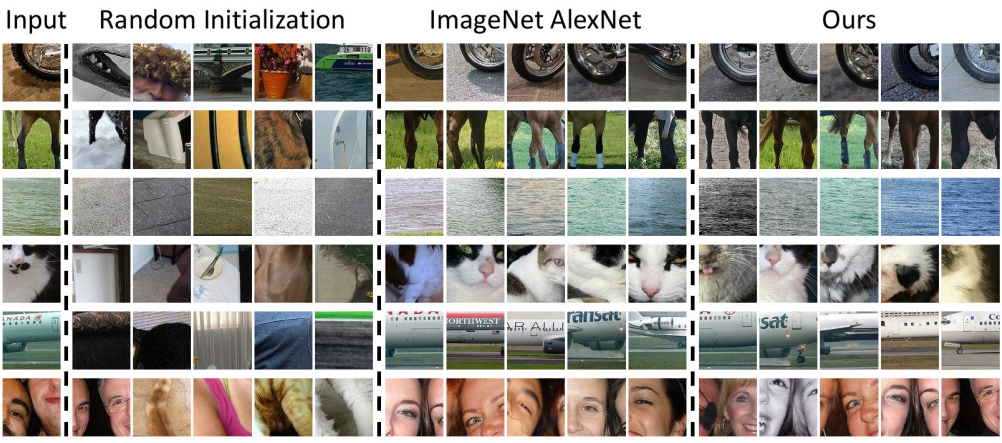
Surface-normal Estimation



	Error (Lower Better)		% Good Pixels (Higher Better)		
Method	Mean	Median	11.25°	22.5°	30.0°
No Pretraining	38.6	26.5	33.1	46.8	52.5
Ours	**33.2**	21.3	36.0	51.2	57.8
ImageNet Labels	33.3	**20.8**	**36.7**	**51.7**	**58.1**

Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

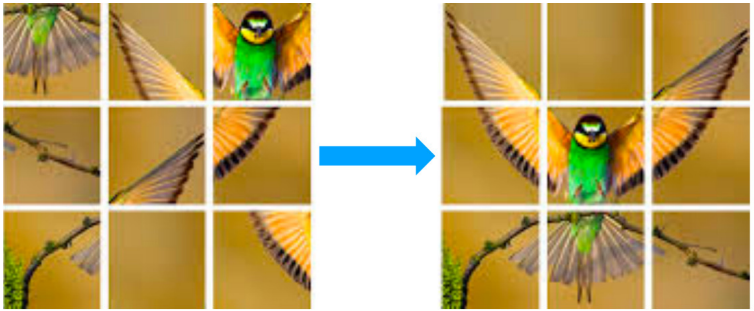
Visual Representation Learning by Context Prediction



- **Nearest neighbor retrieval** results for a query image based on fc6 features

Doersch, Gupta and Efros: Unsupervised Visual Representation Learning by Context Prediction. ICCV, 2015.

Visual Representation Learning by Solving Jigsaw Puzzles



Input: nine patches
Permute using one of N permutations

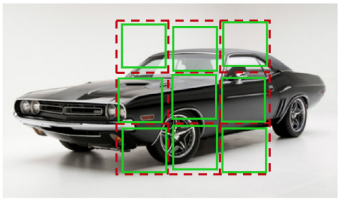
Output: N-way classification

Set N << 9!

- Jigsaw puzzle task: predict one out of 1000 possible random permutations
- Permutations chosen based on Hamming distance to increase difficulty

Noroozi and Favaro: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV, 2016.

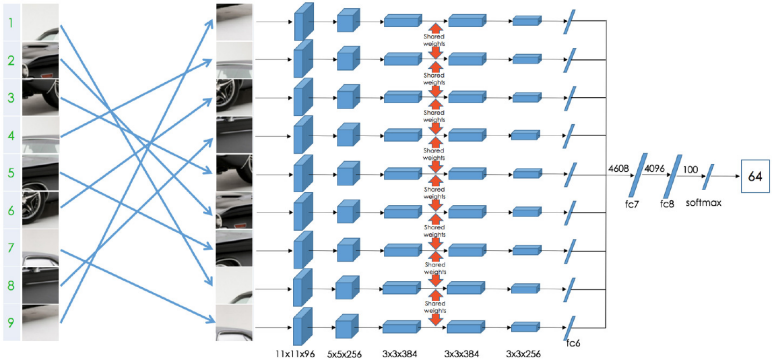
Visual Representation Learning by Solving Jigsaw Puzzles



