Nested Invariance Pooling and RBM Hashing for Image Instance Retrieval

Olivier Morère
Université Pierre et Marie Curie, Paris, France
olivier.morere@gmail.com

Jie Lin
Institute for Infocomm Research, Singapore
lin-j@i2r.a-star.edu.sg

Antoine Veillard
Université Pierre et Marie Curie, Paris, France
antoine.veillard@gmail.com

Ling-Yu Duan
School of EE&CS, Peking University, China
lingyu@pku.edu.cn

Vijay Chandrasekhar
Institute for Infocomm Research / Nanyang Technological University, Singapore
vijay@i2r.a-star.edu.sg

Tomaso Poggio
Massachusetts Institute of Technology, USA
tp@ai.mit.edu

ABSTRACT

The goal of this work is the computation of very compact binary hashes for image instance retrieval. Our approach has two novel contributions. The first one is Nested Invariance Pooling (NIP), a method inspired from *i-theory*, a mathematical theory for computing group invariant transformations with feed-forward neural networks. NIP is able to produce compact and well-performing descriptors with visual representations extracted from convolutional neural networks. We specifically incorporate scale, translation and rotation invariances but the scheme can be extended to any arbitrary sets of transformations. We also show that using moments of increasing order throughout nesting is important. The NIP descriptors are then hashed to the target code size (32-256 bits) with a Restricted Boltzmann Machine with a novel batch-level regularization scheme specifically designed for the purpose of hashing (RBMH). A thorough empirical evaluation with state-of-the-art shows that the results obtained both with the NIP descriptors and the NIP+RBMH hashes are consistently outstanding across a wide range of datasets.

CCS CONCEPTS

•Information systems → Image search;

KEYWORDS

Image Instance Retrieval, CNN, Invariant Representation, Hashing, Unsupervised Learning, Regularization

1 INTRODUCTION

Small binary image representations such as 64-bit hashes are a definite must for fast image instance retrieval. Compact hashes provide more than enough capacity for any practical purposes, including internet-scale problems. In addition, a compact hash is directly addressable in RAM and enables fast matching using ultra-fast Hamming distances.

State-of-the-art global image descriptors such as Fisher Vectors (FV) [30], Vector of Locally Aggregated Descriptors (VLAD) [22] and Convolutional Neural Network (CNN) features [6, 24] allow for robust image matching. However, the dimensionality of such descriptors is typically very high: 4096 to 65536 floating point numbers for FVs [30] and 4096 for CNNs [24]. Bringing such high-dimensional representations down to compact hashes is a considerable challenge.

Deep learning has achieved remarkable success in many visual tasks such as image classification [24, 36], image retrieval [6], face recognition [37, 38] and pose estimation [41]. Here,we propose a deep learning framework for binary hashing that generates extremely compact, yet discriminative descriptors. A series of nested pooling layers introduced in the pipeline, provide higher invariance and robustness to common transformations like rotation and scale. A RBM layer for hashing is introduced at the end of the pipeline to map real-valued data to binary hashes. The proposed deep learning pipeline generates hashes that consistently and significantly outperform other state-of-the-art methods at code size from 256 down to very small sizes like 32 bits, on several popular benchmarks.

2 BACKGROUND AND RELATED WORK

Our image instance retrieval pipeline starts with the computation of high-dimensional vectors referred to as *global descriptors*, followed by a hashing step to obtain compact representations. Here, we review the state-of-the-art in *global descriptors* and hashing methods.

Global Descriptors. State-of-the-art *global descriptors* for image instance retrieval are based on either FV [30]/VLAD [22] or Convolutional Neural Networks (CNN) [6]. Several variants of FV/VLAD [8, 23, 26, 39] have been proposed, since it was first proposed for instance retrieval [30]. In recent work, CNN descriptors have begun being applied to the computation of global descriptors for image retrieval [5, 6, 34, 35, 40, 43].

Razavian et al. [34] evaluate the performance of CNN activations from fully connected layer on a wide range of tasks including instance retrieval, and show initial promising results. After that, Babenko et al. [6] show that a pre-trained CNN can be fine tuned with domain specific data (objects, scenes, etc.) to improve retrieval

ICMR '17, June 6–9, 2017, Bucharest, Romania.
 ISBN 978-1-4503-4701-3/17/06…$15.00.
DOI: http://dx.doi.org/10.1145/XXXXXXX.XXXXXXX

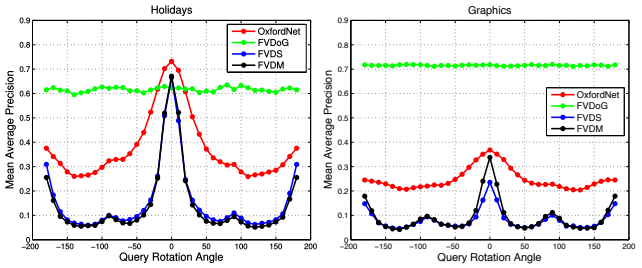


Figure 1: Comparison of CNN and FV descriptors retrieval performance with rotated queries on *Holidays* and *Graphics* datasets (see experimental section for details on datasets). FVDoG, FVDS, and FVDM are Fisher Vectors based on DoG interest points, Dense interest points at Single scale, and Dense interest points at Multiple scales respectively: all use the SIFT descriptor. FVDoG is robust to rotation, while CNN, FVDS and FVDM suffer a sharp drop in performance as query image is rotated.

performance on relevant datasets. In [43], the authors propose extracting activations of fully connected layer from multiple regions sampled in an image, followed by aggregating VLAD descriptors on these local CNN activations. While this results in highly performant descriptors, the starting representations are orders of magnitude larger than descriptors proposed in this work. [4, 35] show that spatial max pooling of intermediate maps is an effective representation and higher performance can be achieved compared to using the fully connected layers. Babenko et al [5] in their very recent work, show that sum-pooling of intermediate feature maps performs better than max-pooling, when the image representation is whitened. Note that the approach in [5] provide limited invariance to translation, but not to scale or rotation. Another very recent work [40] proposes pooling across regional bounding boxes in the image, similar to the popular R-CNN approach [12] used for object detection.

