Fundamentals of Machine Learning:

Convolutional Neural Networks: Advanced Topics

Prof. Andrew D. Bagdanov (andrew.bagdanov AT unifi.it)

UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

Outline

Introduction

Annotating images is expensive, laborious, and noisy

- Commercial rates for image annotation are about USD 0.08 per annotation.
- Let's do some napkin calculations...

Large-scale training is expensive and time-consuming

- Even with massive amounts of labeled data, training a state-of-the-art architecture can take weeks.
- For one training run. If you're optimizing hyperparameters for a complex model, even longer.
- Some of this can be parallelized, but then you have GPU and energy costs to factor in.

Model adaptation

- As usual, there is no silver bullet for these issues.
- However, we can at least mitigate somewhat via:
  - Transfer learning: can we exploit learned representations to derive solutions to new problems?
  - Self-supervised learning: can we mitigate the labeling burden via derived proxy tasks?

Lecture Objectives

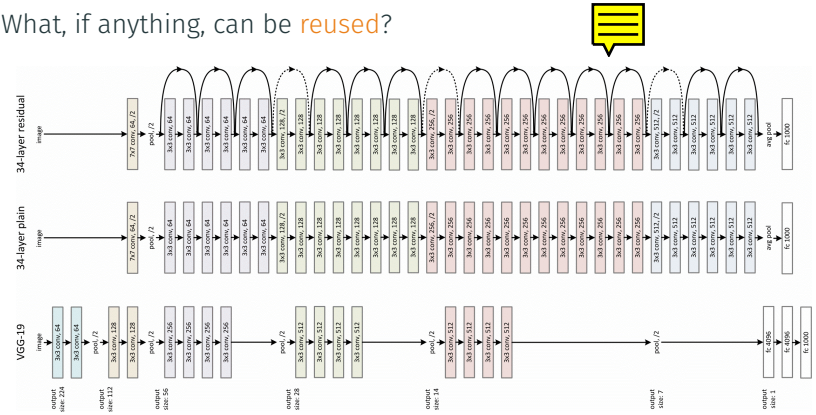
At the end of this lecture you should:

- Understand how transfer learning can be used to exploit pre-trained backbones via fine-tuning.
- Understand how the Convolutional Neural Network architecture can be repurposed to solve complex tasks.
- Understand how self-supervised learning can mitigate entirely the need for supervision.

Transfer Learning and Domain Adaptation

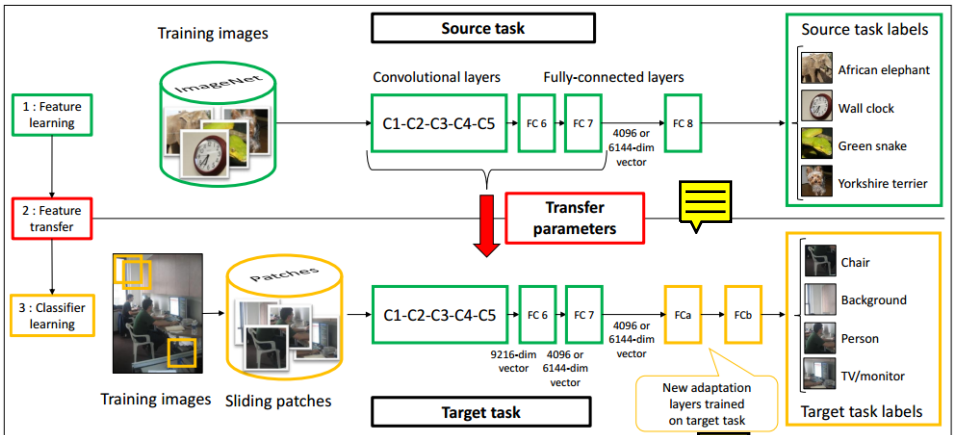
Transfer Learning: Work and reuse

- If we look at a state-of-the-art CNN, let's ask ourselves:
  - What are we investing in when training?
  - Where are the features in the network?
  - What, if anything, can be reused?



