DIGITAL TWIN AI and Machine Learning: Deep Learning II: Model Adaptation

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

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze

27 November 2020

Outline

Overview

Annotation and large-scale training are expensive

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.

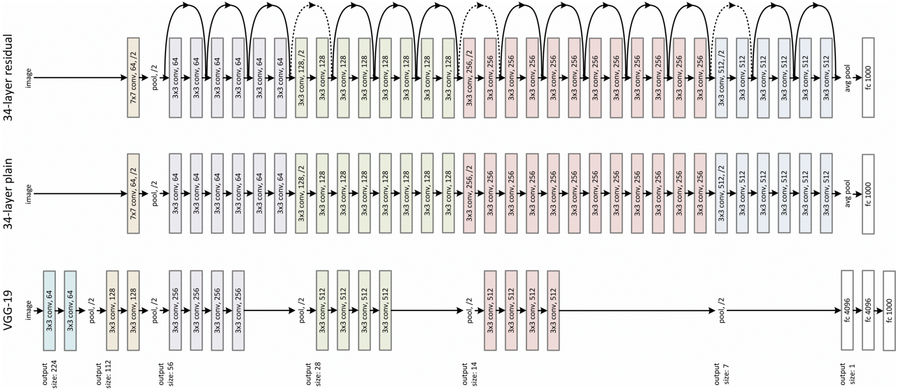
Model adaptation

- 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?
  - Few-shot learning: what if available training data is extremely limited?
  - Meta-learning: can we learn how to learn new visual recognition tasks?
  - Zero-shot learning: what if I have zero examples of some training classes?

Transfer and self-supervised learning

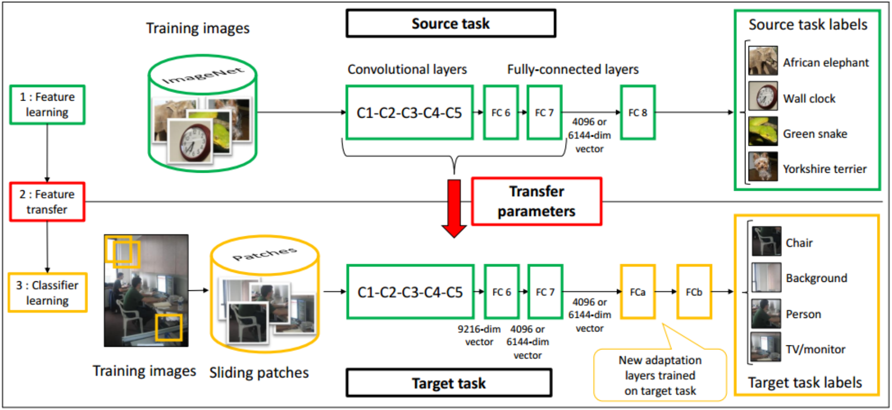
TL: Work and reuse

- 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?



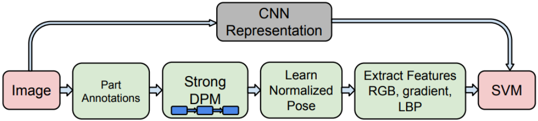
TL: The basic idea

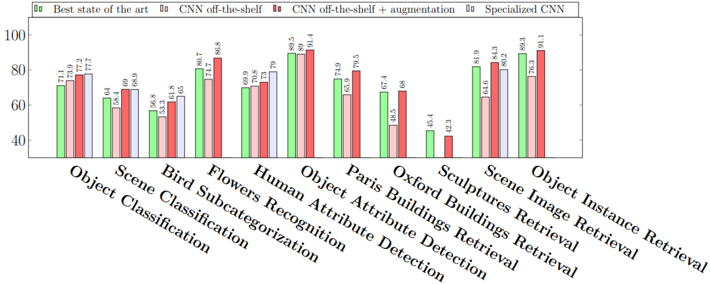
- TL;DR: why on earth start from scratch?



TL: Back to basics

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





CNN features off-the-shelf: an astounding baseline for recognition

TL: Fine-grained recognition

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

CNN features off-the-shelf: an astounding baseline for recognition

TL: Fine-grained recognition

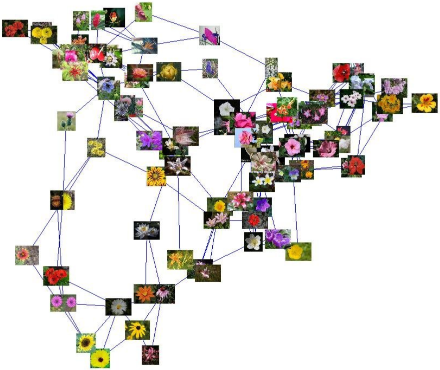
- Not all recognition problems are created equal.
- Fine-grained recognition should require different features.

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**

CNN features off-the-shelf: an astounding baseline for recognition

TL: Fine-grained recognition

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



CNN features off-the-shelf: an astounding baseline for recognition

TL: Fine-grained recognition

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

Method	mean Accuracy
HSV [27]	43.0
SIFT internal [27]	55.1
SIFT boundary [27]	32.0
HOG [27]	49.6
HSV+SIFTi+SIFTb+HOG(MKL) [27]	72.8
BOW(4000) [14]	65.5
SPM(4000) [14]	67.4
FLH(100) [14]	72.7
BiCos seg [7]	79.4
Dense HOG+Coding+Pooling[2] w/o seg	76.7
Seg+Dense HOG+Coding+Pooling[2]	80.7
CNN-SVM w/o seg	74.7
CNNaug-SVM w/o seg	**86.8**

