Self-Supervised Feature Learning by Learning to Spot Artifacts

Wonbin Kim

Self-Supervised Learning

- “To exploit different labelings that are freely available besides or within visual data, and to use them as intrinsic reward signals to learn general-purpose features.”[5]

Object Detection, Semantic Segmentation, Classification

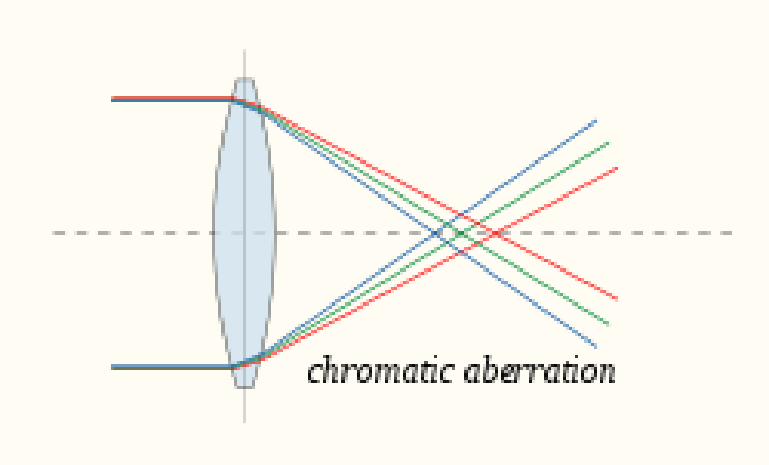
Shortcut

- "Trivial" and undesirable Solution to Self-Supervised Learning
  - Low-level statistics
    - Boundary Pattern or Texture
    - Edge continuity
    - The pixel intensity/color distribution
  - Chromatic aberration

Wikipedia : Industrial_National_Bank_Building

Chromatic aberration

- It cause different wavelengths of light to have differing focal lengths.



Wikipedia : Chromatic aberration

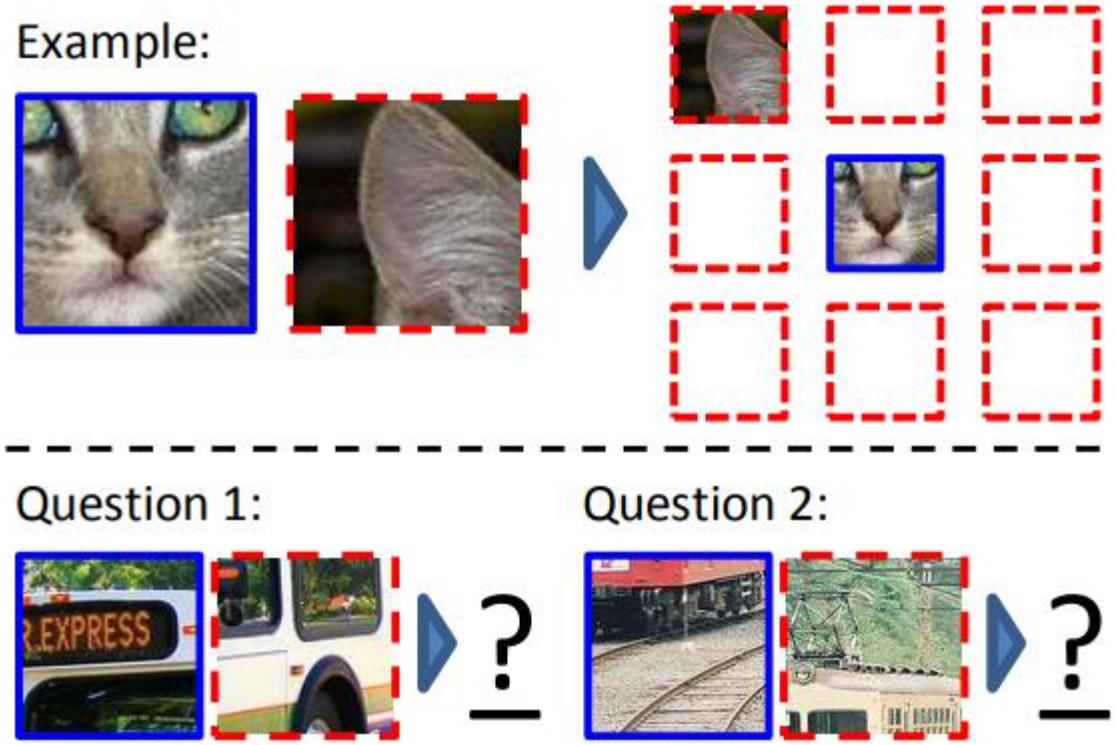
Motivation

- What words will come to the blank?