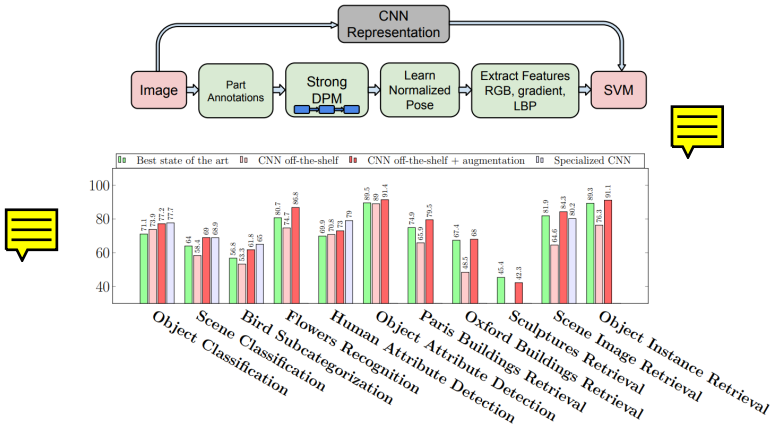
Transfer Learning: The basic idea

- TL;DR: why on earth start from scratch?



Transfer Learning: Back to basics

- Trained CNNs are feature extractors [Sharif Razavian et al., 2014]:



Transfer Learning: Fine-grained recognition

- Not all recognition problems are created equal.
- Fine-grained recognition should require different features.

Transfer Learning: Fine-grained recognition

- Not all recognition problems are created equal.
- Fine-grained recognition should require different features.

Method	Part info	mean Accuracy
Sift+Color+SVM[45]	✗	17.3
Pose pooling kernel[49]	✓	28.2
RF[47]	✓	19.2
DPD[50]	✓	51.0
Poof[5]	✓	56.8
CNN-SVM	✗	53.3
CNNaug-SVM	✗	**61.8**
DPD+CNN(DeCaf)+LogReg[10]	✓	**65.0**

Transfer Learning: Instance recognition

- Instance recognition, kind of a limit of fine-grained:

Transfer Learning: Instance recognition

- Instance recognition, kind of a limit of fine-grained:

	Dim	Oxford5k	Paris6k	Sculp6k	Holidays	UKBench
BoB[3]	N/A	N/A	N/A	**45.4**[3]	N/A	N/A
BoW	200k	36.4[20]	46.0[35]	8.1[3]	54.0[4]	70.3[20]
IFV[33]	2k	41.8[20]	-	-	62.6[20]	83.8[20]
VLAD[4]	32k	55.5 [4]	-	-	64.6[4]	-
CVLAD[52]	64k	47.8[52]	-	-	81.9[52]	89.3[52]
HE+burst[17]	64k	64.5[42]	-	-	78.0[42]	-
AHE+burst[17]	64k	66.6[42]	-	-	79.4[42]	-
Fine vocab[26]	64k	74.2[26]	74.9[26]	-	74.9[26]	-
ASMK*+MA[42]	64k	80.4[42]	77.0[42]	-	81.0[42]	-
ASMK+MA[42]	64k	**81.7**[42]	78.2[42]	-	82.2[42]	-
CNN	4k	32.2	49.5	24.1	64.2	76.0
CNN-ss	32-120k	55.6	69.7	31.1	76.9	86.9
CNNaug-ss	4-15k	**68.0**	**79.5**	42.3	**84.3**	**91.1**
CNN+BOW[16]	2k	-	-	-	**80.2**	-

Transfer Learning: All the devilish details

- The VGG group has an excellent and thorough exploration of transfer learning (and not only) in CNNs [Chatfield et al., 2014].