CNN features off-the-shelf: an astounding baseline for recognition

TL: Instance recognition

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

CNN features off-the-shelf: an astounding baseline for recognition

TL: 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**	-

CNN features off-the-shelf: an astounding baseline for recognition

TL: 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
(II) DECAF	–	(C)	t	t	327K	**73.41**	87.4	79.3	84.1	78.4	42.3	73.7
(a) FK	spm	–			327K	**63.66**	83.4	68.8	59.6	74.1	35.7	71.2
(b) FK IN	spm	–			327K	**64.18**	82.1	69.7	59.7	75.2	35.7	71.3
(c) FK	(x,y)	–			42K	**63.51**	83.2	69.4	60.6	73.9	36.3	68.6
(d) FK IN	(x,y)	–			42K	**64.36**	83.1	70.4	62.4	75.2	37.1	69.1
(e) FK IN	(x,y)	(F)	f	-	42K	**64.35**	83.1	70.5	62.3	75.4	37.1	69.1
(f) FK IN	(x,y)	(C)	f	s	42K	**67.17**	85.5	71.6	64.6	77.2	39.0	70.8
(g) FK IN	(x,y)	(C)	s	s	42K	**66.68**	84.9	70.1	64.7	76.3	39.2	69.8
(h) FK IN 512	(x,y)	–			84K	**65.36**	84.1	70.4	65.0	76.7	37.2	71.3
(i) FK IN 512	(x,y)	(C)	f	s	84K	**68.02**	85.9	71.8	67.1	77.1	38.8	72.3
(j) FK IN COL 512	–	–			82K	**52.18**	69.5	52.1	47.5	64.0	24.6	49.8
(k) FK IN 512 COL+	(x,y)	–			166K	**66.37**	82.9	70.1	67.0	77.0	36.1	70.0
(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
(m) CNN F	–	(C)	f	s	4K	**77.38**	88.7	83.9	87.0	84.7	46.9	77.5
(n) CNN S	–	(C)	f	s	4K	**79.74**	90.7	85.7	88.9	86.6	50.5	80.1
(o) CNN M	–	–			4K	**76.97**	89.5	84.3	88.8	83.2	48.4	77.0
(p) CNN M	–	(C)	f	s	4K	**79.89**	91.7	85.4	89.5	86.6	51.6	79.3
(q) CNN M	–	(C)	f	m	4K	**79.50**	90.9	84.6	89.4	85.8	50.3	78.4
(r) CNN M	–	(C)	s	s	4K	**79.44**	91.4	85.2	89.1	86.1	52.1	78.0
(s) CNN M	–	(C)	t	t	41K	**78.77**	90.7	85.0	89.2	85.8	51.0	77.8
(t) CNN M	–	(C)	f	-	4K	**77.78**	90.5	84.3	88.8	84.5	47.9	78.0
(u) CNN M	–	(F)	f	-	4K	**76.99**	90.1	84.2	89.0	83.5	48.1	77.2
(v) CNN M GS	–	–			4K	**73.59**	87.4	80.8	82.4	82.1	44.5	73.5
(w) CNN M GS	–	(C)	f	s	4K	**77.00**	89.4	83.8	85.1	84.4	49.4	77.6
(x) CNN M 2048	–	(C)	f	s	2K	**80.10**	91.3	85.8	89.9	86.7	52.4	79.7
(y) CNN M 1024	–	(C)	f	s	1K	**79.91**	91.4	86.9	89.3	85.8	53.3	79.8
(z) CNN M 128	–	(C)	f	s	128	**78.60**	91.3	83.9	89.2	86.9	52.1	81.0
(α) FK+CNN F	(x,y)	(C)	f	s	88K	**77.95**	89.6	83.1	87.1	84.5	48.0	79.4
(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
(γ) CNN S TUNE-RNK	–	(C)	f	s	4K	**82.42**	95.3	90.4	92.5	89.6	54.4	81.9

(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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	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)	87.2	86.5	59.5	72.4	74.1	81.7	86.0	82.3	90.8	48.9	73.7	66.8	89.6	71.0
(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)	87.8	88.6	59.0	77.2	73.1	85.9	88.3	83.5	91.8	59.9	81.4	68.3	93.0	74.1
(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)	86.8	85.6	59.9	72.0	73.4	81.4	88.6	80.5	92.1	60.6	77.3	66.4	89.3	73.3
(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)	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

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

TL: 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

TL: Not just recognition

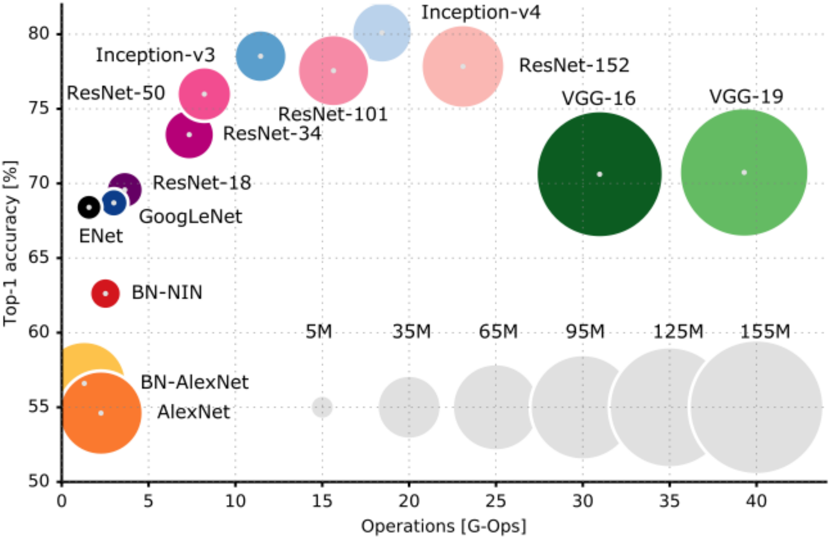
- Transfer learning can be applied to almost any visual recognition or estimation task.
- This includes object detection [Ren et al., 2015], semantic segmentation [Long et al., 2015], crowd counting [Liu et al., 2018], you name it.

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!

Deep residual learning for image recognition

CNNs are really big



CNNs are really big

- One of the first observations one can make about CNNs is that they have a HUGE number of parameters.
- Even modestly-sized, state-of-the-art networks have on the order of *150 million trainable parameters*.
- Fitting such models of course requires massive amounts of labeled training data.

Data collection is expensive

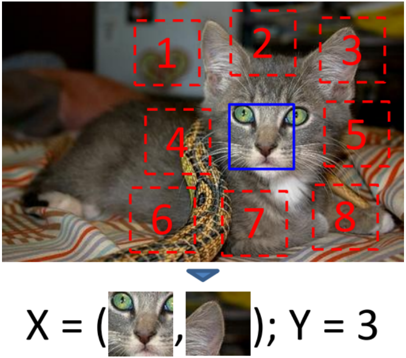
- Such data can be enormously expensive to collect.
- For basic image recognition problems (e.g. cats versus dogs), labeled data is relatively easy to crowdsource.
- For other problems, the annotation task is significantly more tedious and requires careful supervision and annotator corroboration.
- This, of course, translates into higher annotation costs.
- Self-supervision offers the prospect of synthesizing training signal for free and has been applied to representation learning for object recognition:
  - Context prediction: force CNNs to learn how to predict local image (or video) context.
  - Low-level semantics: use the basic building blocks of images to learn useful representations.
  - Niche problems: especially in cases where data is especially scarce, define proxy tasks related to the primary goal.