"There is nothing noble in being superior to your fellow man; true nobility is being _____ to your former self."

- Ernest Hemingway

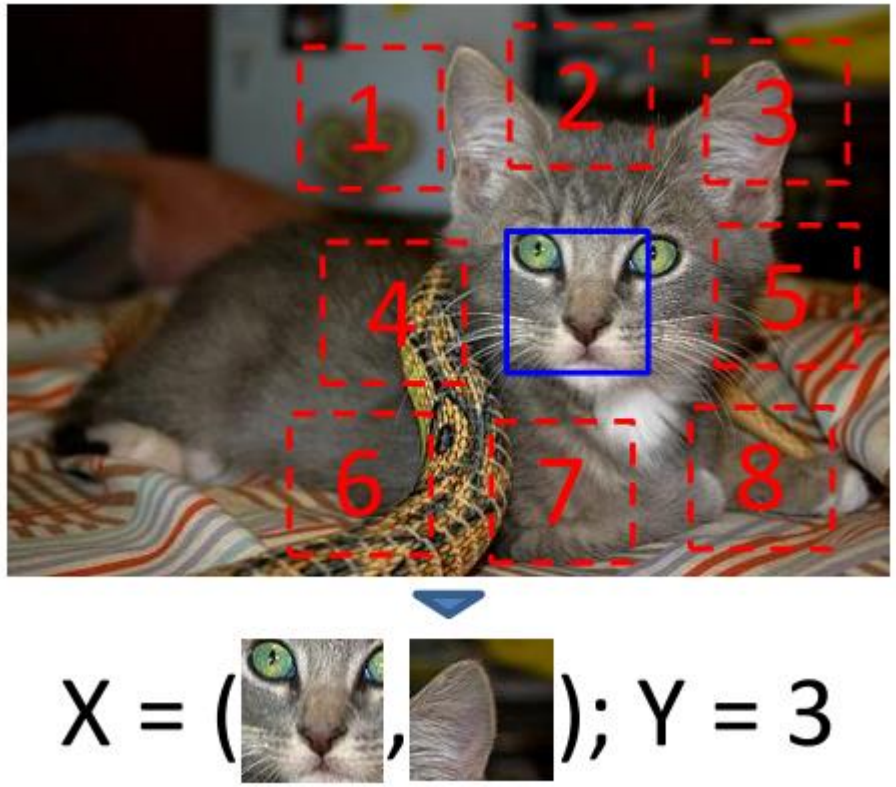
1. Context prediction



Carl Doersch et al., "Unsupervised visual representation learning by context prediction.", ICCV, (2015)

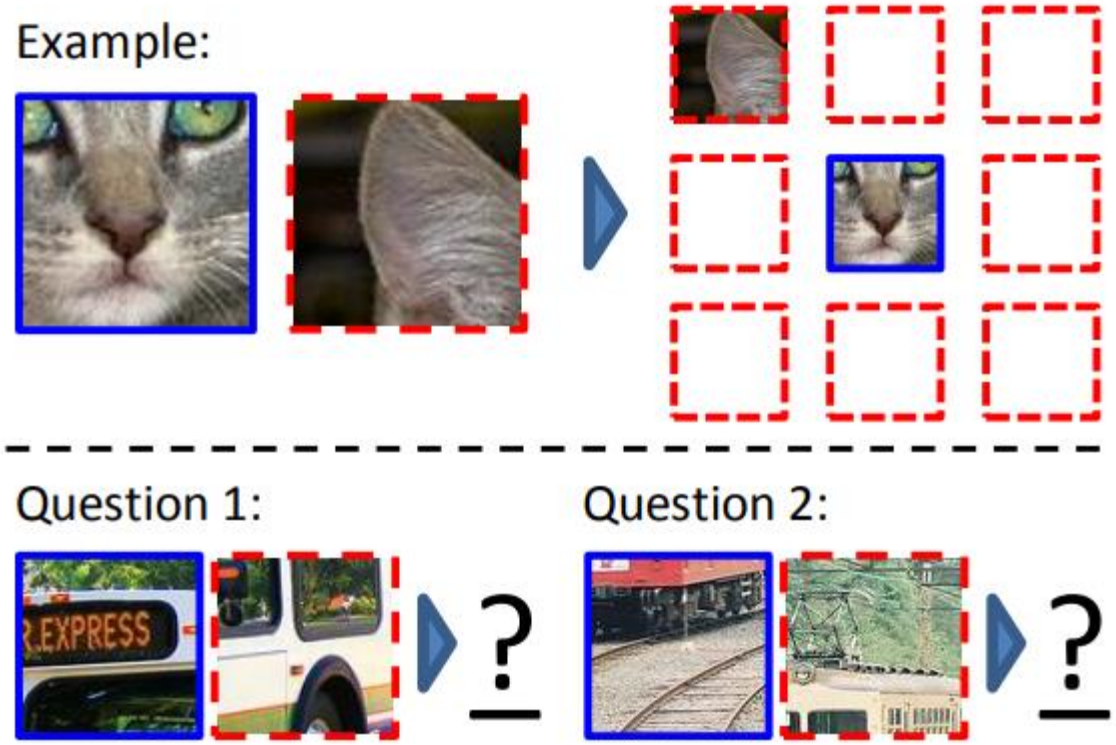
1. Context prediction – avoiding shortcuts