Method	SPool	Image		Aug.		Dim	mAP															
(I) FK BL	spm	–				327K	**61.69**	79.0	67.4	51.9	70.9	30.8	72.2	(I)	79.9	61.4	56.0	49.6	58.4	44.8	78.8	70.8	85.0	31.7	51.0	56.4	80.2	57.5
(II) DECAF	–	(C)	t	t		327K	**73.41**	87.4	79.3	84.1	78.4	42.3	73.7	(II)	83.7	83.7	54.3	61.9	70.2	79.5	85.3	77.2	90.9	51.1	73.8	57.0	86.4	68.0
(a) FK	spm	–				327K	**63.66**	83.4	68.8	59.6	74.1	35.7	71.2	(a)	80.7	64.4	53.8	53.8	60.2	47.8	79.9	68.9	86.1	37.3	51.1	55.8	83.7	56.9
(b) FK IN	spm	–				327K	**64.18**	82.1	69.7	59.7	75.2	35.7	71.3	(b)	80.6	64.8	53.9	54.9	60.7	50.5	80.4	69.5	86.2	38.3	54.4	56.3	82.7	56.7
(c) FK	(x,y)	–				42K	**63.51**	83.2	69.4	60.6	73.9	36.3	68.6	(c)	81.1	64.2	51.1	53.4	61.9	50.0	80.0	67.5	85.3	35.7	51.9	53.8	83.5	58.9
(d) FK IN	(x,y)	–				42K	**64.36**	83.1	70.4	62.4	75.2	37.1	69.1	(d)	80.5	66.9	50.9	53.9	62.1	51.5	80.5	68.5	85.9	37.2	55.2	54.3	83.3	59.2
(e) FK IN	(x,y)	(F)	f	-		42K	**64.35**	83.1	70.5	62.3	75.4	37.1	69.1	(e)	80.5	66.8	51.0	54.1	62.2	51.5	80.4	68.2	86.0	37.3	55.1	54.2	83.3	59.2
(f) FK IN	(x,y)	(C)	f	s		42K	**67.17**	85.5	71.6	64.6	77.2	39.0	70.8	(f)	82.4	71.6	52.8	62.4	63.4	57.1	81.6	70.9	86.9	41.2	61.2	56.9	85.2	61.5
(g) FK IN	(x,y)	(C)	s	s		42K	**66.68**	84.9	70.1	64.7	76.3	39.2	69.8	(g)	81.9	71.0	52.8	61.6	62.2	56.8	81.8	70.0	86.5	41.5	61.0	56.5	84.3	60.9
(h) FK IN 512	(x,y)	–				84K	**65.36**	84.1	70.4	65.0	76.7	37.2	71.3	(h)	81.1	67.9	52.6	55.4	61.4	51.2	80.5	69.1	86.4	41.2	56.0	56.2	83.7	59.9
(i) FK IN 512	(x,y)	(C)	f	s		84K	**68.02**	85.9	71.8	67.1	77.1	38.8	72.3	(i)	82.5	73.2	54.7	62.7	64.5	56.6	82.2	71.3	87.5	43.0	62.0	59.3	85.7	62.4
(j) FK IN COL 512	–	–				82K	**52.18**	69.5	52.1	47.5	64.0	24.6	49.8	(j)	66.1	46.6	42.5	35.8	41.1	45.5	75.4	58.3	83.9	39.8	47.3	35.6	69.2	49.0
(k) FK IN 512 COL+	(x,y)	–				166K	**66.37**	82.9	70.1	67.0	77.0	36.1	70.0	(k)	80.0	65.9	52.8	56.1	61.0	56.9	81.4	69.6	88.4	49.0	59.2	56.4	84.7	62.8
(l) FK IN 512 COL+	(x,y)	(C)	f	s		166K	**67.93**	85.1	70.5	67.5	77.4	35.7	71.2	(l)	81.6	70.8	52.9	59.6	63.1	59.9	82.1	70.5	88.9	50.6	63.7	57.5	86.1	64.1
(m) CNN F	–	(C)	f	s		4K	**77.38**	88.7	83.9	87.0	84.7	46.9	77.5	(m)	86.3	85.4	58.6	71.0	72.6	82.0	87.9	80.7	91.8	58.5	77.4	66.3	89.1	71.3
