ON THE IMPORTANCE OF DATA-DRIVEN FEATURES FOR UNSUPERVISED VISUAL REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A recent line of work showed that various forms of convolutional kernel methods can be competitive with standard supervised deep convolutional networks on datasets like CIFAR-10, obtaining accuracies in the range of $87-90\%$ while being more amenable to theoretical analysis. Here, we highlight that in order to get good performances, these methods rely on a step that extracts features based on data from the training set, like dictionary of patches. Here we extensively study its effect, demonstrating it is the key ingredient for high performance of these methods. Specifically, we show that one of the simplest data-driven kernel, a method based on a single layer of dictionary of patches, combined solely with a linear classifier is already obtaining classification accuracies in this range on CIFAR-10. We scale this method to the challenging ImageNet dataset, showing such a simple approach can exceed all existing non-learned representation methods. This is a new baseline for object recognition without representation learning methods, that initiates the investigation of convolutional kernel models on ImageNet. We conduct experiments to analyze the dictionary that we used, our ablations showing that whitened patches are a key ingredient for the success of this type of method.

1 INTRODUCTION

Understanding the success of deep convolutional neural networks on images has received a plethora of interest from the machine learning community. This is challenging because images are high-dimensional signals and deep neural networks are highly-non linear models with a substantial amount of parameters: yet, the curse of dimensionality is seemingly avoided by these models. One approach taken by several authors (Mairal, 2016; Li et al., 2019; Shankar et al., 2020; Lu et al., 2014) has been to construct simpler models that can achieve similar performance with a model that has more tractable analytical properties (Jacot et al., 2018; Rahimi and Recht, 2008), but at the same time shares various elements with standard deep learning models. In general, these works are able to achieve reasonable performance on the CIFAR-10 dataset but lack of large scale experiments on datasets such as ImageNet.

Our work is primarily motivated by this recent line of research which focuses on convolutional kernels methods. All these works rely possibly on a common pre-processing step to construct a similarity measure at the image patch level, and we emphasize that there is a significant gap of performances if this preprocessing step is not used. We thus investigate to what extent this common step is responsible for the success of those methods. Without clear ablation experiments, it remains difficult to know which of engineering choices here are important, between the kernel design and the patch similarity measure. In our work, we decompose and analyze each step of our feature design, on gold-standard datasets and find that a method based solely on patches K-Nearest Neighbors encoding in a dictionary of randomly selected patches is actually a strong baseline for image classification. Those findings are aligned with empirical studies that relate deep learning models decisions with visual interpretations at small and large patch level (Zeiler and Fergus, 2014; Brendel et al., 2019).

While the literature provides a detailed analysis of the behavior of a dictionary of patches for image compression (Wallace, 1992), texture synthesis (Efros and Leung, 1999) or image inpainting (Criminisi et al., 2004), we have a limited knowledge and understanding of it in the context of image classification. For instance, in the context of image compression, it is known that patches can be modeled as locally stationary processes obtained from images, and it can be empirically verified that

they are sparsified in a wavelet basis (Mallat, 1999). However, the behavior of those dictionaries of patches in some classification methods is still not well understood, despite often being the very first component of many classic vision pipelines (Perronnin et al., 2010; Lowe, 2004; Brendel et al., 2019; Oyallon et al., 2018b).

We investigate the effect of patch-based pre-processing for image classification through a simple baseline representation that does not involve learning (up to a linear classifier) on both CIFAR-10 and ImageNet datasets: the path from CIFAR-10 to ImageNet had never been explored until now in this context. Thus, we believe our baseline to be of high interest for understanding non-deep learning methods, which almost systematically rely on a patch (or descriptor of a patch) encoding step. Our work allows to understand the relative improvement of such encoding step and we show that our method is a challenging baseline for classification on Imagenet: we outperform by a large margin the classification accuracy of former attempts to get rid of representation learning on the large-scale ImageNet dataset.

Our representation is based on a dictionary of whitened patches: pair-wise distances between this dictionary and the patches of each images are computed, binarized and then fed to a linear classifier. This method is straightforward and involves limited ad-hoc feature engineering compared to deep learning approach: here, we employ modern techniques that are necessary for scalability (from thousands to million of samples) but can be understood through the lens of kernel methods (e.g., convolutional classifier, data augmentation, ...). Other papers which work at the patch level can be understood as linear embeddings that rely on some Euclidean distances between patches: our results explicitely indicate that a Hamming distance between well-chosen set of patches is a surprisingly good baseline (see Sec. 3.2), given that the loss of information due to a quantization usually goes in pair with an accuracy degradation (Coates et al., 2011).

Our paper is structured as follows: first, we discuss the related works in Sec. 2. Then, Sec. 3 explains precisely how our visual representation is built. In Sec. 4, we present experimental results on the vision datasets CIFAR-10 and the large scale ImageNet. The final Sec. 4.3 is a collection of numerical experiments to understand better our dictionary of patches. At the time of publication, we will release our code online as well as the commands to reproduce exactly our results.

2 Related work

The seminal works by Coates et al. (2011) and Coates and Ng (2011) study patch-based representations for classification on CIFAR-10. They set the first baseline for a single-layer convolutional network initialized with random patches, and they show it can achieve a non-trivial performance ($\sim 80\%$) on the CIFAR-10 dataset. Recht et al. (2019) published an implementation of this technique and conducted numerous experiments with hundreds of thousands of random patches, improving the accuracy ($\sim 85\%$) on this dataset. However, both works lack two key ingredients: online optimization procedure (which allows to scale up to ImageNet) and well-designed linear classifier (as we propose).

Recently, (Li et al., 2019; Shankar et al., 2020) proposed to handcraft kernels, combined with deep learning tools, in order to obtain high-performances on CIFAR-10. Those performances match standard supervised methods ($\sim 90\%$) which involve end-to-end learning of deep neural networks. Note that the line of work (Li et al., 2019; Shankar et al., 2020; Mairal, 2016) employs a well-engineered combination of patch-extracted representation and a cascade of kernels (possibly some neural tangent kernels). While their works suggest that patch extraction is crucial, the relative improvement due to basic-hyper parameters such as the number of patches or the classifier choice is unclear, as well as the limit of their approach to more challenging dataset. We address those issues.

