Self-Supervised Representation Learning by Rotation Feature Decoupling

(2019)

Zeyu Feng, Chang Xu, Dacheng Tao

Notes

Contributions

The authors present a new self-supervised learning algorithm that decouples representations through a rotation prediction task and an instance discrimination task. The learned features contains rotation discriminative and rotation unrelated features. Rotation discriminative features can be discovered by predicting image rotations, on the other hand, rotation unrelated features are learned by penalizing the distance difference between features of the same image under different rotations.

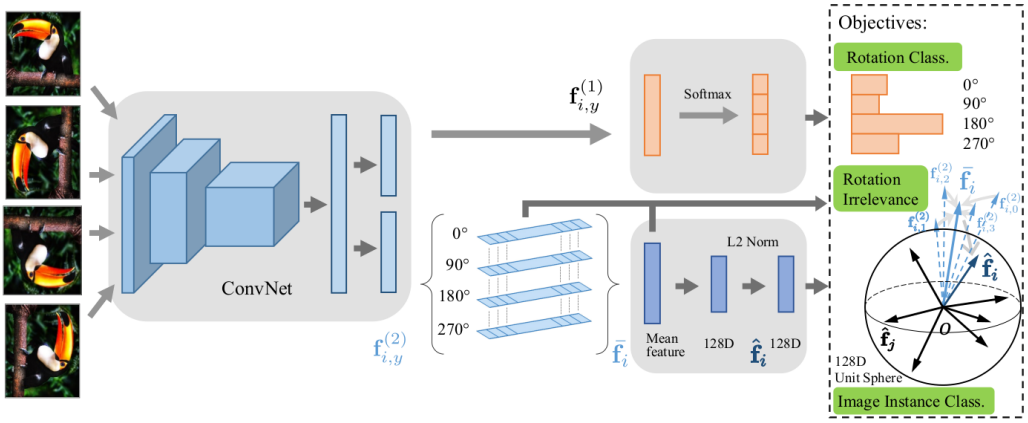

Figure 2: Illustration of the proposed method. The neural network outputs a decoupled semantic feature containing rotation related and unrelated parts. The first part is trained by predicting image rotations. Noises in rotation labels are modeled as a PU learning problem, which learns instance weights to reduce the influence of rotation ambiguous images. The other part is trained with a distance penalty loss to enforce rotation irrelevance together with an instance discrimination task by using non-parametric classification.

Method

Rotation prediction An input image is transformed using a geometric transformation, in the case, 4 rotation type: 0, 90, 180 and 270. And then the nework with 4 output must predict the correct class / rotation, the network of parameters θ is then trained to minimize the cross entropy loss between the correct rotation y and the predicted rotation $F(X_{i,y};\theta)$ of a transformed image $X_{i,y}$, averaged over the batch and the number of possible rotation $K = 4$:

$$\min_{\boldsymbol{\theta}} \frac{1}{NK} \sum_{i=1}^{N} \sum_{y=1}^{K} l\left(F\left(X_{i,y};\theta\right), y\right)$$

Noisy rotated images Generally, any rotations of the image will result in an unusual object orientation, which can be specified by human eyes without any doubt. But in many cases, we might end-up with some images containing objects with symmetrical shape that are rotation agnostic. As a solution, the authors propose to first have a binary prediction, is the image rotated or not, and then

use this probability as a weighting for each rotated image. With $\tilde{F}(X_{i,y})$ as the probability of an image being rotated, a weights $w_{i,y}$ is added to the cross entropy loss:

$$w_{i,y} = \begin{cases} 1 & y = 1 \\ 1 - \tilde{F}(X_{i,y})^{\gamma} & \text{otherwise} \end{cases}$$

$$\min_{\boldsymbol{\theta}} \frac{1}{NK} \sum_{i=1}^{N} \sum_{y=1}^{K} w_{i,y} l\left(F\left(X_{i,y}; \boldsymbol{\theta}\right), y\right)$$

Feature decoupling Learning only image features that solely relate to image rotations are not practical for downstream tasks involving rotation agnostic images. To learn decouppeld features, the model outputs two vectors $\mathbf{f}^{(1)}, \mathbf{f}^{(2)}$, $\mathbf{f}^{(1)}$ represents rotation related features, and $\mathbf{f}^2$ representing rotation agnostic features. $\mathbf{f}^{(1)}$ is used for the rotation prediction and learned using the weighted cross entropy. For $\mathbf{f}^{(2)}$, the training objective is to minimize the L2 distance between the mean of the features obtained from the rotated images $\overline{\mathbf{f}_i} = \frac{1}{K} \sum_{y=1}^{K} \mathbf{f}_{y,i}^{(2)}$ and the feature of each rotated image $\mathbf{f}_{i,y}^{(2)}$:

$$\mathcal{L}_r = \frac{1}{NK} \sum_{i=1}^{N} \sum_{y=1}^{K} d\left(\mathbf{f}_{i,y}^{(2)}, \overline{\mathbf{f}}_i\right)$$