Self-supervised learning

- Self-supervised learning (SSL) offers the promise of learning generically useful features.
- The main idea is to synthesize training signal using domain (or other) knowledge.
- This synthetic supervisory signal should be obtainable for free or for very little cost.
- Most of the work on self-supervision has concentrated on learning generic representation for recognition.
- This representations can then be fine-tuned for specific tasks on limited, fully-supervised training data.
- Let's look at some representative works in this direction.

SSL: Spatial context prediction

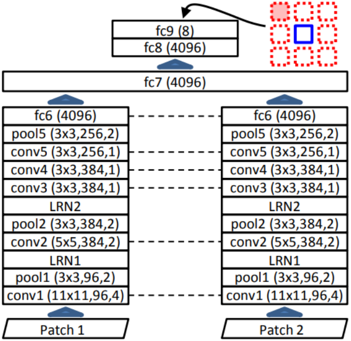
- An appealing approach is to train a network to predict the local context of image patches.
- Feed a network a pair of patches, train to predict which neighbor the second one is wrt the first.
- Local context prediction: [Doersch et al., 2015]



Unsupervised visual representation learning by context prediction

SSL: Spatial context prediction

- The network architecture is Siamese.
- Note that you must always be sure the network can't cheat.
- In this case, the authors discovered that chromatic aberration is a decisive factor.



Unsupervised visual representation learning by context prediction

SSL: 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

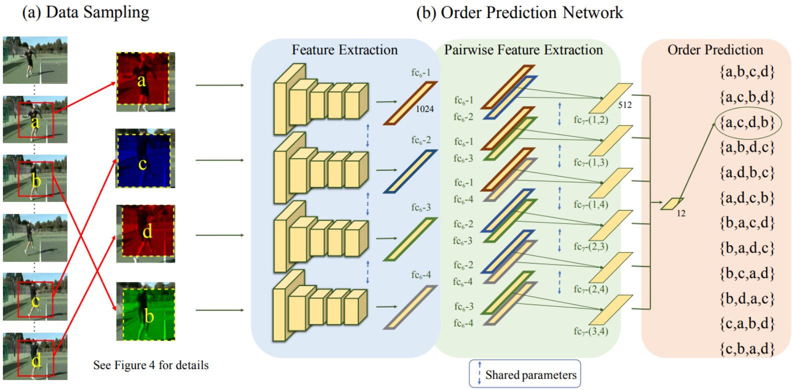
SSL: The key idea behind temporal context

- If unlabeled images can provide self-supervisory signal, what about video?
- Formulate it like a classical proxy task for self-supervised learning.
- The proxy needs no semantic labels – you can sample as many sequences as you like from arbitrary video [Lee et al., 2017].

Unsupervised representation learning by sorting sequences [Lee et al., 2017]

The model

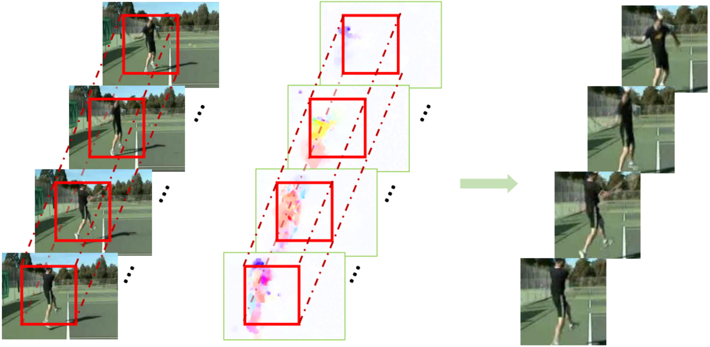
- In this paper, the authors train a network to order input frames.
- Input: n frames in shuffled order.
- Output: probability distribution over the $n!/2$ orders.



Unsupervised representation learning by sorting sequences [Lee et al., 2017]

Tuple sampling: motion awareness

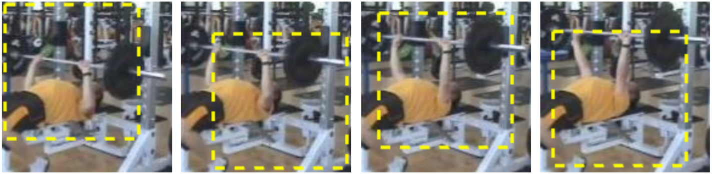
- The authors use the magnitude of optical flow to select frames with large motion regions.
- This flow magnitude is also used to select spatial patches with large motion.



Unsupervised representation learning by sorting sequences [Lee et al., 2017]

Tuple sampling: spatial jittering

- If the same spatial region of frames is sampled, the network can just learn to subtract them.
- This is a coarse estimate of optical flow (spatio-temporal gradient).
- It is easy to sort from this (up to complete reversal), but requires no semantics.
- The solution: spatial jittering.



Unsupervised representation learning by sorting sequences [Lee et al., 2017]

Results: Action Recognition

- Some observations:
  - Self-supervision is superior to random initialization.
  - Self-supervision is inferior to pre-training on ImageNet.
  - Order-prediction works better than other self-supervision approaches.

Initialization	CaffeNet	VGG-M-2048
random	47.8	51.1
ImageNet	67.7	70.8
Misra et al. [24]	50.2	-
Purushwalkam et al. [30]*	-	55.4
Vondrick et al. [39]†	52.1	-
binary	51.6	56.8
3-tuple Concat	52.8	57.0
3-tuple OPN	53.2	58.3
4-tuple Concat	55.2	59.0
4-tuple OPN	**56.3**	**59.8**

Unsupervised representation learning by sorting sequences [Lee et al., 2017]

Results: Image Recognition on PASCAL 2007

- Idea:
  1. use three video datasets for action recognition for pre-training;
  2. fine-tune the backbone on the 20 PASCAL classes (using training images).
- Observations:
  - Self-supervision is still worse than pre-training on ImageNet.
  - OPN works very well and is pretty efficient.