We note the links between methods based on whitened dictionary of patches and Independent Component Analysis methods such as (Ngiam et al., 2010), as a whitening procedure decorrelates the components of a dictionary of patches. Here we use the whitening to define a Euclidian distance between patches and we show that the decision boundary between image classes can be approximated using a rough description of the image patches neighborhood for this Euclidian distance in a set of randomly selected patches. It suggests that patches topological information based on neighborhood already contains a huge amount of information, which is implied for instance by the fame low-dimensional manifold hypothesis (Fefferman et al., 2016).

Recall that a SIFT extraction Lowe (2004) followed by a Fisher Vectors encoding (Sánchez et al., 2013) and a linear SVM was the state of the art on ImageNet ($\sim 75\%$ top-5 accuracy), before the supremacy ($> 80\%$ top-5) of deep neural networks Krizhevsky et al. (2012). Major differences with our work are modern techniques: we use data-augmentation and we do not need dimensionality reduction steps thanks to GPU-acceleration. Our classification pipeline is believed to help to tackle the curse of dimensionality by reducing the image dimensionality: several of our analysis suggests that even the patches of large raw images are quite low-dimensional, which is aligned with other observations (Oyallon, 2017). This is a surprising fact: while this seems natural for small images like CIFAR-10, our work shows that this low-dimensionality property also surprisingly holds for larger images like ImageNet which have a lot of high-frequency components and much larger variabilities.

From a kernel methods perspective, a dictionary of random patches can be viewed as the building block of a random features method (Rahimi and Recht, 2008) that makes kernel methods computationally tractable. Rudi et al. (2017) provided convergence rates and released an efficient implementation of such a method. However, previously mentioned kernel methods (Mairal, 2016; Li et al., 2019; Shankar et al., 2020) have not been tested on ImageNet to our knowledge.

Simple methods involving solely a single-layer of features have been tested on the ImageNet-2010 dataset[1], using for example SIFT, color histogram and Gabor texture encoding of the image with K-nearest neighbors, yet there is a substential gap in accuracy that we attempt to fill in this work on ImageNet-2012 (or simply ImageNet). We note also that Convolutional Neural Networks (CNNs) with random weights have been tested on ImageNet, yielding to low accuracies ($\sim 20\%$ top-1, (Arandjelovic et al., 2017)).

The Scattering Transform (Mallat, 2012) is also a deep non-linear operator that does not involve representation learning, which has been tested on ImageNet ($\sim 45\%$ top-5 accuracy (Zarka et al., 2019) and CIFAR-10 ($\sim 80\%$, (Oyallon and Mallat, 2015)) and is related to the HoG and SIFT transforms (Oyallon et al., 2018a). Some works also study directly patch encoders that achieve competitive accuracy on ImageNet but involve deep cascade of layers that are difficult to interpret (Oyallon et al., 2017; Zarka et al., 2019; Brendel et al., 2019). Here, we focus on shallow classifiers.

3 METHODS

We first discuss in 3.1 how recent methods such as NTK rely on **data-driven convolutional kernels** which we believe are responsible for the competitive performance on simple image classification tasks. We then propose a simple method in 3.2 to obtain an effective data-driven convolutional kernels for image classification while still being scalable.

3.1 DATA-DRIVEN CONVOLUTIONAL KERNELS

A convolutional kernel $K(x, y)$ computes a similarity between two images x and y based on local patches of x and y. It is said to be data-dependent if, in addition, the similarity measure depends on statistics of the data **without any supervision**. In particular, we will be interested in kernels of the form $K(x, y) = \widetilde{K}(\Phi(x), \Phi(y))$, where $\Phi(x)$ is a data-driven representation and $\widetilde{K}$ is a convolutional kernel which might not be data-driven. In Shankar et al. (2020), $\widetilde{K}$ is obtained by composition of convolutional kernel operations and $\Phi(x)$ consists of a ZCA whitening of the image x. In Coates et al. (2011); Recht et al. (2019), $\Phi(x)$ is obtained by convolving the image x with a dictionary of pre-processed patches extracted from the dataset. Li et al. (2019) uses a similar dictionary based representation for Φ and a CNTK for $\widetilde{K}$. Finally, Mairal (2016) performs a whitening for Φ and considers a convolutional kernel $\widetilde{K}$ based on extracted dictionary of patches. In 4, we show that using data-driven kernels improves accuracy compared to their non-data-driven counterpart. Next, we present a simple and scalable method to obtain data-driven convolutional kernel which is shown in 4 to be competitive on Cifar10 while still being amenable to large scale image classification on ImageNet.

[1] As one can see on the Imagenet2010 leaderboard http://image-net.org/challenges/LSVRC/2010/results, and the accuracies on ImageNet2010 and ImageNet2012 are comparable.

3.2 K-NEAREST NEIGHBORS CONVOLUTIONAL KERNEL

We first introduce our main notations. A patch p of size Q^2 of a larger image x, is a restriction of that image to a squared domain of surface Q^2. We denote by N^2 the size of the natural image x and require that $Q <= N$. Hence, for a spatial index i of the image, $p_{i,x}$ represents the patch of image x located at i. We further introduce the collection of all overlapping patches of that image, denoted by: $\mathcal{P}_x = \{p_{i,x}, i \in \mathcal{I}\}$ where $\mathcal{I}$ is a spatial index set such that $|\mathcal{I}| = (N - Q + 1)^2$.

Whitening We describe the single pre-processing step that we used on our image data, namely a whitening procedure on patches. Here, we view natural image patches of size Q^2 as samples from a random vector of mean μ and covariance Σ. We then consider whitening operators which act at the level of each image patch by first subtracting its mean μ then applying the linear transformation $W = (\lambda \mathbf{I} + \Sigma)^{-1/2}$ to the centered patch. The additional whitening regularization with parameter λ was used to avoid ill-conditioning effects.

