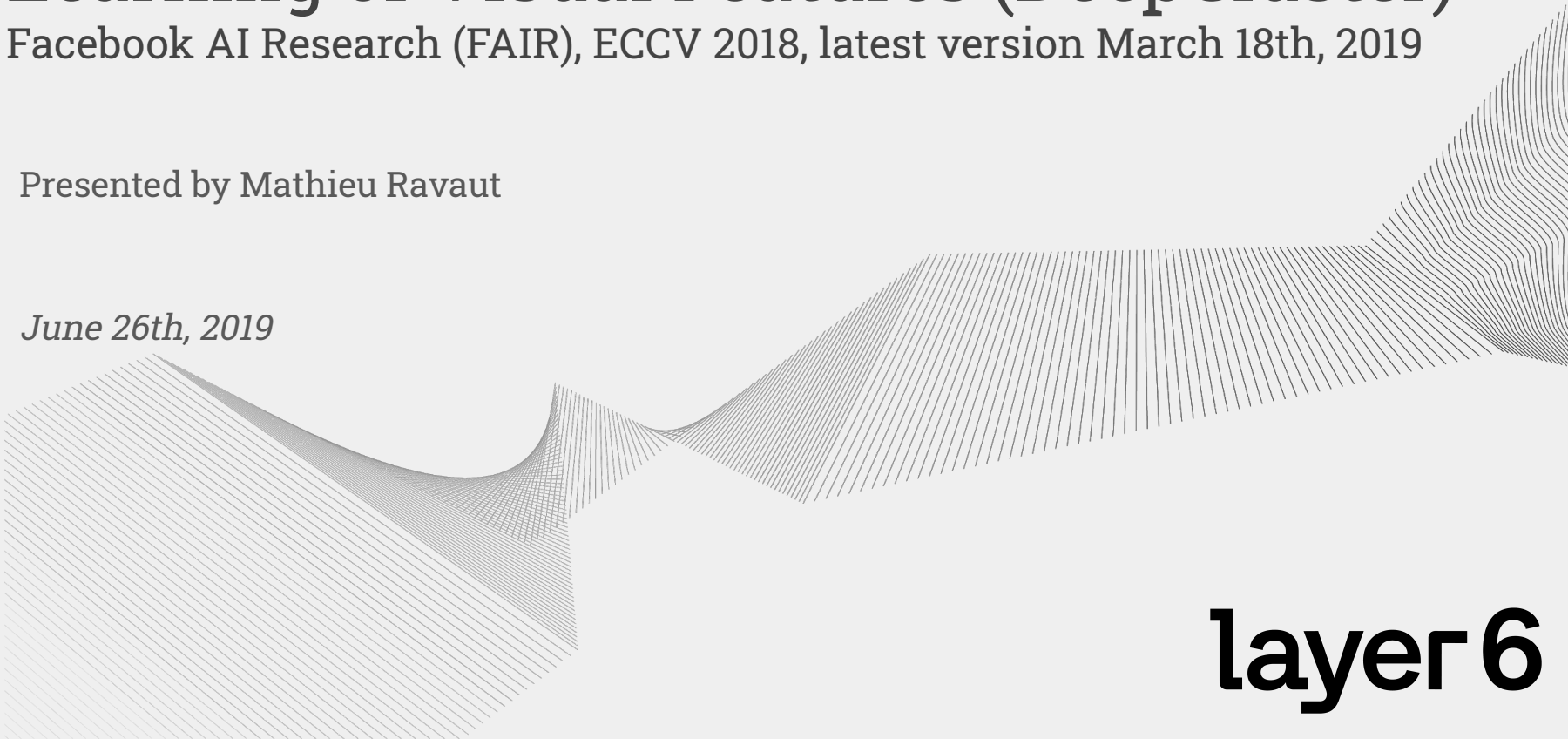
Deep Clustering for Unsupervised Learning of Visual Features (DeepCluster)