Method	Pretraining time	Source	Supervision	Classification	Detection
Krizhevsky et al. [17]	3 days	ImageNet	labeled classes	78.2	56.8
Doerch et al. [6]	4 weeks	ImageNet	context	55.3	46.6
Pathak et al. [29]	14 hours	ImagetNet+StreetView	context	56.5	44.5
Norrozi et al. [26]	2.5 days	ImageNet	context	**68.6**	**51.8**
Zhang et al. [43]	-	ImageNet	reconstruction	67.1	46.7
Wang and Gupta (color) [41]	1 weeks	100k videos, VOC2012	motion	58.4	44.0
Wang and Gupta (grayscale) [41]	1 weeks	100k videos, VOC2012	motion	62.8	**47.4**
Agrawal et al. [2]	-	KITTI, SF	motion	52.9	41.8
Misra et al. [24]	-	$<$ 10k videos	motion	54.3	39.9
Ours (OPN)	$<$ 3 days	$<$ 30k videos	motion	**63.8**	46.9

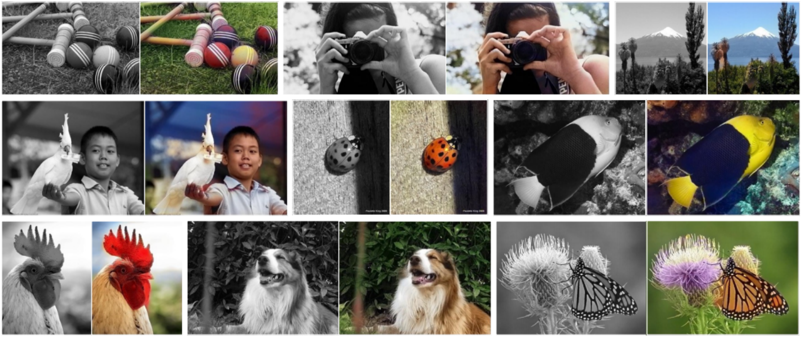
Unsupervised representation learning by sorting sequences [Lee et al., 2017]

SSL: Low-level semantics

- So far we have posed high-level semantic tasks to networks for self-supervised learning (e.g. spatial or temporal context [Noroozi et al., 2017; Kim et al., 2018; Lee et al., 2017]).
- There is another body of self-supervised learning work that instead looks at low-level cues.
- Natural images have statistical properties that influence their perceptions and how we learn internal representations [Simoncelli and Olshausen, 2001].
- Training deep CNNs to reconstruct these statistics can be a rich source of self-supervisory signal.

SSL: Colorization

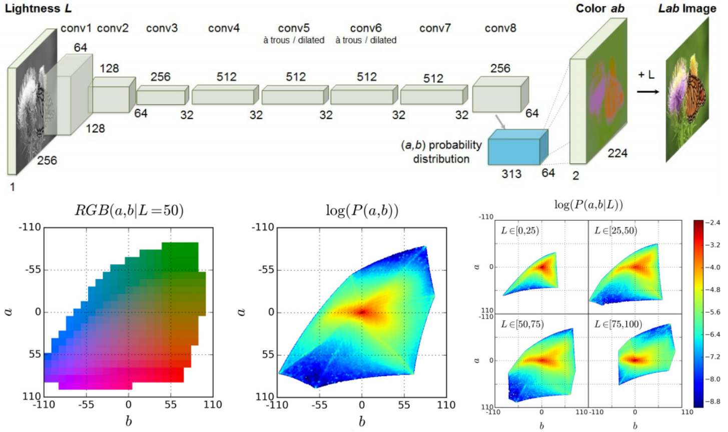
- Color is a strong semantic cue for humans and colorization of grayscale images can be used for self-supervision [Zhang et al., 2016].



Colorful image colorization

SSL: Colorization

- Color is a strong semantic cue for humans and colorization of grayscale images can be used for self-supervision [Zhang et al., 2016].



Colorful image colorization

SSL: Colorization results

- Colorization indeed seems to be a frontrunner as a self-supervised proxy task [Larsson et al., 2017].

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**

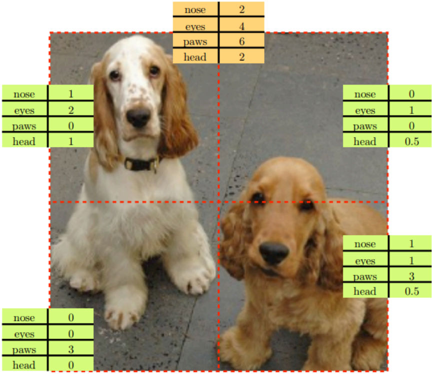
[Zhang et al., 2016]

Initialization		Architecture	Class. %mAP	Seg. %mIU
ImageNet	(+FoV)	VGG-16	86.9	69.5
Random (ours)		AlexNet	46.2	23.5
Random [32]		AlexNet	53.3	19.8
k-means [20, 5]		AlexNet	56.6	32.6
k-means [20]		VGG-16	56.5	-
k-means [20]		GoogLeNet	55.0	-
Pathak *et al.* [32]		AlexNet	56.5	29.7
Wang & Gupta [39]		AlexNet	58.7	-
Donahue *et al.* [5]		AlexNet	60.1	35.2
Doersch *et al.* [4, 5]		AlexNet	65.3	-
Zhang *et al.* (col) [43]		AlexNet	65.6	35.6
Zhang *et al.* (s-b) [44]		AlexNet	67.1	36.0
Noroozi & Favaro [29]		Mod. AlexNet	68.6	-
Larsson *et al.* [21]		VGG-16	-	50.2
Our method		AlexNet	65.9	38.4
	(+FoV)	VGG-16	**77.2**	56.0
	(+FoV)	ResNet-152	**77.3**	**60.0**

[Larsson et al., 2017]

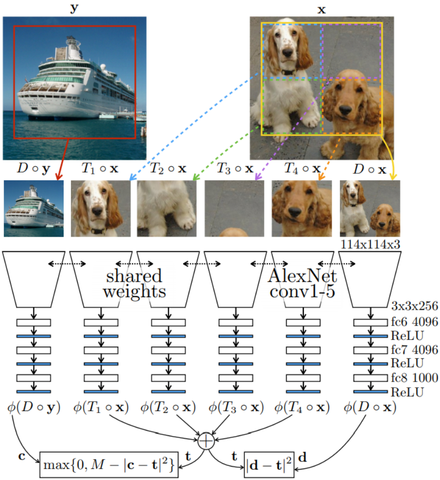
SSL: Counting low-level visual primitives

- Counting unsupervised image features is a powerful way to learn image representations.
- This takes advantage of that fact that anything counted in sub-images must sum to the total count.
- You must be careful that what is being counted, though, is meaningful [Noroozi et al., 2017].



Representation learning by learning to count

SSL: Counting low-level visual primitives

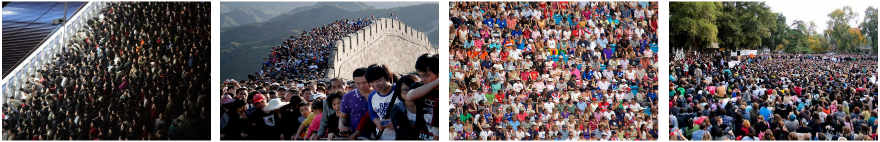


Method	Part A MAE	Part A MSE	Part B MAE	Part B MSE
Cross-scene [34]	181.8	277.7	32.0	49.8
MCNN [10]	110.2	173.2	26.4	41.3
Switching-CNN [11]	90.4	135.0	21.6	33.4
CP-CNN [33]	73.6	**106.4**	20.1	30.1
ACSCP [38]	75.7	**102.7**	17.2	27.4
CSRNet [36]	**68.2**	115.0	**10.6**	**16.0**
ic-CNN [41]	**68.5**	116.2	**10.7**	**16.0**
Ours: Multi-task (Query-by-example)	**72.0**	**106.6**	14.4	23.8
Ours: Multi-task (Keyword)	73.6	112.0	**13.7**	**21.4**

Representation learning by learning to count

SSL: Niche problems

- A problem for which annotation is tedious and expensive is crowd counting.
- Given an image, the task is to estimate the number of persons present in the scene.
- Given the difficulty of annotation, most crowd counting datasets have at most around 1000 labeled images.



