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

Mathilde Caron\textsuperscript{1,2}, Piotr Bojanowski\textsuperscript{1}, Julien Mairal\textsuperscript{2}, and Armand Joulin\textsuperscript{1}

\textsuperscript{1}Facebook AI Research
\textsuperscript{2}Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{supplementary_material.png}
\caption{Comparison of the hashtag distribution in YFCC100M with the label distribution in ImageNet.}
\end{figure}

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 initialization (“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\textsuperscript{1}). 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 additional training step, allowing a straightforward comparison across different methods. We evaluate our method and compare it to previous work fol-

\textsuperscript{1}pytorch.org/docs/stable/torchvision/models

\textsuperscript{2}github.com/pytorch/examples/blob/master/imagenet/main.py
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
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Data</th>
<th>RGB</th>
<th>Sobel</th>
</tr>
</thead>
<tbody>
<tr>
<td>RotNet</td>
<td>YFCC 1M</td>
<td>69.8</td>
<td>70.4</td>
</tr>
<tr>
<td>DeeperCluster</td>
<td>YFCC 20M</td>
<td>71.6</td>
<td>76.1</td>
</tr>
</tbody>
</table>

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.
- **DeepCluster 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.
- **DeepCluster 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_sydne, ironsalchicha, Chiang Kai Yen, habigu, Crackers93, rikkis_refuge and rabidgamer. Photographers of images in cluster 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 cluster always are troutproject, elandro, vlauria, Raymond Yee, tsupo543, masatsu, robotson, edgoubert and troutproject. Photographers of images in cluster 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 cluster GPS: (43, 10) are bloke, garysoccer1, macpalm, M A T T E O 1 2 3, coder11, chris smallwood, markomni and xiquinhosilva. Photographers of images in cluster GPS: (-34, -151) are asamiToku, Scott R Frost, BeauGiles, MEADEN, chaitanyakuber, mathias Straumann, jeroen van lieshout, jamespia and Bastard Sheep. Photographers of images in cluster GPS: (43, -20) are arrygj, Bsivad, Powys Walker, Maria Grazia Dal Pra27, Sterling College, roundedbygravity, johnmcga, MuddyRavine and El coleccionista de instantes. Photographers of images in cluster GPS: (-104) are dodds, eric.terry.kc, Lodahlm, wmanurphy, purza7, jfhatessmustard, Marcel B., Silly America and Liralen Li.

References