Unlike FVs based on interest point detectors like the Difference-of-Gaussian (DoG) detector, CNN does not have a built-in mechanism to ensure resilience to geometric transformations like scale and rotation. In particular, the performance of CNN descriptors quickly degrade when the objects in the query and the database image are rotated or scaled differently. To illustrate this, in Figure 1, we show retrieval results when query images are rotated with respect to database images for descriptors: (a) Fisher Vectors based on Difference of Gaussian interest points and SIFT descriptors (FVDoG), (b) Fisher Vectors based on dense interest points and SIFT descriptors, at just one scale (FVDS), (c) Fisher Vectors based on dense interest points and SIFT descriptors, at multiple scales in the image (FVDM), and (d) CNN descriptors based on the first fully connected layer of *OxfordNet* [36]. Schemes apart from FVDoG suffer a sharp drop in performance as geometric transforms are applied. In this work, we focus on how to systematically incorporate groups of invariance into the CNN pipeline.

Hashing. In this work, we are focused on image instance retrieval with compact descriptors produced by unsupervised hashing on global descriptors. Semantic image retrieval with supervised hashing is outside the scope of this work. Examples of popular unsupervised hashing methods include Locality Sensitive Hashing (LSH) [9], Iterative Quantization (ITQ) [15], Spectral Hashing (SH) [42] and Restricted Boltzmann Machines (RBM) [18, 28]. Gong et al. propose the popular ITQ [15]. ITQ first performs Principal Component Analysis (PCA) to reduce dimensionality, then applies rotations to distribute variance across dimensions, and finally binarizes each dimension according to its sign. Besides hashing, quantization based methods such as Product Quantization (PQ) [22, 45, 46] divide the raw descriptor into smaller blocks and vector quantization is performed on each block. While this results in highly compact descriptors composed of sub-quantizer indices, the resulting representation is not binary and cannot be compared with Hamming distance.

3 CONTRIBUTIONS

The goal of this work is the computation of very compact binary hashes for image instance retrieval. To that end, we propose a multi-stage pipeline as shown on Figure 2 with the following contributions:

- First, we propose Nested Invariance Pooling (NIP), a method to produce compact global image descriptors from visual representations extracted from CNNs. Our method draws its inspiration from the *i-theory* [1–3], a mathematical theory for computing group invariant transformations with feed-forward neural networks. We specifically incorporate scale, translation and rotation invariance but the scheme can be extended to any arbitrary sets of transformations. We also show that using moments of increasing order throughout nesting is important. Resulting NIP descriptors are invariant to various types of image transformations and we show that the process significantly improves retrieval results while keeping dimensionality low (512 dimensions).
- Then, the NIP descriptors are hashed to the target code size (32-256 bits) with a Restricted Boltzmann Machine (RBM). We propose a novel batch-level regularization scheme specifically designed for the purpose of hashing, a scheme we refer to as RBMH from hereon.
- A thorough empirical evaluation with state-of-the-art shows that the results obtained both with the NIP descriptors and the NIP+RBMH hashes are consistently outstanding across a wide range of datasets. To the best of our knowledge, the results reported at 128 bits hashes are the highest reported results in state-of-the-art literature.

4 METHOD

4.1 Nested Invariance Pooling

Let an image $x \in E$ and a group G of transformations acting over E with group action $G \times E \rightarrow E$ denoted with a dot (.). The orbit of x by G is the subset of E defined as $O_x = \{g.x \in E | g \in G\}$. The orbit corresponds to the set of transformations of x under groups such as rotations, translations and scale changes. It can be easily shown that O_x is globally invariant to the action of any element of G and thus any descriptor computed directly from O_x would be globally invariant to G.

The *i-theory* builds invariant representations for a given object $x \in E$ in relation with a predefined template $t \in E$ from the distribution of the dot products $D_{x,t} = \{< g.x, t > \in \mathbb{R} | g \in G\} =$

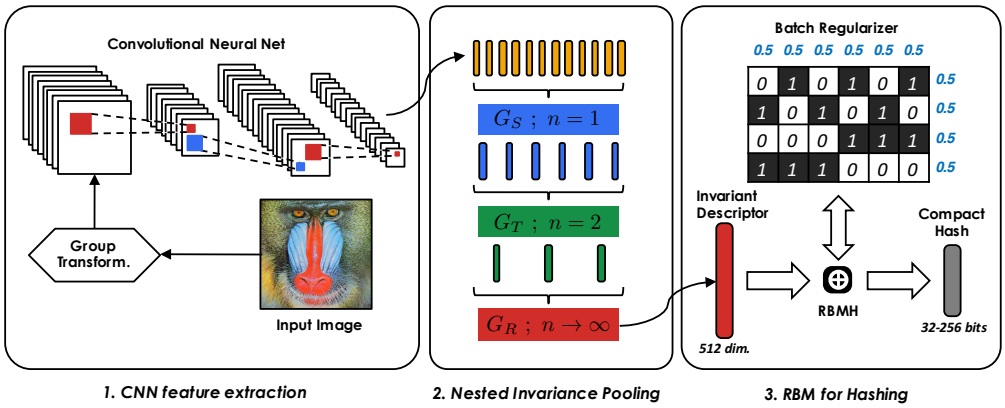


Figure 2: **Our proposed pipeline for image instance retrieval applies Nested Invariance Pooling (NIP) to produce robust and compact descriptors from CNNs followed by an RBM specially regularized for Hashing (RBMH).**

