Self-Supervised Learning

Marina Munkhoeva
Research Scientist, AIRI

HSE, 2025-10-02

00

Outline

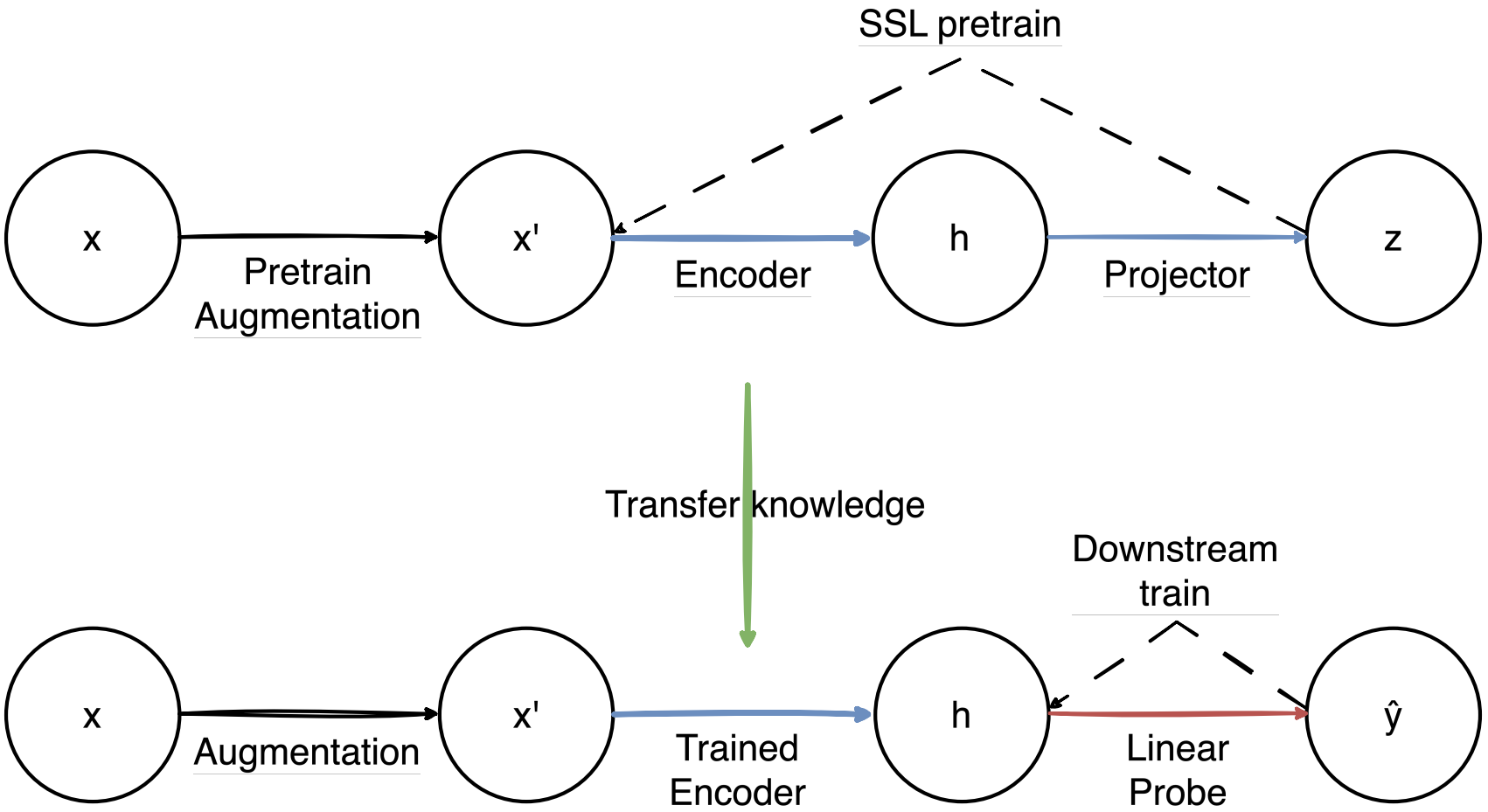
Outline

Recap.....4
Early SSL methods.....10
Bibliography.....52

01

Recap

SSL Train Pipeline



Evaluation / Diagnostics

Some questions pop

- Linearly separability?
- Transferability?
- Robustness?
- Invariances versus shortcuts?

Standard ways to assess representation quality:

1. Linear probe
2. kNN
3. Fine-tuning
4. Classification/Detection/Segmentation transfer
5. Clustering/Retrieval

Query Image

Ours Cos Sim: 0.831
VICReg Cos Sim: 0.571

Ours Cos Sim: 0.471
VICReg Cos Sim: 0.785

Query Image

Ours Cos Sim:0.798
VICReg Cos Sim:0.612

Ours Cos Sim:0.521
VICReg Cos Sim:0.763

Linear probing & kNN

Setup:

- Freeze backbone at pretext weights.

- Head: 1–2 layer MLP (or logistic regression).
- Augs: label-preserving; avoid heavy color jitter.
- Train head on labelled data

- Compute features for training and test sets.
- Predict test targets based on nearest neighbours.

Report top-1 (w/ top-5 for ImageNet-style), seed-avg (3–5 seeds).

Both can be used as **online sanity check**.

Isolating **representation quality**:

- global linear separability
- local metric quality

Fine-tuning

SSL promises **sample efficiency** $\rightarrow$ few-shot/low-data fine-tuning

Use

- few-shot train sets per class, {1, 5, 10, 20}
- low portion of train set, {1%, 10%}

No freeze: full finetune of the backbone + head

Partial freeze: freeze backbone up to a point, leaving last layer blocks + head

Visual Pretext Tasks

Self-supervision central idea — obtain pseudo-labels from structure of data

Pretext task is an **auxiliary** task that uses cheap pseudo-labels

Earlier approaches – **careful** task design

- temporal coherence
- spatial coherence
- domain knowledge

NB not all methods below are representation learning per se, but rather an example of auxiliary task

See (Jing & Tian, 2020) for comprehensive survey on earlier approaches

02

Early SSL methods

Origins of SSL

(Schmidhuber, 1990) yet again!

Making the World Differentiable: On Using Self-Supervised Fully Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environments

Jürgen Schmidhuber*
Institut für Informatik
Technische Universität München
Arcisstr. 21, 8000 München 2, Germany
schmidhu@tumult.informatik.tu-muenchen.de

Temporal Signal

Slow Feature Analysis (Wiskott & Sejnowski, 2002):

Learning problem. Given an I-dimensional input signal $\mathbf{x}(t) = [x_1(t), \ldots, x_I(t)]^T$ with $t \in [t_0, t_1]$ indicating time and $[\ldots]^T$ indicating the transpose of $[\ldots]$. Find an input-output function $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_J(\mathbf{x})]^T$ generating the J-dimensional output signal $\mathbf{y}(t) = [y_1(t), \ldots, y_J(t)]^T$ with $y_j(t) := g_j(\mathbf{x}(t))$ such that for each $j \in \{1, \ldots, J\}$

$$\Delta_j := \Delta(y_j) := \langle \dot{y}_j^2 \rangle \quad \text{is minimal} \tag{2.1}$$

under the constraints

$$\langle y_j \rangle = 0 \qquad \text{(zero mean)}, \tag{2.2}$$

$$\langle y_j^2 \rangle = 1 \qquad \text{(unit variance)}, \tag{2.3}$$

$$\forall\, j' < j: \quad \langle y_{j'}\, y_j \rangle = 0 \qquad \text{(decorrelation)}, \tag{2.4}$$

where the angle brackets indicate temporal averaging, that is,

$$\langle f \rangle := \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t)\, dt.$$

Temporal Signal in Videos

Useful features change slowly in time!

Sample Contrastive formulation (Mobahi et al., 2009)

$$L\left(z_{t_1}, z_{t_2}, W\right) = \begin{cases} D\left(z_{t_1}, z_{t_2}\right) \text{ if } |t_1 - t_2| \leq T \\ 1 - \max\left(0, m - D\left(z_{t_1}, z_{t_2}\right)\right) \text{ otherwise} \end{cases}$$

D - L2 distance, z_{t_i} are extracted features at time t_i with model parameterized by W
T - predefined threshold, m - predefined margin

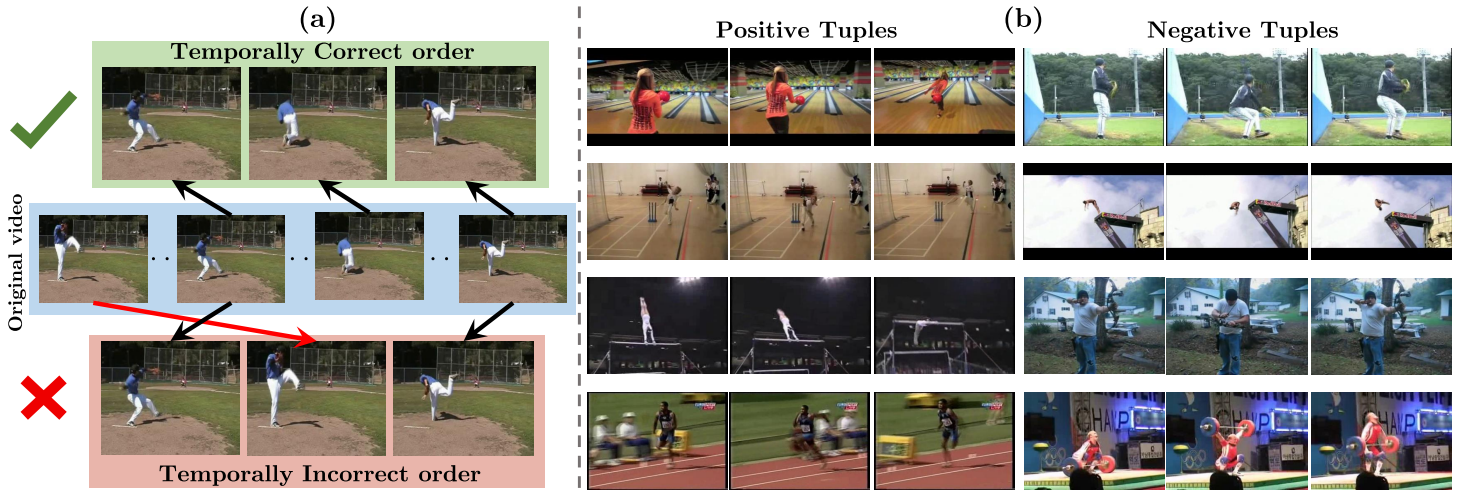
Temporal Order Verification

Consider a set of frames $f_1, ..., f_n$

Classification problem

Positive tuple (f_b, f_c, f_d) if either $b < c < d$ or $d < c < b$ holds

Negative tuple (f_d, f_b, f_c)



(Misra et al., 2016)