The whitening operation is defined up to an isometry, but the Euclidean distance between whitened patches (i.e., the Mahanobolis distance (Chandra et al., 1936)) is not affected by the choice of such isometry (choices leading to PCA, ZCA, ...), as discussed in Appendix A. In practice, the mean and covariance are estimated empirically from the training set to construct the whitening operators. For the sake of simplicity, we only consider whitened patches, and unless explicitly stated, we assume that each patch p is already whitened, which holds in particular for the collection of patches in $\mathcal{P}_x$ of any image x. Once this whitening step is performed, the Euclidean distance over patches is approximatively isotropic and is used in the next section to represent our signals.

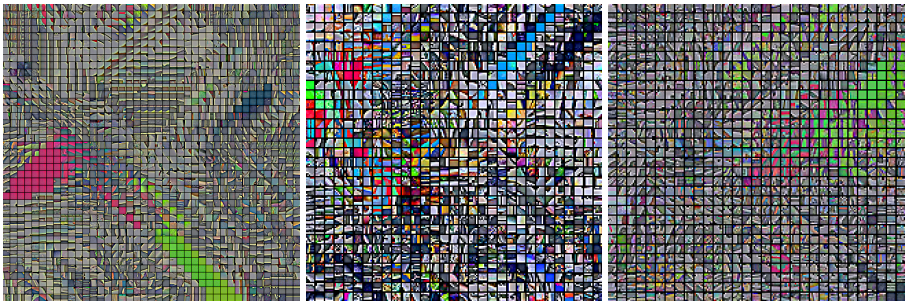

Figure 1: An example of whitened dictionary $\mathcal{D}$ with patch size $Q = 6$ from ImageNet-128 (Left), ImageNet-64 (Middle), CIFAR-10 (Right). The atoms have been reordered via a topographic algorithm from Montobbio et al. (2019) and contrast adjusted.

K-Nearest Neighbors on patches The core idea of our algorithm is to compare the distances between each patch of an image and a fixed dictionary of patches $\mathcal{D}$, with size $|\mathcal{D}|$. For a fixed dataset, this dictionary $\mathcal{D}$ is obtained by uniformly sampling patches from images over the whole training set. We augment $\mathcal{D}$ into $\cup_{d \in \mathcal{D}} \{d, -d\}$ because it allows the dictionary of patches to be contrast invariant and we observe it leads to better classification accuracies; we still refer to it as $\mathcal{D}$. An illustration is given by Fig. 1. Once the dictionary $\mathcal{D}$ is fixed, for each patch $p_{i,x}$ we consider the set $\mathcal{C}_{i,x}$ of pairwise distances $\mathcal{C}_{i,x} = \{\|p_{i,x} - d\|, d \in \mathcal{D}\}$. For each whitened patch we encode the K-Nearest Neighbors of $p_{i,x}$ from the set $\mathcal{D}$, for some $K \in \mathbb{N}$. More formally, we consider $\tau_{i,x}$ the K-th smallest element of $\mathcal{C}_{i,x}$, and we define the K-Nearest Neighbors binary encoding as follow, for $(d, i) \in \mathcal{D} \times \mathcal{I}$:

$$\phi(x)_{d,i} = \begin{cases} 1, & \text{if } \|p_{i,x} - d\| \leq \tau_{i,x} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Equation 1 can be viewed as a Vector Quantization (VQ) step with hard-assignment (Coates and Ng, 2011). The representation ϕ encodes the patch neighborhood in a subset of randomly selected patches and can be seen as a crude description of the topological geometry of the image patches. Moreover, it allows to view the distance between two images x, y as a Hamming distance between the patches neighborhood encoding as:

$$\|\phi(x) - \phi(y)\|^2 = \sum_{i,d} \mathbf{1}_{\phi(x)_{d,i} \neq \phi(y)_{d,i}} \,.$$

In order to reduce the computational burden of our method, we perform an intermediary average-pooling step. Indeed, we subdivide $\mathcal{I}$ in squared overlapping regions $\mathcal{I}_j \subset \mathcal{I}$, leading to the representation Φ defined, for $d \in \mathcal{D}, j$ by:

$$\Phi(x)_{d,j} = \sum_{i \in \mathcal{I}_j} \phi(x)_{d,i} \,. \tag{2}$$

Hence, the resulting kernel is simply given by $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$. Implementation details can be found in Appendix B. The next section describes our classification pipeline, as we feed our representation Φ to a linear classifier on challenging datasets.

4 EXPERIMENTS

We train shallow classifiers, i.e. linear classifier and 1-hidden layer CNN (1-CNN) on top of our representation Φ on two major image classification datasets, CIFAR-10 and ImageNet, which consist respectively of 5×10^5 small and 1.2×10^6 large color images divided respectively into 10 and 10^3 classes. For training, we systematically used mini-batch SGD with momentum of 0.9, no weight decay and using the cross-entropy loss.

Classifier parametrization In each experiments, the spatial subdivisions $\mathcal{I}_j$ are implemented as an average pooling with kernel size k_1 and stride s_1. We then apply a 2D batch-normalization (Ioffe and Szegedy, 2015) in order to standardize our features on the fly before feeding them to a linear classifier. In order to reduce the memory footprint of this linear classifier (following the same line of idea of a "bottleneck" (He et al., 2016)), we factorize it into two convolutional operators. The first one with kernel size k_2 and stride 1 reduces the number of channels from $\mathcal{D}$ to c_2 and the second one with kernel size k_3 and stride 1 outputs a number of channel equal to the number of image classes. Then we apply a global average pooling. For the 1-hidden layer experiment, we simply add a ReLU non linearity between the first and the second convolutional layer.