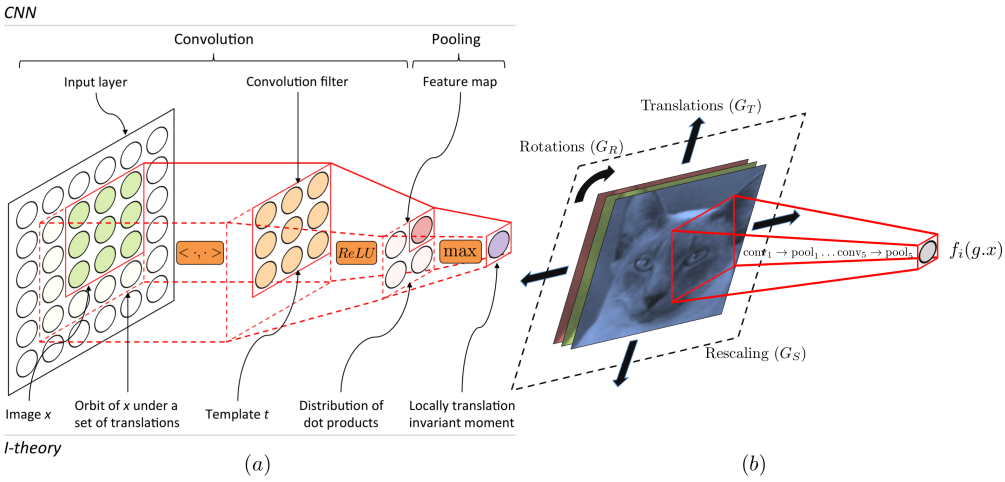


Figure 3: **(a) A single convolution-pooling operation from a CNN schematized for a single input layer and single output unit. The parallel with *i-theory* shows that the universal building block of CNNs is compatible with the incorporation of invariance to local translations of the input according to the theory. (b) A specific succession of convolution and pooling operations learnt by the CNN (depicted in red) computes the *pool5* feature f_i for each feature map i from the RGB image data. A number of transformations g can be applied to the input x in order to vary the response $f_i(g.x)$.**

$\{< x, g.t >\in \mathbb{R} | g \in G\}$ over the orbit. The following representation (for any $n \in \mathbb{N}^*$) is proven to have proper invariance and selectivity properties provided that the group is compact or locally compact:

$$\mu_{G,t,n}(x) = \frac{1}{\int_G dg} \left(\int_G | < g.x, t > |^n dg \right)^{\frac{1}{n}} \quad (1)$$

Note that the sequence $(\mu_{G,t,n}(x))_{n \in \mathbb{N}^*}$ is analogous to a histogram. In practice, the theory extends well (with approximate invariance) to non-locally compact groups and even to continuous non-group transformations (e.g. out-of-plane rotations, elastic deformations) provided that proper class-specific templates can be chosen [2]. Recent work on face verification [25] and music classification [44] apply the theory to non-compact groups with good results.

Popular CNN architectures for classification such as *AlexNet* [24] and *OxfordNet* [36] share a common building block: a succession of convolution-pooling operations designed to model increasingly high-level visual representations of the data. As shown in Figure 3 (a), the succession of convolution and pooling operations in a typical CNN is in fact a way to incorporate local translation invariance strictly compliant with the framework proposed by the *i-theory*. The network architecture provides the robustness as predicted by the *i-theory*, while parameter tuning via back propagation ensures a proper choice of templates.

We build our NIP descriptors starting from the already locally robust *pool5* feature maps of *OxfordNet*. Global invariance to several transformation groups are then sequentially incorporated following the *i-theory* framework. The specific transformation groups considered in this study are translations G_T, rotations G_R and scale changes G_S. For every feature map i of the *pool5* layer ($0 \leq i < 512$), we denote $f_i(x)$ the corresponding unit's output. As shown on Figure 3 (b), transformations g are applied on the input image x varying the output of the *pool5* feature $f_i(g.x)$. Note that the transformation f_i is non-linear due to multiple convolution-pooling

operations thus not strictly a mathematical dot product but can still be viewed as an inner product. Accordingly, the pooling scheme used by NIP with $G \in \{G_T, G_R, G_S\}$ is:

$$\mathcal{X}_{G,i,n}(x) = \frac{1}{\int_G dg}\left(\int_G f_i(g.x)^n dg\right)^{\frac{1}{n}} = \frac{1}{m}\left(\sum_{j=0}^{m-1} f_i(g_j.x)^n\right)^{\frac{1}{n}} \tag{2}$$

when O_x is discretized into m samples. The corresponding global image descriptors are obtained after each pooling step by concatenating the moments for the individual features:

$$\mathcal{X}_{G,n}(x) = (\mathcal{X}_{G,i,n}(x))_{0 \le i < 512} \tag{3}$$

As shown in Equation 2, the pooling operation has an order parameter n defining the "hardness" of the pooling. $n = 1$ is average pooling while $n \to +\infty$ on the other extreme is max-pooling. $n = 2$ is analogous to standard deviation. Subsequently, we refer to the moments for $n = 1, 2, +\infty$ as $\mathcal{A}_G$, $\mathcal{S}_G$ and $\mathcal{M}_G$.

Work on *i-theory* [44] has shown that it is possible to chain multiple types of group invariances one after the other [44]. We apply this principle on our NIP descriptors by making them invariant to several transformations. For instance, following scale invariance with average ($n = 1$) by translation invariance with hard max-pooling ($n \to +\infty$) is done by:

$$\begin{aligned}&\max_{g_t \in G_T}\left(\frac{1}{\int_{g_s \in G_S} dg_s}\int_{g_s \in G_S} f_i(g_t g_s.x) dg_s\right)\\ &= \max_{j \in [0, m_t - 1]}\left(\frac{1}{m_s}\sum_{i=0}^{m_s-1} f_i(g_{t,j} g_{s,i} t.x)\right)\end{aligned} \tag{4}$$

Operations are sometimes commutable (e.g. $\mathcal{A}_G$ and $\mathcal{A}_{G'}$) and sometimes not (e.g. $\mathcal{A}_G$ and $\mathcal{M}_{G'}$) depending on the specific combination of moments so the sequence of transformations does matter for NIP. The hardness parameter n must also be chosen carefully. Empirically, we found pooling progressively with increasing moments (e.g. $\mathcal{A}_G$, then $\mathcal{S}_G$, then $\mathcal{M}_G$) to work well as presented in the experiments section.