An observation

- For many niche problems, we can define much more specific and useful proxy tasks for self-supervised learning.
- These domain-specific proxy tasks allow us to address the lack-of-data problem in application areas where it is much more acute.

SSL: Niche problems

- We can create a proxy ranking task for which we can automatically generate training signal.
- After pre-training a CNN using this proxy task, we transfer the weights to a CNN that estimates absolute crowd counts on images [Liu et al., 2019].
- Note: we are no longer learning general image representations – the proxy task should have some direct link to the primary task.

Exploiting Unlabeled Data in CNNs by Self-supervised Learning to Rank

SSL: Niche problems

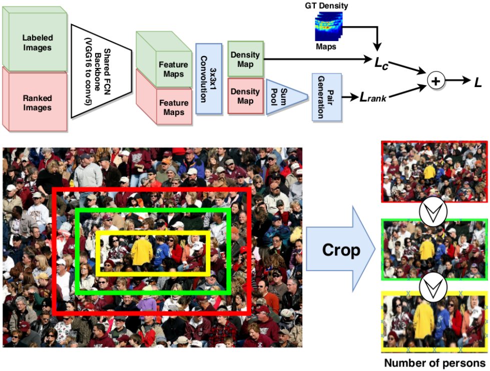
- We can use the observation that sub-images must contain the same number or fewer persons that the super-image.
- Key to this approach is a multi-task training objective.

$$C(I_1) \geq C(I_2) \geq C(I_3)$$

Exploiting Unlabeled Data in CNNs by Self-supervised Learning to Rank

SSL: Niche problems

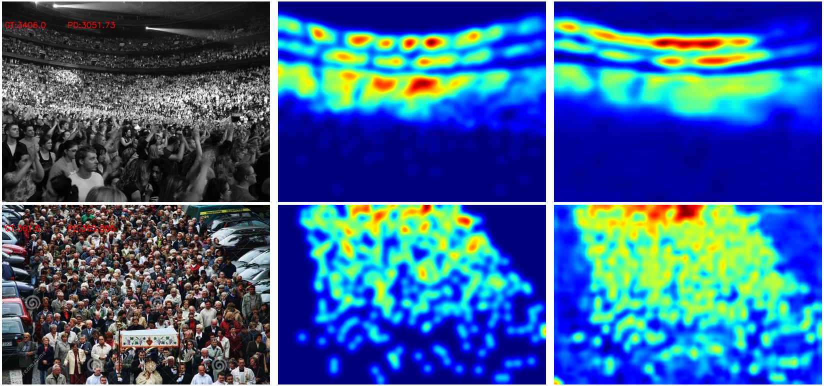
- Using the multi-task loss we can simultaneously train the self-supervised and fully-supervised tasks.
- This helps guarantee that the network is counting useful features, instead of just random correlations (loosely speaking).



Exploiting Unlabeled Data in CNNs by Self-supervised Learning to Rank

SSL: Niche problems

- The result is a high-quality density map that can be integrated to estimate counts:



Exploiting Unlabeled Data in CNNs by Self-supervised Learning to Rank

SSL: Niche problems

- An ablation study demonstrates effectiveness:

Method	Split 1	Split 2	Split 3	Split 4	Split 5	Ave MAE
Basic CNN	701.41	394.52	497.57	263.56	415.23	454.45
+ Pre-trained model	570.01	350.63	334.89	184.79	202.41	328.54
+ multi-scale	532.85	307.43	266.75	216.96	216.35	308.06
Ranking+FT	552.68	375.38	241.28	211.66	247.70	325.73
Multi-task (Random)	462.71	345.31	218.71	226.44	210.19	292.67
Multi-task (Hard)	460.35	343.91	208.23	221.75	205.57	287.96
Multi-task (Ours)	443.68	340.31	196.76	218.48	199.54	**279.60**

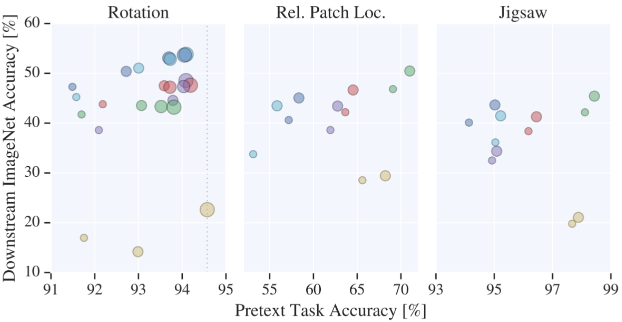
- And results are comparable to the state-of-the-art:

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Cross-scene [34]	181.8	277.7	32.0	49.8
MCNN [10]	110.2	173.2	26.4	41.3
Switching-CNN [11]	90.4	135.0	21.6	33.4
CP-CNN [33]	73.6	**106.4**	20.1	30.1
ACSCP [38]	75.7	**102.7**	17.2	27.4
CSRNet [36]	**68.2**	115.0	**10.6**	**16.0**
ic-CNN [41]	**68.5**	116.2	**10.7**	**16.0**
Ours: Multi-task (Query-by-example)	**72.0**	**106.6**	14.4	23.8
Ours: Multi-task (Keyword)	73.6	112.0	**13.7**	**21.4**

Exploiting Unlabeled Data in CNNs by Self-supervised Learning to Rank

SSL: Retrospective

- 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.



Revisiting Self-Supervised Visual Representation Learning

Few-shot learning

Few-shot learning: Overview

- Sometimes the poverty of data is more acute; or rather it is built right into the problem description.
- Few-shot learning (FSL) refers to learning problems where the number of examples per class is extremely limited.
- One usually refers to M-way, N-shot classification, where M is the number of classes and N is the number of training samples per class.
- Typical configurations: $M = 5$, $N \in \{1, 5\}$.

FSL: Setup

- Few-shot learning often uses meta-learning, or learning to learn.