4.1 CIFAR-10

Implementation details Our data augmentation consists in horizontal random flips and random crops of size 32^2 after reflect-padding with 4 pixels. For the dictionary, we choose a patch size of $Q = 6$ and tested various sizes of the dictionary $|\mathcal{D}|$ and whitening regularization $\lambda = 0.001$. In all cases, we used $K = 0.4|\mathcal{D}|$.The classifier is trained for 175 epoch with a learning rate decay of 0.1 at epochs 100 and 150. The initial learning rate is 0.003 for $|\mathcal{D}| = 2.10^3$ and 0.001 for larger $|\mathcal{D}|$.

Linear classification experiments For the linear classification experiments, we used an average pooling of size $k_1 = 5$ and stride $s_1 = 3$, $k_2 = 1$ and $c_2 = 128$ for the first convolutional operator and $k_3 = 6$ for the second one. Our results are reported and compared in Tab. 1. First, note that contrary to experiments done by Coates et al. (2011) our methods has surprisingly good accuracy despite the hard-assignment in Vector Quantization. Sparse coding, soft-thresholding and orthogonal matching pursuit based representations used by Coates and Ng (2011); Recht et al. (2019) can be seen as soft-assignment VQ and yield comparable classification accuracy (resp. 81.5% with 6.10^3 patches and 85.6% with 2.10^5 patches). However, these representations contain much more information than hard-assignment VQ as they allow to reconstruct a large part of the signal. Our representation yields better accuracy with only coarse topological information on the image patches, suggesting that this information is highly relevant for classification. To obtain comparable accuracies with a linear classifier, we use a single binary encoding step compared to Mairal (2016) and we need a much smaller number of patches than Recht et al. (2019); Coates and Ng (2011). Moreover, Recht et al. (2019) is the only work in the litterature, besides us, that achieves good performance using solely a linear model with depth one. To test the VQ importance, we replace the hard-assignment VQ implemented with a binary non-linearity $\mathbf{1}_{\|p_{i,x}-d\| \leq \tau_{i,x}}$ (see Eq. 1) by a soft-assignment VQ with a sigmoid function $(1 + e^{\|p_{i,x}-d\|-\tau_{i,x}})^{-1}$. The accuracy increases by 0.2%, showing that the use soft-assignment in VQ which is crucial for performance in Coates and Ng (2011) does not affect much the performances of our representation.

Table 1: One layer patch-based classification accuracies on CIFAR-10. We compare methods relying on patch dictionaries. Q is the patch size, $|\mathcal{D}|$ the size of the patch dictionary, VQ indicates whether vector quantization with hard-assignment is applied. Amongst methods relying on random patches ours is the only approach operating online (and therefore allowing for scalable training).

Method	$\lvert\mathcal{D}\rvert$	VQ	Online	Q	Acc.
SimplePatch (Ours)	$1 \cdot 10^4$	✓	✓	6	85.6
SimplePatch (Ours)	$6 \cdot 10^4$	✓	✓	6	86.7
SimplePatch (Ours)	$6 \cdot 10^4$	×	✓	6	**86.9**
??? Ba and Caruana (2014)	$4 \cdot 10^3$	×	✓	???	81.6
Coates et al. (2011)	$1 \cdot 10^3$	✓	×	6	68.6
Scatt. (Oyallon and Mallat, 2015)	-	×	×	8	82.2
Recht et al. (2019)	$2 \cdot 10^5$	×	×	6	85.6

Non-linear classification experiments As a non linear classifier, we simply use 1-hidden layer classifier with ReLU non linearity using an average pooling of size $k_1 = 3$ and stride $s_1 = 2$, $k_2 = 3$, $c_2 = 2048$ and $k_3 = 7$. Our results are reported and compared with other non-linear classification methods in Tab. 3. With this 1-hidden layer classifier, our accuracy is competitive with deep kernel methods (Li et al., 2019; Shankar et al., 2020) and deep supervised convolutional networks (Krizhevsky et al., 2012). This further indicates the relevance of patches neighborhood information for classification task.

Table 2: Accuracies on CIFAR-10 with Handcrafted Kernels classifiers . We compare methods relying on patch dictionaries. Q is the patch size, $|\mathcal{D}|$ the size of the patch dictionary, VQ indicates whether vector quantization with hard-assignment is applied and Classif. stands for classifier. Amongst methods relying on random patches ours is the only approach operating online (and therefore allowing for scalable training).

Method	VQ	Online	Depth	Accuracy (**not** data driven)	Data used	Improvement (data driven)
SimplePatch (Ours)	✓	✓	1	???	Patches	???
NK (Shankar et al., 2020)	×	×	5	77.7	ZCA	8.1
CKN (Mairal, 2016)	×	×	2	81.1	Patches	5.1
NTK (Li et al., 2019)	×	×	8	82.2	Patches	6.7

Table 3: Accuracies on CIFAR-10 with shallow supervised classifiers . We compare methods relying on patch dictionaries. Q is the patch size, $|\mathcal{D}|$ the size of the patch dictionary, VQ indicates whether vector quantization with hard-assignment is applied and Classif. stands for classifier. Amongst methods relying on random patches ours is the only approach operating online (and therefore allowing for scalable training).

Method	VQ (1st layer)	Depth	Classif.	Acc.
SimplePatch (Ours)	✓	2	1-CNN	88.5
NK (Shankar et al., 2020)	×	5	CNN	89.8
CKN (Mairal, 2016)	×	9	CNN	89.8
AlexNet (Krizhevsky et al., 2012)	×	5	CNN	89.1

Ablation experiments CIFAR-10 is a relatively small dataset that allows fast benchmarking, thus we conducted several ablation experiments in order to understand the relative improvement due to each hyper-parameter of our pipeline. We thus vary the size of the dictionary $|\mathcal{D}|$, the patch size Q, the number of nearest neighbors K and the whitening regularization λ which are the hyper-parameters of Φ. Results are shown in Fig. 2.