Figure 4 provides an insight on how adding different types of invariance with our NIP scheme will affect the matching distance on different image pairs of matching objects. With the incorporation of each new transformation group, we notice that the relative reduction in matching distance is the most significant with the image pair which is the most affected by the transformation group.

4.2 Restricted Boltzmann Machine for Hashing

The NIP descriptors are subsequently hashed to the target dimentionality with an RBM layer. The motivation is to obtain mutually independent dimensions while distributing variance evenly across them in a way similar to ITQ [15]. The main originality of our RBM is its batch-level regularization scheme which is specifically designed for hashing. We subsequently refer to this variant as RBMH.

An RBM is a bipartite Markov random field with the input layer $x \in R^I$ connected to a latent layer $z \in R^J$ via a set of undirected weights $W \in R^{IJ}$. The input and latent layers are also parameterised by their corresponding biases c and b, respectively. Since

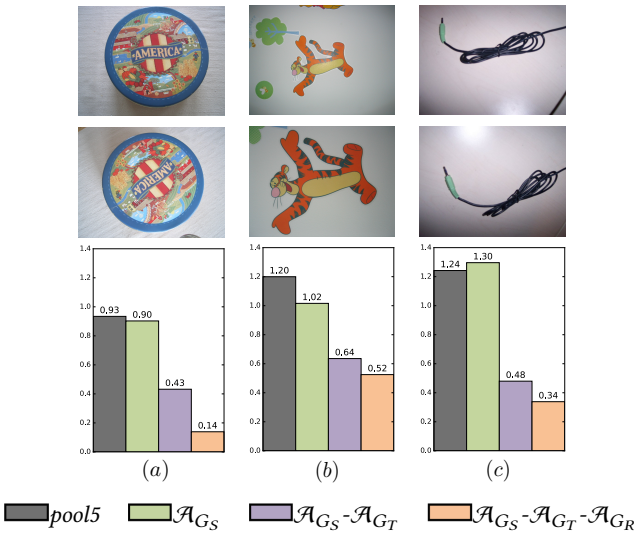


Figure 4: Distances for 3 matching pairs from the *UKB* dataset. For each pair, 4 pairwise distances (L_2-normalized) are computed corresponding to the following descriptors: *pool5*, $\mathcal{A}_{G_S}$, $\mathcal{A}_{G_S}$-$\mathcal{A}_{G_T}$ and $\mathcal{A}_{G_S}$-$\mathcal{A}_{G_T}$-$\mathcal{A}_{G_R}$. Adding scale invariance makes the most difference on (b), translation invariance on (c), and rotation on (a) which is consistent with the scenarios suggested by respective image pairs.

the units within a layer are conditionally independent pairwise, the activation probabilities of one layer can be sampled by fixing the states of the other layer, and using distributions given by logistic functions (a sigmoid activation function is chosen since binary hashes are desired):

$$P(z_j|x) = 1/(1 + \exp(-w_j x - b_j)), \tag{5}$$

$$P(x_i|z) = 1/(1 + \exp(-w_i^\top z - c_i)). \tag{6}$$

As a result, alternating Gibbs sampling can be performed between the two layers. The sampled states are used to update the parameters $\{W, b, c\}$ through batch gradient descent using the contrastive divergence algorithm [19] to approximate the maximum likelihood of the input distribution. The hashed descriptors are obtained by binarizing the latent units at 0.5.

Proper regularization is key during the training of RBM. The popular RBM proposed by Nair and Hinton [28] encourages latent representations to be sparse. This improves separability which is desirable for classification task. For hashing, it is desirable to encourage the representation to make efficient use of the limited latent subspace. RBMH achieves this goal by controlling sparsity in a way to maximize the entropy not only within every hash but also between the same bit of different hashes. This effectively encourages (a) half the bits to be active for a given hash, and (b) each hash bit to be equiprobable across images. We introduce a regularization term at the batch as in [13]. For a batch B, we define a regularization term:

$$h(B) = \sum_{x_\alpha \in B}\sum_{j\alpha} t_{j\alpha} \log z_{j\alpha} + (1 - t_{j\alpha}) \log(1 - z_{j\alpha}), \tag{7}$$

where t_α are the target activations for each data sample α. We choose the $t^l_{j\alpha}$ such that each $\{t^l_{j\alpha}\}_j$ for fixed α and each $\{t^l_{j\alpha}\}_\alpha$ for fixed j is distributed according to the uniform distribution $U(0, 1)$

effectively maximizing entropy. The overall objective function becomes:

$$\arg\min_{\{W,b,c\}} -\sum_{\alpha} \log\Big(\sum_{z_\alpha \in B} P(x_\alpha, z_\alpha) + \lambda h(B)\Big), \tag{8}$$

with λ the regularization constant.

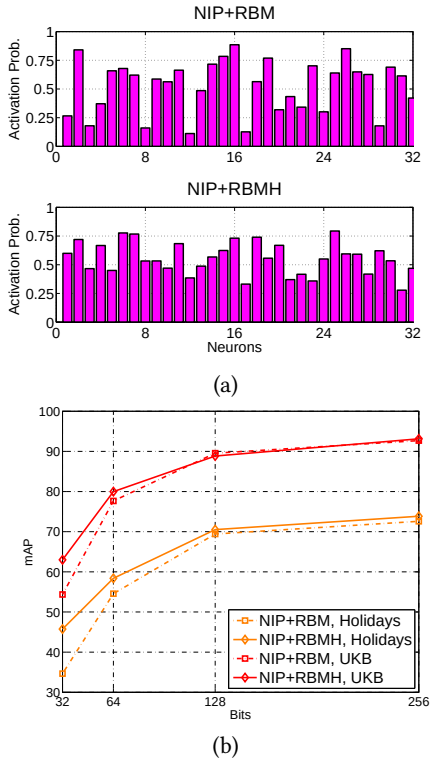


Figure 5: (a) Activation probabilities of hash bits between RBMH and the RBM proposed by Nair&Hinton [28]. We compute the statistics with 32 bits binary hashes on *Holidays* dataset (1491 images in total). (b) Comparison of our RBMH with RBM [28] in terms of mAP on *Holidays* and *UKB*. Both schemes are built upon the best NIP descriptors. RBMH outperforms RBM at very low code sizes.

