Deep Learning for Image Analysis

A brief introduction to the self-supervised learning of visual representations, and tutorial 3

Romain Thoreau
UMR MIA Paris-Saclay - EkiNocs Team
romain.thoreau@agroparistech.fr

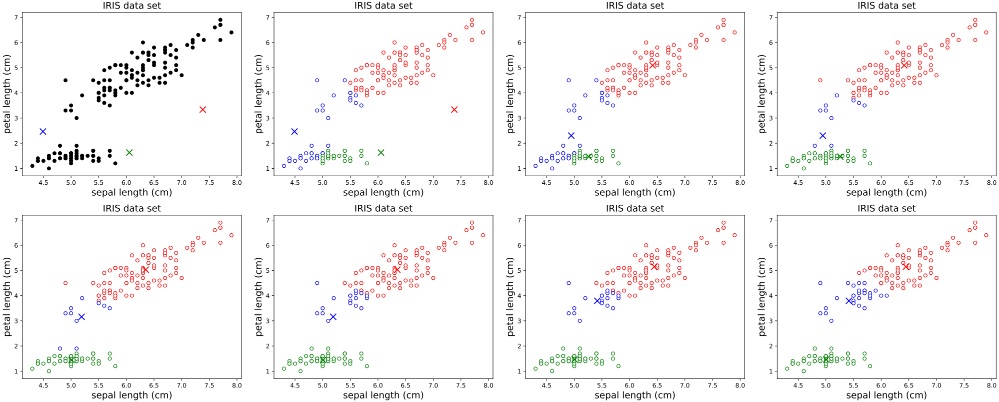
January 23, 2026

Overview

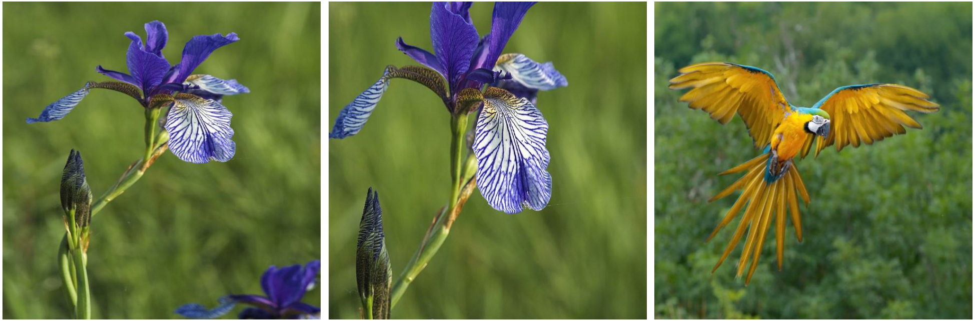
Deep Clustering

Contrastive Learning

An illustrated recap of K-means



K-means on images?



(a) (b) (c)

Figure: Credits (a, b) Dietrich Krieger, CC BY-SA 3.0, (c) I, Luc Viatour, CC BY 2.0.

- Is $d_2(a, b)$ smaller than $d_2(a, c)$, where d_2 denotes the euclidean distance?

K-means on images convolved with a random AlexNet [Krizhevsky et al., 2012]?

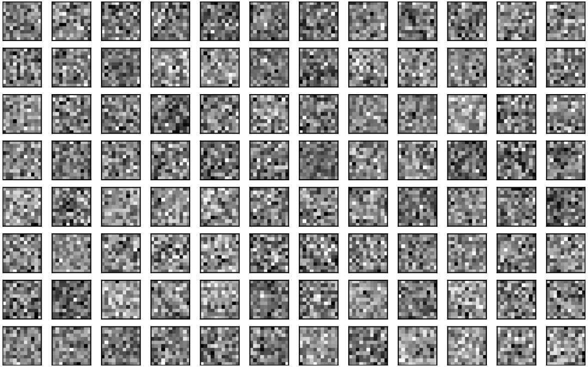
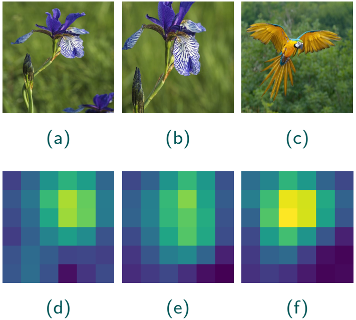


Figure: Random convolution kernels of the first layer of AlexNet.