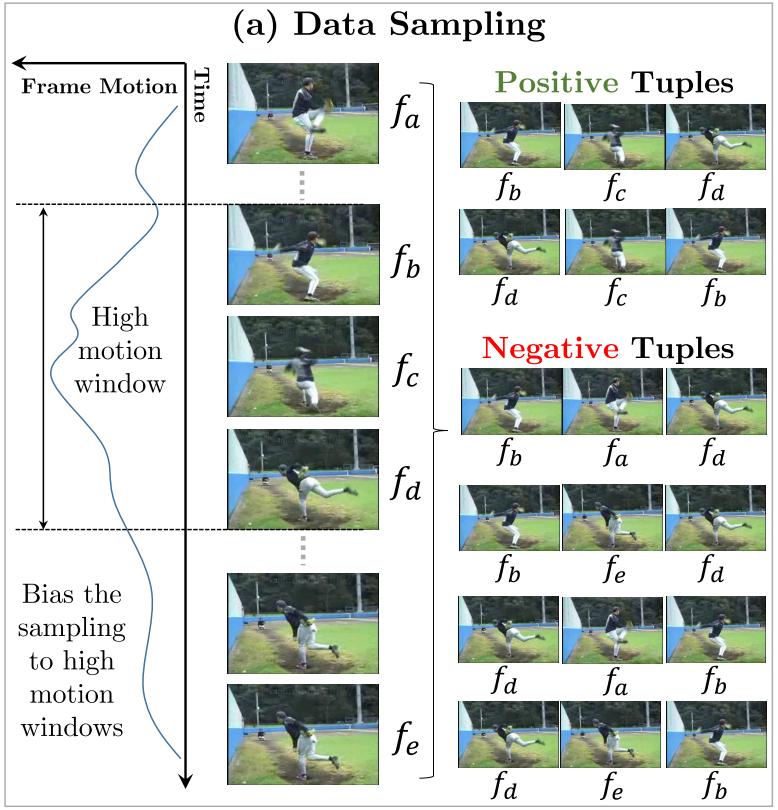
Frames Sampling and Architecture

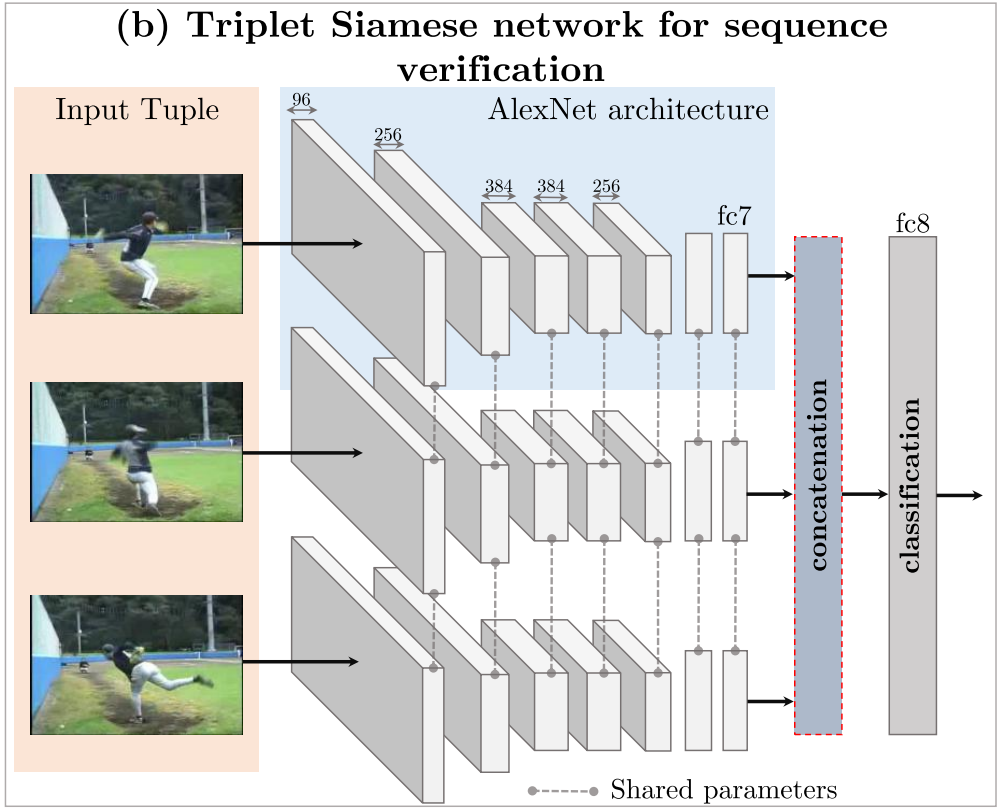
Sample 5 frames from high motion window

Form positive and negative tuples

Process with AlexNet and stack for single representation

Cross-entropy loss for positive and negative classes





(Misra et al., 2016)

Action Recognition

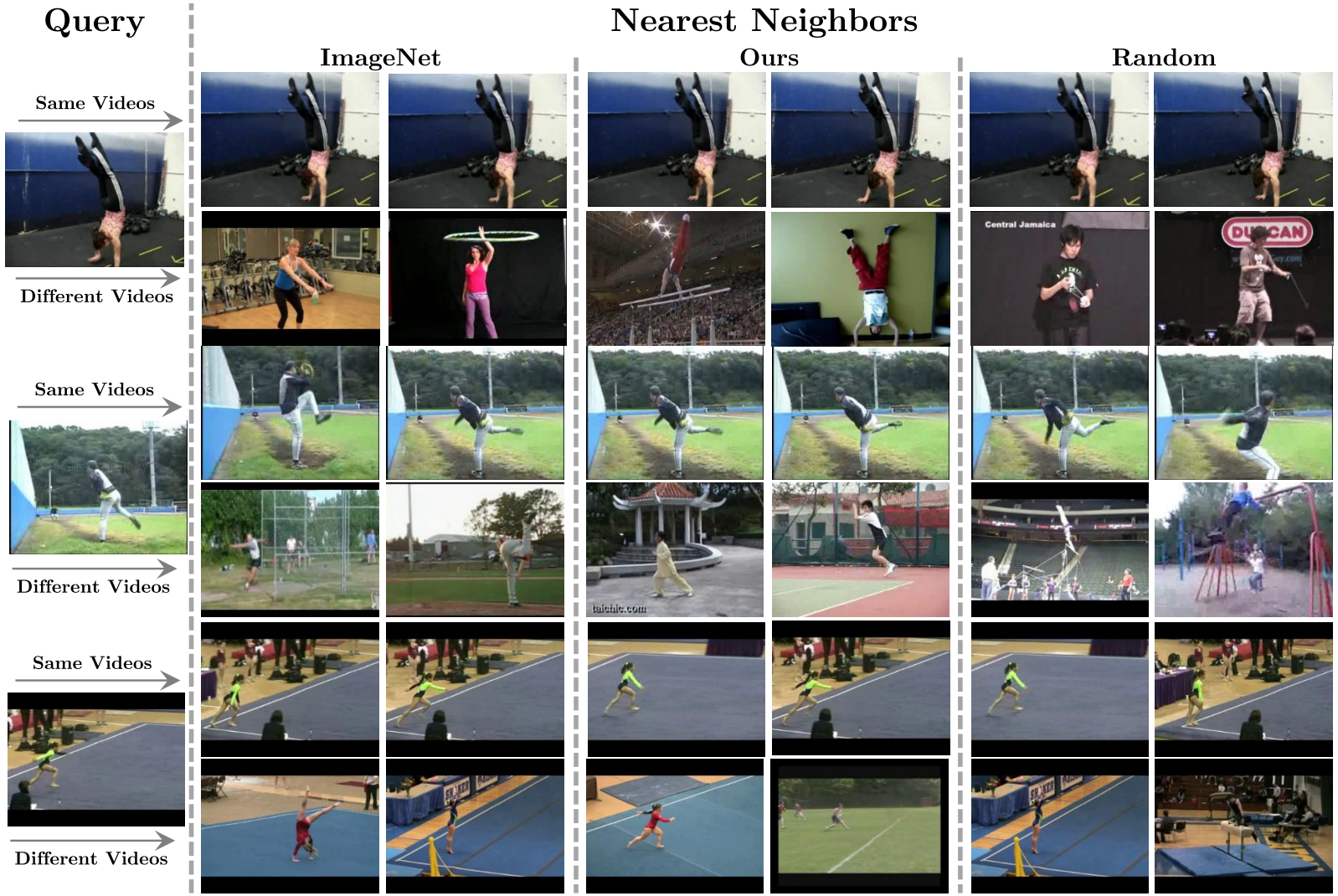
Pretrain datasets

- UCF101
- HMDB51

designed for learning human action recognition

Imagenet pretrained network recognises **scene semantics**

Temporal order verification focuses on **human pose**



(Misra et al., 2016)

Learning to See by Moving

Pretrain datasets with odometry information

- KITTI
- SF

Predict camera transformation between f_i, f_j

Bin transformation values into m bins

Perform classification into m classes

Evaluate on:

- scene recognition
- object recognition
- keypoint matching
- visual odometry

(Agrawal et al., 2015)

Proof of Concept and Architecture

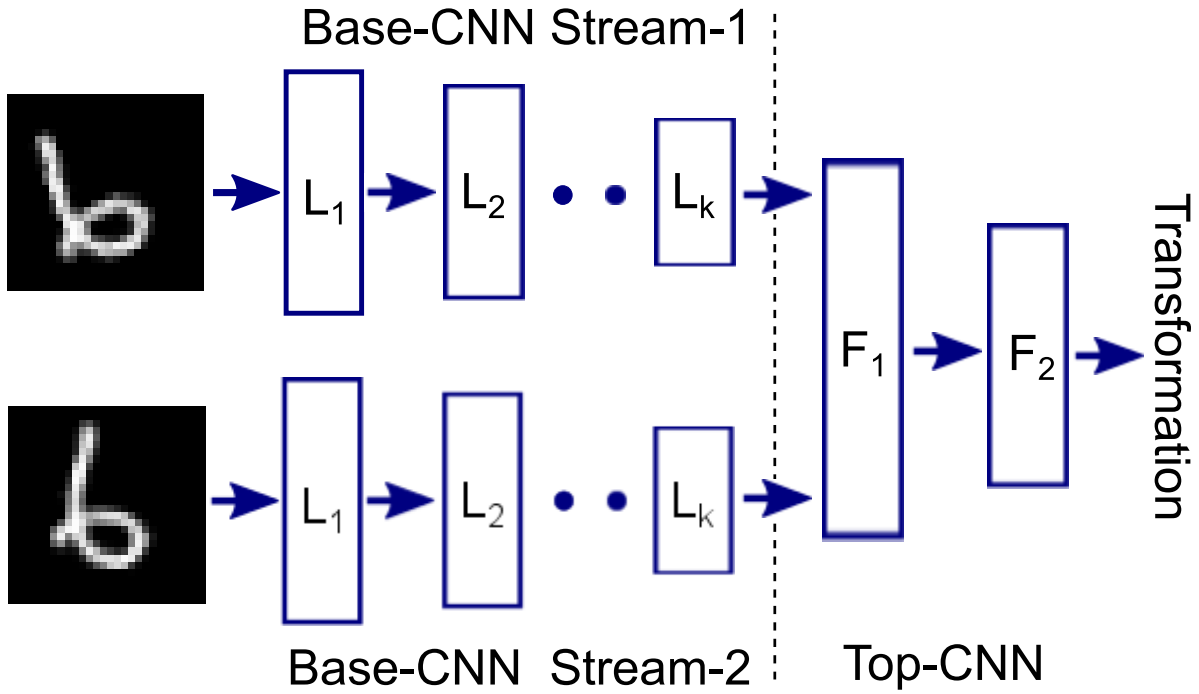
Predict MNIST rotation angle (binned)

Evaluate on label classification

Siamese networks

Top-CCN is disposed of after pretraining

Top-CNN similar to **projection head** used today



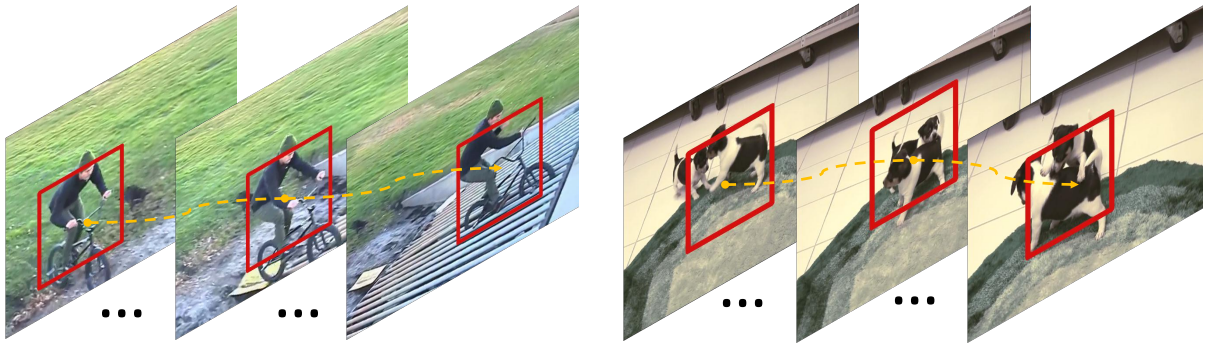
(Agrawal et al., 2015)

Visual tracking

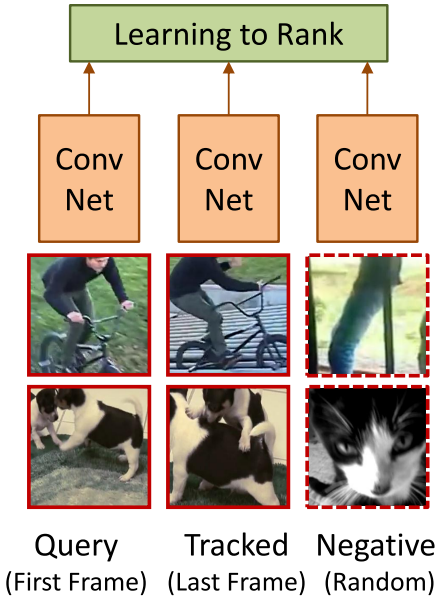
Use video processing tools to extract **patches with motion**

Use unsupervised **tracking** tools to get positive patch

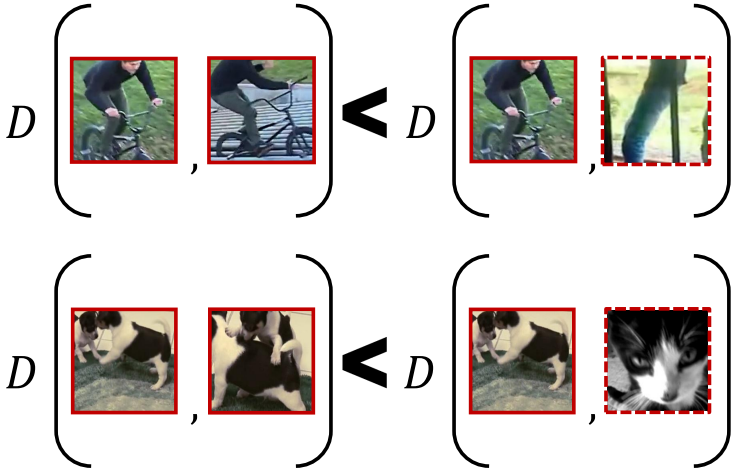
Laborious data processing for mining triplets!