(n) CNN S	–	(C)	f	s		4K	**79.74**	90.7	85.7	88.9	86.6	50.5	80.1	(n)	87.8	88.3	61.3	74.8	74.7	87.2	89.0	83.7	92.3	58.8	80.5	69.4	90.5	74.0
(o) CNN M	–	–				4K	**76.97**	89.5	84.3	88.8	83.2	48.4	77.0	(o)	85.1	87.4	58.1	70.4	73.1	83.5	85.5	80.9	90.8	54.1	78.9	61.1	89.0	70.4
(p) CNN M	–	(C)	f	s		4K	**79.89**	91.7	85.4	89.5	86.6	51.6	79.3	(p)	87.7	88.6	60.3	80.1	74.4	85.9	88.2	84.6	92.1	60.3	80.5	66.2	91.3	73.5
(q) CNN M	–	(C)	f	m		4K	**79.50**	90.9	84.6	89.4	85.8	50.3	78.4	(q)	87.6	88.6	60.7	78.2	73.6	86.0	87.4	83.8	92.3	59.3	81.0	66.8	91.3	74.0
(r) CNN M	–	(C)	s	s		4K	**79.44**	91.4	85.2	89.1	86.1	52.1	78.0	(r)	87.5	88.1	60.4	76.9	74.8	85.8	88.1	84.3	92.2	59.5	79.3	65.8	90.8	73.5
(s) CNN M	–	(C)	t	t		41K	**78.77**	90.7	85.0	89.2	85.8	51.0	77.8	(s)	87.3	87.6	60.1	72.3	75.3	85.2	86.9	82.6	91.9	58.5	77.9	66.5	90.5	73.4
(t) CNN M	–	(C)	f	-		4K	**77.78**	90.5	84.3	88.8	84.5	47.9	78.0	(t)	85.7	87.9	58.3	74.2	73.9	84.7	86.6	82.0	91.0	55.8	79.2	62.1	89.3	71.0
(u) CNN M	–	(F)	f	-		4K	**76.99**	90.1	84.2	89.0	83.5	48.1	77.2	(u)	85.3	87.3	58.1	70.0	73.4	83.5	86.0	80.8	90.9	53.9	78.1	61.2	88.8	70.6
(v) CNN M GS	–	–				4K	**73.59**	87.4	80.8	82.4	82.1	44.5	73.5	(v)	85.0	84.9	57.8	65.9	69.8	79.5	82.9	77.4	89.2	42.8	71.7	60.2	86.3	67.8
(w) CNN M GS	–	(C)	f	s		4K	**77.00**	89.4	83.8	85.1	84.4	49.4	77.6	(w)	87.2	86.5	59.5	72.4	74.1	84.7	86.0	82.3	90.8	48.9	73.7	66.8	90.1	72.0
(x) CNN M 2048	–	(C)	f	s		2K	**80.10**	91.3	85.8	89.9	86.7	52.4	79.7	(x)	87.6	88.4	60.2	76.9	75.4	85.5	88.0	83.4	92.1	61.1	83.1	68.5	91.9	74.2
(y) CNN M 1024	–	(C)	f	s		1K	**79.91**	91.4	86.9	89.3	85.8	53.3	79.8	(y)	87.8	88.6	59.0	77.2	73.1	85.9	88.1	83.5	91.8	59.9	81.4	68.3	93.0	74.1
(z) CNN M 128	–	(C)	f	s		128	**78.60**	91.3	83.9	89.2	86.9	52.1	81.0	(z)	86.6	87.5	59.1	70.0	72.9	84.6	86.7	83.6	89.4	57.0	81.5	64.8	90.4	73.4
(α) FK+CNN F	(x,y)	(C)	f	s		88K	**77.95**	89.6	83.1	87.1	84.5	48.0	79.4	(α)	86.8	85.6	59.9	72.0	73.4	81.4	86.4	80.5	92.1	60.6	77.3	66.4	89.3	73.3
(β) FK+CNN M 2048	(x,y)	(C)	f	s		86K	**80.14**	90.9	85.9	88.8	85.5	52.3	81.4	(β)	87.7	88.4	61.2	76.9	76.6	84.9	89.1	82.9	92.4	61.9	80.9	68.7	91.5	75.1
(γ) CNN S TUNE-RNK	–	(C)	f	s		4K	**82.42**	95.3	90.4	92.5	89.6	54.4	81.9	(γ)	91.5	91.9	64.1	76.3	74.9	89.7	92.2	86.9	95.2	60.7	82.9	68.0	95.5	74.4