Note that even a relatively small number of patches is competitive with much more complicated representations, such as Oyallon and Mallat (2015). While it is possible to slightly optimize the performances according to K or Q, the fluctuations remain minor compared to other factors, which

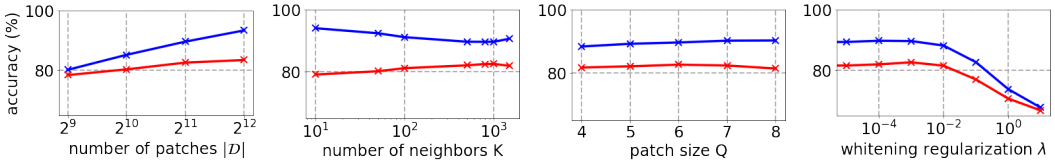

Figure 2: CIFAR-10 ablation experiments, train accuracies in blue, test accuracies in red. Higher resolution plots can be found in Appendix C.

indicate that the performances of our method are relatively stable w.r.t. this set of hyper-parameters. The whitening regularization behaves similarly to a thresholding operator on the eigenvalues of $\Sigma^{1/2}$: the larger it is, the more it penalizes larger eigenvalues. Interestingly, we note that under a certain threshold, this hyper-parameter does almost not affect the classification performances. This goes in hand with both a fast eigenvalue decay and a stability to noise, that we discuss further in Sec. 4.3.

Classifier factorization In order to test the inductive bias of our classifier, we replace it with a simple fully connected layer, with 6 times more parameters than ours: the train and test accuracies are 93.0% and 81.6% compared to 88.9% and 82.5%. The use of this factorized classifier significantly reduces overfitting, while reducing the number of computations. We note that using convolutions is well motivated by the structure of natural images whose class is relatively invariant to translation.

Whitening and Gaussian random filters As observed by some works from Tab. 1 and 3, removing the whitening step for both the dictionary and image patches leads to a performance drop of about 17%. Moreover, replacing the whitened dictionary of patches by patches drawn from a white Gaussian distribution leads to a drop of about 6%. This indicates that both the choice of patches and the whitening step are crucial for obtaining our performances.

4.2 IMAGENET

Implementation details To reduce the computational overhead of our method on ImageNet, we followed the same approach as Chrabaszcz et al. (2017). We reduce the resolution to 64^2 as in Chrabaszcz et al. (2017), instead of the standard 224 length. Note that Chrabaszcz et al. (2017) observed that this does not alter much the top-performances of standard models (5% to 10% drop of accuracy on average), and we also believe it introduces a useful dimensionality reduction, as it removes high-frequency part of images that are unstable Mallat (1999). We set the patch size to $Q = 6$ and the whitening regularization to $\lambda = 10^{-2}$. Since ImageNet is a much larger dataset than CIFAR-10, we restricted to $|\mathcal{D}| = 2048$ patches. As for CIFAR-10 experiments, we set $K = 0.4|\mathcal{D}|$. The parameters of the linear convolutional classifier are chosen to be: $k_1 = 5, s_1 = 3, k_2 = 1, c_2 = 256, k_3 = 12$. For the 1-hidden layer experiment, we used kernel size of $k_2 = 3$ for the first convolution. Our models are trained during 60 epochs with an initial learning rate of 0.003 decayed by a factor 10 at epochs 40 and 50. During training, similarly to Chrabaszcz et al. (2017) we use random flip and we select random crops of size 64, after a reflect-padding of size 8. At testing, we simply resize the image to 64. Note that this procedure differs slightly from the usual data-augmentation, which consists in resizing images while maintaining ratios, before a random cropping.

Classification experiments Tab. 5 reports the accuracy of our method, as well as the accuracy of comparable methods. Using a smaller resolution for the images still allows our method to outperform by a large margin (about $\sim 10\%$ Top-5) the Scattering Transform (Mallat, 2012), which was the previous state-of-the-art-method in the context of no-representation learning. Note that our representation uses only 2.10^3 randomly selected patches which is a tiny fraction of the billions of ImageNet patches.

Sánchez et al. (2013) obtain 72.0% top-5 accuracy with a patch representation computed in three steps (SIFT, Fisher kernel, power-normalization/compression) and of dimension 6.10^4. Using a representation of dimension 4.10^3 they obtain 64.1% top-5 accuracy. The performance of our method tested on lower resolution images (128^2) using a representation of dimension 2.10^3 ($= |\mathcal{D}|$) is relatively close to the performance of the 4.10^3 dimensional Fisher Vectors, but further large scale

experiments would be needed to confirm if this holds for higher dimensions. Note that this visual representation involves the learning of a Gaussian mixture model and several PCA dimensionality reductions that are crucial for performance.

We now compare our performances with supervised models trained end-to-end. BagNets (Brendel et al., 2019) have shown that competitive classification accuracies can be obtained with patch-encoding that consists of 50 layers. The performance obtained by our shallow experiment with a 1-hidden layer classifier reduces significantly the gap of performance between our method and a BagNet with similar patch-size. This suggests once again that hard-assignment VQ does not degrade much of the classification information. We also note that our approach with a linear classifier outperforms supervised shallow baselines that consists of 1 or 2 hidden-layers CNN (Belilovsky et al., 2018), which indicates that a patch based representation is a non-trivial baseline.

To measure the importance of the resolution on the performances, we run a linear classification experiment on ImageNet images with twice bigger resolution (128^2 images , $Q = 12, k_1 = 10, s_1 = 6$). We observe that it improves classification performances. Note that the patches used are in a space of dimension $432 \gg 1$: this improvement is surprising since distance to nearest neighbors are known to be meaningless in high-dimension (Beyer et al., 1999). This shows a form of low-dimensionality in the natural image patches, that we study in the next Section.

Table 4: Handcrafted kernel accuracies on ImageNet, where no ad-hoc explicit loss is optimized. D., Res. and Classif. stand respectively for Depth, Resolution and Classifier. Dim. stands for the dimension of the patch representation equal to $|\mathcal{D}|$ in our setting.