(a) Unsupervised Tracking in Videos



(b) Siamese-triplet Network



D: Distance in deep feature space

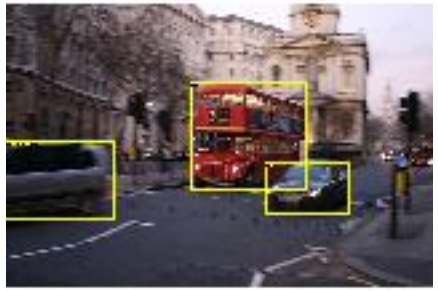
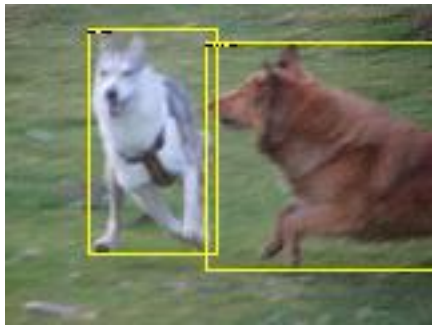
(c) Ranking Objective

(Wang & Gupta, 2015)

Spatial Signal in Images

Pascal VOC 2007

3 tasks: classification (20 classes), object detection (predict bounding box), segmentation

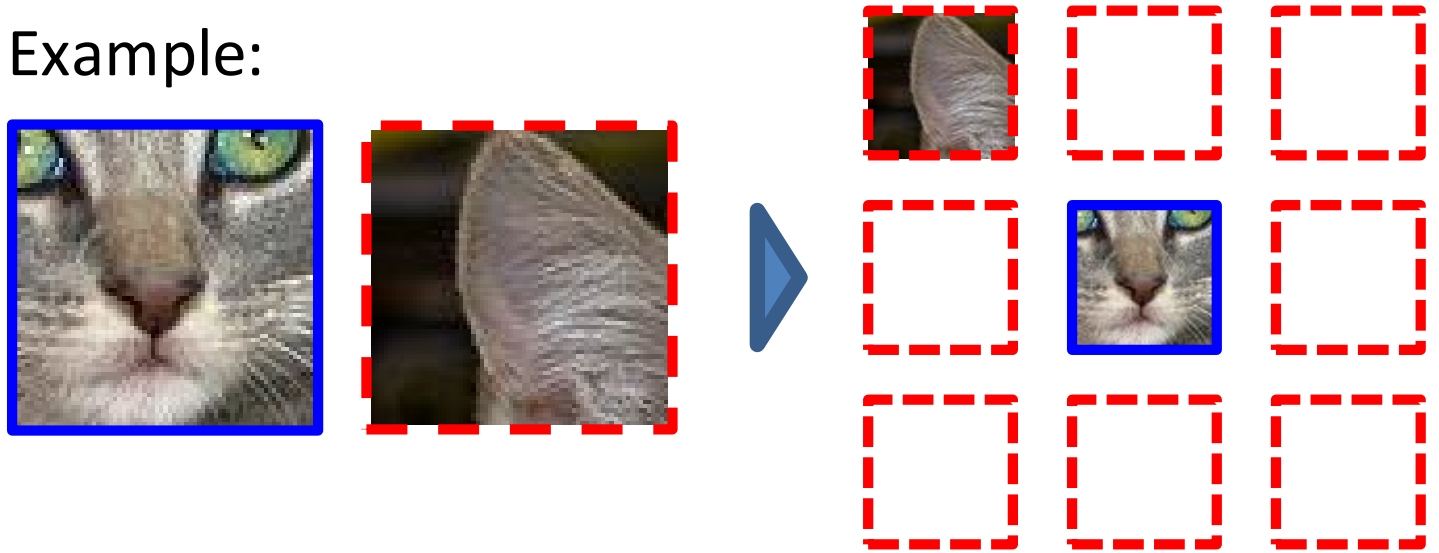




In total there are 9,963 images, containing 24,640 annotated objects.

contest page

Relative Position

Example:



Question 1:

?

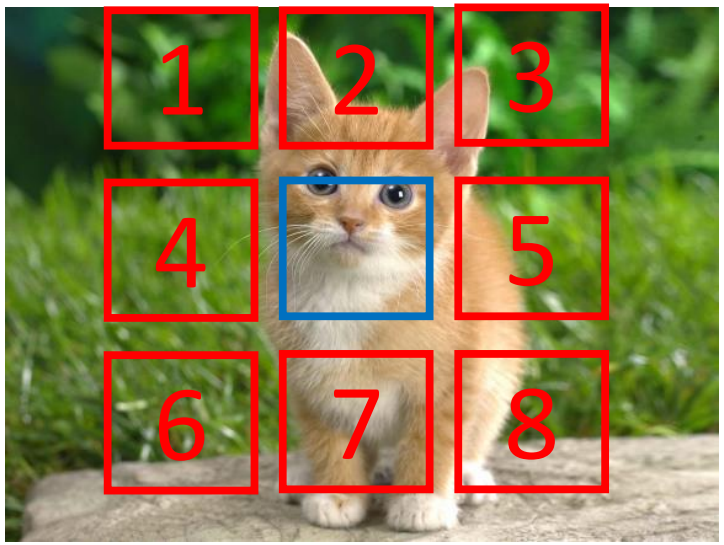
Question 2:

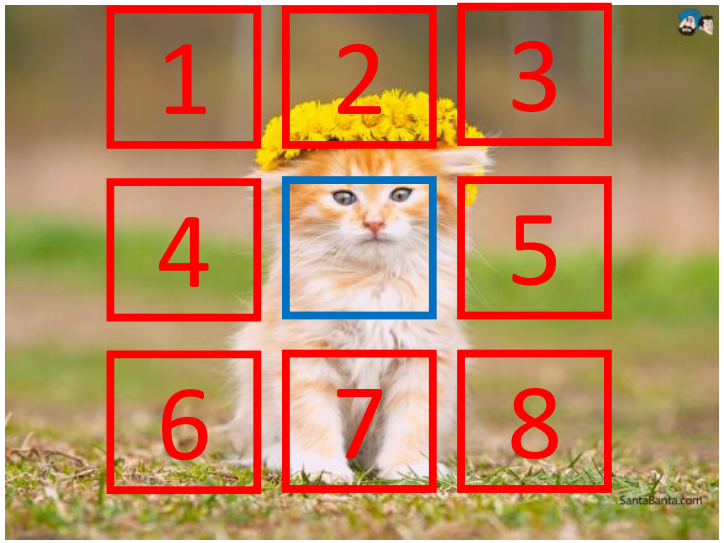
?

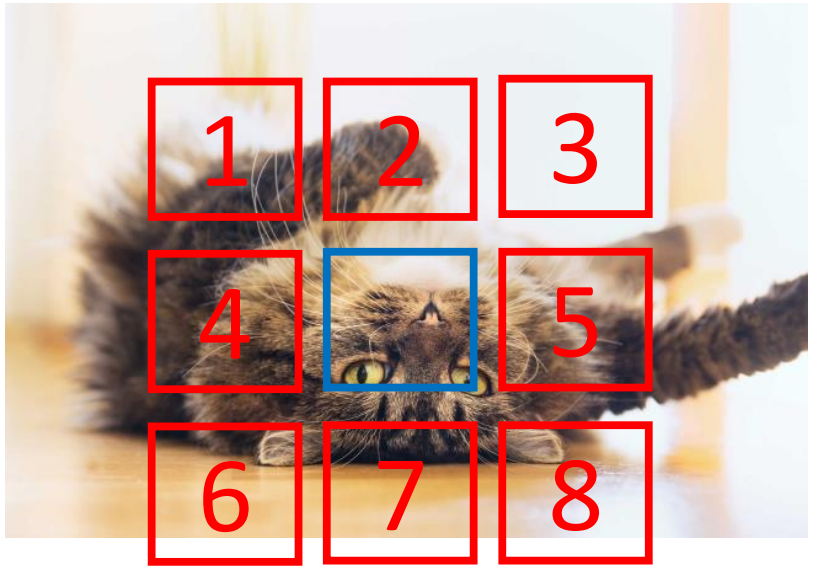
(Doersch et al., 2015)

Relative Position

Pretext task: recover spatial arrangement





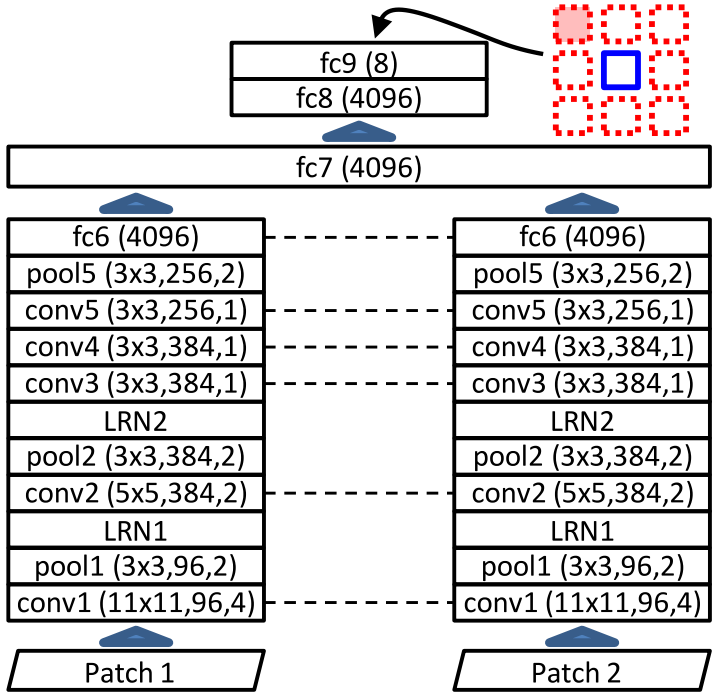


X=( ),Y=7

X=( ),Y=7

X=( ),Y=2

Siamese Architecture



(Doersch et al., 2015)