Figure 5 shows the activation probabilities of the hash bits between RBMH and the RBM proposed by Nair and Hinton [28]. The comparison is for 32-bits hashes. In Figure 5 (a), the mean probability of activation is nearly 0.5 in both cases. Nevertheless, we can see that probabilities are much more evenly distributed across bits with BRMH. In Figure 5 (b), retrieval results on *UKB* and *Holidays* show that the RBMH is able to outperform the standard RBM specially at lower code sizes (32 or 64 bits).

5 EXPERIMENTS

5.1 Evaluation Framework

We evaluate the performances on 4 popular datasets for image instance retrieval: (1) **Holidays**. The INRIA Holidays dataset [21] consists of outdoor holiday pictures. There are 500 queries and 991 database images. (2) **UKB**. The University of Kentucky Benchmark dataset [29] consists of 2550 groups of common objects, 4 images per object. All 10200 images are used as queries. (3) **Oxford5K**. The Oxford buildings dataset [32] consists of 5063 images representing landmark buildings in Oxford. The query set contains 11 different landmarks, each represented by 5 queries. (4) **Graphics**. The Graphics dataset is part of the Stanford Mobile Visual Search dataset [7], which was used in the MPEG standard titled Compact Descriptors for Visual Search (CDVS) [20]. This dataset contains objects like CDs, DVDs, books, prints, business cards. There are 500 unique objects, 1500 queries, and 1000 database images. Note that *Graphics* is different from the other datasets as it contains images of rigid objects captured under widely varying scale and rotation changes.

For large-scale experiments, we present results on the 4 datasets combined with the 1 million MIR-FLICKR distractor images [27]. Most schemes, including our approach, require an unsupervised training step. We train on a randomly sampled set of 150K images from the 1.2 million *ImageNet* dataset [10]. No class labels are used in this work.

For the starting global descriptor representation, we use the *pool5* layer from the 16-layer *OxfordNet* [36], which is widely adopted in instance retrieval literature [5, 35, 40]. The input image size for *OxfordNet* is fixed at 224×224. The dimensionality of the *pool5* descriptor is 25088, organized as 512 feature maps of size 7×7.

For rotation invariance, rotated input images are padded with the mean pixel value computed from the ImageNet dataset. The step size for rotations is 10 degrees yielding 36 rotated images per orbit. For scale changes, 10 different center crops are considered varying from 50% to 100% of the total image. For translations, the entire feature map is used for every feature, resulting in an orbit size of $7 \times 7 = 49$.

For retrieval with floating point descriptors, L_2 normalization is first applied followed by L2 distance computation. For retrieval with binary descriptors, we use hamming distance computation. We evaluate retrieval results with mean Average Precision (mAP). To be consistent with the literature, 4× Recall @ $R = 4$ is provided for *UKB*.

Table 1: Retrieval results (mAP) for different sequences of transformation groups and moments. For *UKB*, 4× Recall @ $R = 4$ is shown between parentheses. G_T, G_R, G_S denote the groups of translations, rotations and scale changes respectively. Note that averages commute with other averages so the sequence order of the composition does not matter when only averages are involved. Best results are achieved by choosing specific moments. $\mathcal{A}_{G_S}$-$\mathcal{S}_{G_T}$-$\mathcal{M}_{G_R}$ corresponds to the best average performer. *fc6* and *pool5* are provided as a baseline.

Sequence	Dims	Dataset			
		Oxford5K	Holidays	UKB	Graphics
pool5	25088	0.427	0.707	0.823(3.11)	0.315
fc6	4096	0.461	0.782	0.910(3.50)	0.312
$\mathcal{A}_{G_T}$	512	0.477	0.800	0.924(3.56)	0.322
$\mathcal{A}_{G_R}$	25088	0.462	0.779	0.954(3.72)	0.500
$\mathcal{A}_{G_S}$	25088	0.430	0.716	0.828(3.12)	0.394
$\mathcal{A}_{G_T}$-$\mathcal{A}_{G_R}$	512	0.418	0.796	0.955(3.73)	0.417
$\mathcal{A}_{G_T}$-$\mathcal{A}_{G_S}$	512	0.537	0.811	0.931(3.61)	0.430
$\mathcal{A}_{G_R}$-$\mathcal{A}_{G_S}$	25088	0.494	0.815	0.959(3.75)	0.552
$\mathcal{A}_{G_T}$-$\mathcal{A}_{G_R}$-$\mathcal{A}_{G_S}$	512	0.484	0.833	0.971(3.82)	0.509
$\mathcal{A}_{G_S}$-$\mathcal{S}_{G_T}$-$\mathcal{M}_{G_R}$	512	**0.592**	**0.838**	**0.975(3.84)**	**0.589**

5.2 Results

Evaluation of NIP descriptors. As shown in Table 1, we first study the effects of incorporating various transformation groups and using different moments on NIP descriptors. We present results for all possible combinations of transformation groups for average pooling (order does not matter as averages commute) and for the single best performer which is $\mathcal{A}_{G_S}$-$\mathcal{S}_{G_T}$-$\mathcal{M}_{G_R}$ (order matters). First, we point out the effectiveness of the *pool5* layer. Although it performs notably worse than *fc6* as-is, a simple average pooling over the space of translations $\mathcal{A}_{G_T}$ makes it both better and 8 times more compact than *fc6*. Similar observations have also been reported by [4, 5]. Second, on average, accuracy significantly increases with the number of transformation groups involved. Third, choosing statistical moments different than averages further improve the retrieval results. In Table 1, we observe that $\mathcal{A}_{G_S}$-$\mathcal{S}_{G_T}$-$\mathcal{M}_{G_R}$ performs significantly better than $\mathcal{A}_{G_T}$-$\mathcal{A}_{G_R}$-$\mathcal{A}_{G_S}$. Notably, the best combination corresponds to an increase in the orders of the moments: $\mathcal{A}$ being a first-order moment, $\mathcal{S}$ second order and $\mathcal{M}$ of infinite order. A different way of stating this is that a more invariant representation requires higher and higher orders of pooling.