- Sampling with gap and jittering
- Color dropping



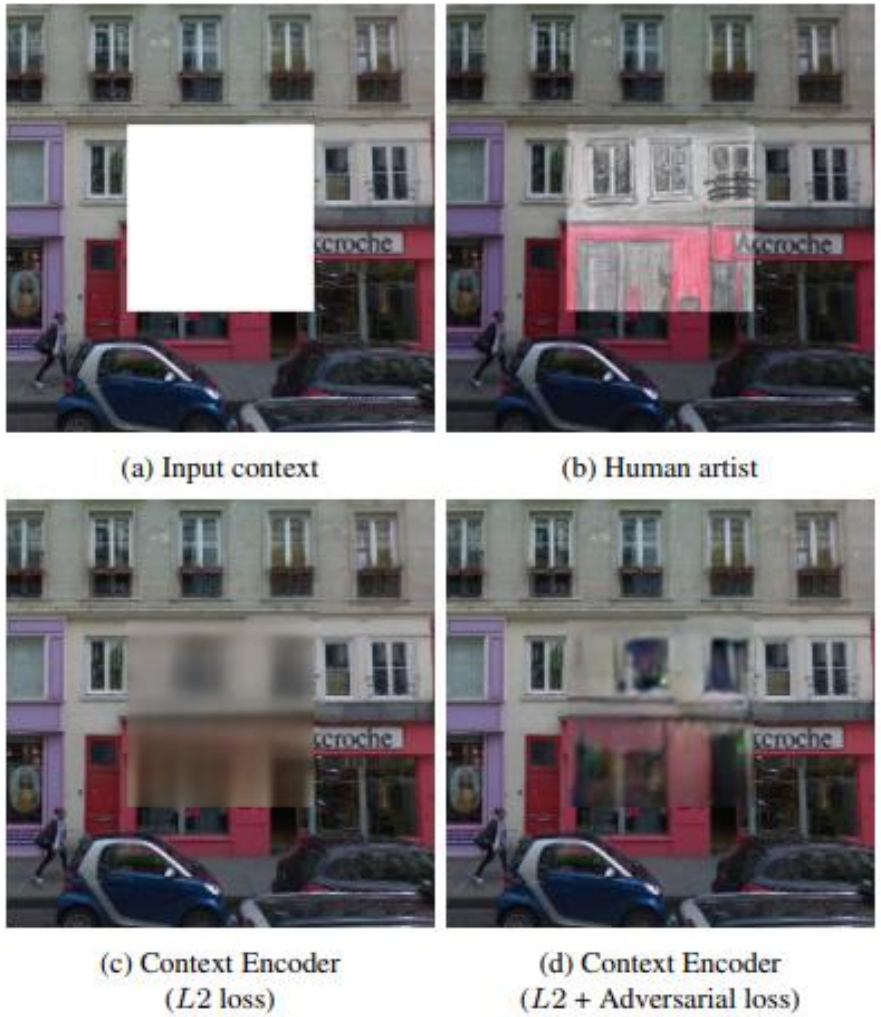
Carl Doersch et al., "Unsupervised visual representation learning by context prediction.", ICCV, (2015)

1. Context prediction[6]



Carl Doersch et al., "Unsupervised visual representation learning by context prediction.", ICCV, (2015)