Transfer Learning: All the devilish details

- Comparison with the state-of-the-art:

	ILSVRC-2012 (top-5 error)	VOC-2007 (mAP)	VOC-2012 (mAP)	Caltech-101 (accuracy)	Caltech-256 (accuracy)
(a) FK IN 512	-	68.0	–	–	–
(b) CNN F	16.7	77.4	79.9	–	–
(c) CNN M	13.7	79.9	82.5	87.15 ± 0.80	77.03 ± 0.46
(d) CNN M 2048	13.5	80.1	82.4	86.64 ± 0.53	76.88 ± 0.35
(e) CNN S	**13.1**	79.7	82.9	87.76 ± 0.66	**77.61 ± 0.12**
(f) CNN S TUNE-CLS	**13.1**	-	83.0	**88.35 ± 0.56**	77.33 ± 0.56
(g) CNN S TUNE-RNK	**13.1**	**82.4**	**83.2**	–	–
(h) Zeiler & Fergus [19]	16.1	-	79.0	86.5 ± 0.5	74.2 ± 0.3
(i) Razavian *et al.* [9], [10]	14.7	77.2	–	–	–
(j) Oquab *et al.* [8]	18	77.7	78.7 (82.8*)	–	–
(k) Oquab *et al.* [16]	-	-	**86.3***	–	–
(l) Wei *et al.* [17]	-	81.5 (**85.2***)	81.7 (**90.3***)	–	–
(m) He *et al.* [29]	13.6	80.1	-	**91.4 ± 0.7**	–

Return of the devil in the details: Delving deep into convolutional nets

Transfer Learning: Not just recognition

- Transfer learning can be applied to almost any visual recognition or estimation task.
- This includes object detection [Ren et al., 2015], semantic segmentation [Long et al., 2015], crowd counting [Liu et al., 2018], you name it.

task	2nd-place winner	MSRA	margin (relative)
ImageNet Localization (top-5 error)	12.0	9.0	**27%**
ImageNet Detection (mAP@.5)	53.6	62.1	**16%**
COCO Detection (mAP@.5:.95)	33.5	37.3	**11%**
COCO Segmentation (mAP@.5:.95)	25.1	28.2	**12%**

absolute 8.5% better!

Backbone Repurposing

Semantic Segmentation: Dense Predictions

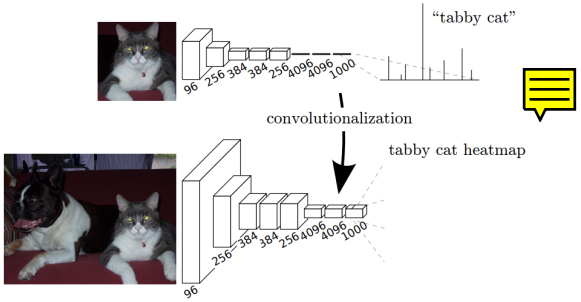
- What if we want to classify at the pixel level?

(a) image (b) semantic segmentation

(c) instance segmentation (d) panoptic segmentation

Fully convolutionalizing a network

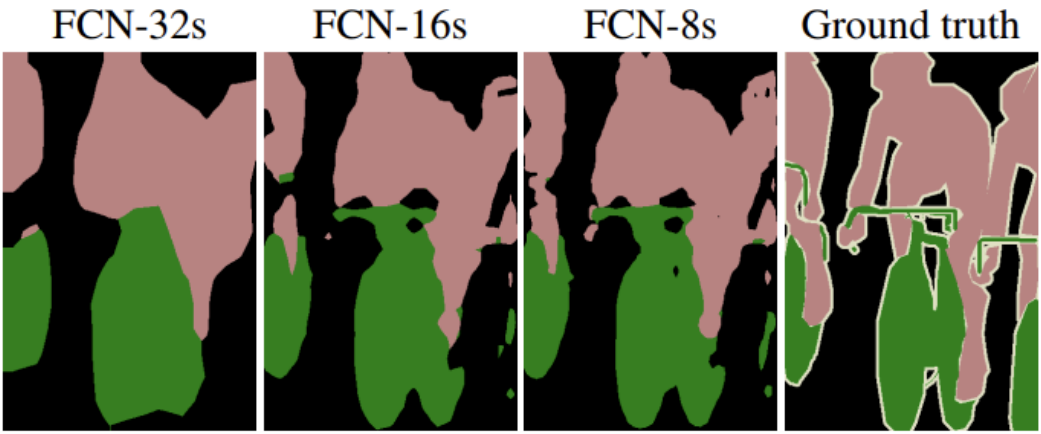
- What the hell does “fully convolutionalizing mean”?