Table 2: Retrieval performance comparing NIP to other state-of-the-art methods. We include results in recent papers with comparable dimensionality of descriptors reported in those papers. L2 distance is used for all methods.

Method	Dims	Dataset		
		Oxford5K	Holidays	UKB
T-embedding [23]	1024	0.560	0.720	3.51
T-embedding [23]	512	0.528	0.700	3.49
FV+Proj [17]	512	-	0.789	3.36
FC+PCAWhitening [34]	500	0.322	0.642	-
FC+VLAD+PCA [43]	512	-	0.784	-
FC+Finetune+PCAWhitening [6]	512	0.557	0.789	3.30
Conv+MaxPooling [35]	256	0.533	0.716	-
FV+FC+PCAWhitening [31]	512	-	0.827	3.37
Conv+SPoC+PCAWhitening [5]	256	0.589	0.802	3.65
R-MAC+PCAWhitening [40]	512	**0.668**	-	-
R-MAC+PCAWhitening [40]	256	0.561	-	-
NIP (Ours)	512	0.592	**0.838**	**3.84**
NIP+PCAWhitening (Ours)	256	**0.609**	**0.836**	**3.83**

Overall, $\mathcal{A}_{G_S}$-$\mathcal{S}_{G_T}$-$\mathcal{M}_{G_R}$ improves results over starting representation *pool5* by 39% (*Oxford5K*) to 87% (*Graphics*) depending on the dataset. Better improvements with *Graphics* can be explained with the presence of many rotations in the dataset (smaller objects taken under different angles) while *Oxford5K* consisting mainly of upright buildings is less significantly helped by incorporating rotation invariance.

Comparing NIP with state-of-the-art including variants of VLAD/FV [17, 23], deep descriptors [5, 6, 35, 40] and descriptors combining deep CNN and VLAD/FV [31, 43]. As shown in Table 2, we observe that 512-D NIP descriptors largely outperform most state-of-the-art methods with 512 or higher dimensions, on all datasets. Following [5, 35, 40], we also perform PCA whitening (PCAW) to reduce the dimensionality of NIP to 256. One can see that the 256-D NIP descriptors yield superior performance to [5, 35, 40] on all datasets.

First, we compare NIP to the most related papers [5,35,40] which propose 256-D deep descriptors by aggregating convolutional features with various pooling operations [1]. [4, 5, 35] can be considered a special case of our work, with just one layer of pooling, which only provided limited levels of translation invariance. The very recently proposed Regional Maximum Activation of Convolutions (R-MAC) [40] reports outstanding results on building dataset *Oxford5K* with very small dimensionality (e.g. 0.668 mAP for 512-D R-MAC and 0.561 mAP for 256-D R-MAC). The authors propose a fast R-CNN type pooling [11], which is effective when the object of interest is in a small portion of the image. Such an approach will be less effective when the object of interest is affected by groups of distortions like rotation and perspective, and located at the centre of the image. Here, we observe that nested pooling over many types of distortions with progressively increasing moments is essential to achieving geometric invariance and high retrieval performance with low dimensional descriptors. Besides, we argue that the technique proposed in [40] can be incorporated with NIP to further improve performance.

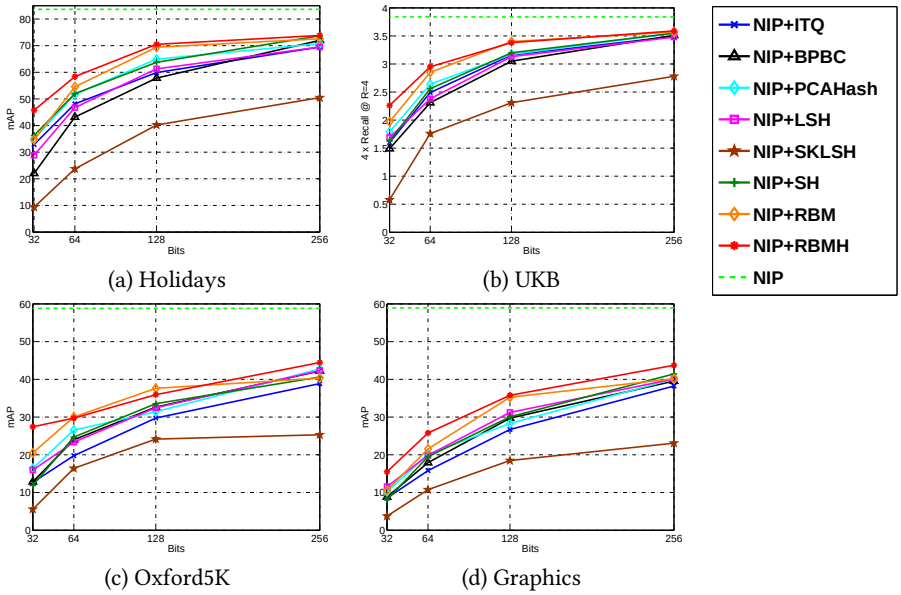
Next, we note that [35] reports better results on *Holidays* (0.881 mAP) and *Oxford5K* (0.844 mAP), with very high-dimensional descriptors (from 10K to 100K). These very high dimensional descriptors are obtained by combining CNN descriptors with spatial max pooling [4]. In contrast, our results are generated using only 256 to 512 dimensional descriptors.

Table 3: Retrieval performance comparing NIP+RBMH to other state-of-the-art methods at small codesizes (from 32 to 512 bits). ADC denotes asymmetric distance computation [16,22]. "FT" stands for fine-tuning.

