The Unreasonable Effectiveness of Patches in Deep Convolutional Kernels Methods.

Louis Thiry[1] , Michael Arbel[2], Eugene Belilovsky[3], Edouard Oyallon[4]
[1]*Departement of Computer Science, DATA Team, ENS, CNRS, PSL*
[2]*Gatsby Computational Neuroscience Unit, UCL*
[3]*Concordia University and Mila Montreal*
[4] *LIP6, Sorbonne Université, CNRS*

Plan

1. Introduction
2. Convolutional kernel methods
3. Our method
4. Results

Introduction

Trends in convolutional kernel methods

Introduction

Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020)

Introduction
Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) : $87 - 90\%$ on CIFAR-10.

Introduction
Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) : $87 - 90\%$ on CIFAR-10.
- Kernels are **data-driven**

Introduction

Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) : $87 - 90\%$ on CIFAR-10.
- Kernels are **data-driven**
- Key ingredient: **whitening**.

Introduction
Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) : $87 - 90\%$ on CIFAR-10.
- Kernels are **data-driven**
- Key ingredient: **whitening**.
- No (published) results on ImageNet.

Introduction

Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) : $87 - 90\%$ on CIFAR-10.
- Kernels are **data-driven**
- Key ingredient: **whitening**.
- No (published) results on ImageNet.

Contributions

- Simple convolutional kernel method: K-nearest-neighbors encoding, Mahanalobis distance, linear kernel.

Introduction
Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) : $87 - 90\%$ on CIFAR-10.
- Kernels are **data-driven**
- Key ingredient: **whitening**.
- No (published) results on ImageNet.

Contributions

- Simple convolutional kernel method: K-nearest-neighbors encoding, Mahanalobis distance, linear kernel.
- Comparable accuracies on CIFAR-10 with shallow classifier.

Introduction

Trends in convolutional kernel methods

- Recent competitive convolutional kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) : $87 - 90\%$ on CIFAR-10.
- Kernels are **data-driven**
- Key ingredient: **whitening**.
- No (published) results on ImageNet.

Contributions

- Simple convolutional kernel method: K-nearest-neighbors encoding, Mahanalobis distance, linear kernel.
- Comparable accuracies on CIFAR-10 with shallow classifier.
- Scalable to ImageNet: S.O.T.A. as non-learned visual representation.

Plan

Convolutional kernel methods

x, y images.

$$K_{k,\Phi,\mathcal{X}}(x,y) = k(\Phi_{\mathcal{X}} L_{\mathcal{X}} x, \Phi_{\mathcal{X}} L_{\mathcal{X}} y)$$

Convolutional kernel methods

x, y images.

$$K_{k,\Phi,\mathcal{X}}(x,y) = k(\Phi_{\mathcal{X}} L_{\mathcal{X}} x, \Phi_{\mathcal{X}} L_{\mathcal{X}} y)$$

Ingredients:

Convolutional kernel methods

x, y images.

$$K_{k,\Phi,\mathcal{X}}(x, y) = k(\Phi_{\mathcal{X}} L_{\mathcal{X}} x, \Phi_{\mathcal{X}} L_{\mathcal{X}} y)$$

Ingredients:

- Training data

$$\mathcal{X}$$

Convolutional kernel methods

x, y images.

$$K_{k,\Phi,\mathcal{X}}(x, y) = k(\Phi_{\mathcal{X}} L_{\mathcal{X}} x, \Phi_{\mathcal{X}} L_{\mathcal{X}} y)$$

Ingredients:

- Training data

$$\mathcal{X}$$

- Shift and rescale (e.g. whitening) operator

$$L_{\mathcal{X}}$$

Convolutional kernel methods

x, y images.

$$K_{k,\Phi,\mathcal{X}}(x, y) = k(\Phi_{\mathcal{X}} L_{\mathcal{X}} x, \Phi_{\mathcal{X}} L_{\mathcal{X}} y)$$