Permutation Set

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

Reorder patches according to the selected permutation



- Siamese architecture, concatenation of features, MLP, 1000-way classification

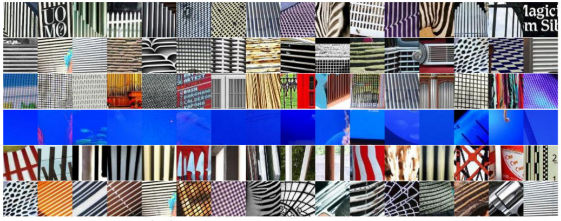
Visual Representation Learning by Solving Jigsaw Puzzles

Preventing Shortcuts: (useful for solving pre-text but not target task)

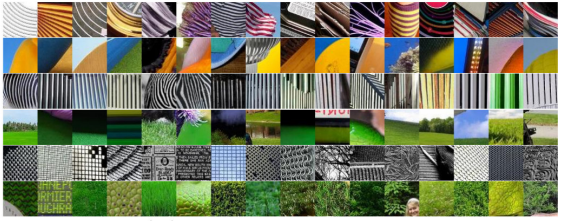
- **Low level statistics:** Adjacent patches include similar low-level statistics (mean and variance). Solution to avoid shortcut: normalize patch mean and variance.
- **Edge continuity:** A strong cue to solve Jigsaw puzzles is the continuity of edges. Solution: select 64×64 pixel tiles randomly from 85×85 pixel cells.
- **Chromatic Aberration:** Chromatic aberration is a relative spatial shift between color channels that increases from the images center to the borders. Solution: use grayscale images or spatial jitter each color channels by few pixels.
- See also: Geirhos et al.: Shortcut Learning in Deep Neural Networks. Arxiv, 2020.

Noroozi and Favaro: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV, 2016.

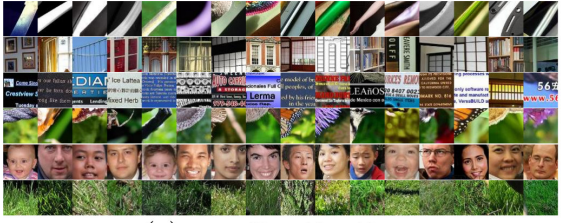
Visual Representation Learning by Solving Jigsaw Puzzles



(a) conv1 activations



(b) conv2 activations



(c) conv3 activations

(d) conv4 activations

Noroozi and Favaro: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV, 2016.

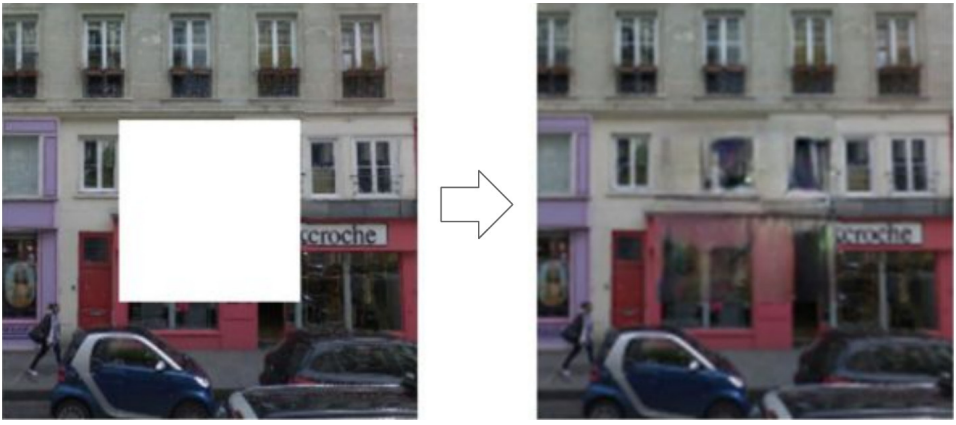
Visual Representation Learning by Solving Jigsaw Puzzles

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%**

- For **transfer learning** pre-trained weights are used as **initialization** of conv layers of AlexNet, the remaining layers are trained from scratch (random. init.)
- Fine-tuning features for PASCAL VOC classification, detection and segmentation
- Performance approaches fully supervised pre-training on ImageNet (first row)

Noroozi and Favaro: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV, 2016.

