Multimodal similarity-preserving hashing

Jonathan Masci, Michael M. Bronstein, *Senior Member, IEEE*, Alexander M. Bronstein, *Senior Member, IEEE*, Jürgen Schmidhuber.

Abstract—We introduce an efficient computational framework for hashing data belonging to multiple modalities into a single representation space where they become mutually comparable. The proposed approach is based on a novel coupled siamese neural network architecture and allows unified treatment of intra- and inter-modality similarity learning. Unlike existing cross-modality similarity learning approaches, our hashing functions are not limited to binarized linear projections and can assume arbitrarily complex forms. We show experimentally that our method significantly outperforms state-of-the-art hashing approaches on multimedia retrieval tasks.

Index Terms—similarity-sensitive hashing, metric learning, feature descriptor

1 Introduction

Efficient computation of similarity between entries in large-scale databases has attracted increasing interest, given the explosive growth of data that has to be collected, processed, stored, and searched for. In particular, in the computer vision and pattern recognition community, this problem arises in applications such as image-based retrieval, ranking, classification, detection, tracking, and registration. In all these problems, given a query object (usually represented as a feature vector), one has to determine the closest entires (nearest neighbors) in a large database.

An even more challenging setting frequently arises in tasks involving multiple media or data coming from different modalities [30], [38]. For example, a medical image of the same organ can be obtained using different physical processes such as CT and MRI; a multimedia search engine may perform queries in a corpus consisting of audio, video, and textual information.

Since the notion of visual objects similarity is rather elusive and cannot be measured explicitly, one often resorts to machine learning techniques that allow constructing similarity from data examples. Such methods are generally referred to as *similarity* or *metric learning*.

J. Masci and J. Schmidhuber are with the Swiss AI Lab, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Manno, Switzerland; M. M. Bronstein is with Università della Svizzera Italiana, Lugano, Switzerland and Intel Semiconductor, Switzerland; A. M. Bronstein is with the School of Electrical Engineering, Tel Aviv University, Israel and Intel Semiconductor, Israel.

Previous work. Traditionally, similarity learning methods can be divided into unsupervised and supervised, with the former relying on the data only without using any side information. PCA-type methods [36] use global structure of the data, while manifold learning techniques such as locally linear embedding [32], eigenmaps [3], and diffusion maps [8] consider data as low-dimensional manifold and use its local intrinsic structure to represent similarity. On the other hand, supervised methods assume additional information is provided together with the data examples. Such information can come in the form of class labels [14], [27], [45], [49], distances [5], similar and dissimilar pairs [9] or order relations [25], [39]. In practice, many similarity learning methods use some representation of the distance, e.g. in the form of a parametric embedding from the original data space to some target space. In the simplest case, such an embedding is a linear projection acting as dimensionality reduction, and the metric of the target space is Euclidean or Mahalanobis distance [39], [45].

More recently, there has been an increased interest in similarity learning methods based on embedding the data in spaces of binary codes with e.g. the Hamming metric [11], [12], [17], [22], [28], [29], [34], [44]. Such an embedding can be considered as a hashing function acting on the data trying to preserve some underlying similarity. Notable examples of the unsupervised setting of this problem include locality sensitive hashing (LSH) [1], [10] and spectral-type hashing [23], [46], which try to approximate some trusted standard similarity such as the Jaccard index or the cosine distance. Shakhnarovich et al. [37] proposed to construct optimal LSH-like hashes (referred to as *similarity-sensitive hashing* or SSH) for data with given binary similarity function using boosting, considering each dimension of the hashing function as a weak classifier. In the same setting, a simple method based on eigendecomposition of covariance matrices of positive and negative samples was proposed by [40]. Masci et al. [24] posed the problem as a neural network learning. Hashing methods have been used successfully in various vision applications such large-scale retrieval [43], feature descriptor learning [40], [24], image matching [15] and alignment [6].

The extension of similarity learning to multimodal data has been addressed in the literature only very recently. Bronstein et al. [6] proposed an extension of the SSH to the crossmodal setting, dubbed CM-SSH. McFee et al. [26] proposed to learn multimodal similarity using ideas from multiple kernel learning [2], [25]. Multimodal kernel learning approaches have been proposed in [21] for medical image registration. Weston et al. [48] used multimodal embeddings for image annotation. The main disadvantage of the latter is the fact that it is limited to linear projections only. The framework proposed in [26] can be kernelized, but it involves the computationally expensive semidefinite programming, which limits scalability. Also, both algorithms produce continuous Mahalanobis metrics, disadvantageous in computational and storage complexity, especially when dealing with large-scale data.

The appealing property of crossmodal similarity-preserving hashing methods like the CM-SSH [6] is the compactness of the representation and the low complexity involved in distance computation. However, CM-SSH is limited to linear projections which may not capture the structure of the data. Furthermore, it accounts only for the similarity *across* modalities, completely ignoring the data similarity *within* each modality. Finally, CM-SSH uses relaxation to solve the underlying optimization problem.

Contributions. We propose a novel multimodal similarity learning framework based on neural networks, that tries to simultaneously learn two (or more) hashing functions that map the different modalities into a common binary space. Our approach has several advantages over the state-of-the-art. First, we combine intra- and inter-modal similarity into a single framework. This allows exploiting richer information about the data and can tolerate missing modalities. We show that several previous works can be considered as particular cases of our model. Second, our approach produces compact binary code representation of the data, thus reducing storage and computational complexity of the similarity function, and is better amenable for efficient indexing. Third, we solve the full optimization problem without resorting to relaxations as in SSH-like methods; it has been recently shown that such a relaxation degrades the hashing performance [24], [40]. Fourth, we introduce a novel coupled siamese neural network architecture to solve the optimization problem underlying our multimodal hashing framework. Finally, the use of neural networks can be very naturally generalized to more complex non-linear projections using multi-layered networks, thus allowing embeddings of arbitrarily high complexity. We show experimental result on several standard multimodal datasets demonstrating that our approach compares favorably to state-of-the-art algorithms.