Performance

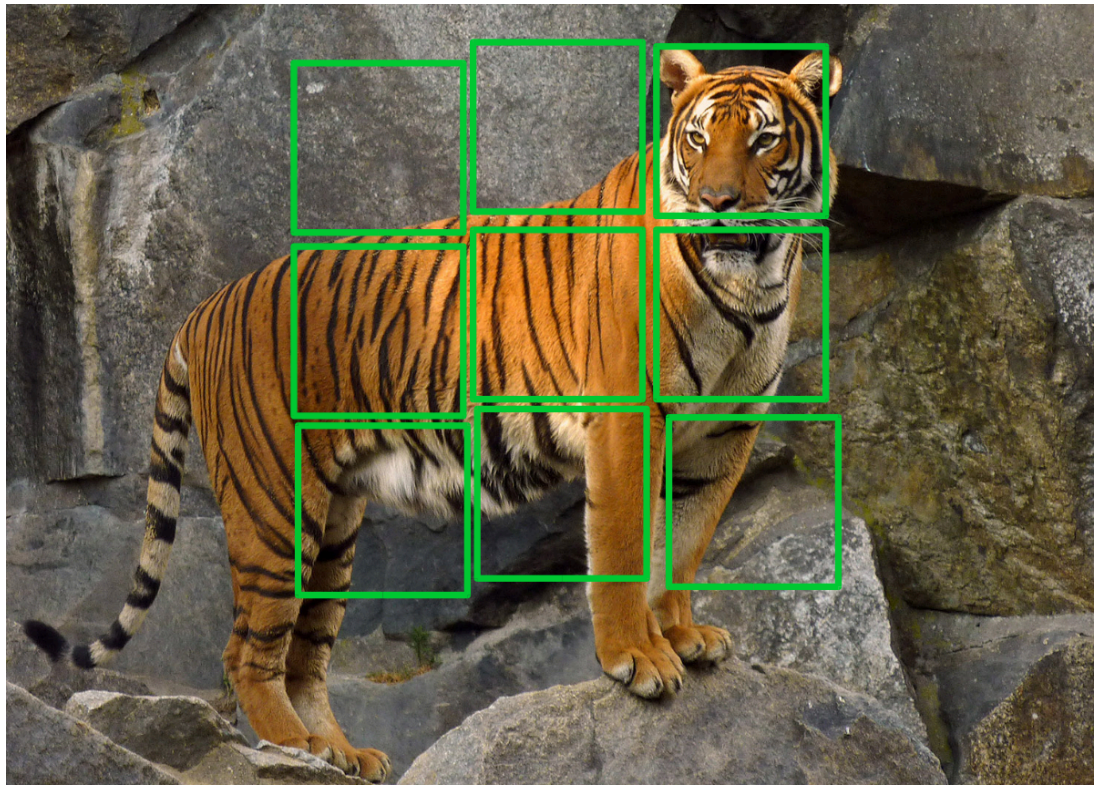
Object detection on PASCAL VOC 2007

VOC-2007 Test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM-v5[17]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
[8] w/o context	52.6	52.6	19.2	25.4	18.7	47.3	56.9	42.1	16.6	41.4	41.9	27.7	47.9	51.5	29.9	20.0	41.1	36.4	48.6	53.2	38.5
Regionlets[58]	54.2	52.0	20.3	24.0	20.1	55.5	68.7	42.6	19.2	44.2	49.1	26.6	57.0	54.5	43.4	16.4	36.6	37.7	59.4	52.3	41.7
Scratch-R-CNN[2]	49.9	60.6	24.7	23.7	20.3	52.5	64.8	32.9	20.4	43.5	34.2	29.9	49.0	60.4	47.5	28.0	42.3	28.6	51.2	50.0	40.7
Scratch-Ours	52.6	60.5	23.8	24.3	18.1	50.6	65.9	29.2	19.5	43.5	35.2	27.6	46.5	59.4	46.5	25.6	42.4	23.5	50.0	50.6	39.8
Ours-projection	58.4	62.8	33.5	27.7	24.4	58.5	68.5	41.2	26.3	49.5	42.6	37.3	55.7	62.5	49.4	29.0	47.5	28.4	54.7	56.8	45.7
Ours-color-dropping	60.5	66.5	29.6	28.5	26.3	56.1	70.4	44.8	24.6	45.5	45.4	35.1	52.2	60.2	50.0	28.1	46.7	42.6	54.8	58.6	46.3
Ours-Yahoo100m	56.2	63.9	29.8	27.8	23.9	57.4	69.8	35.6	23.7	47.4	43.0	29.5	52.9	62.0	48.7	28.4	45.1	33.6	49.0	55.5	44.2
ImageNet-R-CNN[21]	64.2	69.7	50	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
K-means-rescale [31]	55.7	60.9	27.9	30.9	12.0	59.1	63.7	47.0	21.4	45.2	55.8	40.3	67.5	61.2	48.3	21.9	32.8	46.9	61.6	51.7	45.6
Ours-rescale [31]	61.9	63.3	35.8	32.6	17.2	68.0	67.9	54.8	29.6	52.4	62.9	51.3	67.1	64.3	50.5	24.4	43.7	54.9	67.1	52.7	51.1
ImageNet-rescale [31]	64.0	69.6	53.2	44.4	24.9	65.7	69.6	69.2	28.9	63.6	62.8	63.9	73.3	64.6	55.8	25.7	50.5	55.4	69.3	56.4	56.5
VGG-K-means-rescale	56.1	58.6	23.3	25.7	12.8	57.8	61.2	45.2	21.4	47.1	39.5	35.6	60.1	61.4	44.9	17.3	37.7	33.2	57.9	51.2	42.4
VGG-Ours-rescale	71.1	72.4	54.1	48.2	29.9	75.2	78.0	71.9	38.3	60.5	62.3	68.1	74.3	74.2	64.8	32.6	56.5	66.4	74.0	60.3	61.7
VGG-ImageNet-rescale	76.6	79.6	68.5	57.4	40.8	79.9	78.4	85.4	41.7	77.0	69.3	80.1	78.6	74.6	70.1	37.5	66.0	67.5	77.4	64.9	68.6

Table 1. Mean Average Precision on VOC-2007.

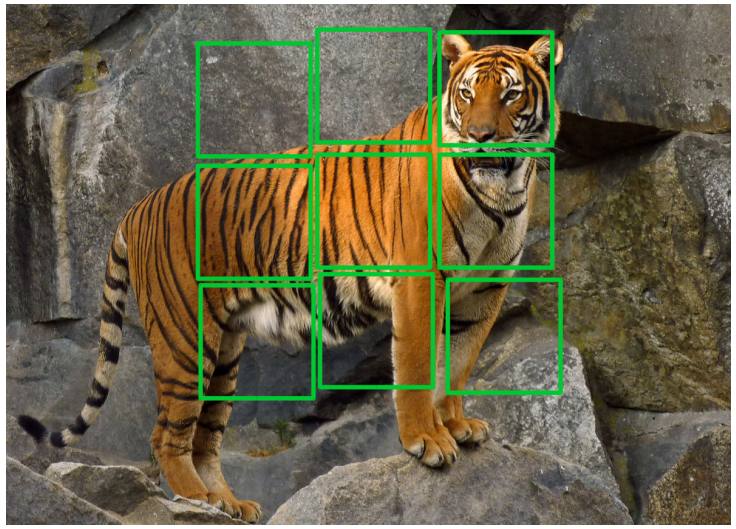
(Doersch et al., 2015)

Jigsaw Puzzle



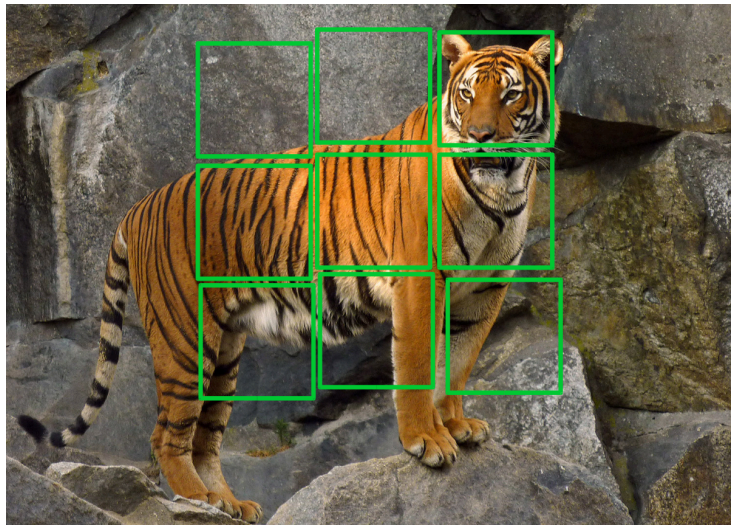
(Noroozi & Favaro, 2016)

Jigsaw Puzzle



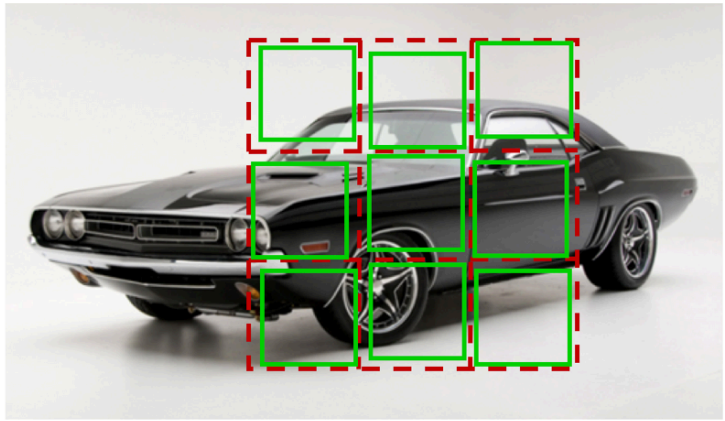
(Noroozi & Favaro, 2016)

Jigsaw Puzzle



(Noroozi & Favaro, 2016)

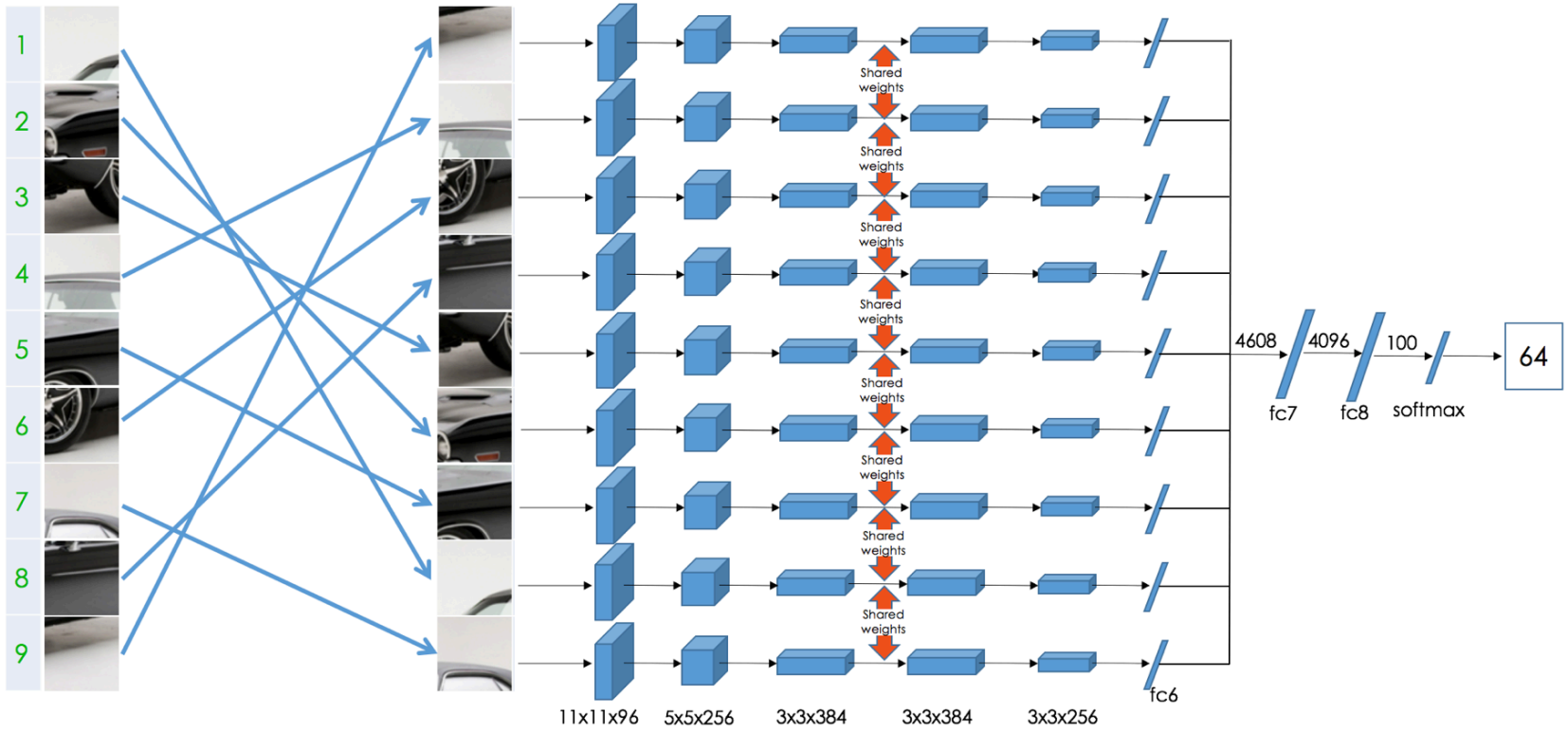
Task Design



Permutation Set

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

Reorder patches according to the selected permutation



(Noroozi & Favaro, 2016)

Close attention to prevent mapping appearance to absolute position: permutation, multiple puzzles per sample, random gap between tiles.

Performance

PASCAL VOC 2007

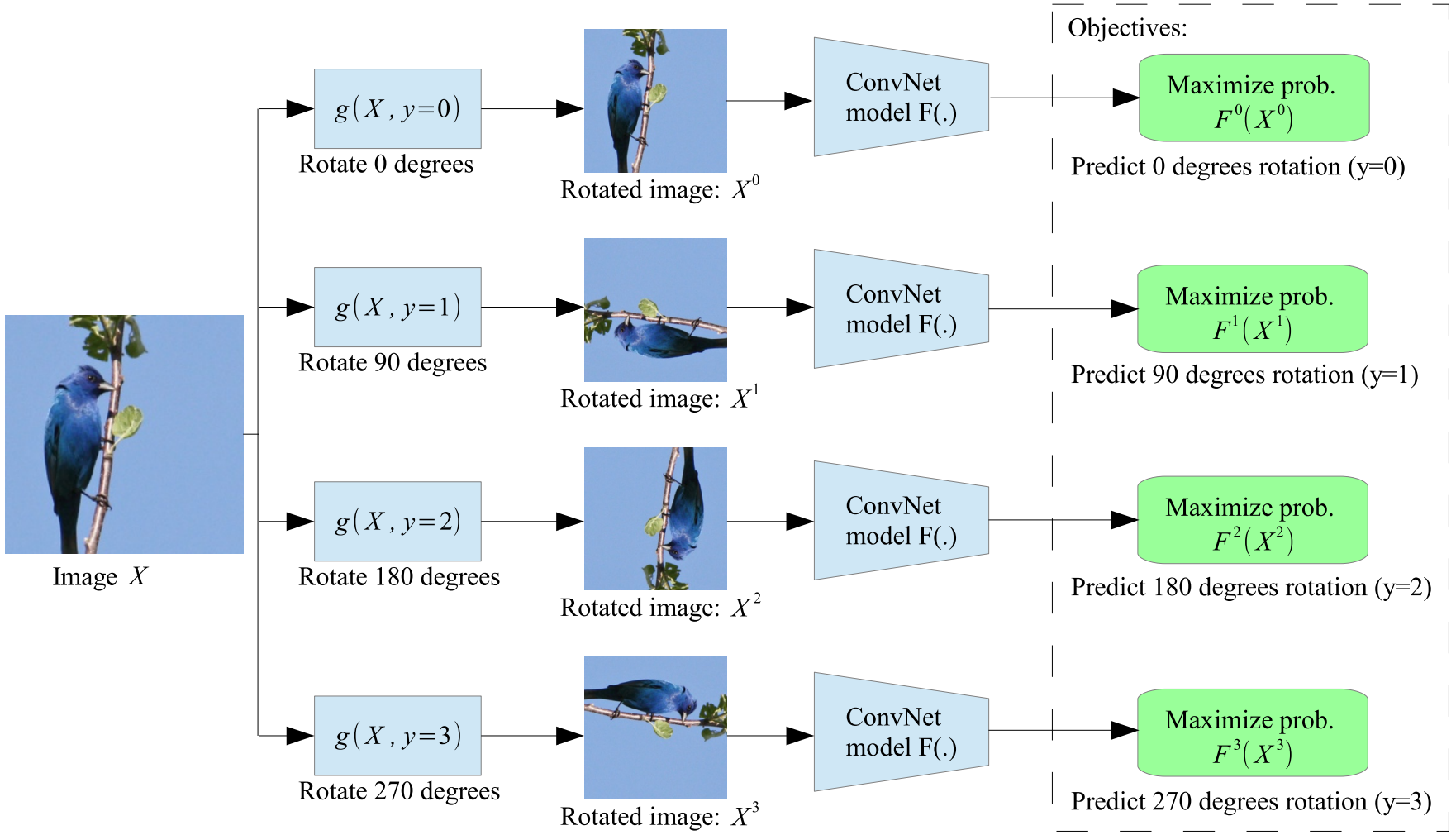
Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky *et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Wang and Gupta [39]	1 week	motion	58.4%	44.0%	-
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%	-
Pathak *et al.* [30]	14 hours	context	56.5%	44.5%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