2. Inpainting



(a) Input context (b) Human artist

(c) Context Encoder ($L2$ loss) (d) Context Encoder ($L2$ + Adversarial loss)

Deepak Pathak et al. "Context encoders: Feature learning by inpainting." ICCV, (2016)

2. Inpainting

- Loss function
  - Reconstruction loss

$$\mathcal{L}_{rec}(x) = \lambda_{rec} \left\| \widehat{M} \odot \left(x - F\left(\left(1 - \widehat{M} \right) \odot x \right) \right) \right\|_2^2$$

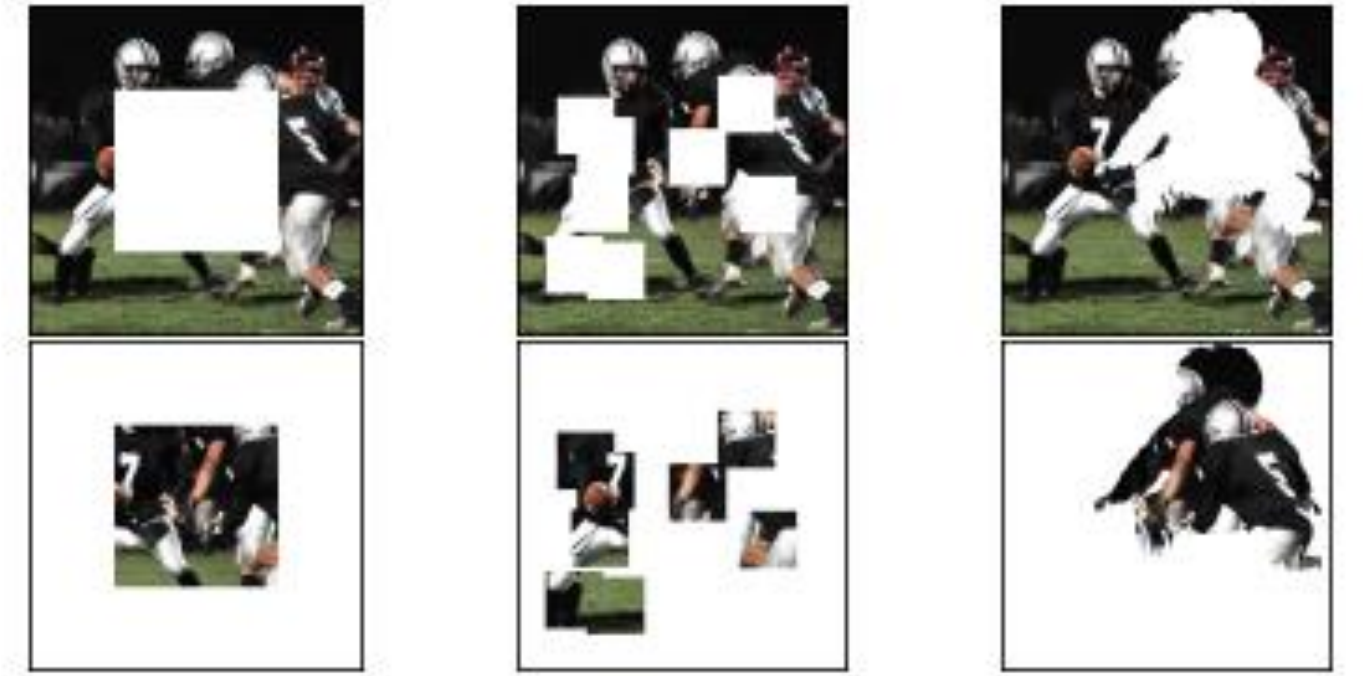
  - Adversarial loss

$$\mathcal{L}_{adv} = \max_D \mathbb{E}_{x \in \chi} \left[\log D(x) + \log \left(1 - D\left(F\left(\left(1 - \widehat{M} \right) \odot \right) \right) \right) \right]$$

  - Aggregated Loss

$$\mathcal{L} = \lambda_{rec} \mathcal{L}_{rec} + \lambda_{adv} \mathcal{L}_{adv}$$

2. Inpainting



(a) Central region (b) Random block (c) Random region

Deepak Pathak et al. "Context encoders: Feature learning by inpainting." ICCV, (2016)