Ingredients:

- Training data

$$\mathcal{X}$$

- Shift and rescale (e.g. whitening) operator

$$L_{\mathcal{X}}$$

- Representation

$$\Phi_{\mathcal{X}}$$

Convolutional kernel methods

x, y images.

$$K_{k,\Phi,\mathcal{X}}(x, y) = k(\Phi_{\mathcal{X}} L_{\mathcal{X}} x, \Phi_{\mathcal{X}} L_{\mathcal{X}} y)$$

Ingredients:

- Training data

$$\mathcal{X}$$

- Shift and rescale (e.g. whitening) operator

$$L_{\mathcal{X}}$$

- Representation

$$\Phi_{\mathcal{X}}$$

- Predefined (e.g. Linear, Gaussian, Neural Tangent) kernel

$$k(x, y)$$

Data-driven convolutional kernel methods

$K(x, y)$ is **data-driven** if Φ or L depend on the training set $\mathcal{X}$,
data-independent otherwise.

Data-driven convolutional kernel methods

$K(x, y)$ is **data-driven** if Φ or L depend on the training set $\mathcal{X}$,
data-independent otherwise.

Examples of Data-driven kernels on CIFAR-10

- Random features (Coates et al., 2011; Recht et al., 2019): 85.6 %

Data-driven convolutional kernel methods

$K(x, y)$ is **data-driven** if Φ or L depend on the training set $\mathcal{X}$,
data-independent otherwise.

Examples of Data-driven kernels on CIFAR-10

- Random features (Coates et al., 2011; Recht et al., 2019): 85.6 %
  - L: whitening of patches
  - Φ: shrinked convolutions with random patches of $\mathcal{X}$
  - k: linear kernel

Data-driven convolutional kernel methods

$K(x, y)$ is **data-driven** if Φ or L depend on the training set $\mathcal{X}$,
data-independent otherwise.

Examples of Data-driven kernels on CIFAR-10

- Random features (Coates et al., 2011; Recht et al., 2019): 85.6 %
  - L: whitening of patches
  - Φ: shrinked convolutions with random patches of $\mathcal{X}$
  - k: linear kernel
- Enhanced convolutional NTK (Li et al., 2019): 88.9 %

Data-driven convolutional kernel methods

$K(x, y)$ is **data-driven** if Φ or L depend on the training set $\mathcal{X}$,
data-independent otherwise.

Examples of Data-driven kernels on CIFAR-10

- Random features (Coates et al., 2011; Recht et al., 2019): 85.6 %
  - L: whitening of patches
  - Φ: shrinked convolutions with random patches of $\mathcal{X}$
  - k: linear kernel
- Enhanced convolutional NTK (Li et al., 2019): 88.9 %
  - L and Φ: same as random features
  - k: Neural Tangent kernel (NTK)

Data-driven convolutional kernel methods

$K(x, y)$ is **data-driven** if Φ or L depend on the training set $\mathcal{X}$, **data-independent** otherwise.

Examples of Data-driven kernels on CIFAR-10

- Random features (Coates et al., 2011; Recht et al., 2019): 85.6 %
  - L: whitening of patches
  - Φ: shrinked convolutions with random patches of $\mathcal{X}$
  - k: linear kernel
- Enhanced convolutional NTK (Li et al., 2019): 88.9 %
  - L and Φ: same as random features
  - k: Neural Tangent kernel (NTK)
- Neural Kernels Without Tangents (Shankar et al., 2020): 89.8 %

Data-driven convolutional kernel methods

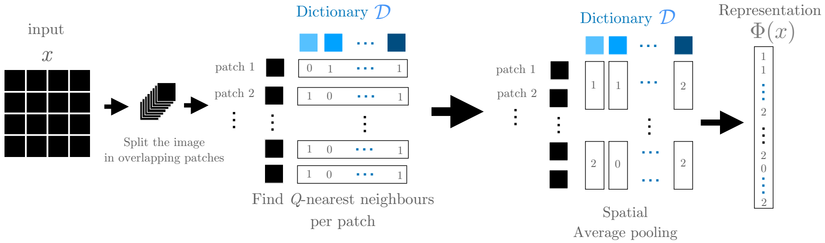
$K(x, y)$ is **data-driven** if Φ or L depend on the training set $\mathcal{X}$,
data-independent otherwise.