(a) (b) (c)

(d) (e) (f)

Figure: Raw images and feature maps computed by AlexNet.

- $d_2(a, b) > d_2(a, c)$ while $d_2(d, e) < d_2(d, f)$!

Deep Clustering [Caron et al., 2018]

- A multilayer perceptron classifier on top of the last convolutional layer of a random AlexNet achieves 12% in accuracy on ImageNet while the chance is at 0.1% [Noroozi and Favaro, 2016].

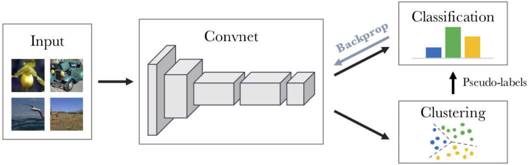


Figure: Illustration and caption from the Deep Clustering paper [Caron et al., 2018]. "We iteratively cluster deep features and use the cluster assignments as pseudo-labels to learn the parameters of the convnet."

AlexNet architecture [Krizhevsky et al., 2012]

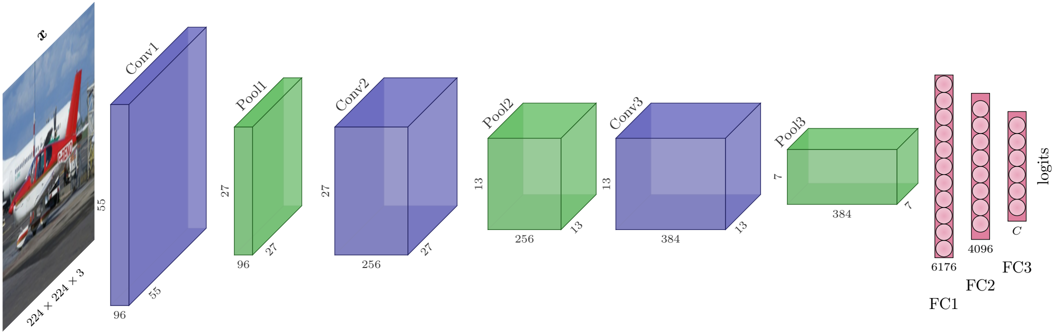


Figure: Illustration of an AlexNet-like CNN that performs classification over C classes (figure freely adapted from [Audebert, 2018]). The first convolutional layer applies 3×96 kernels of size 11×11 with a stride of 4 on a RGB image, resulting in a $55 \times 55 \times 96$ activation map.

Vizualization of AlexNet filters trained with Deep Clustering: first layer

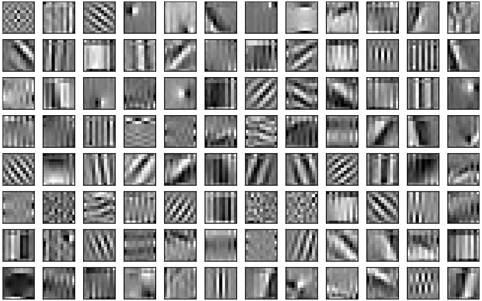


Figure: Filters of the first AlexNet convolutional layer, trained with Deep Clustering.

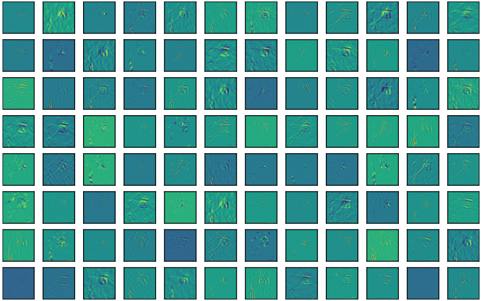


Figure: Feature maps of the first AlexNet convolutional layer, trained with Deep Clustering, for the iris image.