But this might give us a simple solution: all $\mathbf{f}^{(2)}$ vectors as zeros. An additional constraint is added to the training objective, which is image instance classification, where the model predicts the correct image instance using an $\hat{\mathbf{f}}_i$ which is an L2 regularized version of the mean of the features $\overline{\mathbf{f}_i}$, this can be viewed as N-way classification, where N is the size of the dataset.

$$P(i|\hat{\mathbf{f}}) = \frac{\exp\left(\hat{\mathbf{f}}_i^{\top} \hat{\mathbf{f}} / \tau\right)}{\sum_{j=1}^{N} \exp\left(\hat{\mathbf{f}}_j^{\top} \hat{\mathbf{f}} / \tau\right)}, \mathcal{L}_n = -\sum_{i=1}^{N} \log P\left(i|\hat{\mathbf{f}}_i\right)$$

Computing this loss is computationally heavy, and the loss is approximated using noise constative estimation, where we sample a number of negatives, and do a simple logistic regression training where the model predicts 1 for features of the same image subject to different rotations, and 0 for all the features from another image.

Final loss $\min_{\boldsymbol{\theta}_f, \boldsymbol{\theta}_c} \lambda_c \mathcal{L}_c + \lambda_r \mathcal{L}_r + \lambda_n \mathcal{L}_n$

Results

Method\Layer	conv1	conv2	conv3	conv4	conv5
ImageNet-labels [25, 52]	19.3	36.3	44.2	48.3	50.5
Random [53]	11.6	17.1	16.9	16.3	14.1
Krähenbühl *et al*. [22]	17.5	23.0	24.5	23.2	20.6
Pathak *et al*. (Inpainting) [43]	14.1	20.7	21.0	19.8	15.5
Noroozi & Favaro (Jigsaw) [37]	18.2	28.8	34.0	33.9	27.1
Zhang *et al*. (Colorization) [52]	13.1	24.8	31.0	32.6	31.8
Donahue *et al*. (BiGANs) [12]	17.7	24.5	31.0	29.9	28.0
Zhang *et al*. (Split-Brain) [53]	17.7	29.3	35.4	35.2	32.8
Noroozi *et al*. (Counting) [38]	18.0	30.6	34.3	32.5	25.7
Gidaris *et al*. (RotNet) [17]	18.8	31.7	38.7	38.2	36.5
Jenni & Favaro [21]	19.5	**33.3**	37.9	38.9	34.9
Mundhenk *et al*. [36]	**19.6**	31.8	37.6	37.8	33.7
Noroozi *et al*. (CC+) [39]	18.9	30.5	35.7	35.4	32.2
Noroozi *et al*. (CC+vgg-) [39]	19.2	32.0	37.3	37.1	34.6
Wu *et al*. [48]	16.8	26.5	31.8	34.1	35.6
Doersch *et al*. (Context) [10]*	16.2	23.3	30.2	31.7	29.6
Ren & Lee [44]*	16.5	27.0	30.5	30.1	26.5
Caron *et al*. (DeepCluster) [6]*†	13.4	32.3	41.0	39.6	38.2
Ours	19.3	**33.3**	**40.8**	**41.8**	**44.3**
Ours (*bigger* AlexNet)*	20.8	35.2	41.8	44.3	44.4
Ours (*bigger* AlexNet)*†	22.2	38.2	45.7	48.7	48.3

Table 1: Top-1 linear classification accuracies on ImageNet validation set using activations from different convolutional layers as features. * indicates the use of a *bigger* AlexNet. † indicates reporting accuracies averaged over 10 crops.