2 Background

Let $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^{n'}$ be two spaces representing data belonging to different modalities (e.g., X are images and Y are text descriptions). Note that even though we assume that the data can be represented in the Euclidean space, the similarity of the data is not necessarily Euclidean and in general can be described by some metrics $d_X : X \times X \to \mathbb{R}_+$ and $d_Y : Y \times Y \to \mathbb{R}_+$, to which we refer as *intra-modal dissimilarities*. Furthermore, we assume that there exists some *inter-modal dissimilarity* $d_{XY} : X \times Y \to \mathbb{R}_+$ quantifying the "distance" between points in different modality. To deal with these structures in a more convenient way, we try to represent them in a common metric space. In particular, the choice of the Hamming space offers significant advantages in the compact representation of the data as binary vectors and the efficient computation of their similarity.

Unimodal (or *single-modality*) *similarity-preserving hashing* is the problem of representing data from one modality (say, X) in the space $\mathbb{H}^m = \{\pm 1\}^m$ of m-dimensional binary vectors with the Hamming metric $d_{\mathbb{H}^m}(a,b) = \frac{m}{2} - \frac{1}{2}\sum_{i=1}^m a_i b_i$ by means of an embedding, $\xi : X \to \mathbb{H}^m$ mapping similar points as close as possible to each other and dissimilar points as distant as possible from each other, such that $d_{\mathbb{H}^m} \circ (\xi \times \xi) \approx d_X$.

Multimodal similarity-preserving hashing is an extension of the former problem, in which two different modalities X, Y are represented in the common space $\mathbb{H}^m$ by means of two embeddings, $\xi : X \to \mathbb{H}^m$ and $\eta : Y \to \mathbb{H}^m$ mapping similar points as close as possible to each other and dissimilar points as distant as possible from each other, such that $d_{\mathbb{H}^m} \circ (\xi \times \xi) \approx d_X$, $d_{\mathbb{H}^m} \circ (\eta \times \eta) \approx d_Y$, and $d_{\mathbb{H}^m} \circ (\xi \times \eta) \approx d_{XY}$. In a sense, the embeddings act as a *metric coupling*, trying to construct a single metric that preserves the intra- and inter-modal similarities. A simplified setting of the multimodal hashing problem used in [6] is *cross-modality similarity-preserving hashing*, in which only the inter-modal dissimilarity d_{XY} is taken into consideration and d_X, d_Y are ignored.

In the rest of this paper, we assume binary dissimilarities $d_X, d_Y, d_{XY} \in \{0, 1\}$, i.e., a pair of points can be either similar or dissimilar. This dissimilarity is usually unknown and hard to model, however, it should be possible to sample d_X, d_Y, d_{XY} on some subset of the data $X' \subset X, Y' \subset Y$. This sample can be represented as sets of similar pairs of points (*positives*) $\mathcal{P}_X = \{(x \in X', x', \in X') : d_X(x, x') = 0\}$, $\mathcal{P}_Y = \{(y \in Y', y' \in Y') : d_Y(y, y') = 0\}$, and $\mathcal{P}_{XY} = \{(x \in X', y \in Y') : d_{XY}(x, y) = 0\}$, and likely defined sets $\mathcal{N}_X, \mathcal{N}_Y$, and $\mathcal{N}_{XY}$ of dissimilar pairs of points (*negatives*). In many practical applications such as image annotation or text-based image search, it might be hard to get the inter-modal positive and negative pairs, but easy to get the intra-modal ones.

The problem of multimodal similarity-preserving hashing boils down to finding two embeddings $\xi : X \to \mathbb{H}^m$ and $\eta : Y \to \mathbb{H}^m$ minimizing the aggregate of false positive and false negative rates,

$$\begin{aligned}\min_{\xi,\eta} \quad & \mathbb{E}\{d_{\mathbb{H}^m} \circ (\xi \times \xi)|\mathcal{P}_X\} + \mathbb{E}\{d_{\mathbb{H}^m} \circ (\eta \times \eta)|\mathcal{P}_Y\} - \\ & \mathbb{E}\{d_{\mathbb{H}^m} \circ (\xi \times \xi)|\mathcal{N}_X\} - \mathbb{E}\{d_{\mathbb{H}^m} \circ (\eta \times \eta)|\mathcal{N}_Y\} + \\ & \mathbb{E}\{d_{\mathbb{H}^m} \circ (\xi \times \eta)|\mathcal{P}_{XY}\} - \\ & \mathbb{E}\{d_{\mathbb{H}^m} \circ (\xi \times \eta)|\mathcal{N}_{XY}\}. \end{aligned} \quad (1)$$

In what follows, we briefly review the existing approaches to supervised similarity-preserving hashing.