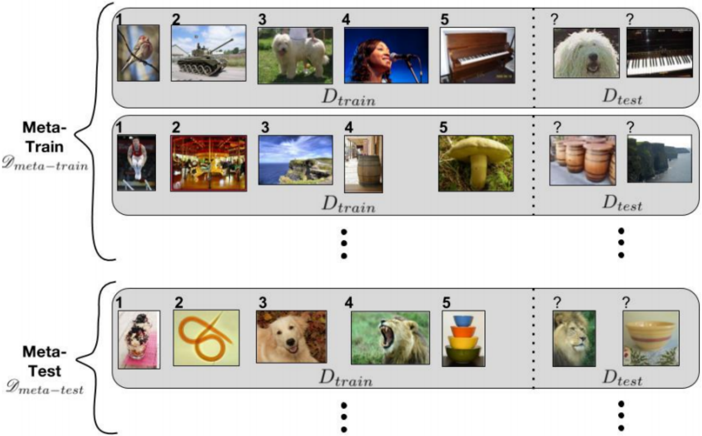


Figure from: [Ravi and Larochelle, 2017]

Zero-shot learning

Zero-shot learning: Overview

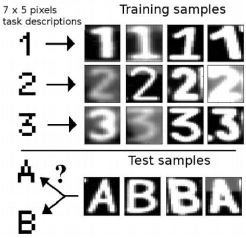
- What if you want to learn visual recognizers but have no training data for classes of interest?
- Are we just plain out of luck?
- An early approach to zero-data task learning uses synthetic task descriptions as a sort of psuedo-label [Larochelle et al., 2008].
- Learning is performed on a disjoint set of semantic prototypes and paired image instances.

$z \rightarrow$	1	2	3	4	5
$d(z) \rightarrow$	1	2	3	A	B
x_t	y_t^1	y_t^2	y_t^3	y_t^4	y_t^5
1	1	0	0	-	-
2	0	1	0	-	-
3	0	0	1	-	-
A	-	-	-	1	0
B	-	-	-	0	1

$\underbrace{\qquad\qquad}_{\text{training data}}$ $\underbrace{\qquad}_{\text{test data}}$

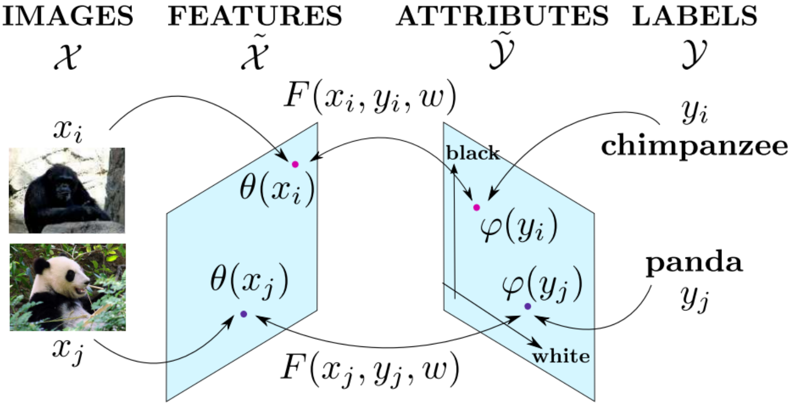
$z \rightarrow$	1	2	3	4	5
$d(z) \rightarrow$	1	2	3	A	B
x_t	y_t^1	y_t^2	y_t^3	y_t^4	y_t^5
1	3	-	-1	0.5	-
2	2.5	1	-	-	-
3	-2	3	-	0.25	2
A	-1	1	-	-2	-3
B	1	-	1.5	3.5	4

$\underbrace{\qquad\qquad}_{\text{training data}}$ $\underbrace{\qquad}_{\text{test data}}$



Zero-shot learning: Overview

- Modern Zero-Shot Learning (ZSL) inherits much from zero-data learning.
- Most ZSL is based on embeddings: semantic feature embedding of images, and semantic attribute embeddings of classes [Lampert et al., 2013; Akata et al., 2013].



Zero-shot learning: Overview

- What do we mean by semantic feature embedding and semantic attribute embedding for zero-shot learning?
- For images, we have seen how CNNs like ResNet and VGG are great feature extractor. So we use them.
- What about class semantics? One option is to use attributes [Xian et al., 2018]:

```
polar bear
black:      no
white:      yes
brown:      no
stripes:    no
water:      yes
eats fish:  yes
```

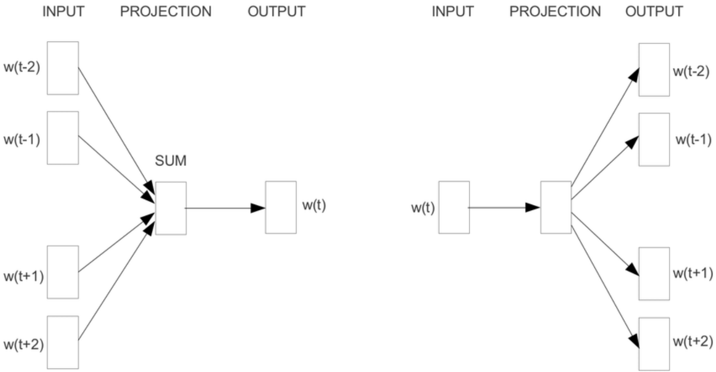
```
zebra
black:      yes
white:      yes
brown:      no
stripes:    yes
water:      no
eats fish:  no
```

Images: changehali (https://www.flickr.com/photos/53371280@N04/4975496109/), lunartique (https://www.flickr.com/photos/97456166@N06/23639651093/), markbyzewski (https://www.flickr.com/photos/markbyzewski/7765659110/), hellobobbigbunny (https://www.flickr.com/photos/bobsteiner/3348643202/)

- Each class is represented by a single binary attribute vector (which could be an average of class instances).
- To learn a new class, we only require its attributes.

Zero-shot learning: Overview

- What if we don't have attribute representations for the categories of interest (attributes, after all, aren't free).
- If we have textual descriptions, we can use word or document embeddings.
- Some choices are word2vec [Mikolov et al., 2013], doc2vec [Le and Mikolov, 2014], or GloVe [Pennington et al., 2014]:



Zero-shot learning: Overview

- Zero-shot Learning (ZSL) in computer vision has seen some explosive growth in interest in recent years.
- Let's take a look at some recent approaches to ZSL and see what makes them tick.
- Much of what we will see here is based on an excellent and comprehensive review published recently [Xian et al., 2018].