Method	Dims (bits)	Dist.	Dataset		
			Oxford5K	Holidays	UKB
Binarized FV [30]	520(520)	Cosine	-	0.460	2.79
FV+SSH [16]	256(256)	ADC	-	0.544	3.08
FV+SSH [16]	128(128)	ADC	-	0.499	2.91
FV+SSH [16]	32(32)	ADC	-	0.334	2.18
FV+PQ [22]	128(128)	ADC	-	0.506	3.10
VLAD+PQ [45]	128(128)	L2	-	0.586	2.88
VLAD+CQ [45]	128(128)	L2	-	0.644	3.19
VLAD+SQ [46]	128(128)	L2	-	0.639	3.06
FC+FT+PCAW [6]	16(512)	L2	0.418	0.609	2.41
Conv+MaxPooling [35]	256(256)	Cosine	0.436	0.578	-
Binarized NIP (Ours)	512(512)	Hamming	**0.477**	**0.781**	**3.70**
NIP+RBMH (Ours)	256(256)	Hamming	**0.445**	**0.739**	**3.59**
NIP+RBMH (Ours)	128(128)	Hamming	**0.359**	**0.705**	**3.38**
NIP+RBMH (Ours)	32(32)	Hamming	**0.274**	**0.458**	**2.26**

Evaluation of NIP+RBMH binary hashes. NIP+RBMH binary hashes are produced by feeding invariant NIP descriptors into the proposed RBM hashing layer. Small scale retrieval results with NIP+RBMH are shown in Figure 6. We compare NIP+RBMH to other popular unsupervised hashing methods at code sizes

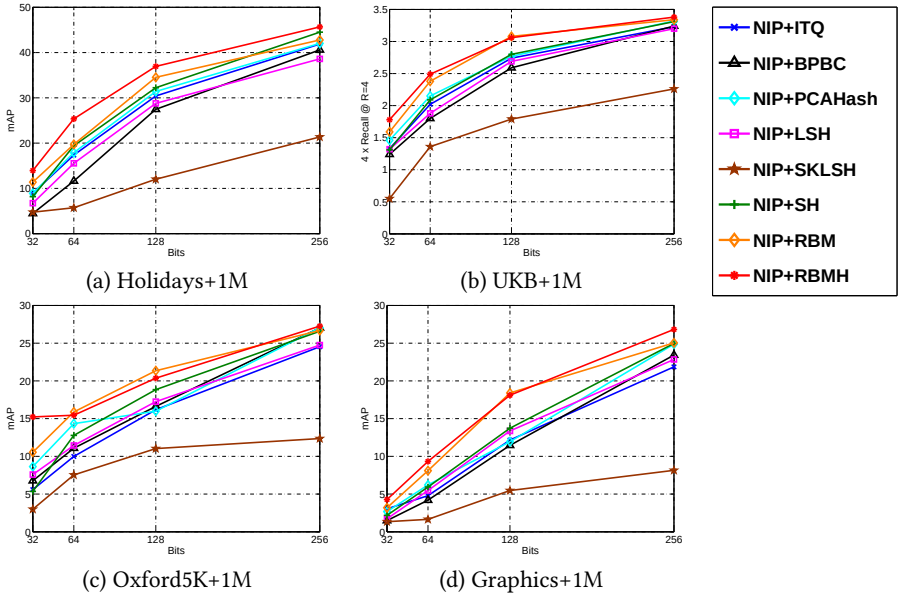
[1] Note that [5, 35, 40] extract deep descriptors from images with size larger than 576×576, while we use 224×224 in this work. As shown in [6], there is potential improvement if larger image size adopted in deep descriptors extraction.



(a) Holidays (b) UKB

(c) Oxford5K (d) Graphics

Figure 6: Comparison of RBMH with other hashing methods on 4 benchmark datasets. All methods are built upon the best NIP descriptors. To examine the effect of compression, we also present retrieval results using uncompressed NIP descriptors.



(a) Holidays+1M (b) UKB+1M

(c) Oxford5K+1M (d) Graphics+1M

Figure 7: Comparison of RBMH with other hashing methods on large scale retrieval experiments. All methods are based on the best NIP descriptors.

from 32 to 256 bits, including *ITQ* [15], Bilinear Projection Binary Codes (*BPBC*) [14], *PCAHash* [15], *LSH* [9], *SKLSH* [33], *SH* [42] and *RBM* [18, 28]. We used the software provided by the authors in [15, 18] to generate results for the baseline hashing methods. In addition, we also include the results of 512-D NIP descriptors, as the baseline uncompressed scheme.

We observe that NIP+RBMH outperforms other methods at most code sizes on all data sets. First, there is a significant improvement at smaller code sizes like 32 bits, due to the proposed batch-level regularization: 0.457 vs. 0.369 in terms of mAP, compared to the second performing method RBM on *Holidays* at 32 bits. Second, the improvements of NIP+RBMH over other methods becomes smaller as code size increases (except SKLSH). For code size larger than 256 bits, the performances of all methods approach the upper bound, i.e., uncompressed NIP descriptors. Finally, compared to uncompressed NIP descriptors, there is a marginal drop for all methods on *UKB* at 256 bits, while performance gap is larger for other datasets.

Comparing NIP+RBMH with state-of-the-art including methods compressing VLAD/FV with direct binarization [30], hashing [16] and PQ [22, 46], methods based on compact deep descriptors [6, 35].

As shown in Table 3, first, a simple binarization strategy applied to our best performing NIP descriptor is sufficient to obtain significantly better accuracy than [6, 30] at comparable code size (512 bits), e.g., 3.7 vs. 2.79 in [30] for 4× Recall @ $R = 4$ on *UKB*. Second, NIP+RBMH outperforms state-of-the-art by a significant margin at comparable code sizes (from 32 to 256 bits). NIP+RBMH achieves the best performance on *Holidays* at small code size (128 bits), 0.705 vs. 0.644 mAP reported in [45] (to our knowledge, the state-of-the-art on this dataset with 128-bit descriptors). Note that Hamming distance is used for our binary descriptors, while other methods like PQ variants employ Euclidean distances (L2 or ADC), which typically result in higher accuracy than Hamming distance, at the expense of higher computational cost.

