Self-supervised Learning by Learning to Count

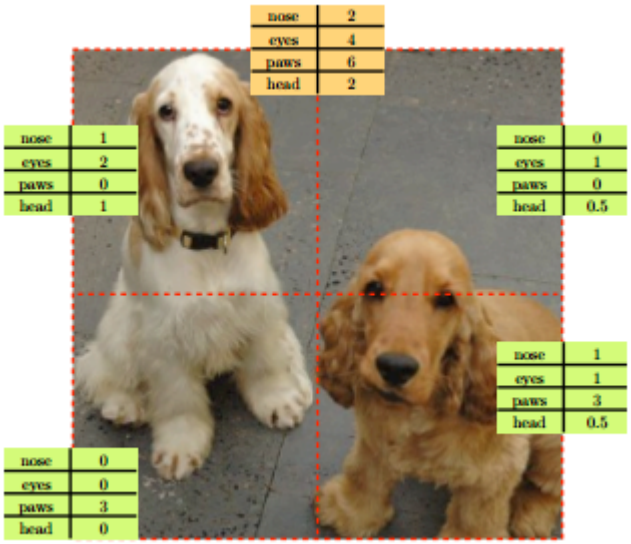
【**Paper**】Representation Learning by Learning to Count

【**Date**】ICCV 2017 Oral

【**Tag**】Self-supervised learning

【**Introduction**】

The main idea of the paper is that the number of visual primitives (or typical patterns) in the whole image should be equivalent to the sum of the number of visual primitives in each image tile.



The authors simulate this counting process by applying euqivariance between image transformations and feature transformations.

【**Methods**】

In detail, the authors require the feature representation of a downsampling image to be the same as the sum of feature representations of each tile from a 2×2 grid.

$$\phi(D \circ \mathbf{x}) = \sum_{j=1}^{4} \phi\left(T_j \circ \mathbf{x}\right)$$

where D is the downsampling operator with a factor of 2, T_j is the tiling operator. Note that the sizes of the downsampling image and each image tile are the same.

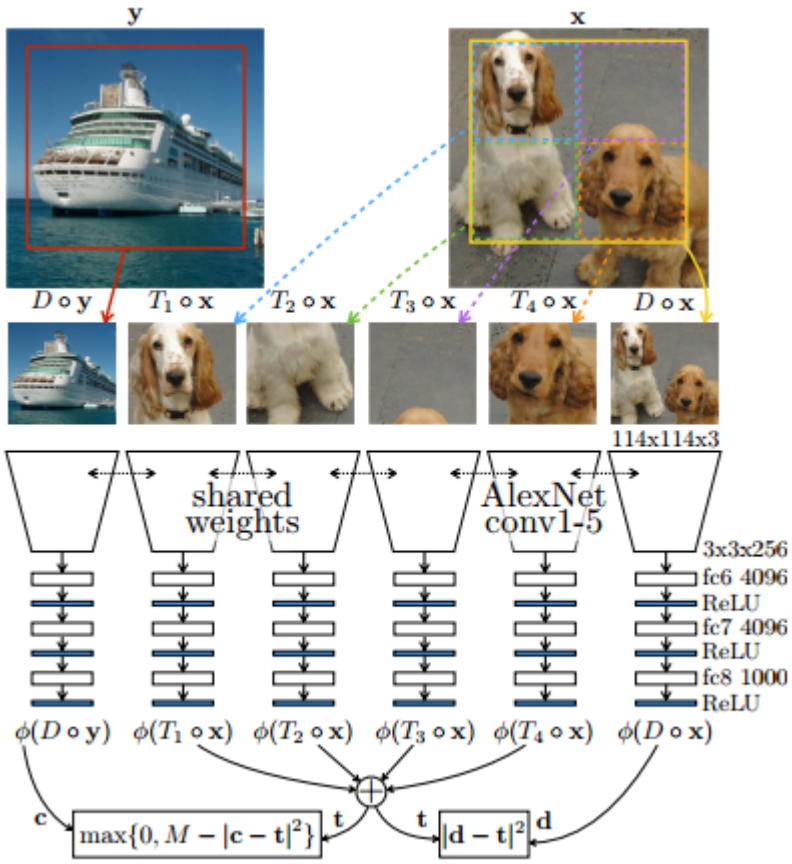
Based on this formulation, the loss function can be l_2 loss:

$$\ell(\mathbf{x}) = \left| \phi(D \circ \mathbf{x}) - \sum_{j=1}^{4} \phi\left(T_j \circ \mathbf{x}\right) \right|^2$$