Facebook AI Research (FAIR), ECCV 2018, latest version March 18th, 2019

Presented by Mathieu Ravaut

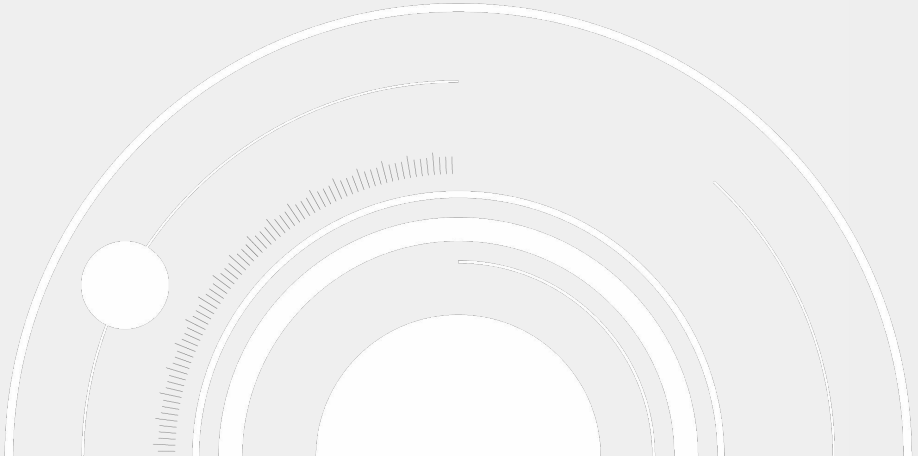
June 26th, 2019



layer 6

Agenda

- Context
- DeepCluster
- Tricks
- Results
- Analysis & discussion
- Other deep clustering approaches



layer 6

Context

layer 6

Context

- **Pre-trained CNNs** (especially on ImageNet) have become a building block in most CV applications: classification, object detection, retrieval, etc.

- They exploit the 1 million image/label pairs of ImageNet.

- How can we exploit larger, Internet-scale datasets for pre-training? (billions)
  At this scale, it would be too costly to build labels.
  ⇨ We need **unsupervised** learning methods

- **Clustering** requires no label nor specific domain knowledge.
  ⇨ How to build an end-to-end deep neural net with clustering?

Context

- We still need to backprop a cross-entropy loss.

- The trick is to use **pretext tasks** to replace human-built labels (*car*, *horse*, *boat*, etc) by ***pseudo-labels***. For example:
  - Predict relative position of patches
  - Shuffle patches and learn to re-arrange them
  - Predict the camera transformation between consecutive frames (for videos)
  - Segment based on motion (for videos)
  - ...

- All these approaches require domain knowledge.
  In this paper: **use cluster assignments as pseudo-labels**

DeepCluster

layer 6

DeepCluster

- CNNs are amazingly well-tailored for images:
  Random, frozen AlexNet (weights sampled from a Gaussian distribution) + MLP gives a test accuracy of 12% (chance is at 0.1%).
  ⇨ **Idea of DeepCluster**: bootstrap this inherent signal
- DeepCluster:
  - Pass mini-batch through encoder θ to get **embeddings**
  - **Cluster** these embeddings with **k-means** to get pseudo-labels yn

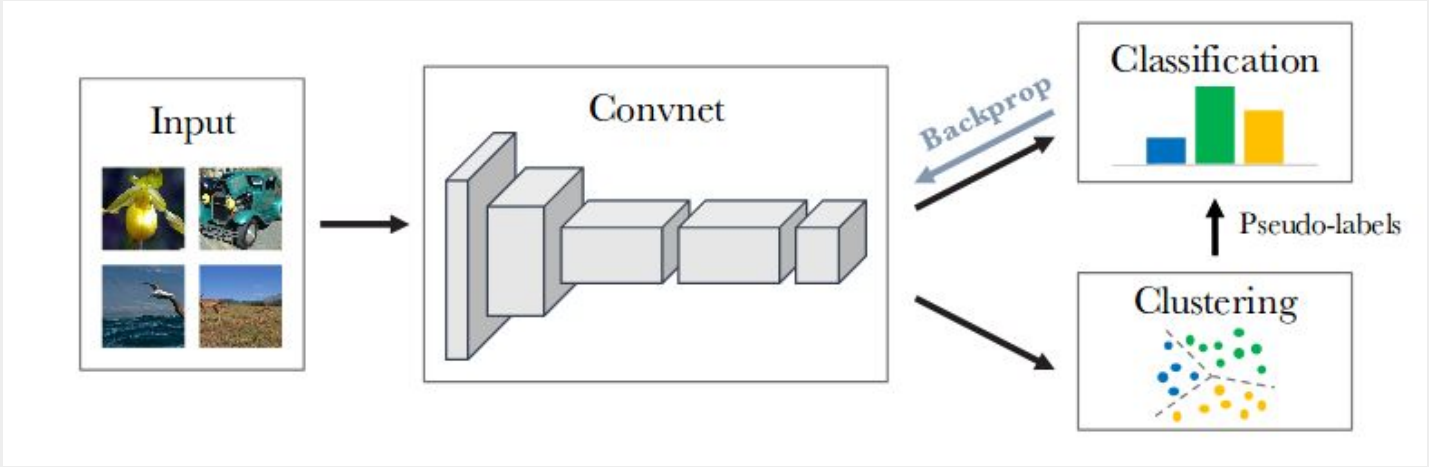
$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \|f_\theta(x_n) - C y_n\|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1.$$