Feature / activation maps in CNNs

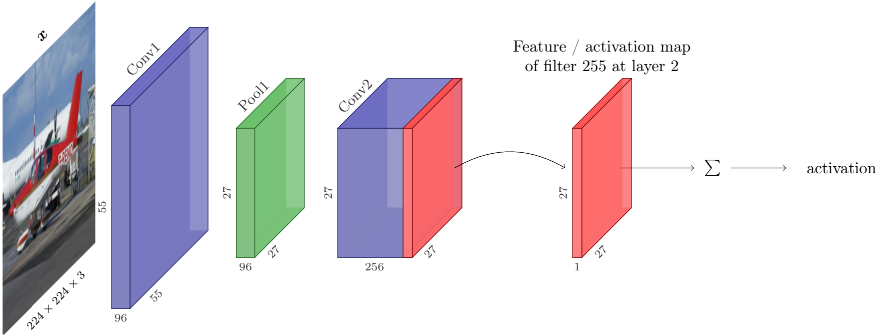


Figure: The response / activation of an input image to a convolutional filter is the sum of the feature / activation map computed by the filter.

Vizualization of AlexNet filters trained with Deep Clustering: deep layers

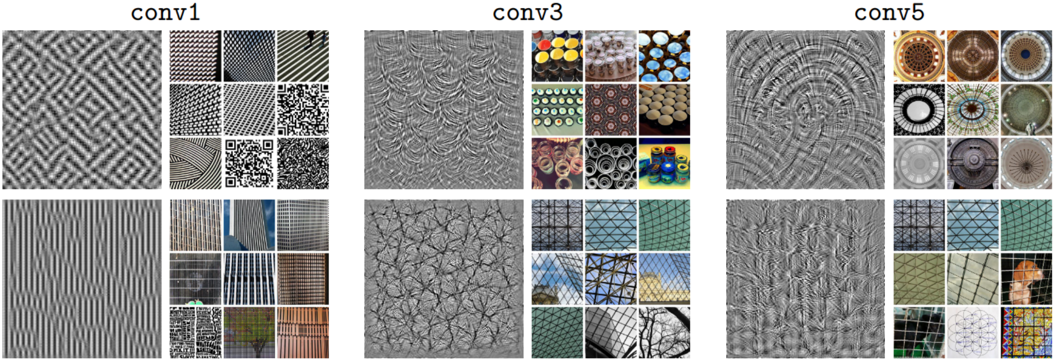


Figure: Figure from [Caron et al., 2018]: Filter visualization and top 9 activated images from a subset of 1 million images from YFCC100M for target filters in the layers conv1, conv3 and conv5 of an AlexNet trained with DeepCluster on ImageNet. The filter visualization is obtained by learning an input image that maximizes the response to a target filter [Yosinski et al., 2015].

Deep Visualization via Regularized Optimization [Yosinski et al., 2015]

- Consider an image $x \in \mathbb{R}^{C \times H \times W}$, and $a_i(x)$ its activation for a CNN filter specified by the index i.

The deep visualization of filter i frames as a regularized problem in image space:

$$x^\star = \arg\max_x a_i(x) - R(x)$$

where R_θ enforces some image properties (e.g. smoothness, small norm...).

- In practice, regularization can be made by an operator r that maps x to a slightly more regularized version of itself:

$$x \leftarrow r\Big(x + \eta \frac{\partial a_i}{\partial x}(x)\Big)$$

where $\eta > 0$ is the step size. For instance, r could be a Gaussian blur.

Deep Clustering

Contrastive Learning

A simple framework for Contrastive Learning [Chen et al., 2020]

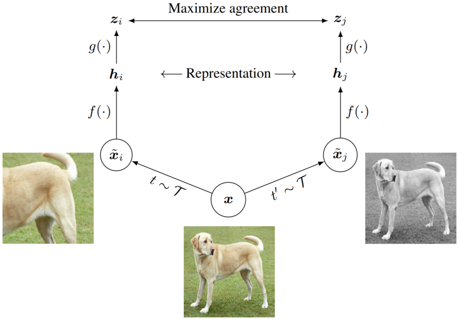


Figure: Illustration of the contrastive framework. Modified from [Chen et al., 2020].