(Noroozi & Favaro, 2016)

Texture and Color Ignored




Image Rotation



(Gidaris et al., 2018)

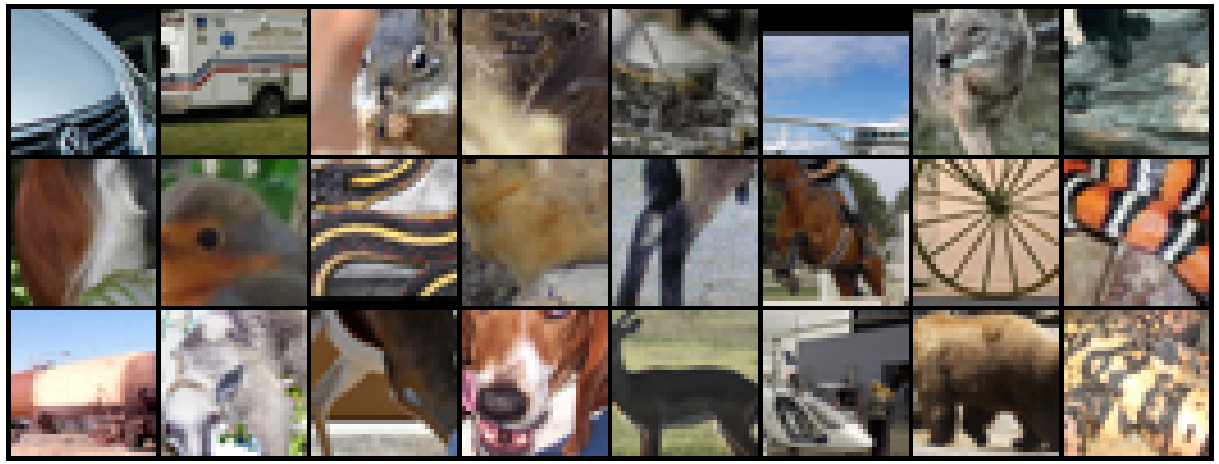
Domain Knowledge

Discriminating among Patches

Sample exemplar patches x_i (high gradient magnitude)

Apply transformations in set τ_i:

- translation
- rotation
- contrast 2
- scaling
- contrast 1
- color



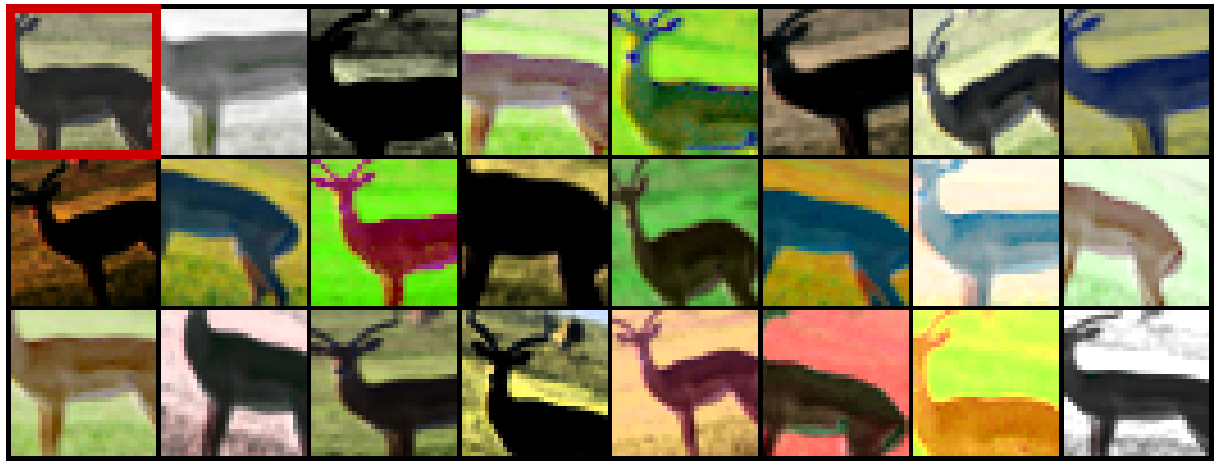
STL dataset high-gradient samples

Classification with pseudo-classes:

$$S_{x_i} = \{\tau_i x_i\} = \{Tx_i \mid T \in \tau_i\}$$

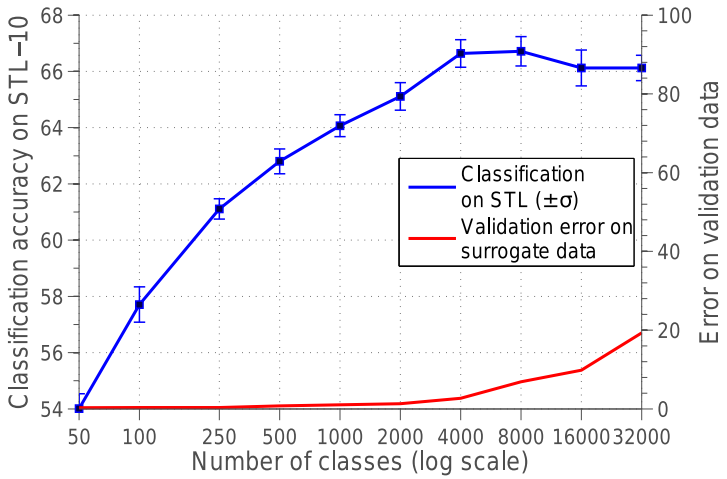
$$L(X) = \sum_{x_i \in X} \sum_{T \in \tau_i} l(i, Tx_i)$$

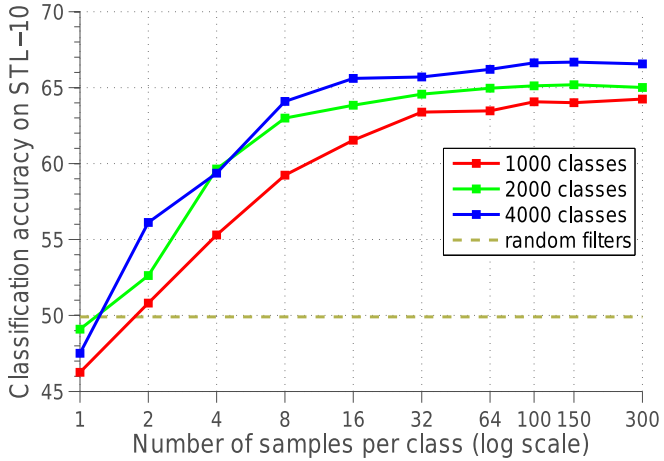
Forces to distinguish different samples x_i



(Dosovitskiy et al., 2014)

Ablation and Performance

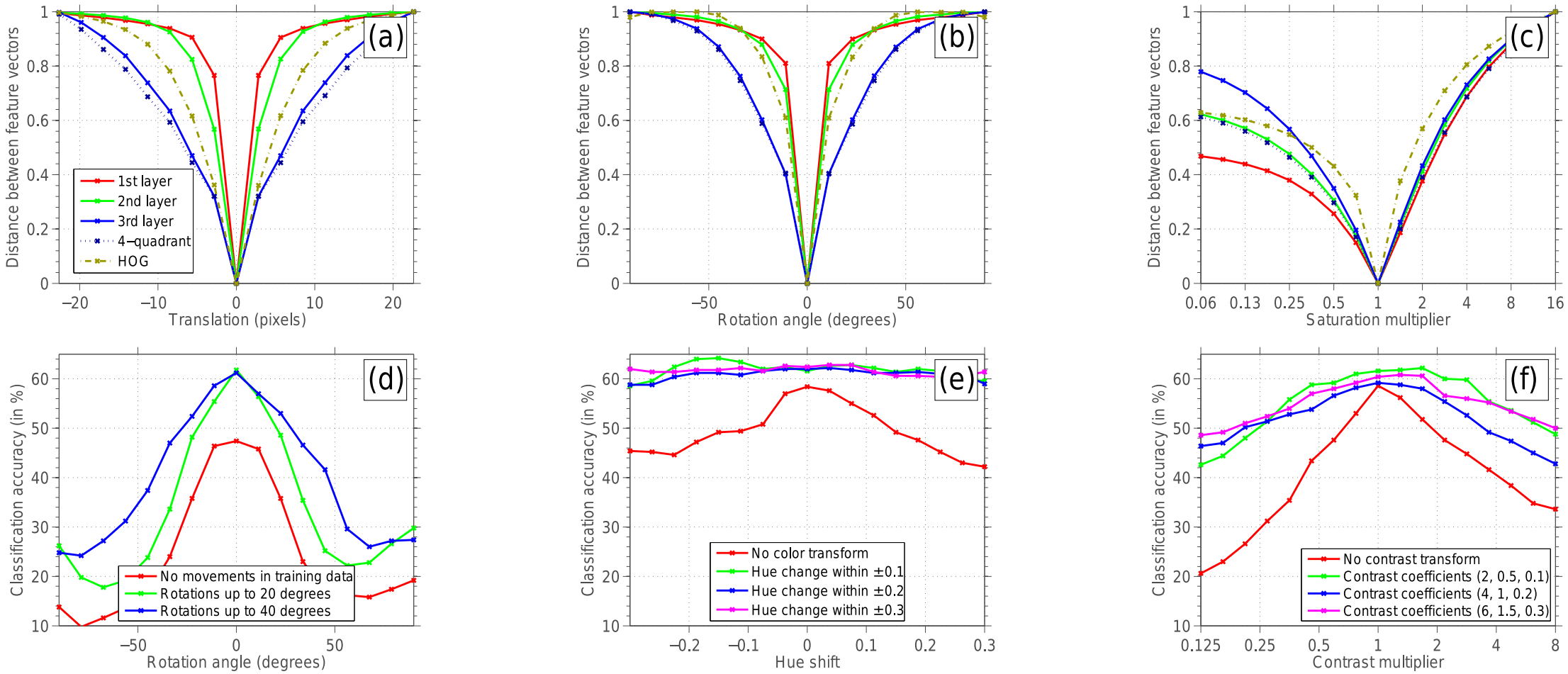




Algorithm	STL-10	CIFAR-10(400)	CIFAR-10	Caltech-101	Caltech-256(30)	#features
Convolutional K-means Network [32]	60.1 ± 1	70.7 ± 0.7	82.0	—	—	8000
Multi-way local pooling [33]	—	—	—	77.3 ± 0.6	41.7	1024×64
Slowness on videos [14]	61.0	—	—	74.6	—	556
Hierarchical Matching Pursuit (HMP) [34]	64.5 ± 1	—	—	—	—	1000
Multipath HMP [35]	—	—	—	82.5 ± 0.5	50.7	5000
View-Invariant K-means [16]	63.7	72.6 ± 0.7	81.9	—	—	6400
Exemplar-CNN (64c5-64c5-128f)	67.1 ± 0.2	69.7 ± 0.3	76.5	$79.8 \pm 0.5^{*}$	42.4 ± 0.3	256
Exemplar-CNN (64c5-128c5-256c5-512f)	72.8 ± 0.4	75.4 ± 0.2	82.2	$86.1 \pm 0.5^{\dagger}$	51.2 ± 0.2	960
Exemplar-CNN (92c5-256c5-512c5-1024f)	$\mathbf{74.2 \pm 0.4}$	$\mathbf{76.6 \pm 0.2}$	**84.3**	$\mathbf{87.1 \pm 0.7}^{\ddagger}$	$\mathbf{53.6 \pm 0.2}$	1884
Supervised state of the art	70.1[36]	—	92.0 [37]	91.44 [38]	70.6 [2]	—

