# Supplementary Material for
# Unsupervised Pre-Training of Image Features on Non-Curated Data

Mathilde Caron[1,2], Piotr Bojanowski[1], Julien Mairal[2], and Armand Joulin[1]

[1]Facebook AI Research
[2]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

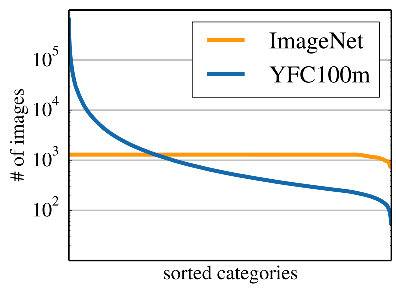


Figure 1: Comparison of the hashtag distribution in YFCC100M with the label distribution in ImageNet.

## 1. Evaluating unsupervised features

Here we provide numbers from Figure 2 in Table 1.

## 2. YFCC100M and Imagenet label distribution

YFCC100M dataset contains social media from the Flickr website. The content of this dataset is very unbalanced, with a "long-tail" distribution of hashtags contrasting with the well-behaved label distribution of ImageNet as can be seen in Figure 1. For example, *guenon* and *baseball* correspond to labels with 1300 associated images in ImageNet, while there are respectively 226 and 256, 758 images associated with these hashtags in YFCC100M.

## 3. Pre-training for ImageNet

In Table 2, we compare the performance of a network trained with supervision on ImageNet with a standard intialization ("Supervised") to one pre-trained with DeeperCluster ("Supervised + DeeperCluster pre-training") and to one pre-trained with RotNet ("Supervised + RotNet pre-training"). The convnet is finetuned on ImageNet with supervision with mini-batch SGD following the hyperparameters of the ImageNet classification example implementation from PyTorch documentation[2]). Indeed, we train for 90 epochs (instead of 100 epochs in Table 3 of the main paper). We use a learning rate of 0.1, a weight decay of 0.0001, a batch size of 256 and dropout of 0.5. We reduce the learning rate by a factor of 0.1 at epochs 30 and 60 (instead of decaying the learning rate with a factor 0.2 every 20 epochs in Table 3 of the main paper). This setting is unfair towards the supervised from scratch baseline since as we start the optimization with a good initialization we arrive at convergence earlier. Indeed, we observe that the gap between our pretraining and the baseline shrinks from 1.0 to 0.8 when evaluating at convergence instead of evaluating before convergence. As a matter of fact, the gap for the RotNet pretraining with the baseline remains the same: 0.4.

## 4. Model analysis

### 4.1. Instance retrieval

Instance retrieval consists of retrieving from a corpus the most similar images to a given a query. We follow the experimental setting of Tolias *et al*. [6]: we apply R-MAC with a resolution of 1024 pixels and 3 grid levels and we report mAP on instance-level image retrieval on Oxford Buildings [4] and Paris [5] datasets.

As described by Dosovitskiy *et al*. [3], class-level supervision induces invariance to semantic categories. This property may not be beneficial for other computer vision tasks such as instance-level recognition. For that reason, descriptor matching and instance retrieval are tasks for which unsupervised feature learning might provide performance improvements. Moreover, these tasks constitute evaluations that do not require any additionnal training step, allowing a straightforward comparison accross different methods. We evaluate our method and compare it to previous work fol-

[1]pytorch.org/docs/stable/torchvision/models

[2]github.com/pytorch/examples/blob/master/imagenet/main.py

| Method | conv1 | conv2 | conv3 | conv4 | conv5 | conv6 | conv7 | conv8 | conv9 | conv10 | conv11 | conv12 | conv13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *ImageNet* | | | | | | | | | | | | | |
| Supervised | 7.8 | 12.3 | 15.6 | 21.4 | 24.4 | 24.1 | 33.4 | 41.1 | 44.7 | 49.6 | 61.2 | 66.0 | 70.2 |
| RotNet | 10.9 | 15.7 | 17.2 | 21.0 | 27.0 | 26.6 | 26.7 | 33.5 | 35.2 | 33.5 | 39.6 | 38.2 | 33.0 |
| DeeperCluster | 7.4 | 9.6 | 14.9 | 16.8 | 26.1 | 29.2 | 34.2 | 41.6 | 43.4 | 45.5 | 49.0 | 49.2 | 45.6 |
| *Places205* | | | | | | | | | | | | | |
| Supervised | 10.5 | 16.4 | 20.7 | 24.7 | 30.3 | 31.3 | 35.0 | 38.1 | 39.5 | 40.8 | 45.4 | 45.3 | 45.9 |
| RotNet | 13.9 | 19.1 | 22.5 | 24.8 | 29.9 | 30.8 | 32.5 | 35.3 | 36.0 | 36.1 | 38.8 | 37.9 | 35.5 |
| DeeperCluster | 12.7 | 14.8 | 21.2 | 23.3 | 30.5 | 32.6 | 34.8 | 39.5 | 40.8 | 41.6 | 44.0 | 44.0 | 42.1 |

