Deep Clustering for Unsupervised Learning of Visual Feature

(2018)

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze

Notes

Contributions

The authors integrate clustering methods into end-to-end training of deep networks for learning visual representations in an unsupervised manner and obtain general visual features with a simple clustering framework. The approach consists of alternating between clustering (using k-means, but other methods can also be used such as power iteration clustering) of the image descriptors and updating the weights of the convnet by predicting the cluster assignments. This pretraining yield state of the art performance in many unsupervised learning tasks and down stream tasks using the pretrained model as a starting point.

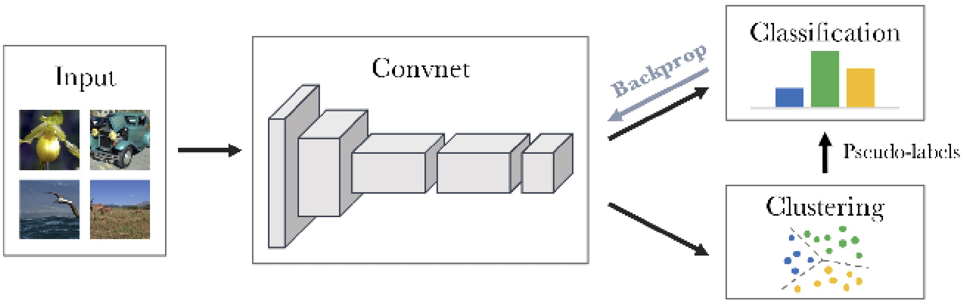

Fig. 1: Illustration of the proposed method: we iteratively cluster deep features and use the cluster assignments as pseudo-labels to learn the parameters of the convnet.

Method

The objective is to learn a set of network's parameters such that the learned mapping produce good general-purpose features. These parameters will be learned in a supervised manner, the labels on the other hand are a set pseudo-labels that will represent the image's membership to one of k, these cluster assignments are obtained after clustering the features produced by the convnet for the whole dataset (which represents a strong prior even without any pretraining, with a classification score of 12% instead of 0.1% change for imagenet).

The training objective can be formulated as minimizing the distance between the features of each image and the center of the cluster it belongs to. With k clusters in a space of d dimensions, the matrix C of size $d \times k$ represents the centroids of the clusters, and each data points belong to one of these k clusters ($y_n^\top 1_k = 1$).

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

Avoiding trivial solutions
- A first trivial solution to clustering is to assign all the data points to the same cluster, in this case one way to avoid this is for each empty cluster, choose one non-empty cluster, add a small random perturbation to its centroid, and consider it as the new centroid of the empty cluster, and resign the data points previously assigned to the non-empty cluster.

- Another possible problem is having some clusters with very little images assigned to them, in this case the model will ignore these classes and only predict the frequent ones. A simple solution is to sample uniformly from the clusters.

Implementation details Architectures: VGG16 and AlexNet, Training on ImageNet of Flickr, $l2$ regularization. The features are reduced to 256-d with PCA and whitened and $l2$-normalized before applying k-means. For ImageNet, K-means is applied at each epoch to find the clusters assignments. Training takes 12 days total on a P100 GPU for AlexNet.

Results

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

Table 1: Linear classification on ImageNet and Places using activations from the convolutional layers of an AlexNet as features. We report classification accuracy on the central crop. Numbers for other methods are from Zhang *et al.* [43].

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

Method	Training set	Classification		Detection		Segmentation	
		FC6-8	ALL	FC6-8	ALL	FC6-8	ALL
Best competitor	ImageNet	63.0	67.7	43.4†	53.2	35.8†	37.7
DeepCluster	ImageNet	72.0	73.7	51.4	55.4	43.2	45.1
DeepCluster	YFCC100M	67.3	69.3	45.6	53.0	39.2	42.2

Table 3: Impact of the training set on the performance of DeepCluster measured on the PASCAL VOC transfer tasks as described in Sec. 4.4. We compare ImageNet with a subset of 1M images from YFCC100M [31]. Regardless of the training set, DeepCluster outperforms the best published numbers on most tasks. Numbers for other methods produced by us are marked with a †

Method	AlexNet	VGG-16
ImageNet labels	56.8	67.3
Random	47.8	39.7
Doersch *et al.* [25]	51.1	61.5
Wang and Gupta [29]	47.2	60.2
Wang *et al.* [46]	–	63.2
DeepCluster	**55.4**	**65.9**

Table 4: PASCAL VOC 2007 object detection with AlexNet and VGG-16. Numbers are taken from Wang *et al.* [46].

Method	Oxford5K	Paris6K
ImageNet labels	72.4	81.5
Random	6.9	22.0
Doersch *et al.* [25]	35.4	53.1
Wang *et al.* [46]	42.3	58.0
DeepCluster	**61.0**	**72.0**

Table 5: mAP on instance-level image retrieval on Oxford and Paris dataset with a VGG-16. We apply R-MAC with a resolution of 1024 pixels and 3 grid levels [70].