2. Inpainting

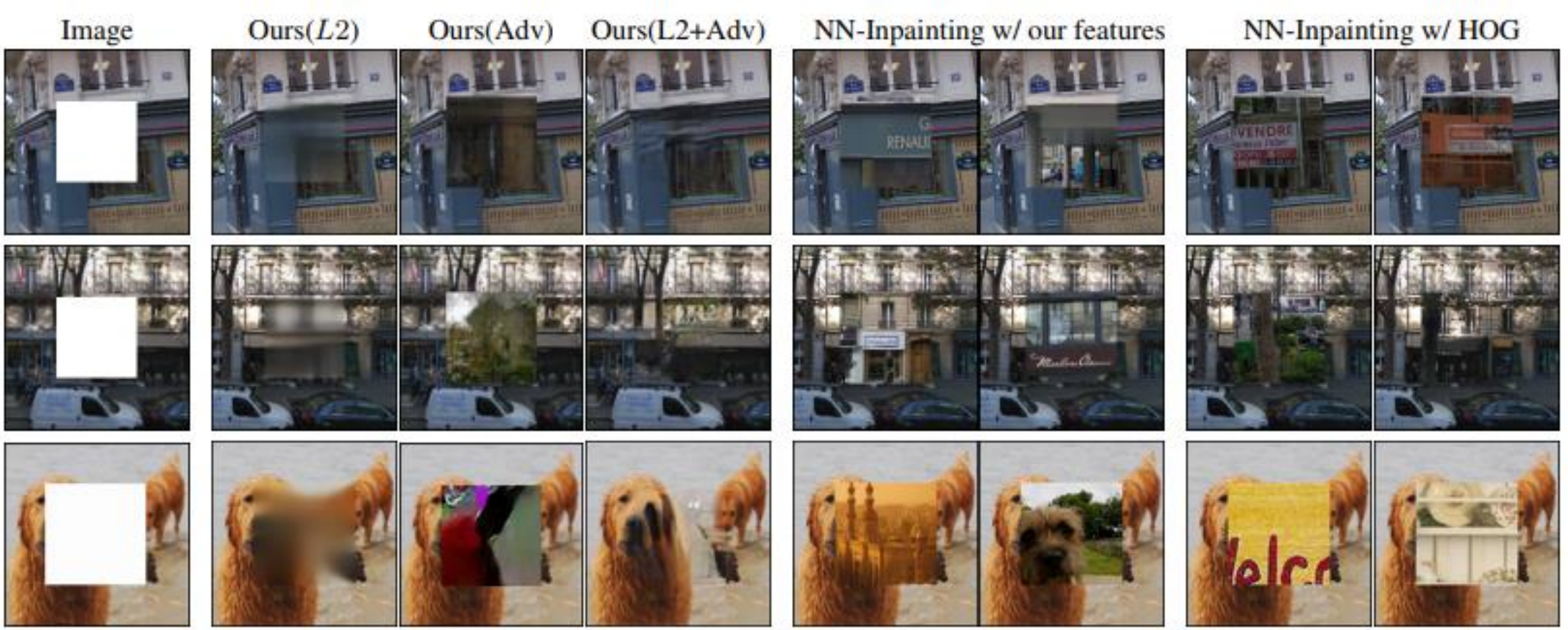


Figure 6: Semantic Inpainting using different methods on *held-out* images. Context Encoder with just L2 are well aligned, but not sharp. Using adversarial loss, results are sharp but not coherent. Joint loss alleviate the weaknesses of each of them. The last two columns are the results if we plug-in the best nearest neighbor (NN) patch in the masked region.

Deepak Pathak et al. "Context encoders: Feature learning by inpainting." ICCV, (2016)

2. Inpainting

Figure 8: Context Nearest Neighbors. Center patches whose context (not shown here) are close in the embedding space of different methods (namely our context encoder, HOG and AlexNet). Note that the appearance of these center patches themselves was never seen by these methods. But our method brings them close just from their context.