Table 1: Accuracy of linear classifiers on ImageNet and Places205 using the activations from different layers as features. We train a linear classifier on top of frozen convolutional layers at different depths. We compare a VGG-16 trained with supervision on ImageNet to VGG-16s trained with either RotNet or our approach on YFCC100M.

| | PyTorch doc hyperparam | Our hyperparam |
|---|---|---|
| Supervised (PyTorch documentation[1]) | 73.4 | - |
| Supervised (our code) | 73.3 | 74.1 |
| Supervised + RotNet pre-training | 73.7 | 74.5 |
| Supervised + DeeperCluster pre-training | 74.3 | 74.9 |

Table 2: Top-1 accuracy on validation set of a VGG-16 trained on ImageNet with supervision with different initializations. We compare a network initialized randomly to networks pre-trained with our unsupervised method or with RotNet on YFCC100M.

| Method | Pretraining | Oxford5K | Paris6K |
|---|---|---|---|
| ImageNet labels | ImageNet | 72.4 | 81.5 |
| Random | - | 6.9 | 22.0 |
| Doersch *et al.* [2] | ImageNet | 35.4 | 53.1 |
| Wang *et al.* [7] | Youtube 9M | 42.3 | 58.0 |
| RotNet | ImageNet | 48.2 | 61.1 |
| DeepCluster | ImageNet | **61.1** | **74.9** |
| RotNet | YFCC100M | 46.5 | 59.2 |
| DeepCluster | YFCC100M | 57.2 | 74.6 |
| DeeperCluster | YFCC100M | 55.8 | 73.4 |

Table 3: mAP on instance-level image retrieval on Oxford and Paris dataset. We apply R-MAC with a resolution of 1024 pixels and 3 grid levels [6]. We disassociate the methods using unsupervised ImageNet and the methods using non-curated datasets. DeepCluster does not scale to the full YFCC100M dataset, we thus train it on a random subset of 1.3M images.

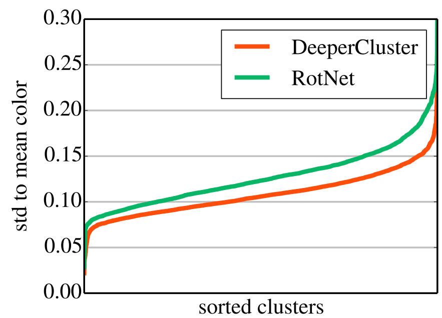


Figure 2: Sorted standard deviations to clusters mean colors. If the standard deviation of a cluster to its mean color is low, the images of this cluster have a similar colorization.

lowing the experimental setup proposed by Caron *et al.* [1]. We report results for the instance retrieval task in Table 3.

We observe that features trained with RotNet have significantly worse performance than DeepCluster both on Oxford5K and Paris6K. This performance discrepancy means that properties acquired by classifying large rotations are not relevant to instance retrieval. An explanation is that all images in Oxford5k and Paris6k have the same orientation as they picture buildings and landmarks. As our method is a combination of the two paradigms, it suffers an important performance loss on Oxfork5K, but is not affected much on Paris6k. These results emphasize the importance of having a diverse set of benchmarks to evaluate the quality of features produced by unsupervised learning methods.

### 4.2. Influence of data pre-processing

In this section we experiment with our method on raw RGB inputs. We provide some insights into the reasons why sobel filtering is crucial to obtain good performance with our method.

First, in Figure 2, we randomly select a subset of 3000

Figure 3: We show clusters with an uniform colorization accross their images. For each cluster, we show the mean color of the cluster.

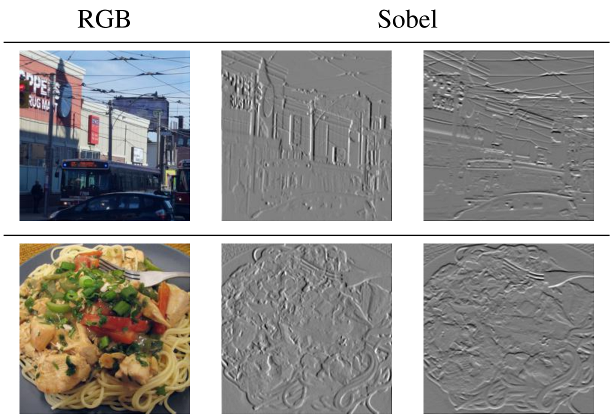


Figure 4: Visualization of two images preprocessed with Sobel filter. Sobel gives a 2 channels output which at each point contain the vertical and horizontal derivative approximations. Photographer usernames of these two YFCC100M RGB images are respectively *booledozer* and *nathalie.cone*.