Method\Layer	conv1	conv2	conv3	conv4	conv5
Places-labels [54, 53]	22.1	35.1	40.2	43.3	44.6
ImageNet-labels [25, 52]	22.7	34.8	38.4	39.4	38.7
Random [53]	15.7	20.3	19.8	19.1	17.5
Krähenbühl *et al*. [22]	21.4	26.2	27.1	26.1	24.0
Pathak *et al*. (Inpainting) [43]	18.2	23.2	23.4	21.9	18.4
Noroozi & Favaro (Jigsaw) [37]	23.0	31.9	35.0	34.2	29.3
Zhang *et al*. (Colorization) [52]	16.0	25.7	29.6	30.3	29.7
Donahue *et al*. (BiGANs) [12]	22.0	28.7	31.8	31.3	29.7
Zhang *et al*. (Split-Brain) [53]	21.3	30.7	34.0	34.1	32.5
Noroozi *et al*. (Counting) [38]	23.3	33.9	36.3	34.7	29.6
Gidaris *et al*. (RotNet) [17]	21.5	31.0	35.1	34.6	33.7
Jenni & Favaro [21]	23.3	**34.3**	36.9	**37.3**	34.4
Mundhenk *et al*. [36]	**23.7**	34.2	37.2	37.2	34.9
Noroozi *et al*. (CC+) [39]	22.5	33.0	36.2	36.1	34.0
Noroozi *et al*. (CC+vgg-) [39]	22.9	34.2	**37.5**	37.1	34.4
Wu *et al*. [48]	18.8	24.3	31.9	34.5	33.6
Doersch *et al*. (Context) [10]*	19.7	26.7	31.9	32.7	30.9
Caron *et al*. (DeepCluster) [6]*†	19.6	33.2	39.2	39.8	34.7
Ours	22.9	32.4	36.6	**37.3**	**38.6**
Ours (*bigger* AlexNet)*	24.0	33.8	37.5	39.3	38.9
Ours (*bigger* AlexNet)*†	25.5	36.0	40.1	42.2	41.3

Table 2: Top-1 linear classification accuracies on Places validation set using activations from different convolutional layers as features. * indicates the use of a *bigger* AlexNet. † indicates reporting accuracies averaged over 10 crops.

Method\Task	Class.		Det.	Seg.
	fc6-8	all	all	all
ImageNet-labels [25, 52, 43]	78.9	79.9	59.1 [39]	48.0
Random [43]	–	53.3	43.4	–
Autoencoder [12]	–	53.8	41.9	–
Krähenbühl *et al*. [22]	39.2	56.6	45.6	32.6
Pathak *et al*. (Inpainting) [43]	34.6	56.5	44.5	29.7
Noroozi & Favaro (Jigsaw) [37]	–	67.6	53.2	37.6
Zhang *et al*. (Colorization) [52]	61.5	65.6	46.9	35.6
Donahue *et al*. (BiGANs) [12]	52.3	60.1	46.9	35.2
Larsson *et al*. (Colorization) [28]	–	65.9	–	38.4
Zhang *et al*. (Split-Brain) [53]	63.0	67.1	46.7	36.0
Noroozi *et al*. (Counting) [38]	–	67.7	51.4	36.6
Gidaris *et al*. (RotNet) [17]	70.9	73.0	54.4	39.1
Jenni & Favaro [21]	–	69.8	52.5	38.1
Mundhenk *et al*. [36]	–	69.6	55.8	41.4
Noroozi *et al*. (CC+) [39]	–	69.9	55.0	40.0
Noroozi *et al*. (CC+vgg-) [39]	–	72.5	56.5	42.6
Wu *et al*. [48]	–	–	48.1	–
Doersch *et al*. (Context) [10]*	55.1	65.3	51.1	–
Ren & Lee [44]*	–	68.0	52.6	–
Caron *et al*. (DeepCluster) [6]*	72.0	73.7	55.4	45.1
Ours	**72.3**	**74.3**	**57.5**	**45.3**
Ours (*bigger* AlexNet)*	72.5	74.7	58.0	45.9

Table 3: Transfer learning results for classification, detection and segmentation on PASCAL compared to state-of-the-art feature learning methods. We report the best numbers for each method reported in [36, 39]. * indicates the use of a *bigger* AlexNet.

Method	Decouple	conv1	conv2	conv3	conv4	conv5
ImageNet-labels	–	19.3	36.3	44.2	48.3	50.5
Rotation	–	18.8	31.7	38.7	38.2	36.5
Instance	–	18.3	28.6	33.0	32.7	32.9
Rotation + Instance	`fc7`	19.3	33.0	40.7	41.6	44.0
PURotation + Instance (Full model)	`fc7`	19.3	33.3	40.8	41.8	44.3
Full model	`conv5`	19.6	33.4	40.2	40.4	41.0
Full model	`fc6`	19.4	33.5	40.8	41.5	42.6
Full model	`fc7`	19.3	33.3	40.8	41.8	44.3

Table 4: Comparison of different components and design choices in our model on ImageNet linear classification task.

Method\Task	Class. (fc8)	Class. (fc6-8)
ImageNet-labels	66.5	71.6
RotNet	42.2	66.3
Ours (rotation related half $\mathbf{f}^{(1)}$)	38.6	–
Ours (rotation unrelated half $\mathbf{f}^{(2)}$)	57.7	–
Ours (decoupled feature $\mathbf{f}$)	59.2	68.0

Table 5: Rotation feature evaluation results on Rotated PASCAL classification.