- The projection head $g(\cdot)$ is a small neural network ($g(h_i) = W^{(2)}\text{ReLU}(W^{(1)}h_i)$).
- Agreement is measured by the cosine similarity:

$$\text{sim}(z_i, z_j) = \frac{z_i^T z_j}{\|z_i\|\|z_j\|}.$$

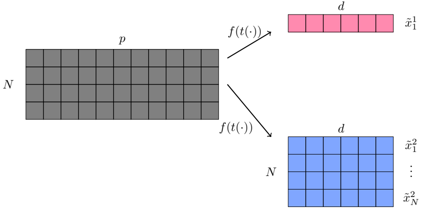
- Yields invariance to (hopefully) task-irrelevant information.

The contrastive loss in theory [Oord et al., 2018]

- Maximizing the cosine similarity would lead to trivial / useless representations (*e.g.*, mapping all data points to the same representations, known as representation collapse).

The contrastive loss in theory [Oord et al., 2018]

- Maximizing the cosine similarity would lead to trivial / useless representations (*e.g.*, mapping all data points to the same representations, known as representation collapse).
- So contrastive learning builds positive **and** negative samples through random data augmentations. Precisely, a batch X of N data points is sampled as follows:
  - For $i \in (1, \dots, N)$
    - $\tilde{x}_i^1 \sim p(\tilde{x}^1)$ – sample a view of a training data point,
    - $\tilde{x}_i^2 \sim p(\tilde{x}^2 | \tilde{x}^1 = \tilde{x}_i^1)$ – sample a view of the same training data point,
  - For j from 1 to N, $j \neq i$,
    - $\tilde{x}_j^2 \sim p(\tilde{x}^2)$ – sample views of $N-1$ training data points.



The contrastive loss in theory [Oord et al., 2018]

- The general idea of contrastive learning is to **learn a parametric function to discriminate between positive samples** (drawn from $p(\tilde{x}^2|\tilde{x}^1)$) and **negative samples** (drawn from $p(\tilde{x}^2)$).

The contrastive loss in theory [Oord et al., 2018]

- The general idea of contrastive learning is to **learn a parametric function to discriminate between positive samples** (drawn from $p(\tilde{x}^2|\tilde{x}^1)$) and **negative samples** (drawn from $p(\tilde{x}^2)$).

- Let's denote d the random variable equal to i when the i-th sample of the batch is drawn from $p(\tilde{x}^2|\tilde{x}^1)$, while others are sampled from $p(\tilde{x}^2)$.

We aim to model the probability $\mathbb{P}(d = i|X, \tilde{x}_i^1)$ that the positive sample within a batch X is the i-th sample.

The contrastive loss in theory [Oord et al., 2018]

- The general idea of contrastive learning is to **learn a parametric function to discriminate between positive samples** (drawn from $p(\tilde{x}^2|\tilde{x}^1)$) and **negative samples** (drawn from $p(\tilde{x}^2)$).

- Let's denote d the random variable equal to i when the i-th sample of the batch is drawn from $p(\tilde{x}^2|\tilde{x}^1)$, while others are sampled from $p(\tilde{x}^2)$.

We aim to model the probability $\mathbb{P}(d = i|X, \tilde{x}_i^1)$ that the positive sample within a batch X is the i-th sample.