Method	Dim.	VQ	Q	D.	Res.	Classif.	Top1	Top5
SimplePatch(Ours)	2.10^3	✓	6	1	64	linear	33.4	54.7
SimplePatch(Ours)	2.10^3	✓	12	1	128	linear	35.4	56.9
Scatt.(Zarka et al., 2019)	-	×	32	2	224	linear	26.1	44.7
Random(Arandjelovic et al., 2017)	-	×	-	9	224	linear	18.9	-

Table 5: Supervised accuracies on ImageNet. D., Res. and Classif. stand respectively for Depth, Resolution and Classifier. Dim. stands for the dimension of the patch representation equal to $|\mathcal{D}|$ in our setting.

Method	Dim.	VQ	Q	D.	Res.	Classif.	Top1	Top5
SimplePatch(Ours)	2.10^3	✓	6	2	64	1-CNN	39.4	62.1
Shallow(Belilovsky et al., 2018)	-	×	-	1	224	CNN	-	26
Shallow(Belilovsky et al., 2018)	-	×	-	2	224	CNN	-	44
BagNet(Brendel et al., 2019)	-	×	9	50	224	CNN	-	70.0

4.3 Dictionary structure

Spectrum of $\mathcal{D}$ Fig. 3 (top) shows the spectrum of $\Sigma^{1/2}$ for several values of Q, normalized by $\|\Sigma^{1/2}\|$ on CIFAR-10 and ImageNet-32. First, note that the spectrum tends to decay at an exponential rate (linear rate in semi-logarithmic scale). This rate decreases as the size of the patch increases (from dark brown to light brown) suggesting an increased linear dimensionality for larger patches. The second observation is that patches from ImageNet-32 dataset tend to be better conditioned than those from CIFAR-10 with a conditioning ratio of 10^2 for ImageNet vs 10^3 for CIFAR-10. This is probably due to the use of more diverse images than on CIFAR-10. From this spectrum, it is straightforward to compute the linear dimensionality of the patches. Fig. 3(bottom-left) shows the number of axis needed to explain 95% of the variance as a function of the extrinsic dimension $d_{\text{ext}} = 3Q^2$, with and without whitening. Before whitening, this linear dimension is much smaller than the ambient dimension: whitening the patches increases the linear dimensionality of the patches, which still increases at a linear growth as a function of Q^2.

Intrinsic dimension of $\mathcal{D}$ We propose to refine our measure of linear dimensionality to a non-linear measure of the intrinsic dimension. Indeed, under the assumption of low-dimensional manifold, the linear dimensionality is simply an upper bound of the true dimensionality of image patches. To do so,

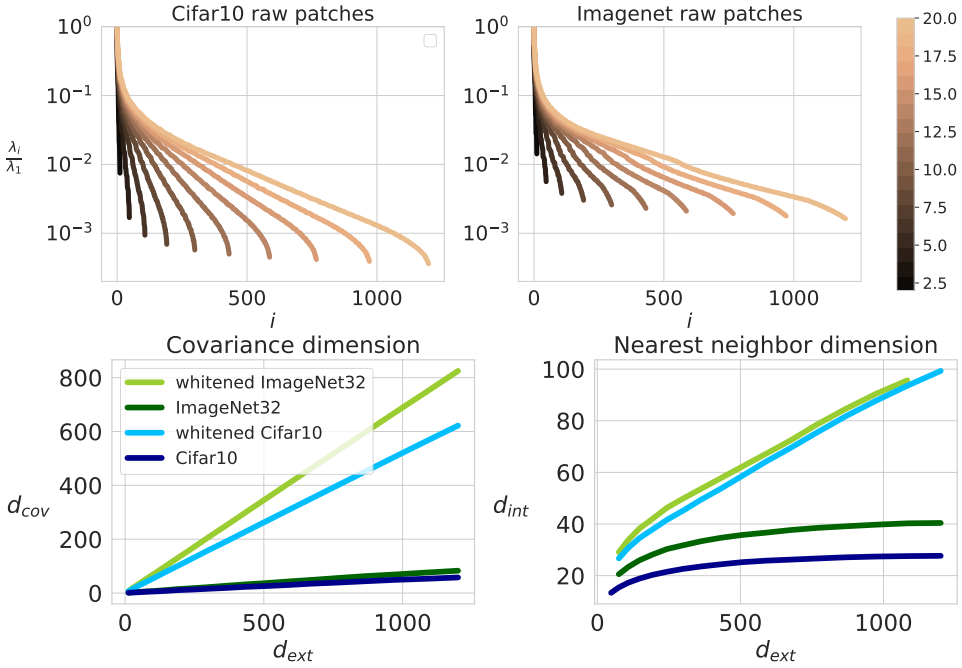

Figure 3: (Top) Spectrum of the square root covariance matrix $\Sigma^{1/2}$ on CIFAR-10 (top-left) and ImageNet-64 (top-right) using small patch sizes in dark-brown to larger patch sizes in light-brown. (Bottom-left) Covariance dimension as a function of the extrinsic dimension of the patches and (bottom right) nearest neighbor dimension as a functions of the extrinsic dimension of the patches.

we use the intrinsic dimension d_{int} introduced in (Levina and Bickel, 2004) (see Appendix D). An overall estimate of the d_{int} is then obtained by averaging the local estimate $d_{\text{int}}(p)$ over all patches. Such estimate depends on the maximum number of neighbors K. However, it converges to the same value when both K and the size of the dataset increases, provided that K remains small compared to it. Fig. 3 (bottom-right) shows the intrinsic dimension estimated using $K = 2000$. in all cases, the estimated intrisic dimension d_{int} is much smaller than the extrinsic dimension $d_{\text{ext}} = 3Q^2$. Moreover, it grows even more slowly than the linear dimension when the patch size Q increases. Finally, even after whitening, d_{int} is only about 10% of the total dimension, which is a strong evidence that the natural image patches are low dimensional.