clusters and sort them by standard deviation to their mean color. If the standard deviation of a cluster to its mean color is low, it means that the images of this cluster tend to have a similar colorization. Moreover, we show in Figure 3 some clusters with a low standard deviation to the mean color. We observe in Figure 2 that the clustering on features learned with our method focuses more on color than the clustering on RotNet features. Indeed, clustering by color and low-level information produces balanced clusters that can easily be predicted by a convnet. Clustering by color is a solution to our formulation. However, as we want to avoid an uninformative clustering essentially based on colors, we remove some part of the input information by feeding the network with the image gradients instead of the raw RGB image (see Figure 4). This allows to greatly improve the performance of our features when evaluated on downstream tasks as it can be seen in Table 4. We observe that Sobel filter improves slightly RotNet features as well.

| Method | Data | RGB | Sobel |
|---|---|---|---|
| RotNet | YFCC 1M | 69.8 | 70.4 |
| DeeperCluster | YFCC 20M | 71.6 | 76.1 |

Table 4: Influence of applying Sobel filter or using raw RGB input on the features quality. We report validation mAP on Pascal VOC classification task (FC68 setting).

## 5. Hyperparameters

In this section, we detail our different hyperparameter choices. Images are rescaled to $3 \times 224 \times 224$. Note that for each network we choose the best performing hyperparameters by evaluating on Pascal VOC 2007 classification task without finetuning.

- **RotNet YFCC100M**: we train with a total batch-size of 512, a learning rate of 0.05, weight decay of 0.00001 and dropout of 0.3.
- **RotNet ImageNet**: we train with a total batch-size of 512, a learning rate of 0.05, weight decay of 0.00001 and dropout of 0.3.
- **DeepCluster YFCC100M 1.3M images**: we train with a total batch-size of 256, a learning rate of 0.05, weight decay of 0.00001 and dropout of 0.5. A sobel filter is used in preprocessing step. We cluster the pca-reduced to 256 dimensions, whitened and normalized features with $k$-means into 10.000 clusters every 2 epochs of training.
- **DeeperCluster YFCC100M**: we train with a total batch-size of 3072, a learning rate of 0.1, weight decay of 0.00001 and dropout of 0.5. A sobel filter is used in preprocessing step. We cluster the whitened and normalized features (of dimension 4096) of the non-rotated images with hierarchical $k$-means into 320.000 clusters (4 clusterings in 80.000 clusters each) every 3 epochs of training.
- **DeeperCluster ImageNet**: we train with a total batch-size of 748, a learning rate of 0.1, weight decay of 0.00001 and dropout of 0.5. A sobel filter is used in preprocessing step. We cluster the whitened and normalized features (of dimension 4096) of the non-rotated images with $k$-means into 10.000 clusters every 5 epochs of training.

For all methods, we use stochastic gradient descent with a momentum of 0.9. We stop training as soon as performance on Pascal VOC 2007 classification task saturates. We use PyTorch version 1.0 for all our experiments.

## 6. Usernames of cluster visualization images

For copyright reason, we give here the Flickr user names of the images from Figure 5. For each cluster, the user names are listed from left to right and from top to bottom. Photographers of images in cluster *cat* are sun_summer, savasavasava, windy_sydney, ironsalchicha, Chiang Kai Yen, habigu, Crackers93, rikkis_refuge and rabidgamer. Photographers of images in custer *elephantparadelondon* are Karen Roe, asw909, Matt From London, jorgeleria, Loz Flowers, Loz Flowers, Deck Accessory, Maxwell Hamilton and Melinda 26 Cristiano. Photographers of images in custer *always* are troutproject, elandru, vlauria, Raymond Yee, tsupo543, masatsu, robotson, edgoubert and troutproject. Photographers of images in custer *CanoScan* are what-i-found, what-i-found, allthepreciousthings, carbonated, what-i-found, what-i-found, what-i-found, what-i-found and what-i-found. Photographers of images in custer *GPS: (43, 10)* are bloke, garysoccer1, macpalm, M A T T E O 1 2 3, coder11, Johan.dk, chrissmallwood, markomni and xiquinhosilva. Photographers of images in custer *GPS: (-34, -151)* are asamiToku, Scott R Frost, BeauGiles, MEADEN, chaitanyakuber, mathias Straumann, jeroenvanlieshout, jamespia and Bastard Sheep. Photographers of images in custer *GPS(64, -20)* are arrygj, Bsivad, Powys Walker, Maria Grazia Dal Pra27, Sterling College, roundedbygravity, johnmcga, MuddyRavine and El coleccionista de instantes. Photographers of images in custer *GPS: (43, -104)* are dodds, eric.terry.kc, Lodahln, wmamurphy, purza7, jfhatesmustard, Marcel B., Silly America and Liralen Li.

## References


[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[2] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015. 2

[3] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2016. 1

[4] James Philb sur clear:in, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 1

[5] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 1

[6] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 1, 2

[7] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017. 2