2.1 Single-modality similarity-preserving hashing

In his Ph.D. dissertation, Shakhnarovich [37] introduced one of the first supervised hashing techniques called similarity-preserving hashing (SSH). The author proposed to regard the construction of an LSH-like similarity-preserving hash as a binary classification problem, in which pairs of points $(\mathbf{x}, \mathbf{x}')$ are assigned positive or negative labels. The minimization of the expected Hamming distance $d_{\mathbb{H}^m}$ on the set of positive pairs (and, respectively, its maximization on the negative set) can be achieved by minimizing the exponential loss of the form

$$\mathbb{E}\left\{\exp(-\ell\xi(\mathbf{x})^{\mathrm{T}}\xi(\mathbf{x}'))\right\} = \mathbb{E}\left\{\prod_{i=1}^{M}\exp(-\ell\xi_i(\mathbf{x})\xi_i(\mathbf{x}'))\right\},$$

where $\ell = +1$ for a positive pair, and $\ell = -1$ for a negative one. Observing the above separability of the exponential loss, the author proposed to train the individual bits ξ_i of the embedding sequentially as *weak learners* using standard boosting techniques. In particular, Shakhnarovich considered linear embeddings of the form $\mathbf{x}_i(\mathbf{x}) = \mathrm{sign}(\mathbf{e}_{k_i}^{\mathrm{T}}\mathbf{x} + a_i)$, where $\mathbf{e}_{k_i}$ is a standard basis vector acting as a feature selector, and a_i is a threshold.

The sequential construction of binary codes is clearly suboptimal. As the result, SSH typically requires relatively long codes to achieve good performance. A remedy to this problem was proposed in the DiffHash scheme introduced by Strecha *et al.* [40]. The authors considered linear embeddings of the form $\xi(\mathbf{x}) = \mathrm{sign}(\mathbf{P}\mathbf{x}+\mathbf{a})$ trained by minimizing a quadratic loss

$$\mathbb{E}\left\{\|\xi(\mathbf{x}) - \xi(\mathbf{x}')\|_2^2|\mathcal{P}\right\} - \alpha\mathbb{E}\left\{\|\xi(\mathbf{x}) - \xi(\mathbf{x}')\|_2^2|\mathcal{N}\right\}, \quad (2)$$

with the parameter α controlling the relative importance of false positives and negatives. By relaxing the problem through the removal of the sign function, $\mathbf{P}$ can be found as the m smallest negative eigenvectors of the difference of the covariance matrices $\mathbf{C}_{\mathcal{P}} - \alpha\mathbf{C}_{\mathcal{N}}$, with $\mathbf{C}_{\mathcal{P}} = \mathbb{E}\{(\mathbf{x}-\mathbf{x}')(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}|\mathcal{P}\}$ and $\mathbf{C}_{\mathcal{N}}$ defined likely on the negative pairs. Once the projection matrix $\mathbf{P}$ has been found, the thresholds $\mathbf{a}$ are found by solving m independent one-dimensional minimization problems. The authors showed that a globally optimal a_i can be computed from the cumulative histograms of $\mathbf{p}_i^{\mathrm{T}}\mathbf{x}$.

Despite its simplicity and computational efficiency, the main drawback of DiffHash is the fact that it is limited to linear projections, which might not be able to properly capture the intricate structure of the data. In machine learning, it is common to introduce non-linearity into linear projection-based schemes via the kernel trick. Generalizing kernelized LSH [18] to the supervised setting, Liu *et al.* [22] proposed the kernelized supervised hashing (KSH) scheme, in which they considered embeddings of the form $\xi(\mathbf{x}) = \mathrm{sign}(\mathbf{P}\mathbf{k}(\mathbf{x}))$, with $\mathbf{P}$ being an $m \times r$ projection matrix, and $\mathbf{k}(\mathbf{x}) = (\kappa(\mathbf{x}, \mathbf{x}_1) - \mu_1, \ldots, \kappa(\mathbf{x}, \mathbf{x}_r) - \mu_r)^{\mathrm{T}}$ a non-linear map created by computing the inner product between $\mathbf{x}$ and a fixed set of r points $\mathbf{x}_1, \ldots, \mathbf{x}_r$ drawn at random from the training set. The inner products are computed via the kernel function κ, which has to satisfy the standard Mercer conditions, and μ_i is precomputed as $\kappa(\mathbf{x}, \mathbf{x}_i)$ averages over all $\mathbf{x}$'s in the training set. In this formulation, the supervised learning of the hash function boils down to minimizing a loss of the form

$$\mathbb{E}\left\{\left(\frac{1}{m}\xi(\mathbf{x})^{\mathrm{T}}\xi(\mathbf{x}') - \ell\right)^2\right\}, \quad (3)$$

where $\ell = +1$ or -1 on $(\mathbf{x}, \mathbf{x}')$ belonging to $\mathcal{P}$ or $\mathcal{N}$, respectively. The authors show that the learning of $\mathbf{P}$ can be performed either via greedy optimization similar to SSH, or by dropping the sign function and resorting to a spectral relaxation closely resembling DiffHash. In fact, depending on the choice of the optimization algorithm, KSH can be viewed as a kernelized version of either SSH or DiffHash. The greedy approximation or the spectral relaxation can be further refined by solving the highly non-convex problem minimizing (3), in which the sign function is replaced by a smooth sigmoid approximation.

2.2 Cross-modality similarity sensitive hashing

To the best of our knowledge, only one attempt has been made to date to generalize supervised hashing techniques to multiple modalities. Bronstein et al. [6] studied the particular case of cross-modal similarity-sensitive hashing (without incorporating intra-modality similarity), with linear embeddings of the form $\xi(\mathbf{x}) = \mathrm{sign}(\mathbf{P}\mathbf{x}+\mathbf{a})$ and $\eta(\mathbf{y}) = \mathrm{sign}(\mathbf{Q}\mathbf{y}+\mathbf{b})$, which can be considered an extension of SSH. The CM-SSH algorithm constructs the dimensions of ξ and η one-by-one using boosting. At each iteration, one-dimensional embeddings $\xi_i(\mathbf{x}) = \mathrm{sign}(\mathbf{p}_i^{\mathrm{T}}\mathbf{x} + a_i)$ and $\eta_i(\mathbf{y}) = \mathrm{sign}(\mathbf{q}_i^{\mathrm{T}}\mathbf{y} + b_i)$ are found using a two-stage scheme: first, the embeddings are linearized as $\xi_i(\mathbf{x}) \approx \mathbf{p}_i^{\mathrm{T}}\mathbf{x}$ and $\eta_i(\mathbf{y}) \approx \mathbf{q}_i\mathbf{y}$ and the resulting objective is minimized to find the projection