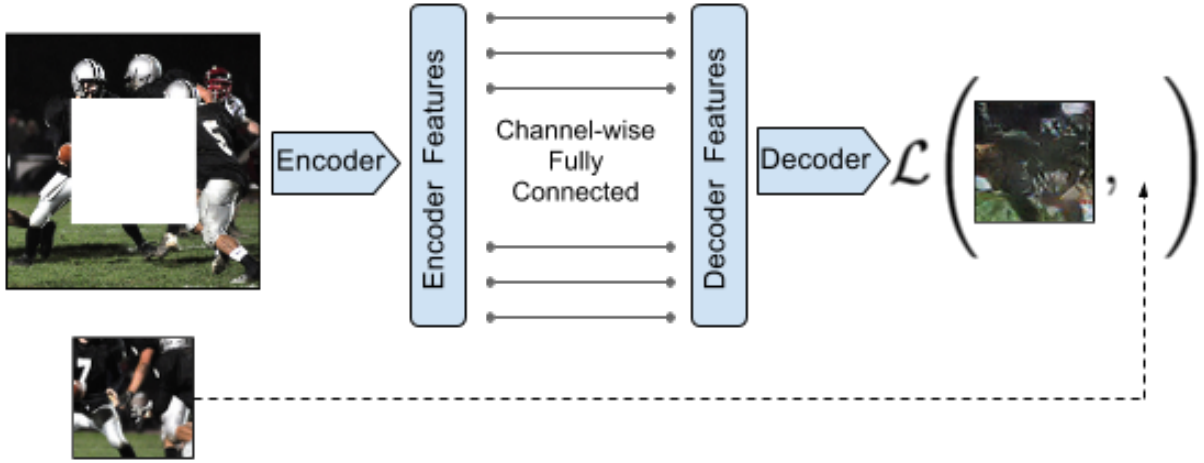
(Dosovitskiy et al., 2014)

Invariance to transformations



(Dosovitskiy et al., 2014)

Inpainting



L2 reconstruction loss

$$\mathcal{L}_{\text{rec}}(x) = \| \widehat{M} \odot \left(x - F\left(\left(1 - \widehat{M}\right) \odot x\right)\right) \|_2^2$$

Adversarial loss

$$\mathcal{L}_{\text{adv}} = \max_D \mathbb{E}_{x \in \mathcal{X}} \Big[\log(D(x)) + \log\Big(1 - D\Big(F\Big(\Big(1 - \widehat{M}\Big) \odot x\Big)\Big)\Big)\Big]$$

input context

human artist

context encoder

context encoder+adv.
(Pathak et al., 2016)

Performance

PASCAL VOC 2007

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%

(Pathak et al., 2016)]

Colorization

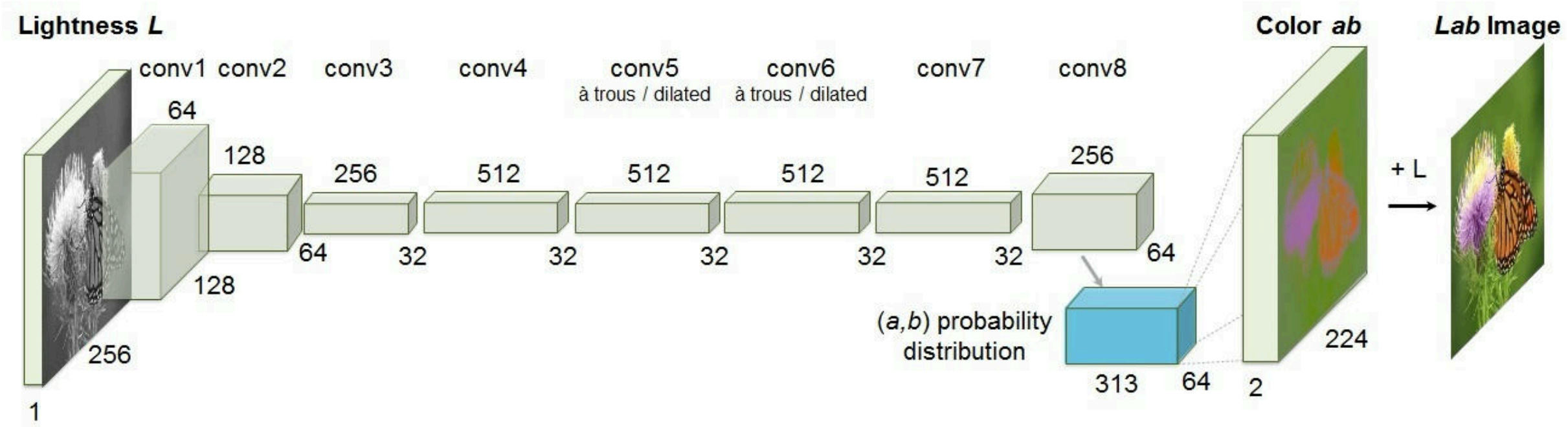
Colorization

Loss formulation requires careful color space treatment

Simple L2 loss is not robust

Averaging effect → desaturated images

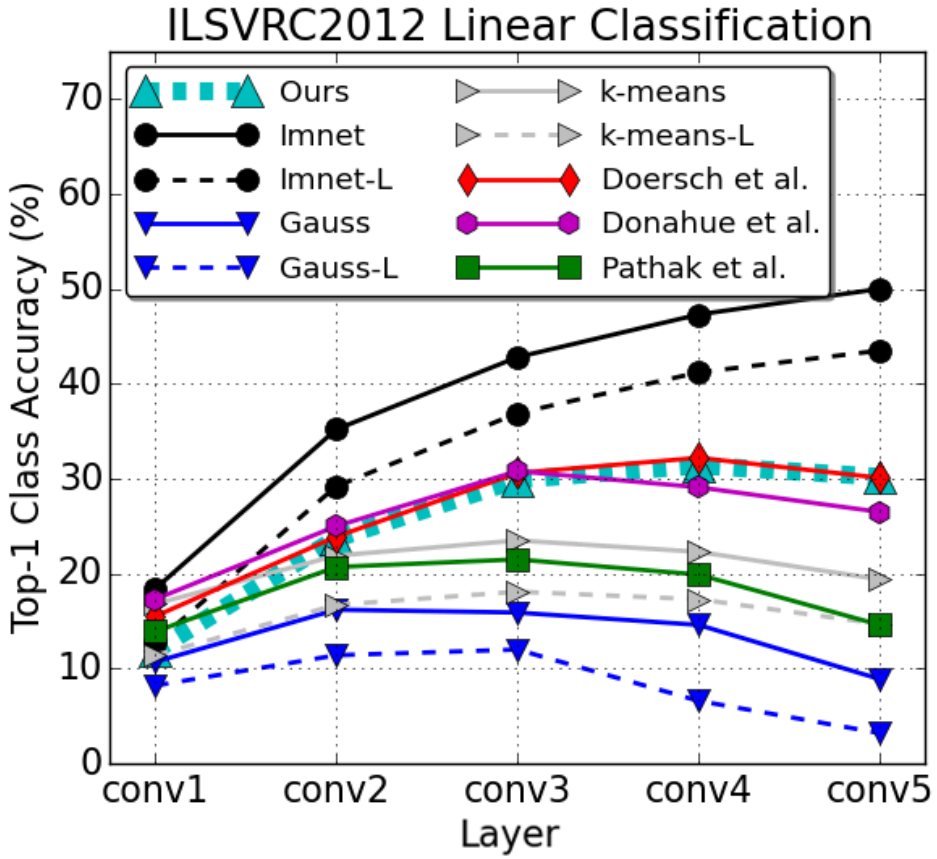
Knowledge of the color space structure required



(Zhang et al., 2016)

Task Generalization

Use intermediate features for classification, object detection and segmentation



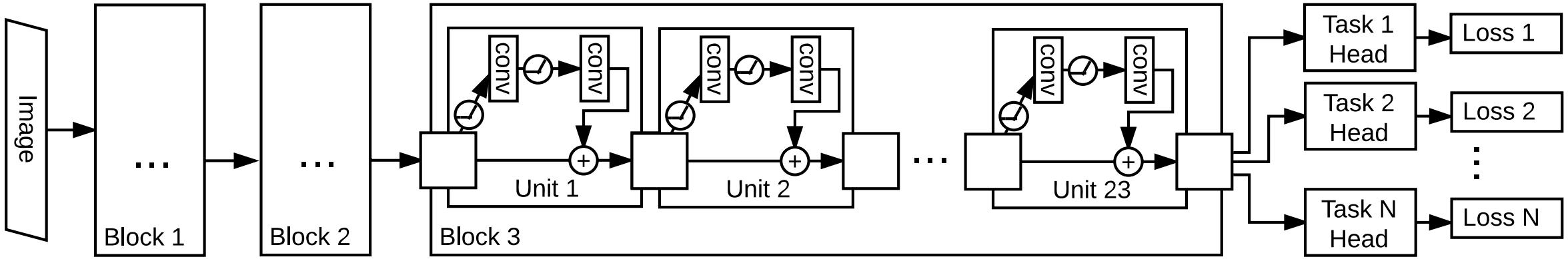
Dataset and Task Generalization on PASCAL [37]		Class. (%mAP)			Det. (%mAP)		Seg. (%mIU)	
fine-tune layers	**[Ref]**	**fc8**	**fc6-8**	**all**	**[Ref]**	**all**	**[Ref]**	**all**
ImageNet [38]	-	76.8	78.9	79.9	[36]	56.8	[42]	48.0
Gaussian	[10]	–	–	53.3	[10]	43.4	[10]	19.8
Autoencoder	[16]	24.8	16.0	53.8	[10]	41.9	[10]	25.2
k-means [36]	[16]	32.0	39.2	56.6	[36]	45.6	[16]	32.6
Agrawal et al. [8]	[16]	31.2	31.0	54.2	[36]	43.9	–	–
Wang & Gupta [15]	–	28.1	52.2	58.7	[36]	47.4	–	–
*Doersch et al. [14]	[16]	44.7	55.1	**65.3**	[36]	**51.1**	–	–
*Pathak et al. [10]	[10]	–	–	56.5	[10]	44.5	[10]	29.7
*Donahue et al. [16]	–	38.2	50.2	58.6	[16]	46.2	[16]	34.9
Ours (gray)	–	**52.4**	**61.5**	**65.9**	–	46.1	–	35.0
Ours (color)	–	**52.4**	**61.5**	**65.6**	–	46.9	–	**35.6**

(Zhang et al., 2016)

Multi-Tasking

By 2017, self-supervised knowledge and deep learning progress accumulated

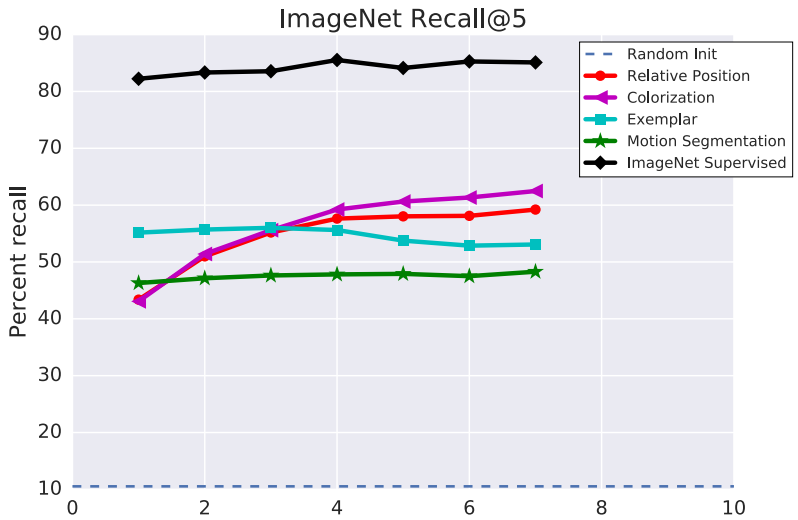
Revision was needed

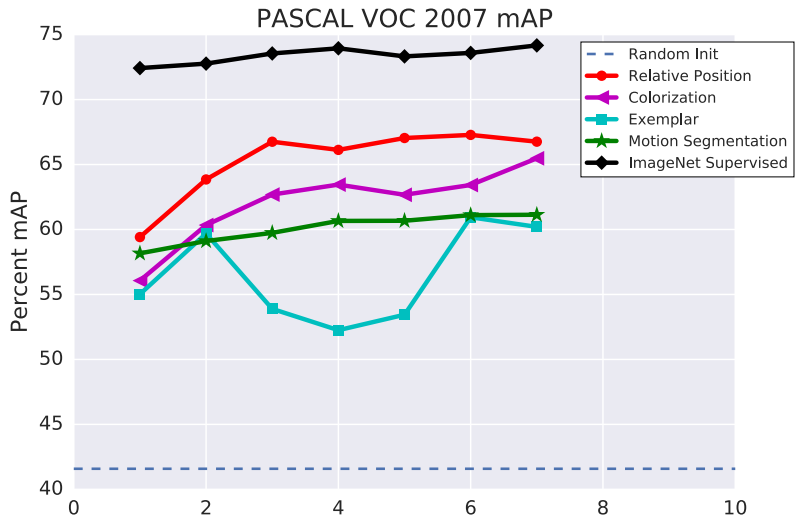


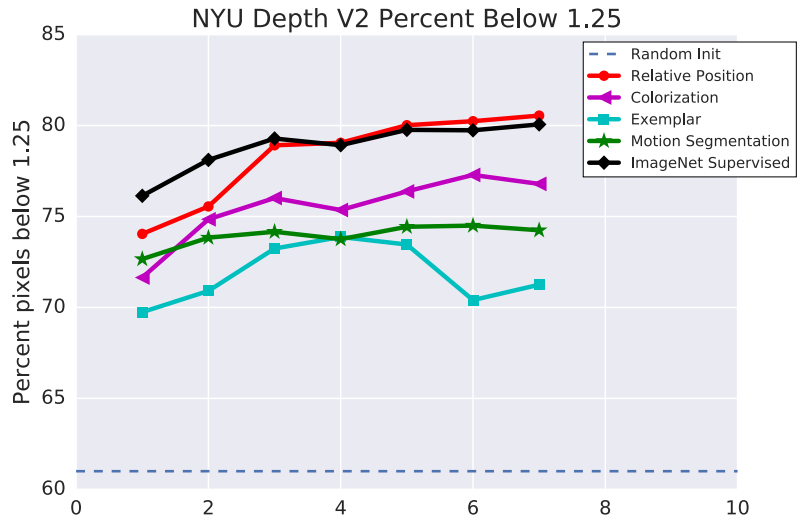
(Doersch & Zisserman, 2017)