Zero-shot learning

- In ZSL we are given a training set of image and class labels:

$$\begin{aligned} \mathcal{S} &= \{ (x_i, y_) \mid i = 1, \ldots N \} \\ y_i &\in \mathcal{Y}^{tr} \ (\mathcal{Y}^{tr} \text{ is the set of training classes}) \end{aligned}$$

- Our goal is to minimize a regularized empirical risk:

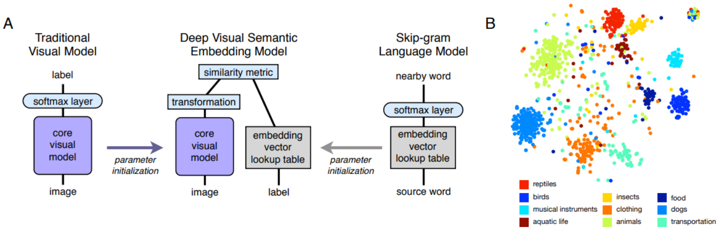
$$\begin{aligned} \mathcal{L} &= \frac{1}{N} \sum_{1}^{N} L(y_n, f(x_n; W)) + \Omega(W) \\ f(x; W) &= \arg\max_{y \in \mathcal{Y}} F(x, y; W) \end{aligned}$$

- At test time, for zero-shot learning we must assign images to an unseen class label $\mathcal{Y}^{ts} \subset \mathcal{Y}$.
- For generalized zero-shot learning test images can be assigned to either seen or unseen classes: $\mathcal{Y}^{tr+ts} \subset \mathcal{Y}$.

ZSL: Linear affinity learning

- Many state-of-the-art approaches use relatively simple linear compatibility functions:
$$F(x, y; W) = \theta(x)^T W \phi(y)$$
- In this θ is a semantic image embedding (e.g. a CNN feature extractor).
- And ϕ is a semantic class embedding (e.g. attributes or text embedding).
- The DEVISE algorithm an unregularized ranking loss [Frome et al., 2013]:

$$\mathcal{L} = \sum_{y \in \mathcal{Y}^{tr}} [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+$$



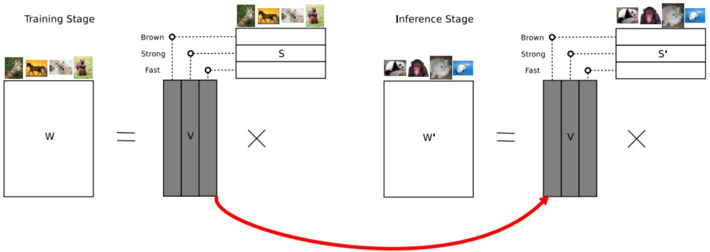
Devise: A deep visual-semantic embedding model

ZSL: Embarrassingly Simple ZSL

- The ESZSL approach uses a linear affinity matrix with square loss and regularizers [Romera-Paredes and Torr, 2015]:

$$\Omega(W) = \gamma||W\phi(y)||^2 + \lambda||\theta(x)^T W||^2 + \beta||W||^T$$

- This bounds the Euclidean norm of projected attributes and image features.
- An advantage of this approach: the objective function is convex and has a closed form solution.



An embarrassingly simple approach to zero-shot learning

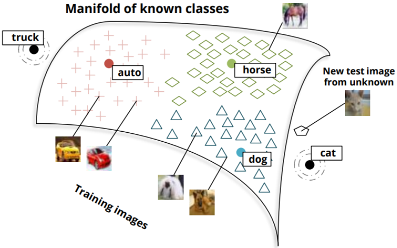
ZSL: Non-linear affinity learning

- The Latent Embedding (LATEM) approach is a natural extension of the linear approach [Xian et al., 2016]:

$$F(x, y; W) = \max_{1 \leq i \leq K} \theta(x)^T W_i \phi(y)$$

- The Cross-Modal Transfer (CMT) approach maps images into the semantic space of attributes [Socher et al., 2013]:

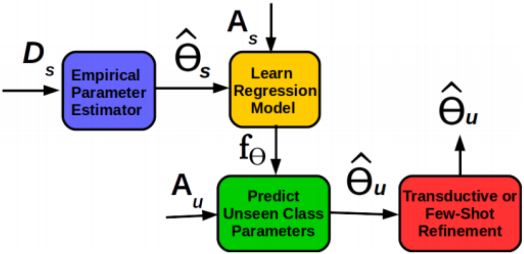
$$\sum_{y \in \mathcal{Y}^{tr}} \sum_{x \in \mathcal{X}_y} ||\phi(y) - W_1 \tanh(W_2 \phi(x))||^2$$



ZSL: GFZSL

- Generative models are sort of the new frontier for ZSL.
- The Generative Framework for Zero-Shot Learning (GFZSL) models class-conditional distributions as multivariate Gaussians [Verma and Rai, 2017].
- Unseen class parameters are computed using two learned regressions:

$$\mu_y = f_\mu(\phi(y)) \text{ and } \sigma_y = f_\sigma(\theta(y))$$



A simple exponential family framework for zero-shot learning

ZSL: Datasets

- It is interesting to look at the ZSL datasets in use and see how the problems are set up.
- Note in particular the splits at evaluation time.

			Number of Classes				Number of Images								
								At Training Time				At Evaluation Time			
								SS		PS		SS		PS	
Dataset	**Size**	**Granularity**	**Att**	$\mathcal{Y}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	**Total**	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$	$\mathcal{Y}^{tr}$	$\mathcal{Y}^{ts}$
SUN [16]	medium	fine	102	717	580 + 65	72	14340	12900	0	10320	0	0	1440	2580	1440
CUB [17]	medium	fine	312	200	100 + 50	50	11788	8855	0	7057	0	0	2933	1764	2967
AWA1 [1]	medium	coarse	85	50	27 + 13	10	30475	24295	0	19832	0	0	6180	4958	5685
AWA2	medium	coarse	85	50	27 + 13	10	37322	30337	0	23527	0	0	6985	5882	7913
aPY [18]	small	coarse	64	32	15 + 5	12	15339	12695	0	5932	0	0	2644	1483	7924

Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly

ZSL: Results

- Generative modeling seems to be taking the lead.
- However, it is interesting that relatively simple and convex approaches still perform respectably.