5 Conclusion

In this work, we considered a visual representation for image classification based on patches K nearest neighbors encoding for Euclidian distance. This non-learned representation achieves a competitive accuracy on CIFAR-10 and can be easily scaled to large datasets such as ImageNet, yielding a non trivial performance. These results, as well as the presented analysis of the image patches, suggest that the image patches live in a much lower dimensional space than their ambient space of dimension $3Q^2$ that we naturally consider. Due to limited computational resources, we restricted ourselves on ImageNet to small image resolutions and relatively small number of patches. Conducting proper large scale experiments is thus one of the next research directions. We hope that the presented method will foster new developments in the field of non-learned visual representations, following the recent success of self-supervised visual representations that are now on par with end-to-end supervised methods. Designing such representations might help to understand the underlying mathematics of image classification problem and to improve understanding of deep learning models.

References

R. Arandjelovic, A. Zisserman, and . Look, listen and learn. In Proceedings of the IEEE International Conference on Computer Vision, pages 609–617, 2017.

J. Ba and R. Caruana. Do deep nets really need to be deep? In Advances in neural information processing systems, pages 2654–2662, 2014.

E. Belilovsky, M. Eickenberg, and E. Oyallon. Greedy layerwise learning can scale to imagenet. arXiv preprint arXiv:1812.11446, 2018.

K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In International conference on database theory, pages 217–235. Springer, 1999.

W. Brendel, M. Bethge, and . Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. arXiv preprint arXiv:1904.00760, 2019.

M. P. Chandra et al. On the generalised distance in statistics. In Proceedings of the National Institute of Sciences of India, volume 2, pages 49–55, 1936.

P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. CoRR, abs/1707.08819, 2017. URL http://arxiv.org/abs/1707.08819.

A. Coates and A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. 2011.

A. Coates, A. Ng, and H. 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.

A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. IEEE Transactions on image processing, 13(9):1200–1212, 2004.

A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In Proceedings of the seventh IEEE international conference on computer vision, volume 2, pages 1033–1038. IEEE, 1999.

C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. Journal of the American Mathematical Society, 29(4):983–1049, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In Advances in neural information processing systems, pages 8571–8580, 2018.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In Advances in neural information processing systems 17, pages 777–784. MIT Press, 2004.

Z. Li, R. Wang, D. Yu, S. S. Du, W. Hu, R. Salakhutdinov, and S. Arora. Enhanced convolutional neural tangent kernels. arXiv preprint arXiv:1911.00809, 2019.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2):91–110, 2004.

Z. Lu, A. May, K. Liu, A. B. Garakani, D. Guo, A. Bellet, L. Fan, M. Collins, B. Kingsbury, M. Picheny, et al. How to scale up kernel methods to be as good as deep neural nets. arXiv preprint arXiv:1411.4000, 2014.

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

S. Mallat. A wavelet tour of signal processing. Elsevier, 1999.

S. Mallat. Group invariant scattering. Communications on Pure and Applied Mathematics, 65(10): 1331–1398, 2012.

G. J. McLachlan. Mahalanobis distance. Resonance, 4(6):20–26, 1999.

N. Montobbio, A. Sarti, and G. Citti. A metric model for the functional architecture of the visual cortex. 2019. URL http://arxiv.org/abs/1807.02479.

J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A. Y. Ng. Tiled convolutional neural networks. In Advances in neural information processing systems, pages 1279–1287, 2010.

E. Oyallon. Building a regular decision boundary with deep networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

E. Oyallon and S. Mallat. Deep roto-translation scattering for object classification. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In Proceedings of the IEEE international conference on computer vision, pages 5618–5627, 2017.

E. Oyallon, E. Belilovsky, S. Zagoruyko, and M. Valko. Compressing the input for cnns with the first-order scattering transform. In The European Conference on Computer Vision (ECCV), September 2018a.

E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. Blaschko, and E. Belilovsky. Scattering networks for hybrid representation learning. IEEE transactions on pattern analysis and machine intelligence, 41(9):2208–2221, 2018b.

F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In European conference on computer vision, pages 143–156. Springer, 2010.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In Advances in neural information processing systems, pages 1177–1184, 2008.

B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? arXiv preprint arXiv:1902.10811, 2019.

A. Rudi, L. Carratino, and L. Rosasco. Falkon: An optimal large scale kernel method. In Advances in Neural Information Processing Systems, pages 3888–3898, 2017.

J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. International journal of computer vision, 105(3):222–245, 2013.

V. Shankar, A. Fang, W. Guo, S. Fridovich-Keil, L. Schmidt, J. Ragan-Kelley, and B. Recht. Neural kernels without tangents. arXiv preprint arXiv:2003.02237, 2020.

G. K. Wallace. The jpeg still picture compression standard. IEEE transactions on consumer electronics, 38(1):xviii–xxxiv, 1992.

J. Zarka, L. Thiry, T. Angles, and S. Mallat. Deep network classification by scattering and homotopy dictionary learning. arXiv preprint arXiv:1910.03561, 2019.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In European conference on computer vision, pages 818–833. Springer, 2014.

A MAHANALOBIS DISTANCE AND WHITENING

The Mahalanobis distance (Chandra et al., 1936; McLachlan, 1999) between two samples x and x' drawn from a random vector X with covariance Σ is defined as

$$D_M(x, x') = \sqrt{(x - x')^T \Sigma^{-1} (x - x')}$$

If the random vector X has identity covariance, it is simply the usual euclidian distance :

$$D_M(x, x') = \|x - x'\| \, .$$

Using the diagonalization of the coraviance matrix, $\Sigma = P \Lambda P^T$, the affine whitening operators of the random vector $\mathbf{X}$ are the operators

$$w : \mathbf{X} \mapsto O \Lambda^{-1/2} P^T (\mathbf{X} - \mu), \quad \forall O \in O_n(\mathbb{R}) \, . \tag{3}$$