But this loss has $\phi(z) = 0, \forall z$, as its trivial solution. Therefore, the authors use a contrastive loss instead to enforce the counting feature to be different between two randomly chosen different images.

$$\ell_{\text{con}}(\mathbf{x}, \mathbf{y}) = \left|\phi(D \circ \mathbf{x}) - \sum_{j=1}^{4} \phi\left(T_j \circ \mathbf{x}\right)\right|^2$$
$$+ \max\left\{0, M - \left|\phi(D \circ \mathbf{y}) - \sum_{j=1}^{4} \phi\left(T_j \circ \mathbf{x}\right)\right|^2\right\}$$

The network architecture is:



The activation of the last layer (i.e., the output feature representation) is called the counting vector. Each dimension of the counting vector denotes a visual primitive desired to count.

【Experiment】

1) Fine-tuning on PASCAL

Method	Ref	Class.	Det.	Segm.
Supervised [20]	[43]	79.9	56.8	48.0
Random	[33]	53.3	43.4	19.8
Context [9]	[19]	55.3	46.6	-
Context [9]*	[19]	65.3	51.1	-
Jigsaw [30]	[30]	67.6	**53.2**	37.6
ego-motion [1]	[1]	52.9	41.8	-
ego-motion [1]*	[1]	54.2	43.9	-
Adversarial [10]*	[10]	58.6	46.2	34.9
ContextEncoder [33]	[33]	56.5	44.5	29.7
Sound [31]	[44]	54.4	44.0	-
Sound [31]*	[44]	61.3	-	-
Video [41]	[19]	62.8	47.4	-
Video [41]*	[19]	63.1	47.2	-
Colorization [43]*	[43]	65.9	46.9	35.6
Split-Brain [44]*	[44]	67.1	46.7	36.0
ColorProxy [22]	[22]	65.9	-	**38.0**
WatchingObjectsMove [32]	[32]	61.0	52.2	-
Counting		**67.7**	51.4	36.6

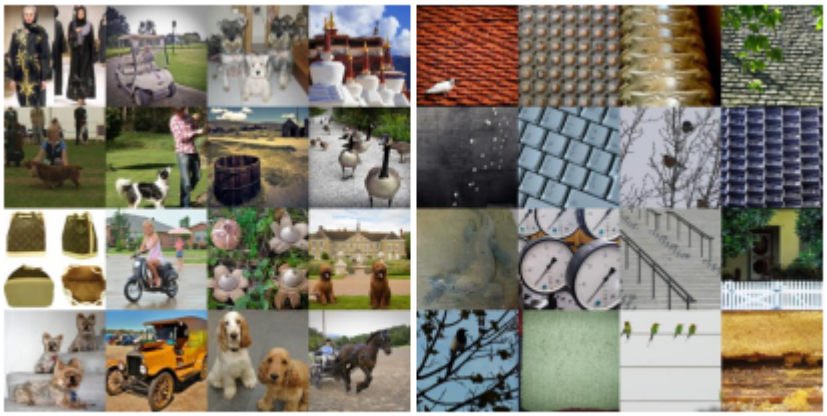
The counting method does not improve much compared to Jigsaw.

2) Ablation Study

Interpolation method	Training size	Color space	Counting dimension	Detection performance
Mixed	1.3M	RGB/Gray	20	50.9
Mixed	128K	Gray	1000	44.9
Mixed	512K	Gray	1000	49.1
Mixed	1.3M	RGB	1000	48.2
Mixed	1.3M	Gray	1000	50.4
Linear	1.3M	RGB/Gray	1000	48.4
Cubic	1.3M	RGB/Gray	1000	48.9
Area	1.3M	RGB/Gray	1000	49.2
Lanczos	1.3M	RGB/Gray	1000	47.3
Mixed	1.3M	RGB/Gray	1000	**51.4**

- The counting method is not sensitive to the dimension of counting vector. (Seeing from the first and last row of the table.)
- The counting method is sensitive to the size of the training set. A large training set is required.
- **Preventing shortcuts**:
  - Mixed interpolation methods: prevent the model from identifying the downsampling image or learning some other artifacts from specific downsampling methods;
  - Mixing RGB / Gray images: prevent the model telling tiles apart from downsampled images by using chromatic aberration.

3) Visualization



(a) (b)

Images with highest magnitude of the counting vector have rich visual primitives (In Fig.(a)), while images with lowest magnitude of the counting vector have some superficial textures (In Fig.(b)).