Deepak Pathak et al. "Context encoders: Feature learning by inpainting." ICCV, (2016)

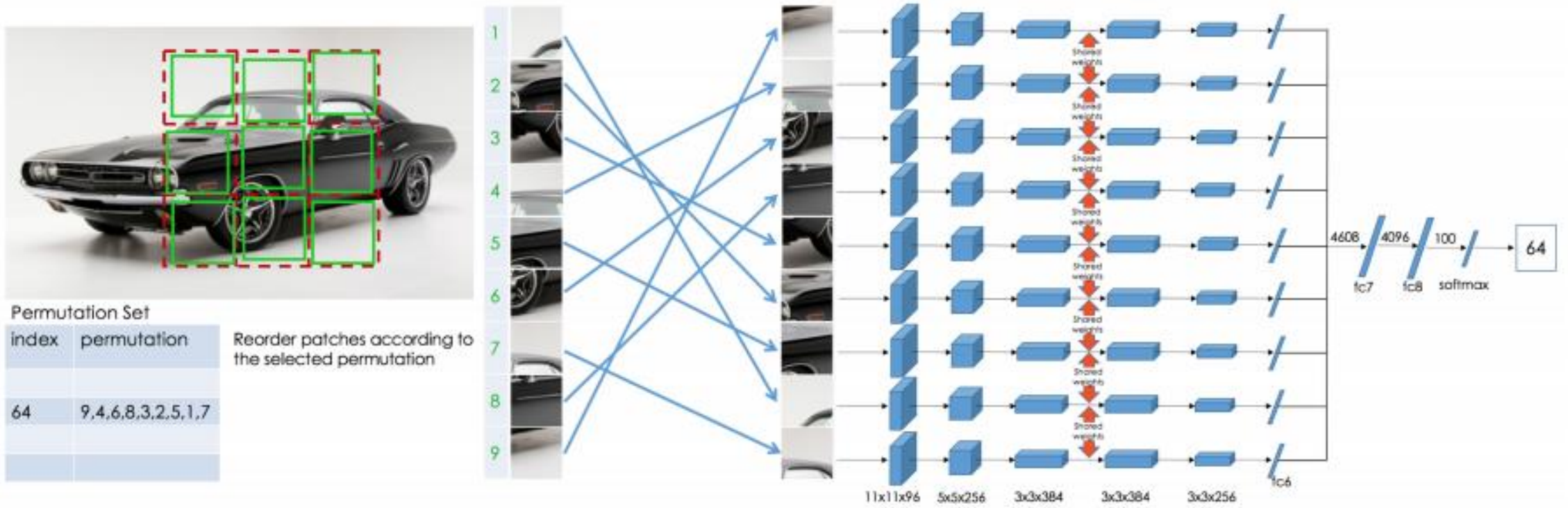
2. Inpainting

Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Wang *et al.* [39]	motion	1 week	58.7%	47.4%	-
Doersch *et al.* [7]	relative context	4 weeks	55.3%	46.6%	-
Ours	context	14 hours	56.5%	44.5%	30.0%

Table 2: Quantitative comparison for classification, detection and semantic segmentation. Classification and Fast-RCNN Detection results are on the PASCAL VOC 2007 test set. Semantic segmentation results are on the PASCAL VOC 2012 validation set from the FCN evaluation described in Section 5.2.3, using the additional training data from [18], and removing overlapping images from the validation set [28].

Deepak Pathak et al. "Context encoders: Feature learning by inpainting." ICCV, (2016)

3. Solving Jigsaw puzzle



Mehdi Noroozi et el., "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.

3. Solving Jigsaw puzzle

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

Mehdi Noroozi et el., "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.

3. Solving Jigsaw puzzle

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

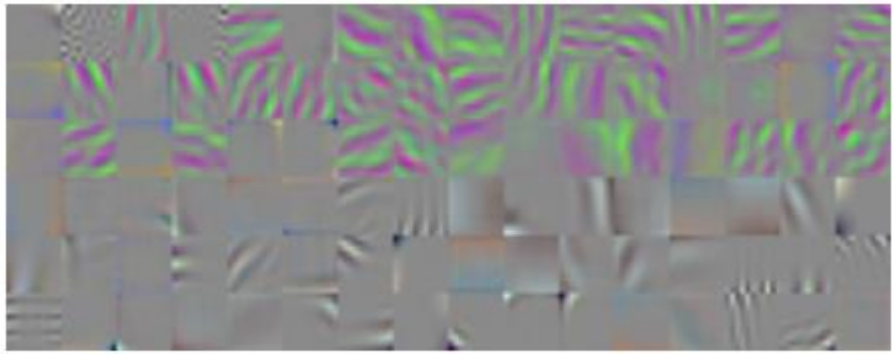
Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Wang and Gupta[39]	1 week	motion	58.4%	44.0%	-
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%	-
Pathak *et al.* [30]	14 hours	context	56.5%	44.5%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

Mehdi Noroozi et el., "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.