Examples of Data-driven kernels on CIFAR-10

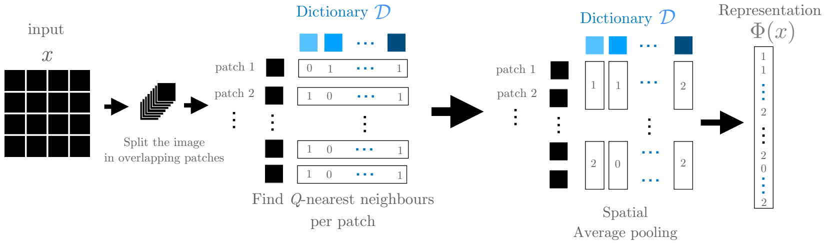
- Random features (Coates et al., 2011; Recht et al., 2019): 85.6 %
  - L: whitening of patches
  - Φ: shrinked convolutions with random patches of $\mathcal{X}$
  - k: linear kernel
- Enhanced convolutional NTK (Li et al., 2019): 88.9 %
  - L and Φ: same as random features
  - k: Neural Tangent kernel (NTK)
- Neural Kernels Without Tangents (Shankar et al., 2020): 89.8 %
  - L: whitening of the image
  - k: Custom *Neural Kernel*

Plan

Our method

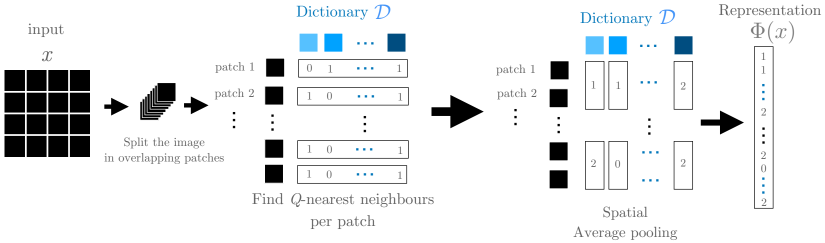

input
x
Split the image
in overlapping patches
Dictionary $\mathcal{D}$
patch 1
patch 2
0 1 ⋯ 1
1 0 ⋯ 1
1 0 ⋯ 1
1 0 ⋯ 1
Find Q-nearest neighbours
per patch
Dictionary $\mathcal{D}$
patch 1
patch 2
1 1 ⋯ 2
2 0 ⋯ 2
Spatial
Average pooling
Representation
$\Phi(x)$
1
1
2
2
0
2

Our method



- x: image viewed as a collection of overlapping patches.

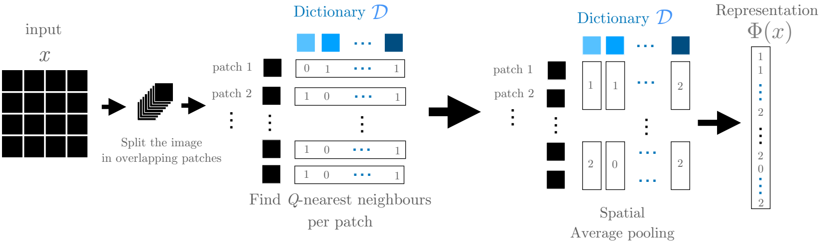
Our method



- x: image viewed as a collection of overlapping patches.
- L: whitening operator