Fully convolutional networks for semantic segmentation [*Long et al., 2015*]

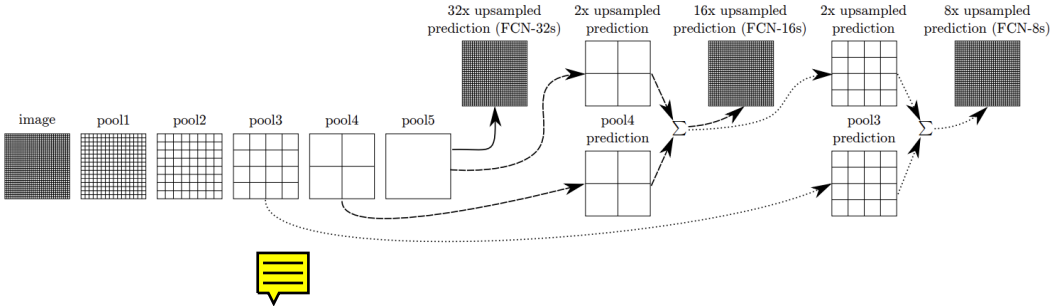
Fully Convolutional Networks for Semantic Segmentation

- Isn't the spatial resolution really low at the end of the network? Yes. Yes it is.



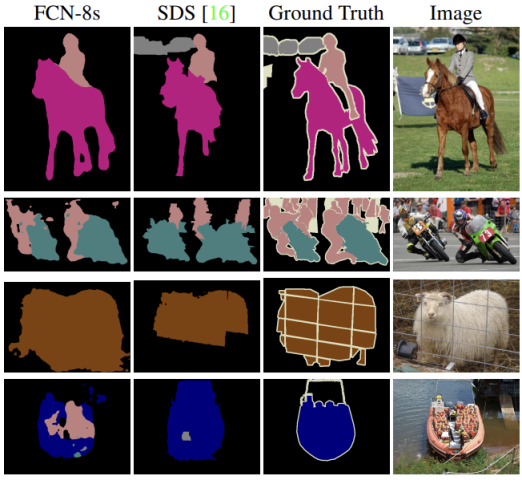
Fully Convolutional Networks for Semantic Segmentation

1. Fully-convolutionalize a model pre-trained for object recognition.
2. Fuse predictions from multiple scales.
3. Profit.



Fully convolutional networks for semantic segmentation [Long et al., 2015]

Fully Convolutional Networks for Semantic Segmentation

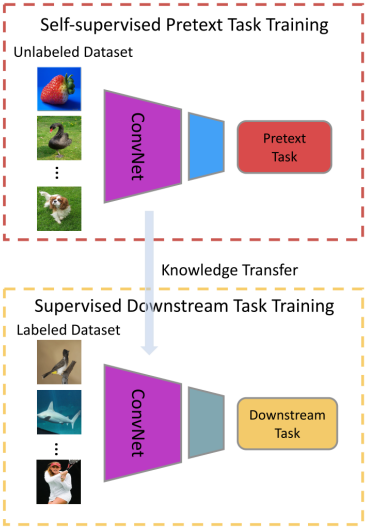


Self-supervised Learning

What's this all about?

- We (myself included) often tend to bi-modalize supervision regimes.
- That is, we think of learning as being supervised or unsupervised.
- Of course, there is a whole spectrum of supervision regimes:
  - unsupervised: where no supervisory signal is used.
  - self-supervised: where supervisory signal is derived from data.
  - weakly-supervised: where supervision is noisy or imprecise
  - semi-supervised: where some samples are supervised
- Here we will look at a few self-supervision techniques for learning general image representations.

What's this all about?

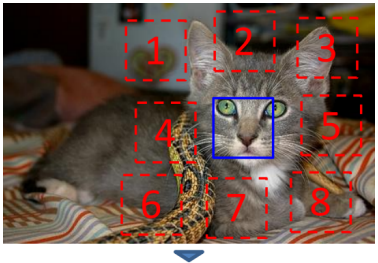


- This is the schema you should keep in mind when thinking about self-supervised learning.
- Step 1: define a meaningful proxy task to stand in for what you really want to do.
- Step 2: train a deep model on unlabeled data using your proxy loss.
- Step 3: transfer learned weights to new model for fine-tuning on limited amounts of labeled data.

Before we begin, *a note on terminology*...