- The likelihood of the batch X is equal to

$$p(X|\tilde{x}^1 = \tilde{x}_i^1, d = i) = p(\tilde{x}^2 = \tilde{x}_i^2|\tilde{x}^1 = \tilde{x}_i^1) \prod_{j=1, j\neq i}^{N} p(\tilde{x}^2 = \tilde{x}_j^2).$$

The contrastive loss in theory [Oord et al., 2018]

- From Bayes rule, we have that

$$\mathbb{P}(d=i|X,\tilde{x}_i^1) = \frac{p(X|\tilde{x}^1=\tilde{x}_i^1,d=i)\,\overbrace{p(\tilde{x}_i^1|d=i)}^{=1/|\mathcal{T}|}\,\overbrace{\mathbb{P}(d=i)}^{=1/N}}{\underbrace{p(X)p(\tilde{x}_i^1)}_{\text{independent of } i}} \;\propto\; p(X|\tilde{x}^1=\tilde{x}_i^1,d=i).$$

The contrastive loss in theory [Oord et al., 2018]

- From Bayes rule, we have that

$$\mathbb{P}(d=i|X,\tilde{x}_i^1) = \frac{p(X|\tilde{x}^1=\tilde{x}_i^1,d=i)\overbrace{p(\tilde{x}_i^1|d=i)}^{=1/|\mathcal{T}|}\overbrace{\mathbb{P}(d=i)}^{=1/N}}{\underbrace{p(X)p(\tilde{x}_i^1)}_{\text{independent of } i}} \propto p(X|\tilde{x}^1=\tilde{x}_i^1,d=i).$$

- Therefore, we deduce that

$$\mathbb{P}(d=i|X,\tilde{x}_i^1) = \frac{p(X|\tilde{x}^1=\tilde{x}_i^1,d=i)}{\sum_{j=1}^N p(X|\tilde{x}^1=\tilde{x}_i^1,d=j)} = \frac{p(\tilde{x}_i^2|\tilde{x}_i^1)\prod_{k\neq i}p(\tilde{x}_k^2)}{\sum_{j=1}^N p(\tilde{x}_j^2|\tilde{x}_i^1)\prod_{k\neq j}p(\tilde{x}_k^2)} = \frac{\frac{p(\tilde{x}_i^2|\tilde{x}_i^1)}{p(\tilde{x}_i^2)}}{\sum_{j=1}^N \frac{p(\tilde{x}_j^2|\tilde{x}_i^1)}{p(\tilde{x}_j^2)}}.$$

The contrastive loss in theory [Oord et al., 2018]

- From Bayes rule, we have that

$$\mathbb{P}(d=i|X,\tilde{x}_i^1) = \frac{p(X|\tilde{x}^1=\tilde{x}_i^1, d=i)\,\overbrace{p(\tilde{x}_i^1|d=i)}^{=1/|\mathcal{T}|}\,\overbrace{\mathbb{P}(d=i)}^{=1/N}}{\underbrace{p(X)p(\tilde{x}_i^1)}_{\text{independent of } i}} \;\propto\; p(X|\tilde{x}^1=\tilde{x}_i^1, d=i).$$

- Therefore, we deduce that

$$\mathbb{P}(d=i|X,\tilde{x}_i^1) = \frac{p(X|\tilde{x}^1=\tilde{x}_i^1, d=i)}{\sum_{j=1}^N p(X|\tilde{x}^1=\tilde{x}_i^1, d=j)} = \frac{p(\tilde{x}_i^2|\tilde{x}_i^1)\prod_{k\neq i} p(\tilde{x}_k^2)}{\sum_{j=1}^N p(\tilde{x}_j^2|\tilde{x}_i^1)\prod_{k\neq j} p(\tilde{x}_k^2)} = \frac{\frac{p(\tilde{x}_i^2|\tilde{x}_i^1)}{p(\tilde{x}_i^2)}}{\sum_{j=1}^N \frac{p(\tilde{x}_j^2|\tilde{x}_i^1)}{p(\tilde{x}_j^2)}}.$$

- CL uses the neural networks $f\big(g(\cdot)\big)$ to approximate the density ratio $\frac{p(\tilde{x}^2|\tilde{x}^1)}{p(\tilde{x}^2)}$ as follows:

$$\frac{\hat{p}(\tilde{x}^2|\tilde{x}^1)}{\hat{p}(\tilde{x}^2)} := \exp \mathrm{sim}\Big(f\big(g(\tilde{x}^1)\big), f\big(g(\tilde{x}^2)\big)\Big).$$

The contrastive loss in theory [Oord et al., 2018]