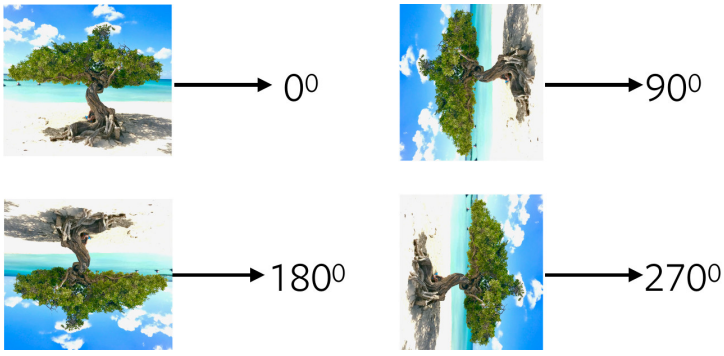
Feature Learning by Inpainting



- **Inpainting task:** try to recover a region that has been removed from context
- Similar to DAE, but requires more semantic knowledge due to large missing region

Pathak, Krähenbühl, Donahue, Darrell and Efros: Context Encoders: Feature Learning by Inpainting. CVPR, 2016.

Visual Representation Learning by Predicting Image Rotations



Input: image rotated by [0, 90, 180, 270]

Output: 4-way classification

- **Rotation task:** try to recover the true orientation (4-way classification)
- Idea: in order to recover the correct rotation, semantic knowledge is required

Gidaris, Singh and Komodakis: Unsupervised Representation Learning by Predicting Image Rotations. ICLR, 2018.

Summary

Pretext Tasks:

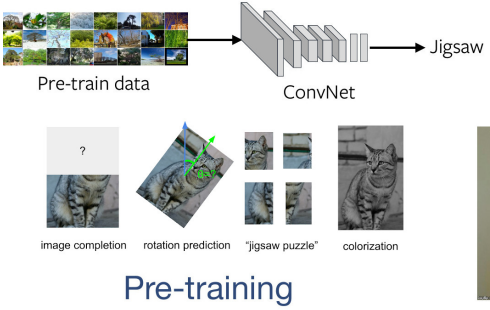
- Pretext tasks focus on “visual common sense”, e.g., rearrangement, predicting rotations, inpainting, colorization, etc.
- The models are forced learn good features about natural images, e.g., semantic representation of an object category, in order to solve the pretext tasks
- We don’t care about pretext task performance, but rather about the utility of the learned features for downstream tasks (classification, detection, segmentation)

Problems:

- Designing good pretext tasks is tedious and some kind of “art”
- The learned representations may not be general

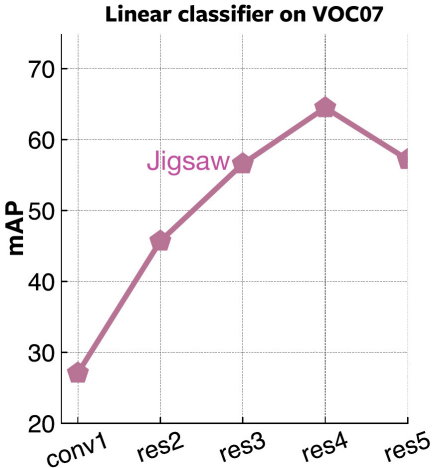
11.4
Contrastive Learning

Hope of Generalization

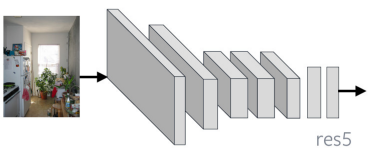


- We really **hope** that the pre-training task and the transfer task are "aligned"

Hope of Generalization



mAP = mean Average Precision
(Higher is better)