$$\min_{\mathbf{p}_i,\mathbf{q}_i} \mathbb{E}\{\mathbf{x}^{\mathrm{T}}\mathbf{p}_i\mathbf{q}_i^{\mathrm{T}}\mathbf{y}|\mathcal{P}_{XY}\} - \mathbb{E}\{\mathbf{x}^{\mathrm{T}}\mathbf{p}_i\mathbf{q}_i^{\mathrm{T}}\mathbf{y}|\mathcal{N}_{XY}\}, \quad (4)$$

(here $\mathbf{p}_i^{\mathrm{T}}$ and $\mathbf{q}_i^{\mathrm{T}}$ are unit vectors representing the ith row of the matrices $\mathbf{P}$ and $\mathbf{Q}$, respectively, and the expectations are weighted by per-sample weights adjusted by the boosting). With such an approximation, the optimal projection directions $\mathbf{p}$ and $\mathbf{q}$ have a closed-form expressions using the SVD of the positive and negative covariance matrices. At the second stage, the thresholds a_i and b_i are found by two-dimensional search.

This approach has several drawbacks. First, CM-SSH solves a particular setting of problem (1) with $\mathcal{P}_{XY}, \mathcal{N}_{XY}$ only, thus ignoring the intra-modality similarities. Second, the assumption of separability (treating each dimension separately) and the linearization of the objective replace the original problem with a relaxed version, whose optimization produces suboptimal solutions. Finally, this approximation is limited to a relatively narrow class of linear embeddings that often do not capture well the structure of the data.

3 Multimodal NN hashing

Our approach for multimodal hashing is related to supervised methods for dimensionality reduction and in particular extends the framework of [13], [35], [41], also known as the *siamese architecture*. These methods learn a mapping onto a usually low-dimensional feature space such that similar observations are mapped to nearby points in the new manifold and dissimilar observations are pulled apart. In our simplest setting, the linear embedding $\xi = \mathrm{sign}(\mathbf{P}\mathbf{x} + \mathbf{a})$ is realized as a neural network with a single layer (where $\mathbf{P}$ represent the linear weights and $\mathbf{a}$ is the bias) and a sign activation function (in practice, we use a smooth approximation $\mathrm{sign}(x) \approx \tanh(\beta x)$). The parameters of the embedding can be learned using the back-propagation algorithm [47] minimizing the loss

$$\begin{aligned}\mathcal{L}_X &= \frac{1}{2}\sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{P}_X} \|\xi(\mathbf{x}) - \xi(\mathbf{x}')\|_2^2 \\ &+ \frac{1}{2}\sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{N}_X} \max\{0, m_X - \|\xi(\mathbf{x}) - \xi(\mathbf{x}')\|_2\}^2\end{aligned} \tag{5}$$

w.r.t. the network parameters $(\mathbf{P}, \mathbf{a})$. Same way, embedding η is learned by minimizing the loss $\mathcal{L}_Y$ w.r.t. parameters $(\mathbf{Q}, \mathbf{b})$. Note that for binary vectors (when $\beta = \infty$), the squared Euclidean distance in (5) is equivalent, up to constants, to the Hamming distance. The second term in (5) is a *hinge-loss* providing robustness to outliers and produces a mapping for which negatives are pulled m_X apart. The system is fed with pairs of samples which share the same parametrization and for which a corresponding dissimilarity is known, 0 for positives and 1 for negatives (thus the name *siamese network*, i.e. two inputs and a common output vector). This approach has been also successfully applied by [41] to problems such as matching people in similar pose and which exhibits invariance to identity, clothing, background, lighting, shift and scale.

3.1 Coupled siamese architecture

In the multimodal setting, we have two embeddings ξ and η, each cast as a siamese network with parameters $(\mathbf{P}, \mathbf{a})$ and $(\mathbf{Q}, \mathbf{b})$, respectively. Such an architecture allows to learn similarity-sensitive hashing for each modality independently by minimizing the loss functions $\mathcal{L}_X, \mathcal{L}_Y$. In order to incorporate inter-modal similarity, we couple the two siamese networks by the cross-modal loss

$$\begin{aligned}\mathcal{L}_{XY} &= \frac{1}{2}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{P}_{XY}} \|\xi(\mathbf{x}) - \eta(\mathbf{y})\|_2^2 \\ &+ \frac{1}{2}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{N}_{XY}} \max\{0, m_{XY} - \|\xi(\mathbf{x}) - \eta(\mathbf{y})\|_2\}^2,\end{aligned} \tag{6}$$

thus jointly learning two sets of parameters for each modality. We refer to this model, which generalizes the siamese framework, as *coupled siamese networks*.

Our implementation differs from the architecture of [13] in the choice of the output activation function (we use $\tanh$ activation that encourages binary representations rather than a linear output layer). This way the maximum distance is bounded by $\sqrt{4m}$ and by simply enlarging the margin between dissimilar pairs we enforce the learning of codes which differ by the sign of their components. Once the model is learned, hashes are produced by thresholding the output.

This architecture can be extended to arbitrarily complex mappings by adding multiple layers of non-linearities. This has the advantage of scaling linearly with the number of activations which is a very desirable property in large scale problems.