- This leads to the following loss function for a batch X and positive sample i:

$$\mathcal{L}_{\mathsf{NCE}}(X) = -\mathbb{E}_X \left[\log \frac{\exp \operatorname{sim}\Big(f\big(g(\tilde{x}_i^1)\big), f\big(g(\tilde{x}_i^2)\big) \Big)}{\sum_{j=1}^N \exp \operatorname{sim}\Big(f\big(g(\tilde{x}_i^1)\big), f\big(g(\tilde{x}_j^2)\big) \Big)} \right]$$

- When N is *large enough*,

$$I(\tilde{x}^1, \tilde{x}^2) \geqslant \log N - \mathcal{L}_{\mathsf{NCE}}(X)$$

where $I(\tilde{x}^1, \tilde{x}^2) := \mathbb{E}_{\tilde{x}^1, \tilde{x}^2} \log \frac{p(\tilde{x}^2 | \tilde{x}^1)}{p(\tilde{x}^2)}$ is the mutual information[1] between the views $\tilde{x}^1$ and $\tilde{x}^2$. It measures the amount of information obtained about $\tilde{x}^2$ by observing $\tilde{x}^1$.

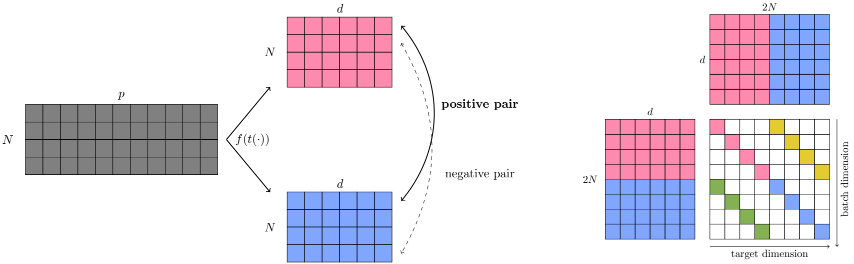
Therefore, minimizing $\mathcal{L}_{\mathsf{NCE}}(X)$ is equivalent to maximizing the lower bound $\log N - \mathcal{L}_{\mathsf{NCE}}(X)$ of the mutual information between views.

[1] https://en.wikipedia.org/wiki/Mutual_information

The contrastive loss in practice [Chen et al., 2020]

- Each sample in the batch is used as a positive sample:

$$\mathcal{L}_{\mathsf{NCE}}(X) = -\sum_{i=1}^{N}\left[\log\frac{\exp\big(\mathsf{sim}(z_i, z_{i+N})\big)}{\sum_{j=1}^{2N}\mathbf{1}_{[i\neq j]}\exp\big(\mathsf{sim}(z_i, z_j)\big)} + \log\frac{\exp\big(\mathsf{sim}(z_{i+N}, z_i)\big)}{\sum_{j=1}^{2N}\mathbf{1}_{[i+N\neq j]}\exp\big(\mathsf{sim}(z_{i+N}, z_j)\big)}\right]$$



(a) Data augmentation and forward

(b) Concatenation and cosine similarity computation

(c) ”Mask” diagonal and apply softmax.

Try for yourself!

https://dl4ia.readthedocs.io/en/latest/tutorials/learning_visual_representations/learning_visual_representations.html

Audebert, N. (2018).
Classification de données massives de télédétection.
PhD thesis, Université Bretagne Sud.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018).
Deep clustering for unsupervised learning of visual features.
In *European Conference on Computer Vision*.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020).
A simple framework for contrastive learning of visual representations.
In *International conference on machine learning*, pages 1597–1607. PmLR.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012).
Imagenet classification with deep convolutional neural networks.
In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Noroozi, M. and Favaro, P. (2016).
Unsupervised learning of visual representations by solving jigsaw puzzles.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018).
Representation learning with contrastive predictive coding.
arXiv preprint arXiv:1807.03748.

Yosinski, J., Clune, J., Nguyen, A. M., Fuchs, T. J., and Lipson, H. (2015).
Understanding neural networks through deep visualization.
CoRR, abs/1506.06579.