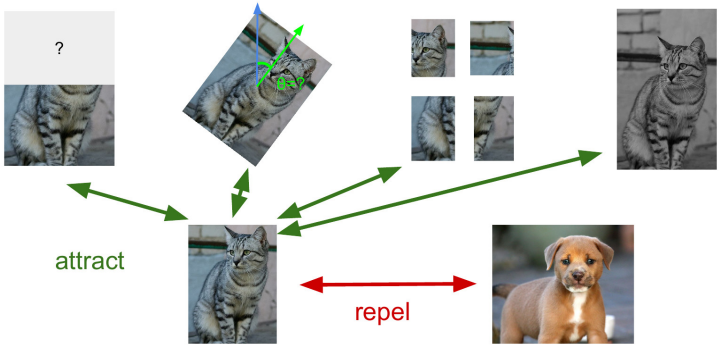


- Pretext feature **performance saturates** (last layers specific to pretext task)

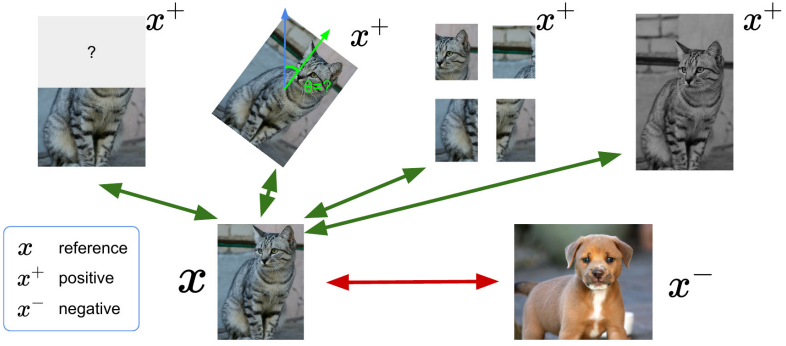
Desiderata

Can we find a more general pretext task?

- Pre-trained features should represent **how images relate** to each other
- They should also be **invariant to nuisance factors** (location, lighting, color)
- Augmentations generated from one reference image are called “views”



Contrastive Learning



- Given a chosen **score function** $s(\cdot, \cdot)$, we want to learn an encoder f that yields **high score for positive pairs** (x, x^+) and **low score for negative pairs** (x, x^-):

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

Contrastive Learning

Assume we have1 reference (x), 1 positive (x^+) and N-1 negative (x_j^-) examples.
Consider the following **multi-class cross entropy loss function:**

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

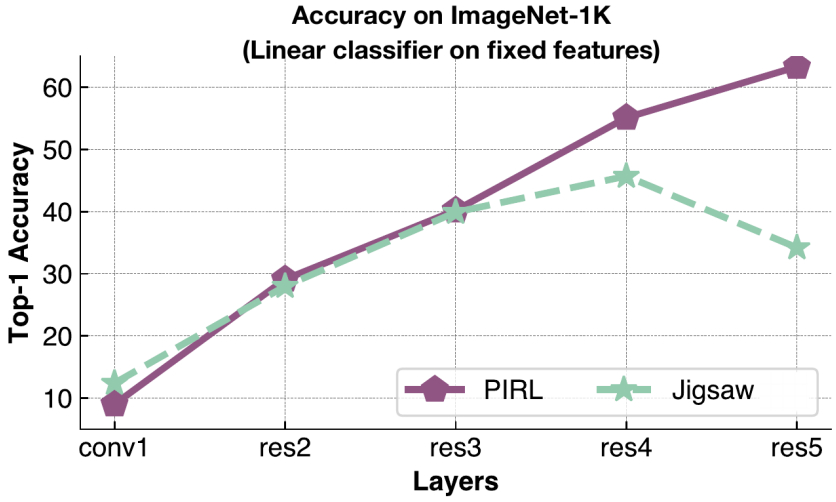
This is commonly known as the **InfoNCE loss** and its negative is a **lower bound** on the **mutual information** between $f(x)$ and $f(x^+)$ (See [Oord, 2018] for details):

$$MI[f(x), f(x^+)] \geq \log(N) - \mathcal{L}$$

Key idea: maximizing mutual information between features extracted from multiple "views" forces the features to capture information about higher-level factors.
Important remark: The larger the negative sample size (N), the tighter the bound.

Oord, Li and Vinyals: Representation Learning with Contrastive Predictive Coding. Arxiv, 2018.

Hope of Generalization



Misra, van der Maaten: Self-Supervised Learning of Pretext-Invariant Representations. CVPR, 2020.