3.2 Training

The training of our coupled siamese network is performed by minimizing

$$\min_{\mathbf{P},\mathbf{a},\mathbf{Q},\mathbf{b}} \mathcal{L}_{XY} + \alpha_X\mathcal{L}_X + \alpha_Y\mathcal{L}_Y, \tag{7}$$

where α_X, α_Y are weights determining the relative importance of each modality. The loss (7), can be considered as a generalization of the loss in (1), which is obtained by setting $\alpha_X = \alpha_Y = 1$, margins = 0, and $\beta = \infty$. We call the setting $\alpha_X, \alpha_Y > 0$ MM-NN. Furthermore, setting $\alpha_X = \alpha_Y = 0$, we obtain the particular setting of cross-modal loss (referred to in the following as CM-NN), whose relaxed version is minimized by the CM-SSH algorithm of [6]. It is also worth repeating that in many practical cases, it is very hard to obtain reliable cross-modal training samples $(\mathcal{P}_{XY}, \mathcal{N}_{XY})$ but much easier to obtain intra-modal samples $(\mathcal{P}_X, \mathcal{N}_X, \mathcal{P}_Y, \mathcal{N}_Y)$. In the full multimodal setting ($\alpha_X, \alpha_Y > 0$), the terms $\mathcal{L}_X, \mathcal{L}_Y$ can be considered as a *regularization*, preventing the algorithm from over fitting.

We apply the back-propagation algorithm [47], [20], [33] to get the gradient of our model w.r.t. the embedding parameters. The gradient of the intra-modal loss function w.r.t. to the parameters of ξ is given by $\nabla\mathcal{L}_X = (\xi(\mathbf{x})-\xi(\mathbf{x}'))(\nabla\xi(\mathbf{x})-\nabla\xi(\mathbf{x}'))$ for $(\mathbf{x},\mathbf{x}') \in \mathcal{P}_X$; $\nabla\mathcal{L}_X = (\xi(\mathbf{x}) - \xi(\mathbf{x}') - m_X)(\nabla\xi(\mathbf{x}) - \nabla\xi(\mathbf{x}'))$ for $(\mathbf{x},\mathbf{x}') \in \mathcal{N}_X$ and $m_X > \|\xi(\mathbf{x}) - \xi(\mathbf{x}')\|_2$; and zero otherwise (here the term $\nabla\xi = \partial\xi/\partial(\mathbf{P},\mathbf{a})$ is the usual back-propagation step of a neural network). The gradient of the inter-modal loss function w.r.t. to the parameters of ξ is given by $\nabla\mathcal{L}_{XY} = (\xi(\mathbf{x})-\eta(\mathbf{y}))\nabla\xi(\mathbf{x})$ for $(\mathbf{x},\mathbf{y}) \in \mathcal{P}_{XY}$; $\nabla\mathcal{L}_{XY} = (\xi(\mathbf{x})-\eta(\mathbf{y})-m_{XY})\nabla\xi(\mathbf{x})$ for $(\mathbf{x},\mathbf{y}) \in \mathcal{N}_{XY}$ and $m_{XY} > \|\xi(\mathbf{x}) - \eta(\mathbf{y})\|_2$; and zero otherwise. Equivalent derivation is done for the parameters of η.

The model can be easily learned jointly using any gradient-based technique such as conjugate gradient or stochastic gradient descent. The latter is the preferred choice for large datasets as it has minimal memory footprint and performs many more updates of the parameters, one per sample in the fully online setting, speeding up convergence of deep architectures.

3.3 Non-linear embeddings

Our model straightforwardly generalizes to non-linear embeddings using multi-layered network architectures. The proposed framework is in fact general and any class of neural networks can be applied to arbitrarily increase the complexity of the embedding. Deep and hierarchical models are able to model highly non-linear embeddings and scale well to large-scale data by means of fully online learning, where the parameters are updated after every input tuple presentation. This allows to sample a very large training set with constant memory requirements.

Learning deep models. To avoid bad local minima a long list of techniques have been proposed, see [4] for an overview. For our purpose we found that the hybrid batch on-line approach of [19] worked the best. We sample batches and train for only 5 iterations using L-BFGS, repeating until convergence. We found that, because all parameters are learned, setting $\beta = 1$ and adjusting the margin dependent on the code length delivered the best results.

4 Results

In this section, we evaluate our approach on several standard multimedia datasets: CIFAR10 [16], NUS [7], and Wiki [31] (see Table 1). All datasets were centered and unit-length normalized. In our experiments, we distinguish between uni- and multi-modal training, where in the former the hash functions are learned on each modality individually without using the other modality, and in the latter, inter-modal information is also used. Furthermore, we distinguish between uni- and cross-modal retrieval. In the former case, both the query and the database are from the same modality; in the latter case, the query and the database belong to different modality.

In the unimodal setting, we compare to the following state-of-the-art hashing methods: Diffhash [40], SSH [37], AGH [23], and KSH [22], using the code provided by the authors. In the cross-modal setting, we used Euclidean embedding by means of canonical correlation analysis (CCA) as a baseline, and compare to CM-SSH [6]. As a 'sanity check', we also tested hash functions trained in the multimodal setting on unimodal retrieval tasks. Ideally, the use of another modality information during training should improve (or at least not deteriorate) the performance of unimodal retrieval.

Our NN hash was tested in single-layer (L1) and two-layered (L2) configurations. We also distinguish between a version trained on inter-modal data only (CM-NN, corresponding to $\alpha_X = \alpha_Y = 0$) and full multimodal version (MM-NN, using $\alpha_X = \alpha_Y = 0.5$) making use of inter- and intra-modal training data. The architecture of CM-NN L1 is directly comparable to CM-SSH.