Spatial context prediction

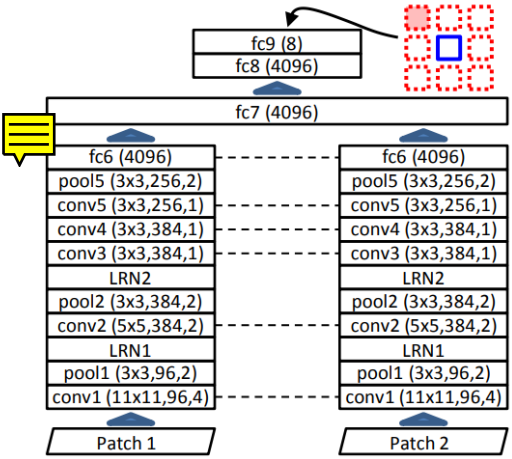
- Idea: train the network to predict the local context of image patches.
- Feed a network a pair of patches, train to predict which neighbor the second one is wrt the first.



Unsupervised visual representation learning by context prediction [Doersch et al., 2015]

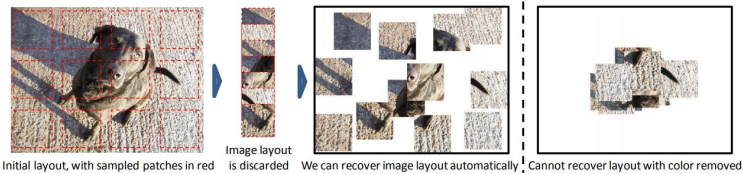
Spatial context prediction

- The network architecture is Siamese – two patches are passed through two (conceptually) different branches
- Context directions are converted into an 8-way classification problem.
- The network can be trained with a standard cross-entropy loss.
- You must always be sure the network can't cheat – in this case:
  - Directly adjacent patches admit trivial solutions; and
  - Chromatic aberration was also a surprising culprit.



Unsupervised visual representation learning by context prediction [Doersch et al., 2015]

Aside: lowdown, cheating CNNs

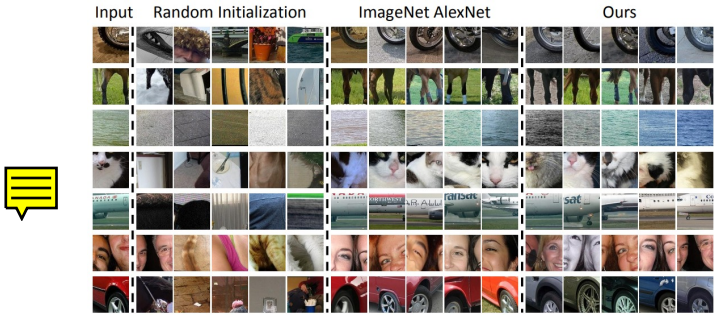


- We must always be careful to understand what the network is learning.
- CCD cameras suffer from chromatic aberration which causes them to compress the range of the green color channel more towards the center of the image.
- The network can exploit this to orient patches with respect to one another.

Unsupervised visual representation learning by context prediction [Doersch et al., 2015]

Learning spatial context learns *semantics*

- How can we tell if our self-supervision is guiding the network to learn something about image semantics?
- One way: compute the network representation for small patches of images, look at nearest neighbors.



Spatial context prediction

- Results on PASCAL 2007 Object Detection:

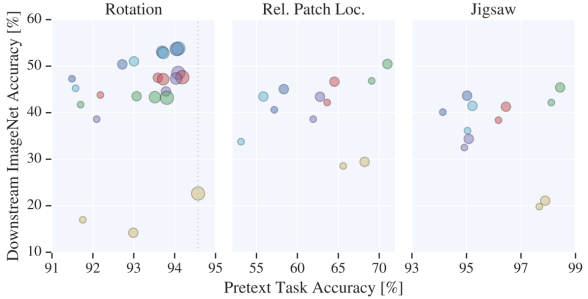
VOC-2007 Test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM-v5[17]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
[8] w/o context	52.6	52.6	19.2	25.4	18.7	47.3	56.9	42.1	16.6	41.4	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

Unsupervised visual representation learning by context prediction [Doersch et al., 2015]