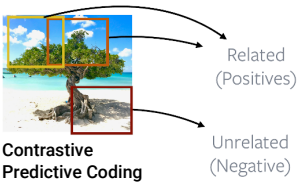
Design Choices

1. Score Function:

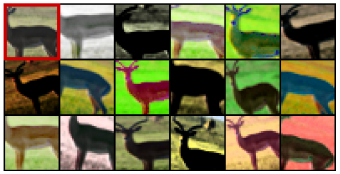
$$s(\mathbf{f}_1, \mathbf{f}_2) = \frac{\mathbf{f}_1^\top \mathbf{f}_2}{\|\mathbf{f}_1\| \, \|\mathbf{f}_2\|}$$

- Cosine similarity
- Commonly used

2. Examples:



3. Augmentations:



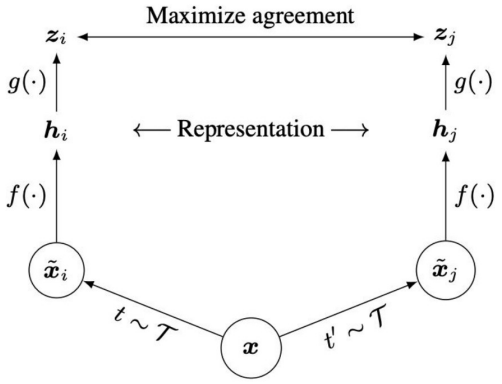
- Crop, resize, flip
- Rotation, cutout
- Color drop/jitter
- Gaussian noise/blur
- Sobel filter

A Simple Framework for Contrastive Learning

- **Cosine similarity** as score function:

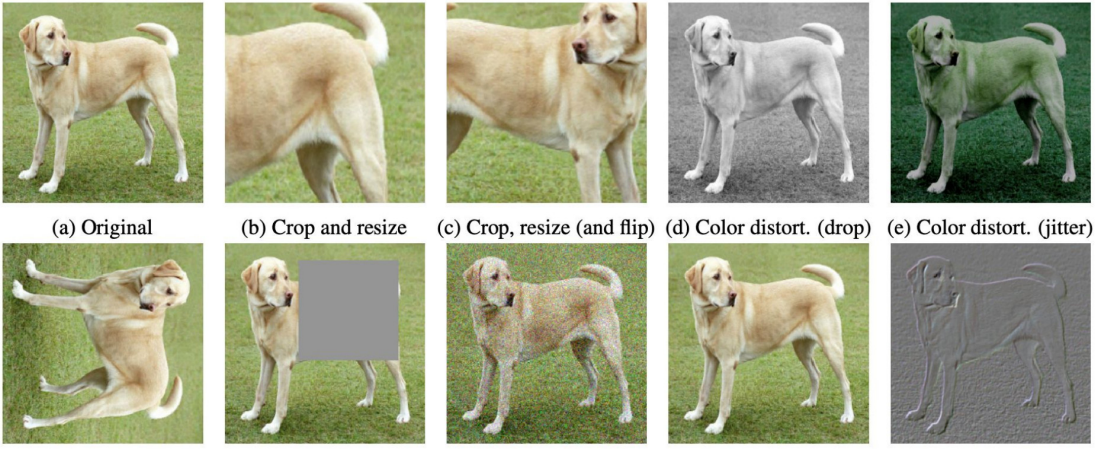
$$s(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\| \, \|\mathbf{z}_j\|}$$

- SimCLR uses a **projection network** $g(\cdot)$ to project features to a space where contrastive learning is applied
- The projection improves learning (more relevant information preserved in $\mathbf{h}$ which is discarded in $\mathbf{z}$)



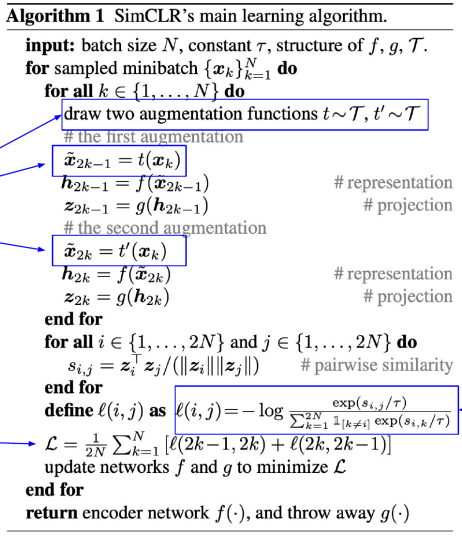
Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