Method	SUN SS	SUN PS	CUB SS	CUB PS	AWA1 SS	AWA1 PS	AWA2 SS	AWA2 PS	aPY SS	aPY PS
DAP [1]	38.9	39.9	37.5	40.0	57.1	44.1	58.7	46.1	35.2	33.8
IAP [1]	17.4	19.4	27.1	24.0	48.1	35.9	46.9	35.9	22.4	36.6
CONSE [15]	44.2	38.8	36.7	34.3	63.6	45.6	67.9	44.5	25.9	26.9
CMT [12]	41.9	39.9	37.3	34.6	58.9	39.5	66.3	37.9	26.9	28.0
SSE [13]	54.5	51.5	43.7	43.9	68.8	60.1	67.5	61.0	31.1	34.0
LATEM [11]	56.9	55.3	49.4	49.3	74.8	55.1	68.7	55.8	34.5	35.2
ALE [30]	59.1	58.1	53.2	54.9	78.6	59.9	80.3	62.5	30.9	39.7
DEVISE [7]	57.5	56.5	53.2	52.0	72.9	54.2	68.6	59.7	35.4	**39.8**
SJE [9]	57.1	53.7	**55.3**	53.9	76.7	65.6	69.5	61.9	32.0	32.9
ESZSL [10]	57.3	54.5	55.1	53.9	74.7	58.2	75.6	58.6	34.4	38.3
SYNC [14]	59.1	56.3	54.1	**55.6**	72.2	54.0	71.2	46.6	39.7	23.9
SAE [33]	42.4	40.3	33.4	33.3	**80.6**	53.0	**80.7**	54.1	8.3	8.3
GFZSL [41]	**62.9**	**60.6**	53.0	49.3	80.5	**68.3**	79.3	**63.8**	**51.3**	38.4

Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly

GZSL: Results

- The situation is completely different for generalized ZSL, however:

	SUN			CUB			AWA1			AWA2			aPY		
Method	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**
DAP [1]	4.2	25.1	7.2	1.7	67.9	3.3	0.0	**88.7**	0.0	0.0	84.7	0.0	4.8	78.3	9.0
IAP [1]	1.0	37.8	1.8	0.2	72.8	0.4	2.1	78.2	4.1	0.9	87.6	1.8	5.7	65.6	10.4
CONSE [15]	6.8	**39.9**	11.6	1.6	**72.2**	3.1	0.4	88.6	0.8	0.5	**90.6**	1.0	0.0	**91.2**	0.0
CMT [12]	8.1	21.8	11.8	7.2	49.8	12.6	0.9	87.6	1.8	0.5	90.0	1.0	1.4	85.2	2.8
CMT* [12]	8.7	28.0	13.3	4.7	60.1	8.7	8.4	86.9	15.3	8.7	89.0	15.9	**10.9**	74.2	**19.0**
SSE [13]	2.1	36.4	4.0	8.5	46.9	14.4	7.0	80.5	12.9	8.1	82.5	14.8	0.2	78.9	0.4
LATEM [11]	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	0.1	73.0	0.2
ALE [30]	**21.8**	33.1	**26.3**	23.7	62.8	**34.4**	**16.8**	76.1	**27.5**	14.0	81.8	23.9	4.6	73.7	8.7
DEVISE [7]	16.9	27.4	20.9	**23.8**	53.0	32.8	13.4	68.7	22.4	**17.1**	74.7	**27.8**	4.9	76.9	9.2
SJE [9]	14.7	30.5	19.8	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	3.7	55.7	6.9
ESZSL [10]	11.0	27.9	15.8	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	2.4	70.1	4.6
SYNC [14]	7.9	43.3	13.4	11.5	70.9	19.8	8.9	87.3	16.2	10.0	90.5	18.0	7.4	66.3	13.3
SAE [33]	8.8	18.0	11.8	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	0.4	80.9	0.9
GFZSL [41]	0.0	39.6	0.0	0.0	45.7	0.0	1.8	80.3	3.5	2.5	80.1	4.8	0.0	83.3	0.0

Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly

Discussion

Discussion: Transfer learning

- In pre-CNN visual recognition, features were everything.
- Features are still everything, however now we can re-use in new ways.
- CNNs are proven semantic feature extractors, and these features can be used in a ton of useful ways:
  - Classical supervised learning: CNN features can be shoved into your favorite classifiers as-is (SVM, K-SVM, logistic regression, etc.)
  - Fine-tuning: the weights of early layers in CNNs can be transferred to a new model with new layers (with randomly initialized weights).
- Transfer learning in this way greatly reduces the amount of labeled data needed.
- Bottom line: you should probably be using a pre-trained network as a starting point.

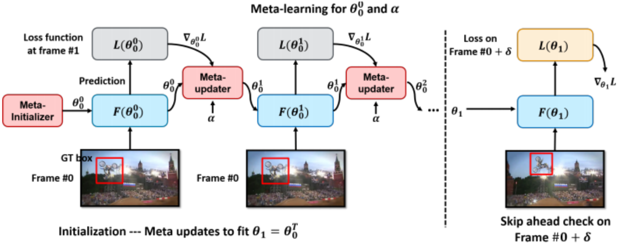
Discussion: Self-supervised learning

- Self-supervised learning is a type of unsupervised representation learning (more on this tomorrow).
- The amount of unsupervised image and video data massively outnumber the amount of supervised data.
- Finding new ways to exploit this is one of the holy grails of computer vision (and machine learning in general).
- If proxy tasks, for which supervisory signal can be derived for free, can be defined, self-supervised learning can be an effective way to learn useful representations.

  "If intelligence is a cake, the bulk of the cake is unsupervised learning, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning (RL)."
  – Yann LeCun @ NIPS 2016

Discussion: Few-shot learning

- Probably the best way to mitigate the data-hungry nature of Convolutional Neural Networks is to understand how to effectively train them with limited data.
- Meta-learning offers some solutions: observe learners in action, and learn how to learn new tasks (on limited data).
- Though important steps have been made, meta-learning is still lacking a killer app in visual recognition.
- However, one candidate might be object tracking [Park and Berg, 2018].



Meta-tracker: Fast and robust online adaptation for visual object trackers

Discussion: Zero-shot learning

- Zero-shot learning is an exciting and (relatively) new application of visual recognition.
- In a way, it harks back to the glory days of Content-based Image Retrieval (CBIR) [Smeulders et al., 2000].
- Think about it: if you can describe a visual category using our rich, semantic and natural language, ZSL can recognize instances of it.
- Generative models seem to be the way forward, but Generalized Zero-shot Learning has a ways to go [Wang et al., 2018; Antoniou et al., 2017].

Bibliography I

Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.

A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

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.

A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

D. Kim, D. Cho, D. Yoo, and I. S. Kweon. Learning image representations by completing damaged jigsaw puzzles. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 793–802. IEEE, 2018.

Bibliography II

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

C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013.

H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008.

G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017.

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.

Bibliography III

X. Liu, J. Van De Weijer, and A. D. Bagdanov. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

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.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

E. Park and A. C. Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 569–585, 2018.

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Bibliography IV

S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

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.

B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 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.

E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001.

A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):1349–1380, 2000.

R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

Bibliography V

V. K. Verma and P. Rai. A simple exponential family framework for zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 792–808. Springer, 2017.

Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018.

Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.