For example, the PCA whitening operator is

$$w_{\mathrm{PCA}} : \mathbf{X} \mapsto \Lambda^{-1/2} P^T (\mathbf{X} - \mu)$$

and the ZCA whitening operator is

$$w_{\mathrm{ZCA}} : \mathbf{X} \mapsto P \Lambda^{-1/2} P^T (\mathbf{X} - \mu) \, .$$

For all whitening operator w we have

$$\|w(x) - w(x')\| = D_M(x, x')$$

since

$$\begin{aligned} \|w(x) - w(x')\| &= \|O \Lambda^{-1/2} P^T (x - x')\| \\ &= \sqrt{(x - x')^T P \Lambda^{-1/2} O^T O \Lambda^{-1/2} P^T (x - x')} \\ &= \sqrt{(x - x')^T P \Lambda^{-1} P^T (x - x')} \\ &= D_M(x, x') \, . \end{aligned}$$

B IMPLEMENTATION OF THE PATCHES K-NEAREST-NEIGHBORS ENCODING

In this section, we explicitly write the whitened patches with the whitening operator W. Recall that we consider the following set of euclidean pairwise distances:

$$\mathcal{C}_{i,x} = \{\|W p_{i,x} - W d\| \, d \in \mathcal{D}\} \, .$$

For each image patch we encode the K nearest neighbors of $W p_{i,x}$ in the set $W d, d \in \mathcal{D}$, for some $K \in 1 \dots |\mathcal{D}|$. We can use the square distance instead of the distance since it doesn't change the K nearest neighbors. We have

$$\|W p_{i,x} - W d\|^2 = \|W p_{i,x}\|^2 - 2 \langle p_{i,x}, W^T W d \rangle + \|W d\|^2$$

The term $\|W p_{i,x}\|^2$ doesn't affect the K nearest neighbors, so the K nearest neighbors are the K smallest values of

$$\left\{ \frac{\|W d\|^2}{2} + \langle p_{i,x}, -W^T W d \rangle, \; d \in \mathcal{D} \right\}$$

This can be implemented in a convolution of the image using $-W^T W d$ as filters and $\|W d\|^2/2$ as bias term, followed by a "vectorwise" non-linearity that binary encodes the K smallest values in the channel dimension. Once this is computed, we can then easily compute

$$\left\{ \frac{\|W d\|^2}{2} + \langle p_{i,x}, W^T W d \rangle, \; d \in \mathcal{D} \right\}$$

which is the quantity needed to compute the K nearest neighbors in the set of negative patches $\overline{\mathcal{D}}$. This is a computationally efficient way of doubling the number of patches while making the representation invariant to negative transform.

C ABLATION STUDY ON CIFAR-10

For this ablation study on CIFAR-10, the reference experiment uses $|\mathcal{D}| = 2048$ patches, a patch size $Q = 6$ a number of neighbors $K = 0.4 \times 2048 = 820$ and a whitening regularizer $\lambda = 1e-3$, and yields 82.5% accuracy. Figure 4 shows the results in high resolution.

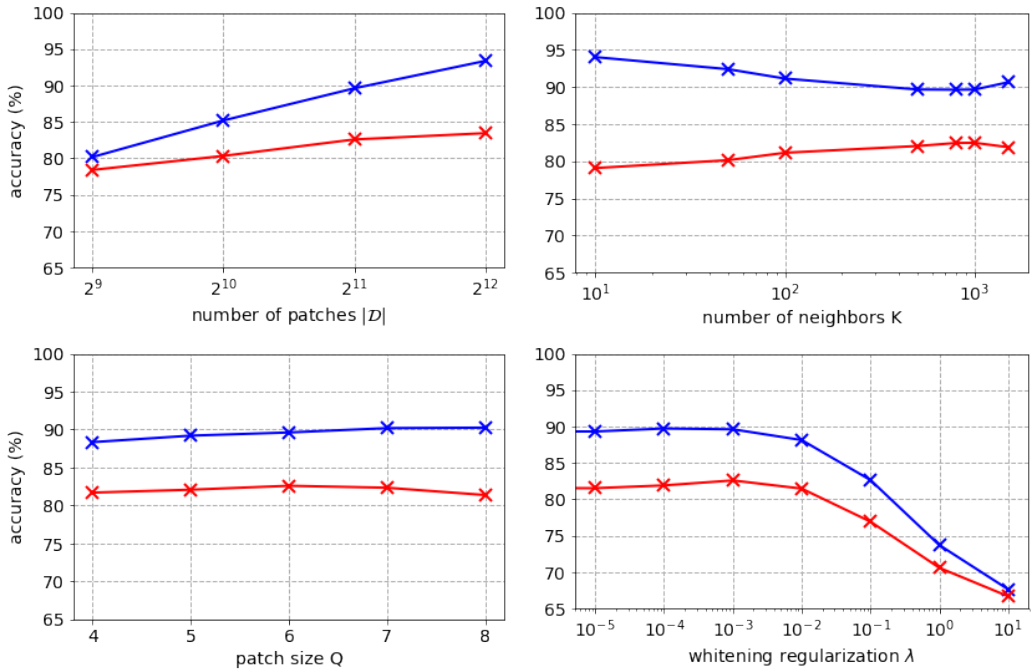

Figure 4: CIFAR-10 ablation experiments, train accuracies in blue, test accuracies in red. Number of patches $|\mathcal{D}|$ varies in $\{512, 1024, 2048, 4096\}$, number of neighbors K varies in $\{10, 50, 100, 500, 800, 1000, 1500\}$, patch size Q varies in $\{4, 5, 6, 7, 8\}$, whitening regularization λ varies in $\{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$.

D INTRINSIC DIMENSION ESTIMATE

The following estimate of the intrinsic dimension d_{int} is introduced in Levina and Bickel (2004) as follows

$$d_{\text{int}}(p) = \left(\frac{1}{K-1} \sum_{k=1}^{K-1} \log \frac{\tau_K(p)}{\tau_k(p)} \right)^{-1}, \tag{4}$$

where $\tau_k(p)$ is the euclidean distance between the patch p and it's k-th nearest neighbor int the training set.