A Simple Framework for Contrastive Learning



Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

A Simple Framework for Contrastive Learning



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 $g(\cdot)$

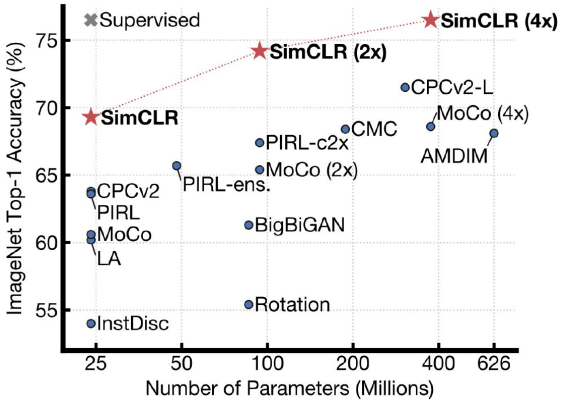
Generate a positive pair by sampling data augmentation functions

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^-

Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

A Simple Framework for Contrastive Learning



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

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

Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

A Simple Framework for Contrastive Learning

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.

Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

A Simple Framework for Contrastive Learning

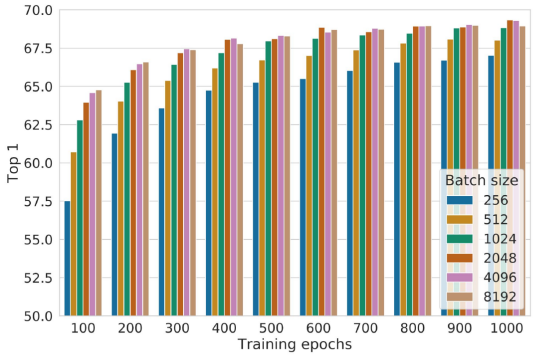


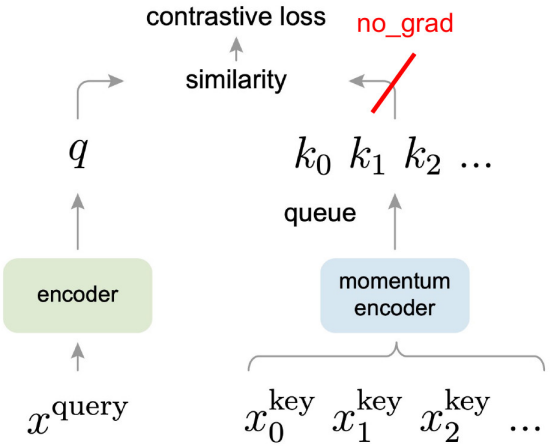
Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

Large training batch size is crucial for SimCLR!

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

Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

Momentum Contrast



Differences to SimCLR:

- Keep **running queue** (dictionary) of keys for negative samples (i.e., ring buffer of minibatches)
- Backpropagate gradients only to query encoder, not queue
- Thus, dictionary can be much larger than minibatch, but: inconsistent as encoder changes during training
- To improve consistency of keys in queue use **momentum update rule:**

$$\theta_k = \beta\,\theta_k + (1-\beta)\,\theta_q$$

He, Fan, Wu, Xie and Girshick: Momentum Contrast for Unsupervised Visual Representation Learning. CVPR, 2020.

Improved Momentum Contrast

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**

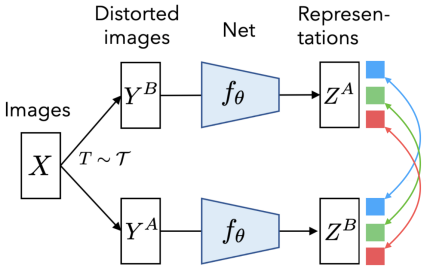
Table 2. **MoCo *vs*. SimCLR**: ImageNet linear classifier accuracy (**ResNet-50, 1-crop 224×224**), trained on features from unsupervised pre-training. "aug+" in SimCLR includes blur and stronger color distortion. SimCLR ablations are from Fig. 9 in [2] (we thank the authors for providing the numerical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

Key insights:

- Non-linear projection head and strong data augmentation are crucial
- Using both, MoCo v2 outperforms SimCLR with much smaller mini-batch sizes (i.e., MoCo v2 runs on a regular 8x V100 GPU node)

Chen, Fan, Girshick and He: Improved Baselines with Momentum Contrastive Learning. Arxiv, 2020.

