Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles
(2017)

Mehdi Noroozi and Paolo Favaro

Notes

Contributions

In a self supervised setting, where we define a pretext task by exploiting the freely available labels that can be extracted from within the visual data, and then train the model in a supervised manner. The authors introduce *solving jigsaw puzzles* as a pretext task. Where the model is presented with a random permutation of 9 patches extracted from an image, and the objective to is predict the correct order. Such a task forces the system to learn the parts an object is made of, and task will hopefully build transferable features for other visual tasks.

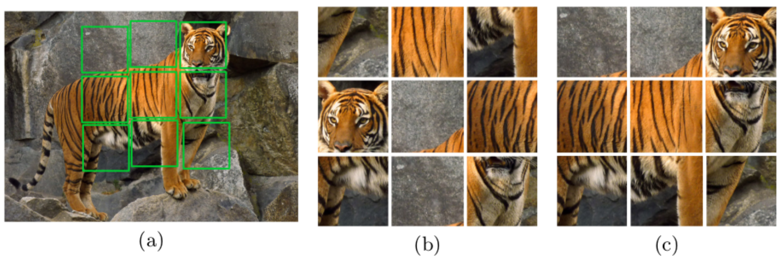

Fig. 1: Learning image representations by solving Jigsaw puzzles. (a) The image from which the tiles (marked with green lines) are extracted. (b) A puzzle obtained by shuffling the tiles. Some tiles might be directly identifiable as object parts, but others are ambiguous (*e.g.*, have similar patterns) and their identification is much more reliable when all tiles are jointly evaluated. In contrast, with reference to (c), determining the relative position between the central tile and the top two tiles from the left can be very challenging [10].

Method

Model A simple design of a network to solve a jigsaw puzzle is to stack the patches over the channel dimension, and feed the new tensor of size `[Batches, 3 x 9, H, W]` to the network. The problem with this approach is the sharing of low level statistics across the patches to solve the puzzle, which does not require any high level understanding of the global appearance. A good alternative is to use the same network over each patch separately. First we pass each patch through the `conv` layers and the first `FC` layer of the network, and then stack all the 9 resulting features and feed them to the last `FC` layers and the classifier. This way we force the network to learn features that are representative and discriminative of each patch that will help the final classification. This architecture is called *context-free network* (CFN).

Shortcuts for solving Jigsaw puzzles CFN can be seen as probability density function over possible permutations S given an the input patches $(A_1 ... A_9)$. Which can be expressed as the pdf over S given the features $p(S|F_1, F_2, \ldots, F_9)$ (modeled by last `FC` layers) and the features computed for each input patch (computed by the `conv` layers) $p(F_i|A_i)$.

$$p\left(S|A_1, A_2, \ldots, A_9\right) = p\left(S|F_1, F_2, \ldots, F_9\right) \prod_{i=1}^{9} p\left(F_i|A_i\right)$$

To solve such a task, the network can take a number of *shortcuts*. If we generate one Jigsaw puzzle per image, the network can encode the 2D position of each patch in the features F_i, and cluster the patches based only of their position and not their content. In this case the tile position of a given patch L_i only depends on its corresponding features F_i: $p\left(L_1, \ldots, L_9|F_1, F_2, \ldots, F_9\right) = \prod_{i=1}^{9} p\left(L_i|F_i\right)$.

Solutions

- To avoid mapping the appearance to an absolute position, we create multiple Jigsaw puzzles per images. For each image we sample 69 possible permutations from 1000 predefined ones, given 9 patches, we have 9! possible permutations. We select the 1000 permutation with the largest Hamming distance between each possible pair. This is done by first creating a matrix of all 9! permutations and a matrix of selected permutation (empty at first), sampling one at each time, calculating the Hamming distance between the selected permutations and the rest of 9! permutations, and adding the permutation with the maximum Hamming distance to the selected permutations.
- To prevent the usage of low level features. The inputs images are resized to 256, cropped to 225 x 225 and split into 3 x 3 grid, each grid cell is of size 75 x 75 patches, and then we randomly select a patch of size 64 x 64 from each grid cell.
- To reduce Chromatic Aberration (using the position of the center of the camera that can be detected from the colors to solve the task). The network is trained on both colored and gray scales images (the dataset contains 30% of gray scale images and the rest is colored). The colors are also jittered by up +/- 2 pixels.

Results

Transfer Learning The CFN weights are used to initialize all the `conv` layers of a standard AlexNet.Then the network in trained on down stream tasks.

Table 1: Results on PASCAL VOC 2007 Detection and Classification. The results of the other methods are taken from Pathak *et al.* [30].

Method	**Pretraining time**	**Supervision**	**Classification**	**Detection**	**Segmentation**
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Wang and Gupta[39]	1 week	motion	58.4%	44.0%	-
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%	-
Pathak *et al.* [30]	14 hours	context	56.5%	44.5%	29.7%
Ours	2.5 days	context	**67.6%**	**53.2%**	**37.6%**

Table 2: Comparison of classification results on ImageNet 2012 [9]. The numbers are obtained by averaging 10 random crops predictions.

	🔒 conv1	🔒 conv2	🔒 conv3	🔒 conv4	🔒 conv5
CFN	**54.7**	**52.8**	**49.7**	45.3	**34.6**
Doersch *et al.* [10]	53.1	47.6	48.7	**45.6**	30.4
Wang and Gupta [39]	51.8	46.9	42.8	38.8	29.8
Random	48.5	41.0	34.8	27.1	12.0

Table 3: Transfer learning of AlexNet from a classification task to the Jigsaw puzzle reassembly problem. The j-th column indicates that all layers from `conv1` to `conv-`j were locked and all subsequent layers were randomly initialized and retrained. Notice how the first 4 layers provide very good features for solving puzzles. This shows that object classification and the Jigsaw puzzle problems are related.

	🔒 conv1	🔒 conv2	🔒 conv3	🔒 conv4	🔒 conv5
AlexNet [25]	88	87	86	83	74

Table 4: Ablation study on the impact of the permutation set.

Number of permutations	**Average hamming distance**	**Minimum hamming distance**	**Jigsaw task accuracy**	**Detection performance**
1000	8.00	2	71	**53.2**
1000	6.35	2	62	51.3
1000	3.99	2	54	50.2
100	8.08	2	88	52.6
95	8.08	3	90	52.4
85	8.07	4	91	52.7
71	8.07	5	92	52.8
35	8.13	6	94	52.6
10	8.57	7	97	49.2
7	8.95	8	98	49.6
6	9	9	99	49.7

Table 5: Ablation study on the impact of the shortcuts.

Gap	**Normalization**	**Color jittering**	**Jigsaw task accuracy**	**Detection performance**
✗	✓	✓	98	47.7
✓	✗	✓	90	43.5
✓	✓	✗	89	51.1
✓	✓	✓	88	52.6

Ablation Studies