We adopted the following rule of thumb for the margins: 3 for 12bit, 5 for 24 and 48 bit, 7 for 64 and 16 for 256bit. For training the neural networks, we used L-BFGS with randomly sampled mini-batches [19], run until convergence.[1]

The hash functions learned by each of the methods were applied to the data in the datasets, and the exact Hamming distance was used to rank the matches. Retrieval performance was evaluated using *mean average precision* $mAP = \sum_{r=1}^{R} P(r)\cdot rel(r)$, where $rel(r)$ is the relevance of a given rank (one if relevant and zero otherwise), R is the number of retrieved results, and $P(r)$ is *precision at* r, defined as the percentage of relevant results in the first r top-ranked retrieved matches.

TABLE 1
Summary of the experiments and datasets.

	Modalities		Classes	Testing	
Dataset	n	n'		queries	database
Wiki	128	10	10	693	2173
CIFAR10	384	486	10	1000	59000
NUS	500	1000	81	2100	193739

CIFAR10 [16] is a set of 60K labeled images belonging to 10 different classes, sampled from the 80M tiny image benchmark [42]. The images are represented using 384-dimensional GIST and 486-dimensional HOG descriptors, used as two different modalities. Following [22], we used a training set of 200 images for each class; for testing, we used a disjoint query set of 100

1. For our methods, as it allows stochastic optimization, we do not run into memory problems when the number of data points grows. In fact we are not bounded at all by the size of the training set which is generated on the fly. This is a crucial difference between our method and other hashing approaches, since real-world datasets are typically orders of magnitude larger than what can be handled by standard batch methods.

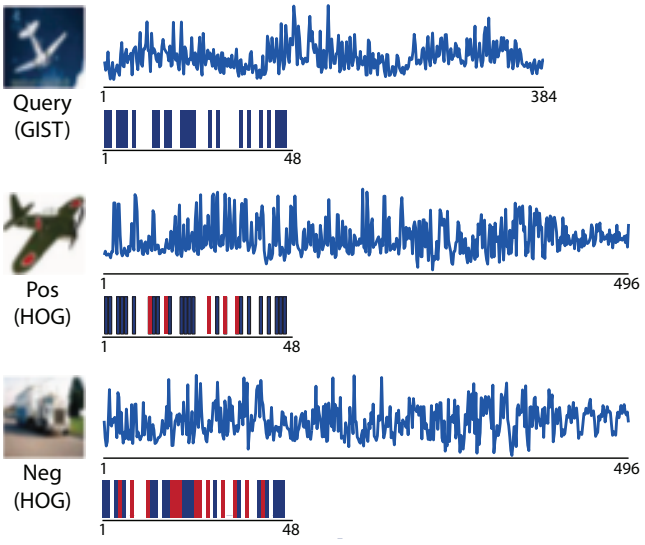

Fig. 2. Example of GIST-HOG matching on CIFAR dataset. Shown are the original descriptors and their 48-bit MM-NN L2 hash codes. Red shows the bits different w.r.t. query.

images per class and the remaining 59K images as database.

Table 2 shows the unimodal (GIST–GIST and HOG–HOG) retrieval performance; examples of a few top matches produced by different hashing algorithms are shown in Figure 1. We can see that our NN-based methods significantly outperform all the rest of the methods, including the previous state-of-the-art AGH and KSH. Further significant improvement is achieved by using a two-layer configuration with 48 $\tanh$ units (NN-L2).

TABLE 2
Unimodal training and retrieval experiment on the CIFAR10 dataset. NN hash was trained on single modality only. Performance is shown as mAP in %.

		GIST – GIST			HOG – HOG		
Method / m		**12**	**24**	**48**	**12**	**24**	**48**
DiffHash		14.72	13.35	12.85	13.05	11.92	11.47
SSH		15.42	16.75	17.06	15.49	16.15	16.71
AGH1		15.59	15.45	14.66	16.82	16.56	16.65
AGH2		15.46	15.29	15.15	16.09	16.74	16.43
KSH		25.79	29.01	30.84	25.70	28.95	30.17
NN	**L1**	31.48	35.41	36.79	31.48	37.24	38.03
	L2	**45.42**	**49.88**	**50.46**	**49.20**	**50.16**	**53.01**
Raw		19.16			19.19		

Table 3 (bottom) shows the performance of cross-modal retrieval. Figure 2 shows examples of query and database descriptors in this setting and their corresponding binary codes. NN-based method significantly outperform CM-SSH. Furthermore, we observe that MM-NN shows superior performance compared to CM-NN, which we explain by the importance of using intra-modal training data in addition to inter-modal one.

Applying the hash functions trained in the multimodal setting to unimodal retrieval (GIST–GIST and HOG–HOG in Table 3), MM hash achieves slightly better performance compared to the corresponding results obtained with unimodal training shown in Table 2. We interpret this result as the usefulness of multimodal information in training as a kind of regularization. Figure 3 (left) shows the precision recall curves for the cross-modal retrieval cases.

TABLE 3
Unimodal and cross-modal retrieval experiment on the CIFAR10 dataset. All methods were trained using multimodal data. CCA produces Euclidean embeddings. Performance is shown as mAP in %.