$$L : x \mapsto (\Sigma + \lambda I)^{-1}(x - \mu)$$

Our method

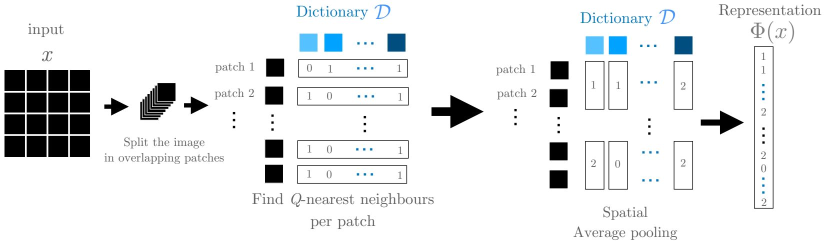


- x: image viewed as a collection of overlapping patches.
- L: whitening operator

$$L : x \mapsto (\Sigma + \lambda I)^{-1}(x - \mu)$$

- Φ: K-nearest-neighbor encoding in a dictionary $\mathcal{D}$ of randomly selected whitened patches.

Our method



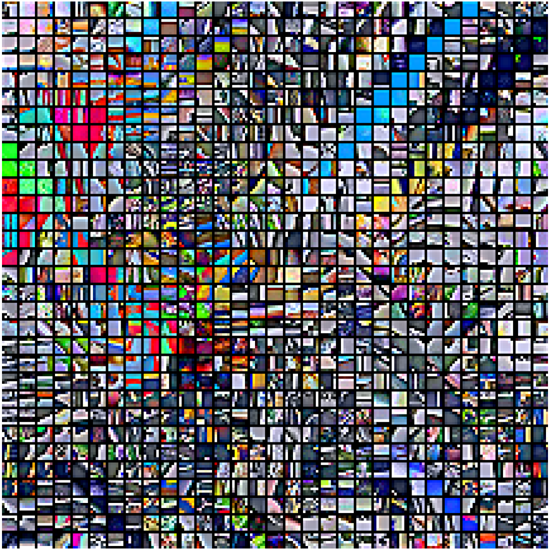
- x: image viewed as a collection of overlapping patches.
- L: whitening operator

$$L : x \mapsto (\Sigma + \lambda I)^{-1}(x - \mu)$$

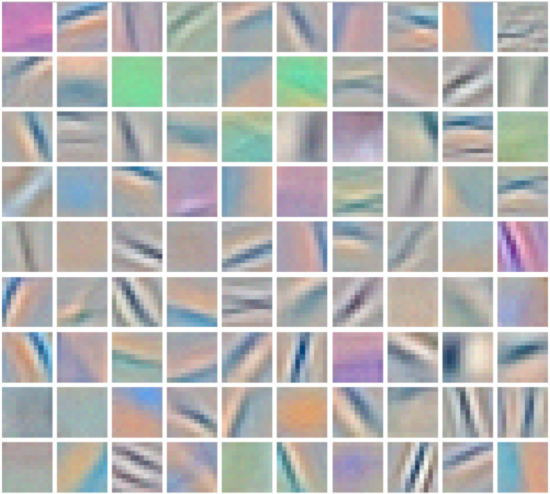
- Φ: K-nearest-neighbor encoding in a dictionary $\mathcal{D}$ of randomly selected whitened patches.
- $k(x, y)$: linear kernel.

CIFAR-10 Dictionary

ImageNet64 Dictionary



First layer of AlexNet



Plan

CIFAR-10

Linear classification

Method	$\lvert\mathcal{D}\rvert$	**VQ**	**Online**	P	**Acc.**
Coates et al. (2011)	$1k$	✓	×	6	68.6
Wavelets (Oyallon et al. 2015)	-	×	×	8	82.2
Recht et al. (2019)	$0.2M$	×	×	6	85.6
SimplePatch (Ours)	$10k$	✓	✓	6	85.6
SimplePatch (Ours)	$60k$	×	✓	6	**86.9**