A self-supervised learning retrospective

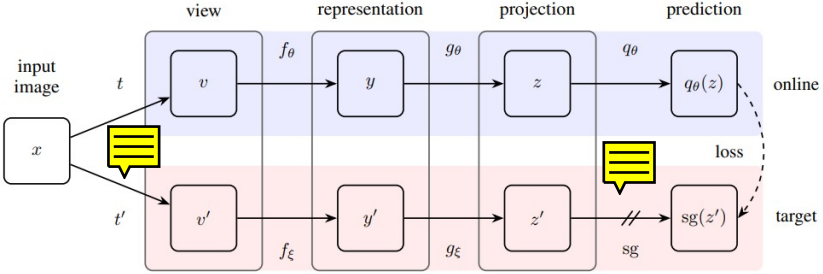
[Kolesnikov et al., 2019]

- A retrospective on self-supervised visual representation learning looks at the design space of self-supervised learners [Kolesnikov et al., 2019].
- Some findings:
  - Architecture matters.
  - Proxy-task performance *does not always correlate downstream*.



BYOL: A new kind of SSL

[Grill et al., 2020]



$$\mathcal{L}_\theta^{\texttt{BYOL}} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}.$$

$$\xi \leftarrow \tau\xi + (1-\tau)\theta.$$

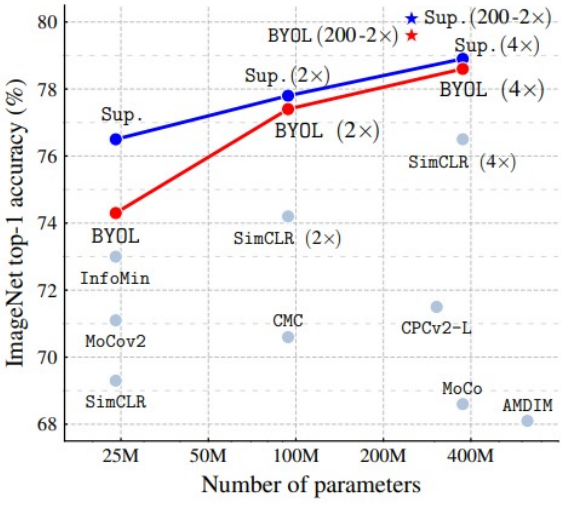
BYOL: A new kind of SSL

[Grill et al., 2020]

- The link between data augmentation and self-supervision:
  - random cropping: a random patch of the image is selected, with an area uniformly sampled between 8% and 100% of that of the original image
  - resizing: to 224×224 + optional left-right flip.
  - color jittering: the brightness, contrast, saturation and hue of the image are shifted by a uniformly random offset applied on all pixels.
  - color dropping: an optional conversion to grayscale.
  - Gaussian blurring: for a 224×224 image, a square Gaussian kernel of size 23×23 is used, with a standard deviation uniformly sampled over $[0.1, 2.0]$
  - solarization: an optional color transformation.

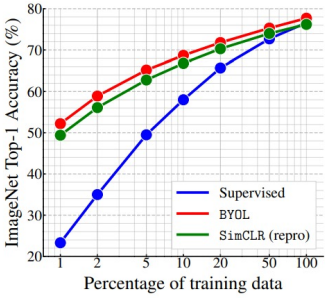
BYOL: A new kind of SSL

[Grill et al., 2020]

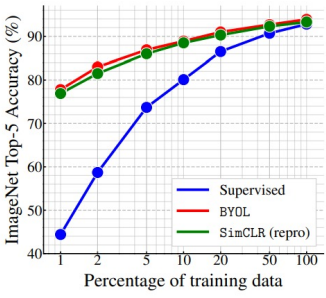


BYOL: A new kind of SSL

[Grill et al., 2020]



(a) Top-1 accuracy



(b) Top-5 accuracy

- My favorite quote from the paper:

  "We use a batch size of 4096 split over 512 Cloud TPU v3 cores. With this setup, training takes approximately 8 hours for a ResNet-50(x1)."

What are image annotations really worth? Yann LeCun's closer slide.

- **"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**
- **Unsupervised/Predictive Learning** (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **(Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)**

Discussion

A true paradigm shift

- The expression *paradigm shift* gets thrown around indiscriminately these days.
- The advent of CNNs and the Deep Learning era of Machine Learning is a *true paradigm shift*.
- Pre-trained backbones can be flexibly repurposed to solve very many difficult problems that needed hand-crafted solutions just a few years ago.
- Pre-training using Self-Supervised Learning can significantly mitigate the cost of annotation.

References

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.

Bibliography ii

X. Liu, J. van de Weijer, and A. D. Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7661–7669, 2018.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.