		GIST – GIST			HOG – HOG		
Method / m		**12**	**24**	**48**	**12**	**24**	**48**
CCA		11.21	11.73	12.36	10.26	10.44	10.83
CM-SSH		16.93	16.78	16.17	17.65	17.60	17.50
CM-NN	**L1**	24.71	28.82	31.34	25.10	29.23	32.55
	L2	41.60	45.23	44.22	47.15	45.11	44.25
MM-NN	**L1**	28.49	34.31	34.33	30.64	36.11	36.01
	L2	**46.62**	**48.62**	**52.00**	**49.46**	**52.34**	**53.40**

		GIST – HOG			HOG – GIST		
Method / m		**12**	**24**	**48**	**12**	**24**	**48**
CCA		10.04	10.06	10.09	10.21	10.40	10.84
CM-SSH		17.21	15.83	14.44	17.28	17.04	16.62
CM-NN	**L1**	24.56	28.38	32.72	25.11	29.30	33.25
	L1	47.89	47.52	47.09	43.05	45.79	45.32
MM-NN	**L1**	29.53	35.00	35.39	29.11	35.26	35.07
	L2	**48.97**	**51.15**	**54.01**	**46.80**	**49.97**	**51.06**

NUS [7] is a multi-class dataset containing annotated images from Flickr. The images are manually categorized into 81 classes (one image can belong to more than a single class) and represented as 500-dimensional bags of SIFT features (BoF, used as the first modality) and 1000-dimensional bags of text tags (Tags, used as the second modality). To produce results consistent with previous state-of-the-art, we follow the dataset generation protocol of [23], which considers only the top-21 frequent classes and used 5K samples for KSH. We used full mAP and mAP@10 as the retrieval quality criteria.

Table 4 shows the unimodal performance of several hashing methods of different lengths, where our method outperforms the best competitor. Due to the ambiguous nature of this multi-class dataset we did not experience improvement using an additional layer. We also notice that KSH performs worse than AGH, a completely unsupervised technique. We attribute this to the inability of binary labels to discriminate the various degrees of similarity given by class intersection; we believe that trivial and less generalizable solutions are favored with such setup. We intend to further investigate the multi-class problem in future work.

Table 5 (bottom) reports the performance of the several methods using hashes up to 256bit. CCA is used as Euclidean baseline also in this case. NN-based methods outperforms CM-SSH by large margin while still keeping almost the same code generation complexity. Figure 3 (right) shows the precision-recall

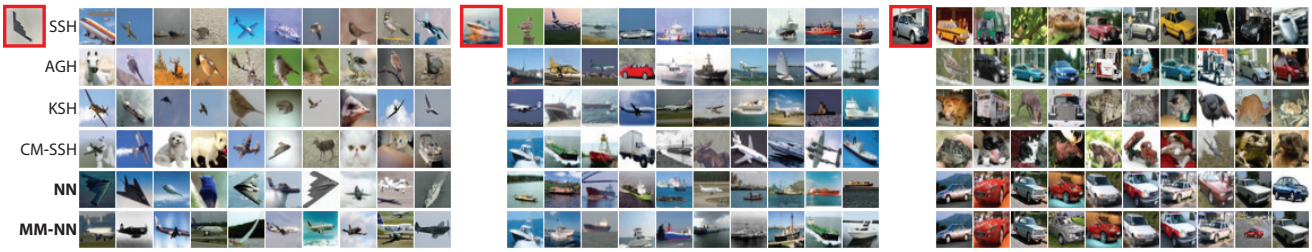

Fig. 1. Unimodal retrieval on CIFAR dataset. Shown are top 10 matches to three different queries (marked in red) using different hashing method with codes of length 48. All NN methods are used in L2 configuration. CM-SSH, CM-NN and MM-NN were trained on multiple modalities, and used in this experiment for single modality retrieval.

TABLE 4
Unimodal training and retrieval experiment on the NUS dataset. NN hash was trained on single modality only. Performance is shown as mAP@10 / mAP in %, (– indicates no convergence was reached).

	BoF – BoF			Tags – Tags		
Method / m	**16**	**64**	**256**	**16**	**64**	**256**
DiffHash	54.70 / 37.40	52.97 / 36.88	53.71 / 36.75	72.85 / 42.45	80.97 / 41.02	79.82 / 39.58
SSH	45.31 / 43.76	59.00 / 43.40	59.58 / 42.21	44.51 / 41.99	61.54 / 44.23	70.26 / 45.09
AGH1	54.53 / 38.31	59.38 / 38.09	– / –	74.60 / 45.37	79.07 / 41.52	– / –
AGH2	53.86 / 38.24	59.56 / 39.08	– / –	67.60 / 47.55	77.99 / 43.29	– / –
KSH	56.25 / 49.84	64.25 / 51.30	66.46 / 51.78	72.25 / 60.11	70.29 / 57.69	84.05 / 62.68
NN	**60.93 / 53.40**	**66.52 / 57.10**	**72.57 / 59.36**	**79.25 / 65.96**	**83.87 / 68.04**	**87.08 / 67.40**
Raw		61.53			83.02	

TABLE 5
Unimodal and cross-modal retrieval experiment on the NUS dataset. All methods were trained using multimodal data. CCA produces Euclidean embeddings. Performance is shown as mAP@10 / mAP in %.

	BoF – BoF			Tags – Tags		
Method / m	**16**	**64**	**256**	**16**	**64**	**256**
CCA	58.72 / 42.26	61.58 / 43.26	63.51 / 43.94	77.66 / 42.50	81.79 / 38.71	81.64 / 37.87
CM-SSH	41.23 / 44.69	50.30 / 43.45	53.23 / 41.56	71.33 / 48.74	80.11 / 49.80	83.00 / 47.62
CM-NN	52.16 / 50.34	64.33 / 51.44	67.55 / 50.14	75.18 / 61.70	79.62 / 61.21	83.44 / 64.68
MM-NN	**60.02 / 53.09**	**64.66 / 51.87**	**70.45 / 57.84**	**78.99 / 65.52**	**83.31 / 64.64**	**86.79 / 69.4**

	BoF – Tags			Tags – BoF		
Method / m	**16**	**64**	**256**	**16**	**64**	**256**
CCA	35.75 / 34.35	39.17 / 35.84	32.00 / 36.79	35.72 / 35.16	48.33 / 40.46	61.52 / 43.11
CM-SSH	61.63 / 47.78	62.08 / 44.61	61.18 / 40.61	55.48 / 45.98	59.10 / 46.87	55.83 / 45.31
CM-NN	**70.07** / 57.16	**72.86 / 58.44**	**74.83** / 60.28	70.10 / **57.17**	71.56 / **57.70**	76.74 / 59.59
MM-NN	64.57 / **57.44**	68.07 / 56.33	73.12 / **61.63**	**73.40** / 56.91	**73.28** / 55.83	**78.77 / 61.09**

curve for the cross modal retrieval, MM-NN delivers the best performance.