CIFAR-10

Linear classification

Method	$\lvert\mathcal{D}\rvert$	**VQ**	**Online**	P	**Acc.**
Coates et al. (2011)	$1k$	✓	×	6	68.6
Wavelets (Oyallon et al. 2015)	-	×	×	8	82.2
Recht et al. (2019)	$0.2M$	×	×	6	85.6
SimplePatch (Ours)	$10k$	✓	✓	6	85.6
SimplePatch (Ours)	$60k$	×	✓	6	**86.9**

Non-linear classification

Method	**VQ**	**Depth**	**Classifier**	**Acc.**
SimplePatch (Ours)	✓	2	1-hidden-layer	88.5
AlexNet (Krizhevsky et al. 2012)	×	5	e2e	89.1
NK (Shankar et al. 2020)	×	5	kernel	89.8
CKN (Mairal et al. 2016)	×	9	kernel	89.8

ImageNet

Method	$\|\mathcal{D}\|$	VQ	P	Depth	Res.	Top1	Top5
Random CNN	-	$\times$	-	9	224	18.9	-
Zarka et al. (19)	-	$\times$	32	2	224	26.1	44.7
Ours	$2k$	✓	12	1	128	35.9	57.4
Ours	$2k$	$\times$	12	1	128	36.0	**57.6**

Linear classification

ImageNet

Linear classification

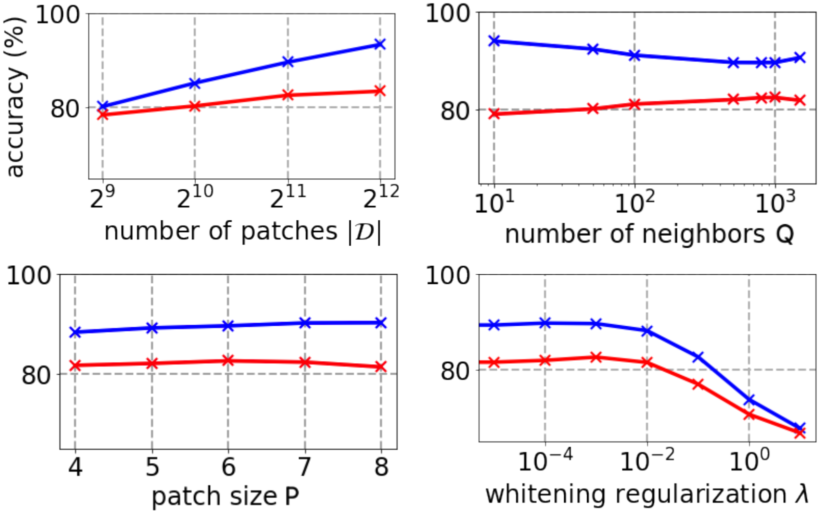
Method	$\lvert\mathcal{D}\rvert$	VQ	P	Depth	Res.	Top1	Top5
Random CNN	-	$\times$	-	9	224	18.9	-
Zarka et al. (19)	-	$\times$	32	2	224	26.1	44.7
Ours	$2k$	✓	12	1	128	35.9	57.4
Ours	$2k$	$\times$	12	1	128	36.0	**57.6**

Non-linear classification

Method	VQ	P	Depth	Res.	Classif.	Top1	Top5
Belilov. al. (18)	$\times$	-	2	224	e2e	-	44
Ours	✓	6	2	64	1-layer	39.4	62.1
Brendel al. (19)	$\times$	9	50	224	e2e	-	70.0

Ablation Study

Train accuracies in blue, test accuracies in red.



Questions ?

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

Zhiyuan Li, Ruosong Wang, Dingli Yu, Simon S Du, Wei Hu, Ruslan Salakhutdinov, and Sanjeev Arora. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

Julien Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in neural information processing systems*, pages 1399–1407, 2016.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Ludwig Schmidt, Jonathan Ragan-Kelley, and Benjamin Recht. Neural kernels without tangents. *arXiv preprint arXiv:2003.02237*, 2020.