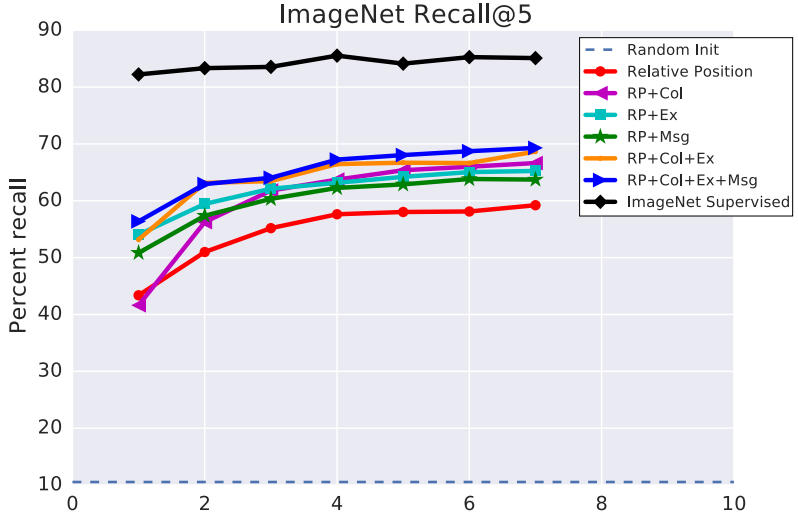
Similar work revisiting video and image approaches (Wang et al., 2017)

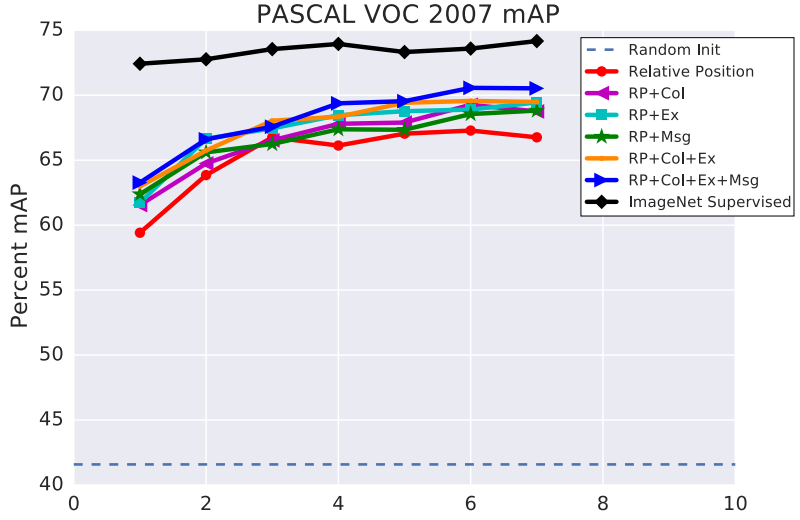
Performance

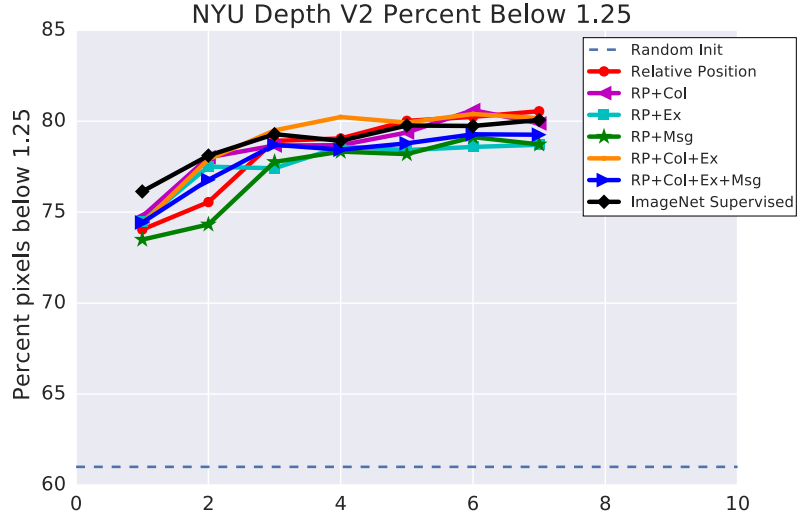












(Doersch & Zisserman, 2017)

Summary

Not hard to come up with pretext task, but

- Careful **task design** needed (though not as much effort as human annotation)
- Attention to avoid **shortcut** solutions (for features to generalize)

Pretext tasks usually cover particular aspect of visual understanding (spatial, color, generation, object recognition)

Combining different aspects generally helps on downstream

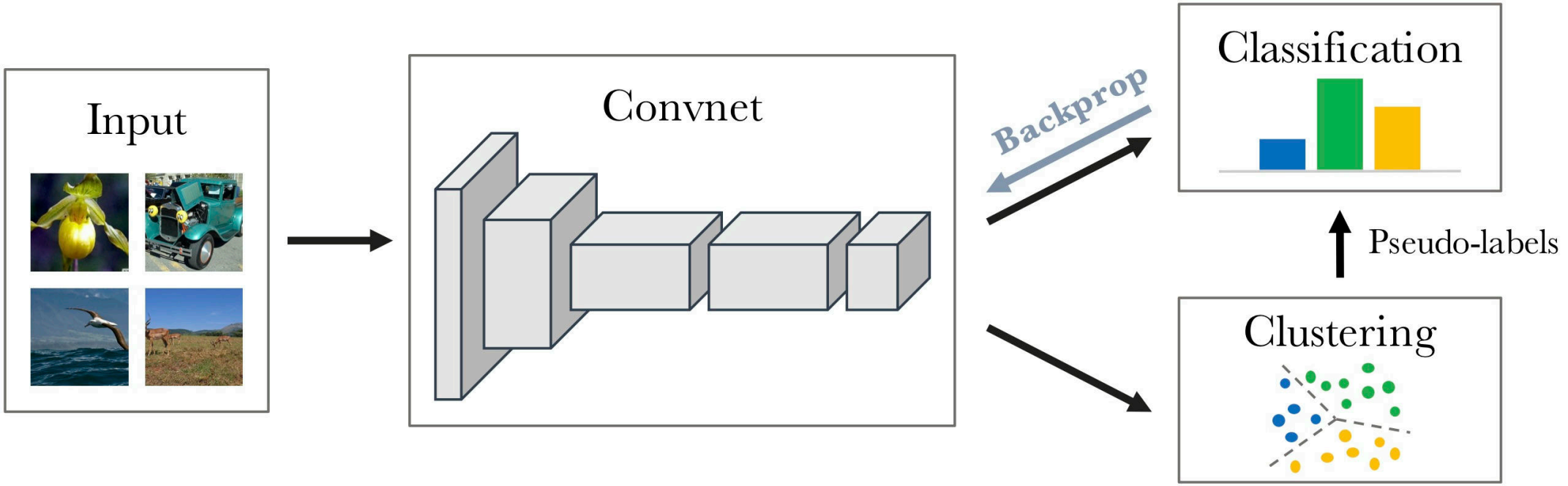
Siamese architecture, projection head, some data augmentations are used to this day

Overall, foundation is laid for further **easier** progress

Deep Cluster

no need for careful task design!

Just iteratively cluster features to get pseudo-labels for classification



(Caron et al., 2018)

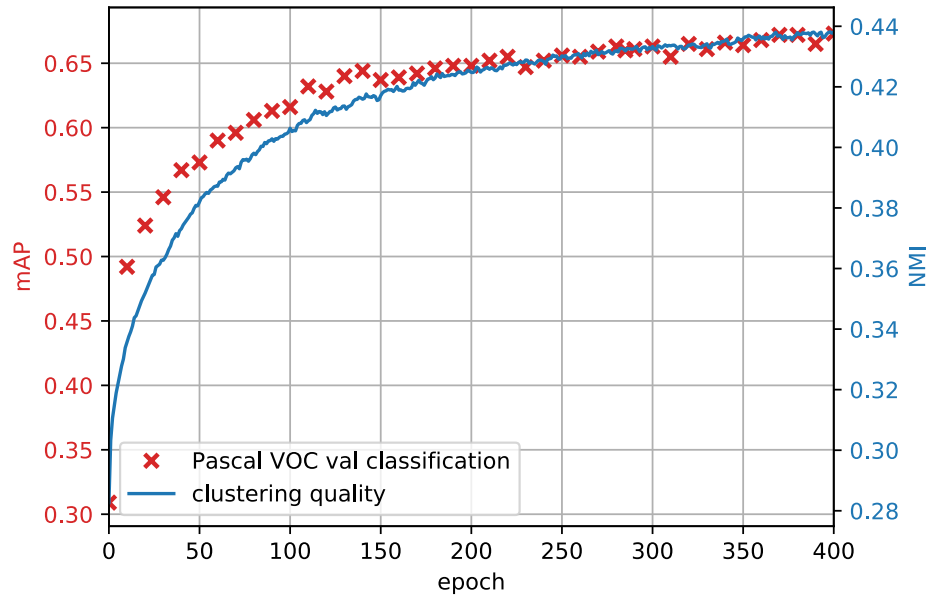
Clustering Quality

Normalized Mutual Information between assignment A and B:

$$\mathrm{NMI}(A;B) := \frac{I(A;B)}{\sqrt{H(A)H(B)}}$$

A, B independent $\rightarrow \mathrm{NMI} = 0$, deterministically predictabilty $\rightarrow \mathrm{NMI} = 1$

Evolution of NMI between cluster assignment and ImageNet labels vs downstream performance



Performance

PASCAL VOC 2007

	Classification		Detection		Segmentation	
Method	FC6-8	ALL	FC6-8	ALL	FC6-8	ALL
ImageNet labels	78.9	79.9	–	56.8	–	48.0
Random-rgb	33.2	57.0	22.2	44.5	15.2	30.1
Random-sobel	29.0	61.9	18.9	47.9	13.0	32.0
Pathak *et al.* [38]	34.6	56.5	–	44.5	–	29.7
Donahue *et al.* [20]*	52.3	60.1	–	46.9	–	35.2
Pathak *et al.* [27]	–	61.0	–	52.2	–	–
Owens *et al.* [44]*	52.3	61.3	–	–	–	–
Wang and Gupta [29]*	55.6	63.1	32.8†	47.2	26.0†	35.4†
Doersch *et al.* [25]*	55.1	65.3	–	51.1	–	–
Bojanowski and Joulin [19]*	56.7	65.3	33.7†	49.4	26.7†	37.1†
Zhang *et al.* [28]*	61.5	65.9	43.4†	46.9	35.8†	35.6
Zhang *et al.* [43]*	63.0	67.1	–	46.7	–	36.0
Noroozi and Favaro [26]	–	67.6	–	53.2	–	37.6
Noroozi *et al.* [45]	–	67.7	–	51.4	–	36.6
DeepCluster	**70.4**	**73.7**	**51.4**	**55.4**	**43.2**	**45.1**

(Caron et al., 2018)

Performance

Probing AlexNet layers

Method	ImageNet					Places				
	conv1	conv2	conv3	conv4	conv5	conv1	conv2	conv3	conv4	conv5
Places labels	–	–	–	–	–	22.1	35.1	40.2	43.3	44.6
ImageNet labels	19.3	36.3	44.2	48.3	50.5	22.7	34.8	38.4	39.4	38.7
Random	11.6	17.1	16.9	16.3	14.1	15.7	20.3	19.8	19.1	17.5
Pathak *et al.* [38]	14.1	20.7	21.0	19.8	15.5	18.2	23.2	23.4	21.9	18.4
Doersch *et al.* [25]	16.2	23.3	30.2	31.7	29.6	19.7	26.7	31.9	32.7	30.9
Zhang *et al.* [28]	12.5	24.5	30.4	31.5	30.3	16.0	25.7	29.6	30.3	29.7
Donahue *et al.* [20]	17.7	24.5	31.0	29.9	28.0	21.4	26.2	27.1	26.1	24.0
Noroozi and Favaro [26]	**18.2**	28.8	34.0	33.9	27.1	23.0	32.1	35.5	34.8	31.3
Noroozi *et al.* [45]	18.0	**30.6**	34.3	32.5	25.7	**23.3**	**33.9**	36.3	34.7	29.6
Zhang *et al.* [43]	17.7	29.3	35.4	35.2	32.8	21.3	30.7	34.0	34.1	32.5
DeepCluster	12.9	29.2	**38.2**	**39.8**	**36.1**	18.6	30.8	**37.0**	**37.5**	**33.1**