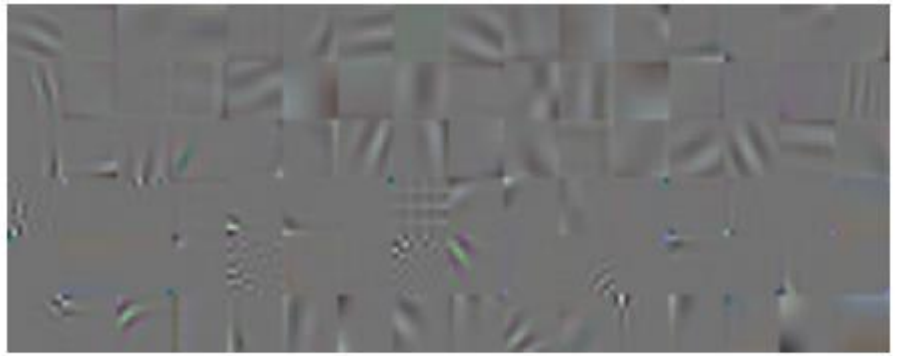
3. Solving Jigsaw puzzle

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



(f) conv1 filters without color jittering



(g) conv1 filters with color jittering

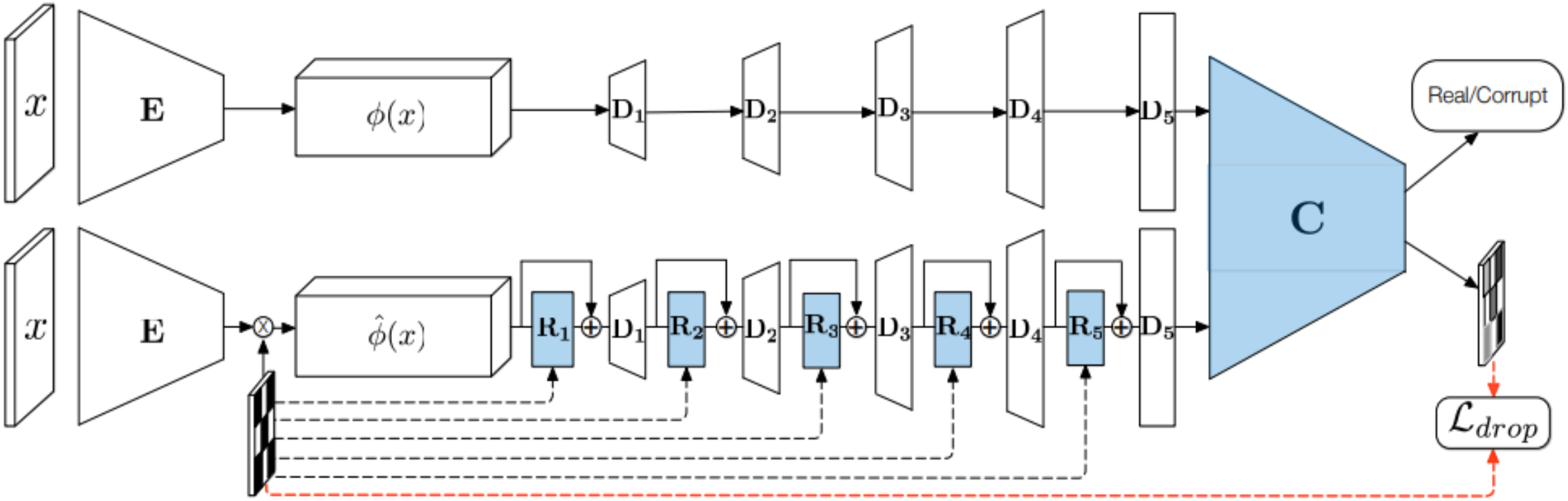
Mehdi Noroozi et el., "Unsupervised learning of visual representations by solving jigsaw puzzles." ECCV, 2016.

3. Distortion

Figure 1. A mixture of real images (green border) and images with synthetic artifacts (red border). Is a good object representation necessary to tell them apart?

Simon Jenni et el., "Self-Supervised Feature Learning by Learning to Spot Artifacts." CVPR (2018).

4. Distortion



Simon Jenni et el., "Self-Supervised Feature Learning by Learning to Spot Artifacts." CVPR (2018).