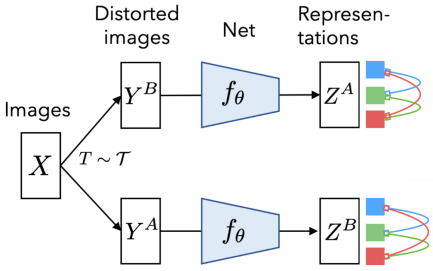
Barlow Twins



- Inspired by information theory: try to **reduce redundancy between neurons**
- Neurons should be invariant to data augmentations but independent of others
- Idea: compute **cross-correlation matrix** $\Rightarrow$ encourage **identity matrix**
- Diagonal: neurons across augmentations correlated, Off-diagonal: no redundancy

Zbontar, Jing, Misra, LeCun and Deny: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. Arxiv, 2021

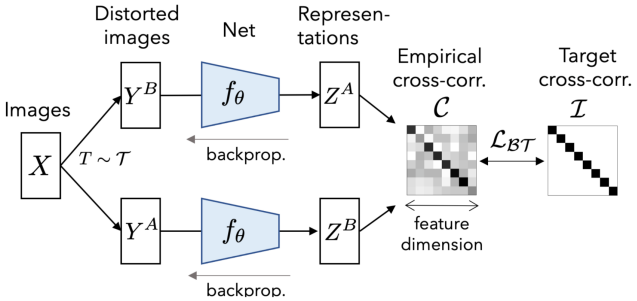
Barlow Twins



- Inspired by information theory: try to **reduce redundancy between neurons**
- Neurons should be invariant to data augmentations but independent of others
- Idea: compute **cross-correlation matrix** $\Rightarrow$ encourage **identity matrix**
- Diagonal: neurons across augmentations correlated, Off-diagonal: no redundancy

Zbontar, Jing, Misra, LeCun and Deny: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. Arxiv, 2021

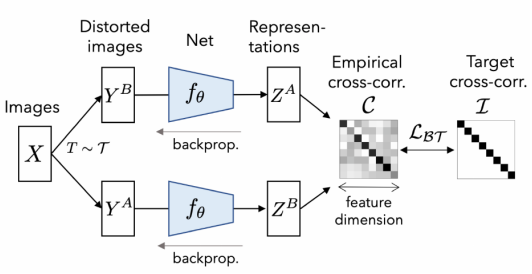
Barlow Twins



- Inspired by information theory: try to **reduce redundancy between neurons**
- Neurons should be invariant to data augmentations but independent of others
- Idea: compute **cross-correlation matrix** $\Rightarrow$ encourage **identity matrix**
- Diagonal: neurons across augmentations correlated, Off-diagonal: no redundancy

Zbontar, Jing, Misra, LeCun and Deny: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. Arxiv, 2021

Barlow Twins



$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b \left(z^A_{b,i}\right)^2} \sqrt{\sum_b \left(z^B_{b,j}\right)^2}}$$

$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}{}^2}_{\text{redundancy reduction term}}$$

- **No negative samples needed** like in contrastive learning!

Barlow Twins

```
# f: encoder network
# lambda: weight on the off-diagonal terms
# N: batch size
# D: dimensionality of the representation
#
# mm: matrix-matrix multiplication
# off_diagonal: off-diagonal elements of a matrix
# eye: identity matrix

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    y_a, y_b = augment(x)

    # compute representations
    z_a = f(y_a) # NxD
    z_b = f(y_b) # NxD

    # normalize repr. along the batch dimension
    z_a_norm = (z_a - z_a.mean(0)) / z_a.std(0) # NxD
    z_b_norm = (z_b - z_b.mean(0)) / z_b.std(0) # NxD

    # cross-correlation matrix
    c = mm(z_a_norm.T, z_b_norm) / N # DxD

    # loss
    c_diff = (c - eye(D)).pow(2) # DxD
    # multiply off-diagonal elems of c_diff by lambda
    off_diagonal(c_diff).mul_(lambda)
    loss = c_diff.sum()

    # optimization step
    loss.backward()
    optimizer.step()
```