  - Train an MLP classifier to map embeddings to pseudo-labels yn.

$$\min_{\theta, W} \frac{1}{N} \sum_{n=1}^{N} \ell\left(g_W\left(f_\theta(x_n)\right), y_n\right)$$

layer 6

DeepCluster



- The classification loss on pseudo labels is back-propagated through both the encoder CNN and the MLP classifier.
- Training is done in one stage.
- Simple and elegant !

layer 6

Tricks

layer 6

Tricks: implementation details

- **Sobel filtering** on images to remove colors.
- How often to compute k-means?
  - They do it **every epoch** on ImageNet (on the full set -> it is slow)
  - One could do it every *n epochs* too.
  - Technically, one could do it every *m batches*.
- How to **choose k**?
  - k does not have to be equal to the number of classes (let that sink in!)
  - Authors vary k on a logarithmic scale.
  - k = 10,000 works well on ImageNet
- 1st k-means should be computed before training the neural net.
- Before each k-means, features are PCA-reduced to size 256, whitened and L2-normalized.

Tricks: empty clusters

- A possible collapse mode of clustering is to have (k-1) **empty clusters**, and one cluster with all the data (or something similar).

- The authors propose to use **re-assignment**:
  When a cluster becomes empty, randomly select a non-empty cluster and use its centroid with a small random perturbation as the new centroid for the empty cluster.
  Basically, remove the empty cluster and cut another cluster in half.

Tricks: trivial parameterization

- If clusters are **highly imbalanced**, the parameters θ will learn to exclusively discriminate between them.

- The authors propose to **weight** the contribution of an input to the loss function by the inverse of the size of its assigned cluster
(thus, samples from small clusters will have higher contribution)

Results

layer 6

Results: image classification

Authors use DeepCluster as an **unsupervised pre-training method**, and compare on downstream tasks (classification, detection, retrieval) with other pre-training methods.

Two different (and quite specific) setup):

- In the paper:
  Pre-train model, then take frozen weights of each convolution layer and train a single layer MLP with sigmoid for any downstream classification task.
  ⇨ Useful to determine which layer of the pre-trained model to use
  ⇨ With that method, we can pre-train with any value of k.

- In the supplementary material:
  Pre-train model, then freeze convolutional layers and train fully connected layers for any downstream classification task.
  ⇨ Useful to have an idea of how good the overall pre-training was.

Results: image classification (paper)

Supervised baseline

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.* [46]	14.1	20.7	21.0	19.8	15.5	18.2	23.2	23.4	21.9	18.4
Doersch *et al.* [13]	16.2	23.3	30.2	31.7	29.6	19.7	26.7	31.9	32.7	30.9
Zhang *et al.* [71]	12.5	24.5	30.4	31.5	30.3	16.0	25.7	29.6	30.3	29.7
Donahue *et al.* [15]	17.7	24.5	31.0	29.9	28.0	21.4	26.2	27.1	26.1	24.0
Noroozi and Favaro [42]	**18.2**	28.8	34.0	33.9	27.1	23.0	32.1	35.5	34.8	31.3
Noroozi *et al.* [43]	18.0	30.6	34.3	32.5	25.7	**23.3**	**33.9**	36.3	34.7	29.6
Zhang *et al.* [72]	17.7	29.3	35.4	35.2	32.8	21.3	30.7	34.0	34.1	32.5
DeepCluster	13.4	**32.3**	**41.0**	**39.6**	**38.2**	19.6	33.2	**39.2**	**39.8**	**34.7**

New SOTA on unsupervised pre-training classification on ImageNet + Places.

layer 6

Results: image classification (supplementary)

Method	Pre-trained dataset	Acc@1
Supervised	ImageNet	59.7
Supervised Sobel	ImageNet	57.8
Random	-	12.0
Wang *et al.* [29]	YouTube100K [74]	29.8
Doersch *et al.* [25]	ImageNet	30.4
Donahue *et al.* [20]	ImageNet	32.2
Noroozi and Favaro [26]	ImageNet	34.6
Zhang *et al.* [28]	ImageNet	35.2
Bojanowski and Joulin [19]	ImageNet	36.0
DeepCluster	ImageNet	44.0
DeepCluster	YFCC100M	39.6

Supervised baselines

New SOTA on unsupervised pre-training classification on ImageNet.

layer 6

Results: Pascal VOC (detection, classif., segment.)