4. Distortion

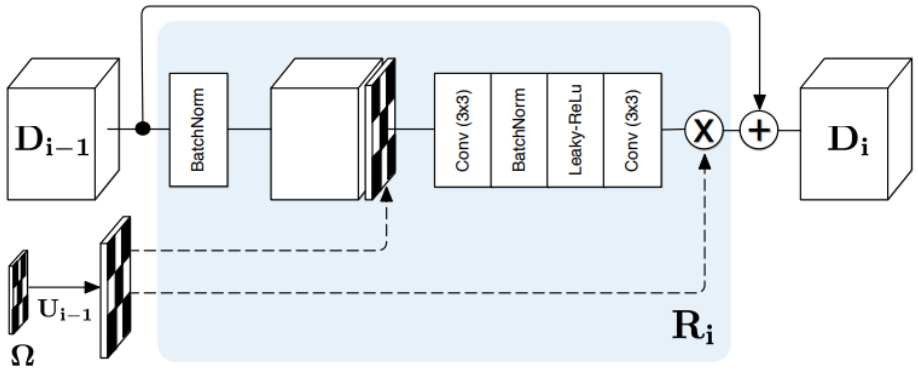
- Decoder with Repair Network.
  - i=1

$$D_1 = \phi(x) + (1 - \Omega) \odot R_1\left(\hat{\phi}(x)\right),$$

Where $\hat{\phi}(\mathrm{x}) = \Omega \odot \phi(\mathrm{x}) + (1 - \Omega) \odot \left(u * \phi(x)\right)$

  - i=2, 3, 4

$$D_i = D_{i-1} + \left(1 - U_{i-1}(\Omega)\right) \odot R_i(D_{i-1})$$



Simon Jenni et el., "Self-Supervised Feature Learning by Learning to Spot Artifacts." CVPR (2018).

4. Distortion

- Loss function
  - Discriminator adversarial Loss : {real , corruption}

$$\mathcal{L}_{class} = \min_{R} \max_{C} \sum_{\mathbf{x} \sim \mathrm{p}(\mathbf{x})} \log C^{class}\left(D\left(\phi(\mathbf{x})\right)\right) + \log\left(1 - C^{class}\left(\widehat{D}\left(\widehat{\phi}\,(\mathbf{x})\right)\right)\right).$$

  - Discriminator binary classification Loss : mask prediction

$$\mathcal{L}_{mask} = \min_{C} \sum_{\hat{\mathbf{x}}} \sum_{ij} \Omega_{ij} \log \sigma\left(C_{ij}^{mask}(\hat{\mathbf{x}})\right) + \left(1 - \Omega_{ij}\right) \log\left(1 - \sigma\left(C_{ij}^{mask}\left(\widehat{D}\left(\widehat{\phi}\,(\mathbf{x})\right)\right)\right)\right)$$

Simon Jenni et el., "Self-Supervised Feature Learning by Learning to Spot Artifacts." CVPR (2018).

4. Distortion

Table 1. Influence of different architectural choices on the classification accuracy on STL-10 [4]. Convolutional layers were pre-trained on the proposed self-supervised task and kept fixed during transfer for classification.

	Ablation experiment	Accuracy
	Baseline (dropping rate = 0.5)	79.94%
(a)	Input image as real	74.99%
(b)	Distributed vs. local repair network	77.51%
(c)	Dropping rate = 0.1	70.92%
(d)	Dropping rate = 0.3	76.26%
(e)	Dropping rate = 0.7	81.06%
(f)	Dropping rate = 0.9	79.60%
(g)	Without mask prediction	78.44%
(h)	3×3 encoder convolutions	79.84%
(i)	No gating in repair layers	79.66%
(j)	No history of corrupted examples	79.76%
(k)	No repair network	54.74%
(l)	GAN instead of damage & repair	56.59%

Figure 6. The *damage & repair* network renderings. The left-most column shows the input image. Column (a) shows results when input images are used as real examples. Note that this introduces commonly observed GAN artifacts. Column (b) shows results with the local instead of distributed repair network. Columns (c)-(f) show results with dropping rates of 0.1, 0.3, 0.7 and 0.9.

Simon Jenni et el., "Self-Supervised Feature Learning by Learning to Spot Artifacts." CVPR (2018).

Reference

- [1] Jenni, Simon, and Paolo Favaro. "Self-Supervised Feature Learning by Learning to Spot Artifacts." CVPR (2018).
- [2] Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
- [3] Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." European Conference on Computer Vision. Springer, Cham, 2016.
- [4] Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE International Conference on Computer Vision. 2015.
- [5] Wikipedia : Chromatic aberration; https://en.wikipedia.org/wiki/Chromatic_aberration