Large scale experiments. In Figure 7, we present large scale retrieval results, combining the 1 million MIR FLICKR distractor images with each data set respectively. Trends consistent with small scale retrieval results in Figure 6 are observed.

6 CONCLUSIONS

In this work, we proposed a method to produce global image descriptors from CNNs which are both compact and robust to typical geometric transformations. The method provides a practical and mathematically proven way for computing invariant object representations with feed-forward neural networks. To achieve global geometric invariance, we introduce a series of nested pooling layers at intermediate levels of the deep CNN network. We further introduce a RBM layer with a novel batch-level regularization scheme for generating compact binary descriptors. Through a thorough evaluation with state-of-the-art, we show that the proposed method outperforms state-of-the-art by a significant margin.

7 ACKNOWLEDGMENTS

This work was partially supported by grants from National Natural Science Foundation of China (U1611461, 61661146005) and National Hightech R&D Program of China (2015AA016302).

REFERENCES

[1] F. Anselmi, J. Z. Leibo, L. Rosasco, J. Mutch, A. Tacchetti, and T. Poggio. Magic materials: a theory of deep hierarchical architectures for learning sensory representations. *CBCL paper*, 2013.

[2] F. Anselmi, J. Z. Leibo, L. Rosasco, J. Mutch, A. Tacchetti, and T. Poggio. Unsupervised learning of invariant representations in hierarchical architectures. *arXiv:1311.4158*, 2013.

[3] F. Anselmi and T. Poggio. Representation learning in sensory cortex: a theory. *CBMM memo n 26*, 2010.

[4] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In *Computer Vision and Pattern Recognition Workshops*, 2015.

[5] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *International Conference on Computer Vision (ICCV)*, 2015.

[6] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural Codes for Image Retrieval. In *European Conference on Computer Vision (ECCV)*, 2014.

[7] V. Chandrasekhar, D.M.Chen, S.S.Tsai, N.M.Cheung, H.Chen, G.Takacs, Y.Reznik, R.Vedantham, R.Grzeszczuk, J.Back, and B.Girod. Stanford Mobile Visual Search Data Set. In *ACM Multimedia Systems Conference (MMSys)*, 2011.

[8] D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod. Residual Enhanced Visual Vector as a Compact Signature for Mobile Visual Search. In *Signal Processing*, 2012.

[9] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-Sensitive Hashing Scheme based on p-stable Distributions. In *Annual Symposium on Computational Geometry*, 2004.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[11] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[13] Hanlin Goh, Nicolas Thome, Matthieu Cord, and Joo-Hwee Lim. Unsupervised and supervised visual codes with restricted Boltzmann machines. In *European Conference on Computer Vision (ECCV)*, 2012.

[14] Yunchao Gong, Sanjiv Kumar, Henry Rowley, and Svetlana Lazebnik. Learning Binary Codes for High-Dimensional Data Using Bilinear Projections. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[15] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-scale Image Retrieval. In *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

[16] Albert Gordo, Florent Perronnin, Yunchao Gong, and Svetlana Lazebnik. Asymmetric distances for binary embeddings. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(1):33–47, 2014.

[17] A. Gordoa, J.A. Rodriguez-Serrano, F. Perronnin, and E. Valveny. Leveraging category-level labels for instance-level image retrieval. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[19] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[20] ISO/IEC-JTC1/SC29/WG11/N12202. *Evaluation Framework for Compact Descriptors for Visual Search*, 2011.

[21] H. Jégou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *European Conference on Computer Vision (ECCV)*, 2008.

[22] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(9):1704–1716, 2012.

[23] Hervé Jégou and Andrew Zisserman. Triangulation embedding and democratic aggregation for image search. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, 2012.

[25] Q. L. Liao, J. Z. Leibo, and T. Poggio. Learning invariant representations and applications to face verification. In *Neural Information Processing Systems (NIPS)*, 2013.

[26] Jie Lin, Ling-Yu Duan, Tiejun Huang, and Wen Gao. Robust Fisher Codes for Large Scale Image Retrieval. In *International Conference on Acoustics and Signal Processing (ICASSP)*, 2013.

[27] B. Thomee Mark J. Huiskes and Michael S. Lew. New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative. In *ACM International Conference on Multimedia Information Retrieval*, 2010.

[28] Vinod Nair and Geoffrey Hinton. 3D Object Recognition with Deep Belief Nets. In *Neural Information Processing Systems (NIPS)*, 2009.

[29] D. Nistér and H. Stewénius. Scalable Recognition with a Vocabulary Tree. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[30] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale Image Retrieval with Compressed Fisher Vectors. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[31] Florent Perronnin and Diane Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[32] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[33] M. Raginsky and S. Lazebnik. Locality-Sensitive Binary Codes from Shift-Invariant Kernels. In *Neural Information Processing Systems (NIPS)*, 2009.

[34] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Computer Vision and Pattern Recognition Workshops*, 2014.

[35] Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. A baseline for visual instance retrieval with deep convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2015.

[36] K Simonyan and A Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[37] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[38] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[39] G. Tolias, Y. Avrithis, and H. Jegou. To aggregate or not to aggregate: Selective match kernels for image search. In *International Conference on Computer Vision (ICCV)*, 2013.

[40] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *arXiv:1511.05879*, 2015.

[41] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[42] Y. Weiss, A. Torralba, and R. Fergus. Spectral Hashing. In *Neural Information Processing Systems (NIPS)*, 2008.

[43] R. Guo Y. Gong, L. Wang and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European Conference on Computer Vision (ECCV)*, 2014.

[44] C. Zhang, G. Evangelopoulos, S. Voinea, L. Rosasco, and T. Poggio. A deep representation for invariance and music classification. In *International Conference on Acoustics and Signal Processing (ICASSP)*, 2014.

[45] Ting Zhang, Chao Du, and Jingdong Wang. Composite quantization for approximate nearest neighbor search. In *International Conference on Machine Learning (ICML)*, 2014.

[46] Ting Zhang, Guo-Jun Qi, Jinhui Tang, and Jingdong Wang. Sparse composite quantization. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.