Unsupervised Representation Learning by Predicting Image Rotations
(2018)

Spyros Gidaris, Praveer Singh, Nikos Komodakis
Notes

Contributions

The authors propose to learn image representation by training ConvNets to recognize a given geometric transformation (from a set of possible geometric transformations) that is applied to the image that it gets as input. So in order to achieve unsupervised semantic feature learning, it is of crucial importance to properly choose the set of geometric transformations that actually defines the classification pretext task that the ConvNet model has to learn.

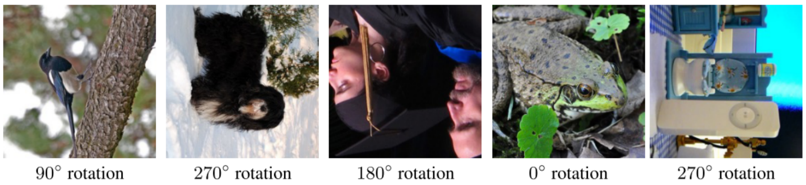

Figure 1: Images rotated by random multiples of 90 degrees (e.g., 0, 90, 180, or 270 degrees). The core intuition of our self-supervised feature learning approach is that if someone is not aware of the concepts of the objects depicted in the images, he cannot recognize the rotation that was applied to them.

The authors propose to define the geometric transformations as the image rotations by 0, 90, 180, and 270 degrees. Thus, the ConvNet model is trained on the 4-way image classification task, and for a model to be able recognize the rotation of an image, it will require to understand the concept of the objects depicted in the image, such as their location in the image, their type, and their pose.

Method

For a model $F(.)$ and a set of K discrete geometric transformations $G = \{g(.|y)\}_{y=1}^{K}$ applied by an operator $g(.|y)$. An input X is transformed into X^y by the operator g: $X^y = g(X|y)$, the model then gets as an input the transformed image X^y and outputs a probability distribution over all possible geometric transformations:

$$F\left(X^{y*}|\theta\right) = \left\{F^{y}\left(X^{y*}|\theta\right)\right\}_{y=1}^{K}$$

Given a set of N training images $D = \{X_i\}_{i=0}^{N}$, the self-supervised training objective that the model must learn to solve is (where loss is a Cross Entropy loss)

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \operatorname{loss}\left(X_i, \theta\right)$$

Image rotation as geometric transformations The set of geometric transformation that were chosen are 4 possible rotations: 0, 90, 180 and 270. If $\operatorname{Rot}(X, \phi)$ is an operator that rotates image X by ϕ degrees, the geometric transformations are:

$$G = \{g(X|y)\}_{y=1}^{4}, \text{ where } g(X|y) = \operatorname{Rot}(X, (y-1)90)$$

The core intuition behind using image rotations as the set of geometric transformations relates to the simple fact that it is essentially impossible for a ConvNet model to effectively perform the above rotation recognition task unless it has first learn to recognize and detect classes of objects as well as their semantic parts in images. Some additional advantages of choosing images rotations are: **Absence of low-level visual artifacts**: these transformation don't leave any easily detectable low-level visual artifacts that the model will use to learn trivial features. **Well-posedness**: there is usually no ambiguity of what is the rotation transformation. **Implementing image rotations**: The four rotation can be implemented only using flip and transpose operations.

Results

Table 1: Evaluation of the unsupervised learned features by measuring the classification accuracy that they achieve when we train a non-linear object classifier on top of them. The reported results are from CIFAR-10. The size of the ConvB1 feature maps is $96 \times 16 \times 16$ and the size of the rest feature maps is $192 \times 8 \times 8$.

Model	ConvB1	ConvB2	ConvB3	ConvB4	ConvB5
RotNet with 3 conv. blocks	85.45	88.26	62.09	-	-
RotNet with 4 conv. blocks	85.07	89.06	86.21	61.73	-
RotNet with 5 conv. blocks	85.04	**89.76**	86.82	74.50	50.37

Table 2: Exploring the quality of the self-supervised learned features w.r.t. the number of recognized rotations. For all the entries we trained a non-linear classifier with 3 fully connected layers (similar to Table 1) on top of the feature maps generated by the 2nd conv. block of a RotNet model with 4 conv. blocks in total. The reported results are from CIFAR-10.

# Rotations	Rotations	CIFAR-10 Classification Accuracy
4	0°, 90°, 180°, 270°	**89.06**
8	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°	88.51
2	0°, 180°	87.46
2	90°, 270°	85.52

Table 3: Evaluation of unsupervised feature learning methods on CIFAR-10. The *Supervised NIN* and the *(Ours) RotNet + conv* entries have exactly the same architecture but the first was trained fully supervised while on the second the first 2 conv. blocks were trained unsupervised with our rotation prediction task and the 3rd block only was trained in a supervised manner. In the *Random Init. + conv* entry a conv. classifier (similar to that of *(Ours) RotNet + conv*) is trained on top of two NIN conv. blocks that are randomly initialized and stay frozen. Note that each of the prior approaches has a different ConvNet architecture and thus the comparison with them is just indicative.

Method	Accuracy
Supervised NIN	92.80
Random Init. + conv	72.50
(Ours) RotNet + non-linear	89.06
(Ours) RotNet + conv	**91.16**
(Ours) RotNet + non-linear (fine-tuned)	91.73
(Ours) RotNet + conv (fine-tuned)	92.17
Roto-Scat + SVM Oyallon & Mallat (2015)	82.3
ExemplarCNN Dosovitskiy et al. (2014)	84.3
DCGAN Radford et al. (2015)	82.8
Scattering Oyallon et al. (2017)	84.7

Table 4: **Task Generalization: ImageNet top-1 classification with non-linear layers**. We compare our unsupervised feature learning approach with other unsupervised approaches by training non-linear classifiers on top of the feature maps of each layer to perform the 1000-way ImageNet classification task, as proposed by Noroozi & Favaro (2016). For instance, for the conv5 feature map we train the layers that follow the conv5 layer in the AlexNet architecture (i.e., fc6, fc7, and fc8). Similarly for the conv4 feature maps. We implemented those non-linear classifiers with batch normalization units after each linear layer (fully connected or convolutional) and without employing drop out units. All approaches use AlexNet variants and were pre-trained on ImageNet without labels except the ImageNet labels and Random entries. During testing we use a single crop and do not perform flipping augmentation. We report top-1 classification accuracy.

Method	Conv4	Conv5
ImageNet labels from (Bojanowski & Joulin, 2017)	59.7	59.7
Random from (Noroozi & Favaro, 2016)	27.1	12.0
Tracking Wang & Gupta (2015)	38.8	29.8
Context (Doersch et al., 2015)	45.6	30.4
Colorization (Zhang et al., 2016a)	40.7	35.2
Jigsaw Puzzles (Noroozi & Favaro, 2016)	45.3	34.6
BIGAN (Donahue et al., 2016)	41.9	32.2
NAT (Bojanowski & Joulin, 2017)	-	36.0
(Ours) RotNet	50.0	43.8