(Caron et al., 2018)

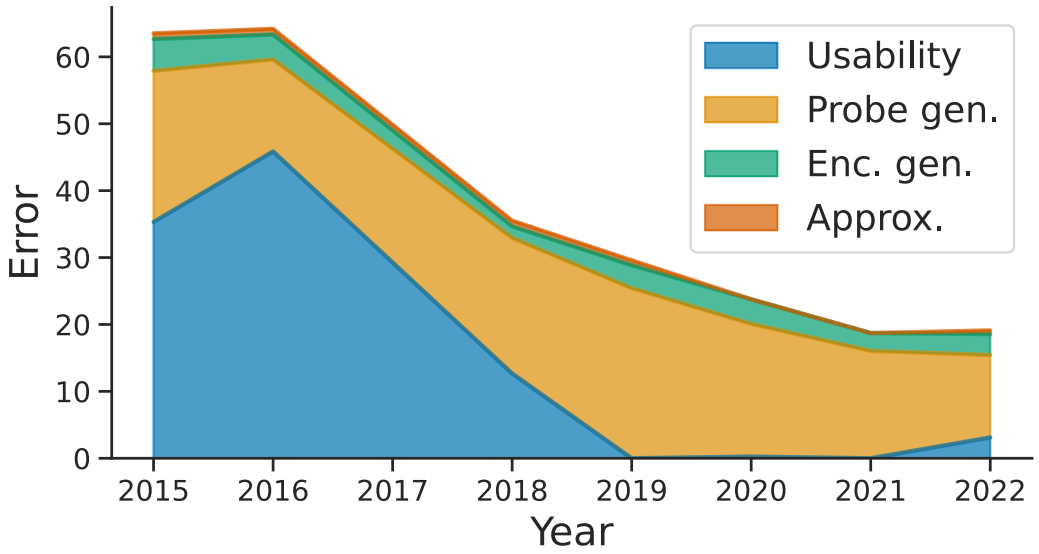

MAE ViT-H14
CLIP ViT-B32
MoCo-v3 ViT-B
ep300
LossyLess ViT-B32
b001
VICRegL ConvNext-B
alpha075
DINO ViT-S16
VICRegL ConvNext-S
alpha075
LossyLess ViT-B32
b005
VICRegL ConvNext-B
alpha09
VICReg RN50w2
VICRegL ConvNext-S
alpha09
MSN ViT-L16
ep600
DeepCluster-v2 RN50
ep800_2x224_6x96
SwAV RN50
LossyLess ViT-B32
b01
MAE ViT-L16
SimCLR RN50w2
DINO RN50
MoCo-v3 RN50
ep1000
SwAV RN50
ep400
BYOL RN50
bs4096
SwAV RN50
ep400_bs256
DeepCluster-v2 RN50
ep400_2x160_4x96
VICReg RN50
SwAV RN50
ep200
DISSL RN50
d8192_e800_m8
MoCo-v3 RN50
ep300
CLIP RN50
DISSL RN50
dnone_e400_m6
SimCLR RN101
BYOL RN50
bs2048
DISSL RN50
d8192_e400_m6
BYOL RN50
bs1024
InfoMin RN50
800ep
SwAV RN50
ep200_bs256
SwAV RN50
ep100
BarlowTwins RN50
BYOL RN50
bs256
BYOL RN50
bs512
SimCLR RN50w4
PIRL RN50w2
headmlp
VICRegL RN50
alpha09
BarlowTwins RN50
ep300
BYOL RN50
augcropcolor
DISSL RN50
dnone_e400_m2
DeepCluster-v2 RN50
ep400_2x224
PIRL RN50
headmlp
MoCo-v3 RN50
ep100
SimCLR RN50w2
ep100
VICRegL RN50
alpha075
SwAV RN50
ep400_2x224
BYOL RN50
bs128
MoCo-v2 RN50
ep800
BYOL RN50
augnogray
PIRL RN50w2
InfoMin RN50
200ep
SimCLR RN50
ep800
SimSiam RN50
bs256_ep200_mmselfsup
SimCLR RN50
ep400
SimSiam RN50
bs256_ep100
SimCLR RN50
DISSL RN50
d4096_e100_m2
SimCLR RN101
ep100
SeLa-v2 RN50
ep400_2x224
SimSiam RN50
bs512_ep100
SimCLR RN50
d8192_e100_m2
SimCLR RN50
ep200
MAE ViT-B16
DISSL RN50
dnone_e100_m2_headtlinslin
DISSL RN50
dnone_e100_m2_headtmlpsmlp
DISSL RN50
d8192_e100_m2
MoCo-v2 RN50
ep200
DISSL RN50
dnone_e100_m2_augsmall
PIRL RN50
ep200_headmlp
DISSL RN50
dnone_e100_m2_auglarge
SimCLR RN50
dnone_e100_m2
SimCLR RN50
dnone_e100_m2_headtmlpsmlp
MoCo-v2 RN50
vissl

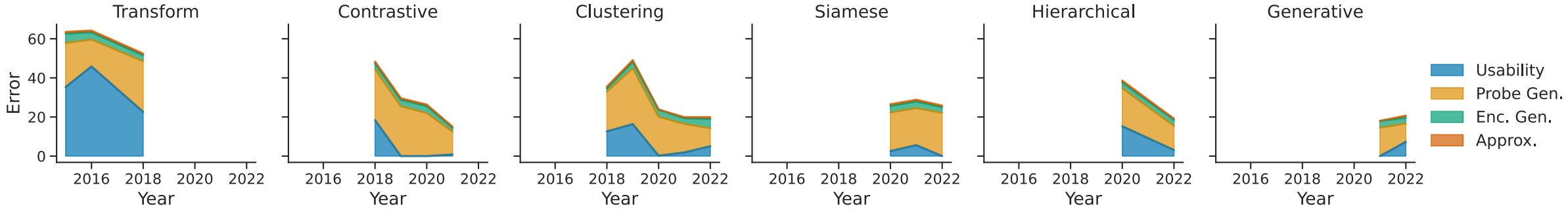
Representation usability

By 2019 SSL algorithms no longer produce representations that are not separable

Representation usability error measures errors due to learning representations via an **SSL pipeline** rather than **supervised learning**.

$R_{\text{repr.usability}} \to 0$ when SSL pretraining yields as much usable information as supervised training.

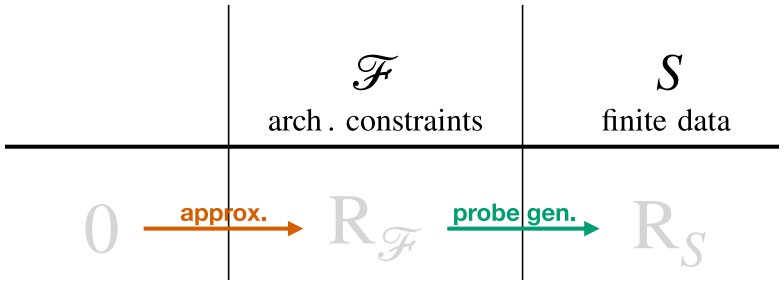




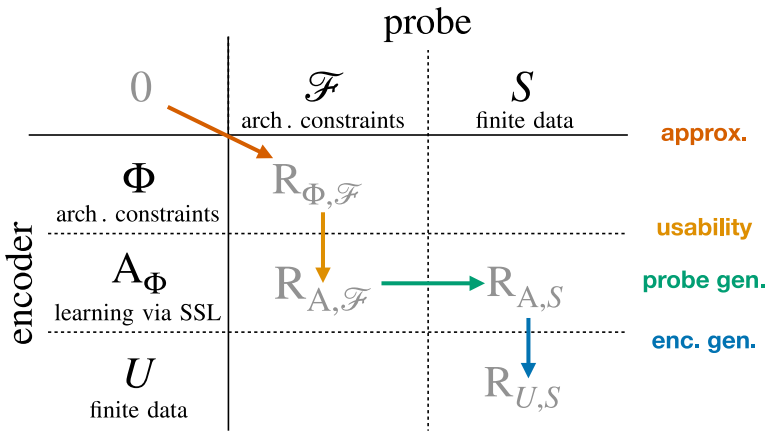
(Dubois et al., 2023)

Representation usability

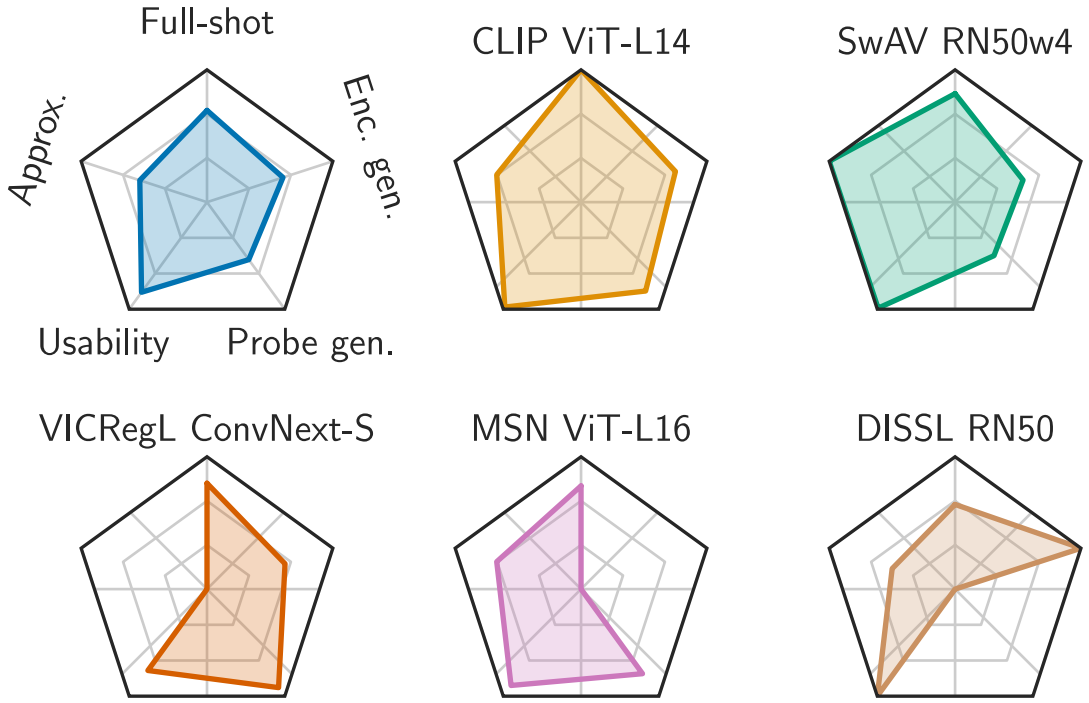
$\varphi \in \Phi$ - encoder (representation function); $f \in \mathcal{F}$ - probe; A_Φ - SSL learning algorithm



supervised setup



SSL setup



risk decomposition for different SSL models

03

Bibliography

Agrawal, P., Carreira, J., & Malik, J. (2015). Learning to see by moving. Proceedings of the IEEE International Conference on Computer Vision, 37–45.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. Proceedings of the European Conference on Computer Vision (ECCV), 132–149.

Doersch, C., & Zisserman, A. (2017). Multi-task self-supervised visual learning. Proceedings of the IEEE International Conference on Computer Vision, 2051–2060.

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. Proceedings of the IEEE International Conference on Computer Vision, 1422–1430.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. Advances in Neural Information Processing Systems, 27.

Dubois, Y., Hashimoto, T., & Liang, P. (2023). Evaluating self-supervised learning via risk decomposition. International Conference on Machine Learning, 8779–8820.

Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. Arxiv Preprint Arxiv:1803.07728.

Jing, L., & Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(11), 4037–4058.

Misra, I., Zitnick, C. L., & Hebert, M. (2016). Shuffle and learn: unsupervised learning using temporal order verification. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, 527–544.

Mobahi, H., Collobert, R., & Weston, J. (2009). Deep learning from temporal coherence in video. Proceedings of the 26th Annual International Conference on Machine Learning, 737–744.

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. European Conference on Computer Vision, 69–84.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2536–2544.

Schmidhuber, J. (1990,). Making the world differentiable: On using self-supervised fully recurrent n eu al networks for dynamic reinforcement learning and planning in non-stationary environments. Citeseer.

Wang, X., & Gupta, A. (2015). Unsupervised learning of visual representations using videos. Proceedings of the IEEE International Conference on Computer Vision, 2794–2802.

Wang, X., He, K., & Gupta, A. (2017). Transitive invariance for self-supervised visual representation learning. Proceedings of the IEEE International Conference on Computer Vision, 1329–1338.

Wiskott, L., & Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. Neural Computation, 14(4), 715–770.

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14, 649–666.

Thank you!