Method	Classification		Detection		Segmentation	
	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.* [46]	34.6	56.5	–	44.5	–	29.7
Donahue *et al.* [15]*	52.3	60.1	–	46.9	–	35.2
Pathak *et al.* [45]	–	61.0	–	52.2	–	–
Owens *et al.* [44]*	52.3	61.3	–	–	–	–
Wang and Gupta [63]*	55.6	63.1	32.8†	47.2	26.0†	35.4†
Doersch *et al.* [13]*	55.1	65.3	–	51.1	–	–
Bojanowski and Joulin [5]*	56.7	65.3	33.7†	49.4	26.7†	37.1†
Zhang *et al.* [71]*	61.5	65.9	43.4†	46.9	35.8†	35.6
Zhang *et al.* [72]*	63.0	67.1	–	46.7	–	36.0
Noroozi and Favaro [42]	–	67.6	–	53.2	–	37.6
Noroozi *et al.* [43]	–	67.7	–	51.4	–	36.6
DeepCluster	**72.0**	**73.7**	**51.4**	**55.4**	**43.2**	**45.1**

Supervised Baseline (pre-training with the ImageNet labels)

New SOTA on all 3 tasks. Getting really close to supervised accuracy.

layer 6

Results: instance-level image retrieval

Method	Oxford5K	Paris6K
ImageNet labels	72.4	81.5
Random	6.9	22.0
Doersch *et al.* [13]	35.4	53.1
Wang *et al.* [64]	42.3	58.0
DeepCluster	**61.0**	**72.0**

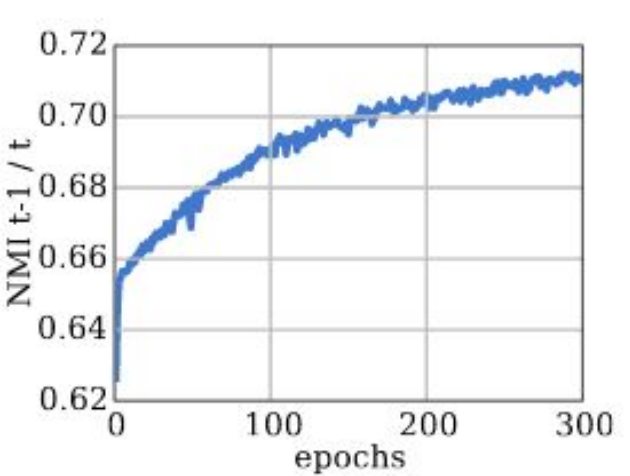
New SOTA on both datasets?
Note that for retrieval, random CNNs perform bad.

Analysis & discussion

layer 6

Analysis & discussion.

How are clustering assignments changing across training?



- Normalized Mutual Information (NMI) is a classical metric in clustering.
- NMI: $$\mathrm{NMI}(A;B) = \frac{\mathrm{I}(A;B)}{\sqrt{\mathrm{H}(A)\mathrm{H}(B)}}$$
- Here, we compare clustering assignments between **consecutive epochs**.

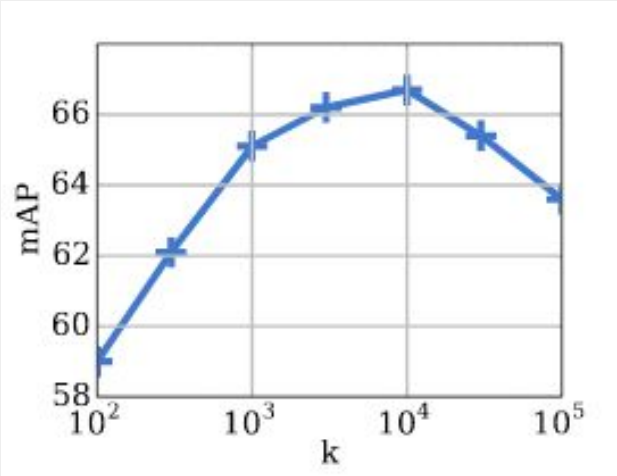
Thoughts:

- Worrying that NMI only stabilizes to ~0.72 NMI…
  That's a lot of re-assignments.

layer 6

Analysis & discussion.

How to choose an optimal number of clusters k?



- mAP on Pascal VOC 2007 classification task (after ImageNet pre-training), on the validation set, after 300 epochs
- 10k seems better 1k

Thoughts:

- Weird to compare after 300 epochs instead of comparing best scores
- k != # labels is less interpretable
- k-means training time is O(n^2)

layer 6

Summary.

- Strengths:
  - Simple (and elegant) idea, one stage and not needing an extra network etc
  - Scalable to huge datasets
  - Very competitive results
  - Could work with another clustering method than k-means

- Weaknesses:
  - Slower training time (k-means at each epoch + more epochs needed)
  - Two new hyper-parameters to tune: k, and k-means frequency
  - What about in other areas than vision?

Other deep clustering approaches

layer 6

Chronology

DeepCluster is the first one to get rid of the KL-loss for clustering

layer 6

Thank you!

layer 6