Zbontar, Jing, Misra, LeCun and Deny: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. Arxiv, 2021

Barlow Twins

Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised	25.4	56.4	48.4	80.4
PIRL	-	-	57.2	83.8
SIMCLR	48.3	65.6	75.5	87.8
BYOL	53.2	68.8	78.4	89.0
SWAV (w/ multi-crop)	53.9	**70.2**	78.5	**89.9**
BARLOW TWINS (ours)	**55.0**	69.7	**79.2**	89.3

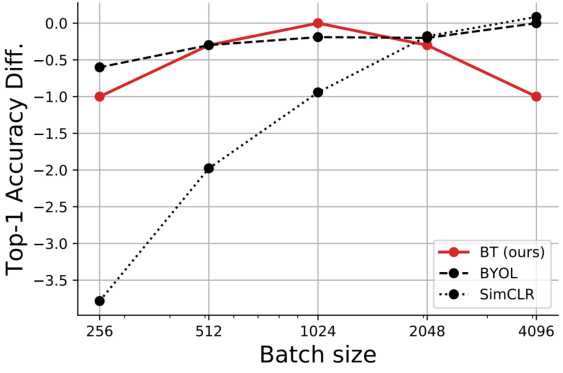
Finetuning on ImageNet

Method	Places-205	VOC07	iNat18
Supervised	53.2	87.5	46.7
SimCLR	52.5	85.5	37.2
MoCo-v2	51.8	86.4	38.6
SwAV	52.8	86.4	39.5
SwAV (w/ multi-crop)	56.7	88.9	48.6
BYOL	54.0	86.6	47.6
BARLOW TWINS (ours)	54.1	86.2	46.5

Linear Classifiers

- **Simpler method** that performs on par with state-of-the-art

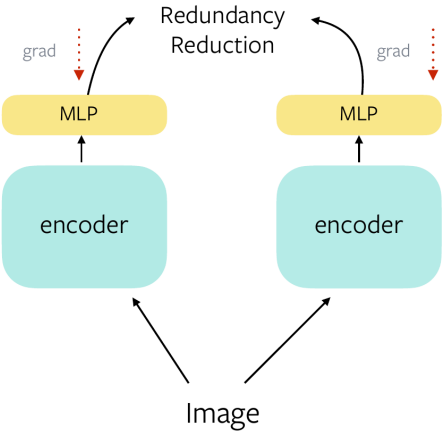
Barlow Twins

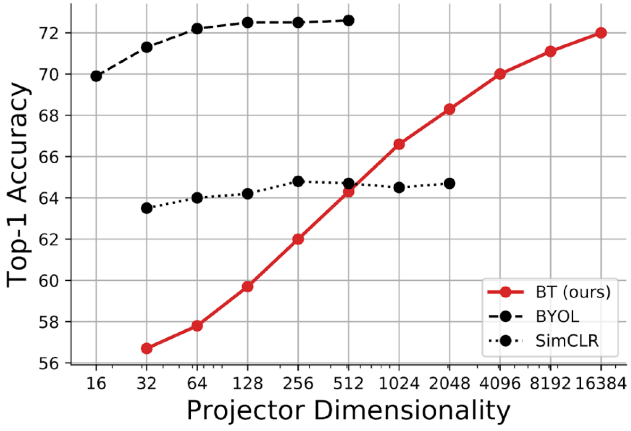


- Mildly affected by **batch size** (does not degrade as much as SimCLR)

Zbontar, Jing, Misra, LeCun and Deny: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. Arxiv, 2021

Barlow Twins





- **Projection** into a high-dimensional space (only pre-training) leads to better results

Summary

- Creating labeled data is time consuming and expensive
- Self-supervised methods overcome this by learning from data alone
- Task-specific models typically minimize photoconsistency measures
- Pretext tasks have been introduced to learn more generic representations
- These generic representations can then be fine-tuned to target task
- However, classical pretext tasks (rotation) often not well aligned with target task
- Contrastive learning and redundancy reduction are better aligned and produce state-of-the-art results, closing the gap to fully supervised ImageNet pretraining