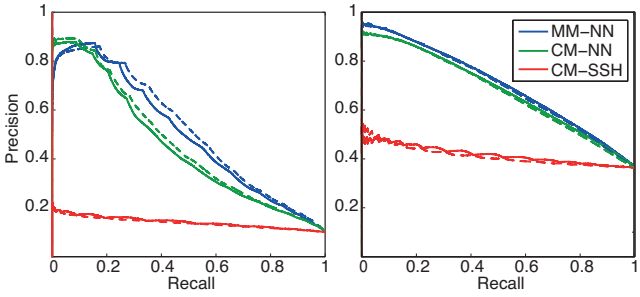

Fig. 3. Precision-Recall curves for the cross-modal retrieval experiments on CIFAR10 (solid: HOG–GIST, dashed: GIST–HOG) and NUS (solid: Tag–Bof, dashed: Bof–Tag).

Figure 4 shows cross-modal retrieval results using as queries artificially created Tag vectors containing specific words. These Tags are hashed using η and matched to BoFs hashed using ξ. The retrieved results are meaningful and most of them belong to the same class. The results produced by NN hash (bottom) are visually more meaningful compared to CM-SSH (top). Figure 5 shows image annotation results. We retrieve the top five Tags matches from a BoF query and assign the corresponding annotations to the image.

Wiki. In the third experiment, we reproduced the results of [31] using the dataset of 2866 annotated images from Wikipedia. The images are categorized in 10 classes and represented as 128-dimensional bags of SIFT features (Image modality) and 10-dimensional LDA topic model (Text modality). Table 6 shows the mAP for the Image–Text and Text–Image cross-modal retrieval experiment. For reference, we also reproduce the results reported in [31] using correlation matching (CM), semantic matching (SM), and semantic correlation matching (SCM). MM-NN largely outperforms SCM in all evaluation criterion with codes that are at least 10× smaller and that can be searched very efficiently. Figure 7 shows a few matching examples.

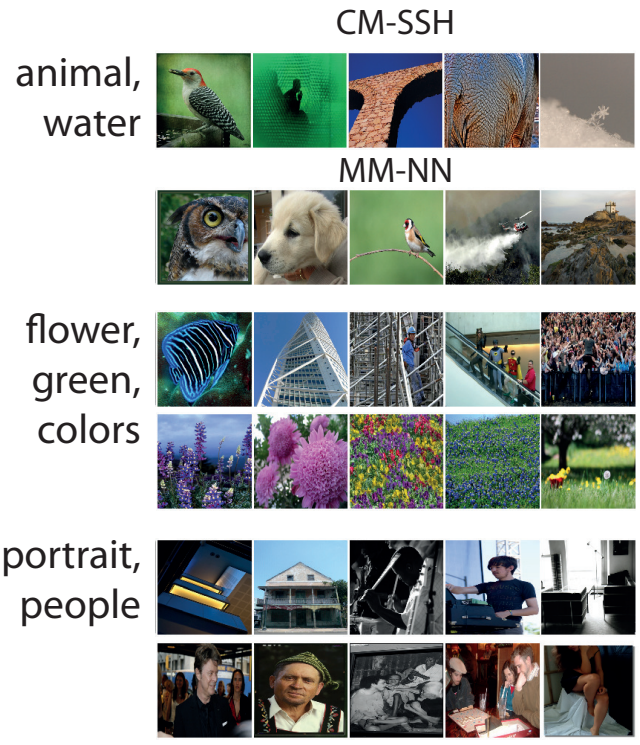

Fig. 4. Example of text-based image retrieval on NUS dataset using multimodal hashing. Shown are top five image matches produced by CM-SSH (odd rows) and MM-NN (even rows) in response to three different queries.

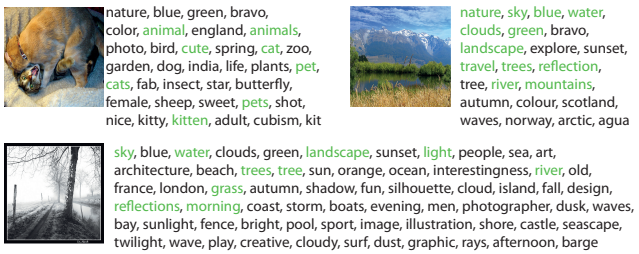

Fig. 5. Example of image annotation on the NUS dataset using multimodal hashing. Shown are Tags returned for the image query on the left. Groundtruth tags are shown in green.

5 Conclusions

We introduced a novel learning framework for multimodal similarity-preserving hashing based on the coupled siamese neural network architecture. Our approach is free from assuming linear projections unlike existing crossmodal similarity learning methods; in fact, by increasing the number of layers in the network, mappings of arbitrary complexity can be trained (our experiments showed that using multilayer architecture results in a significant improvement of performance). We also solve the exact optimization problem during training making no approximations like the boosting-based CM-SSH. Our method does not involve semidefinite programming, and is scalable to a very large number of dimensions and training samples. Experimental results on standard multimedia retrieval datasets showed performance superior to state-of-the-art hashing approaches.

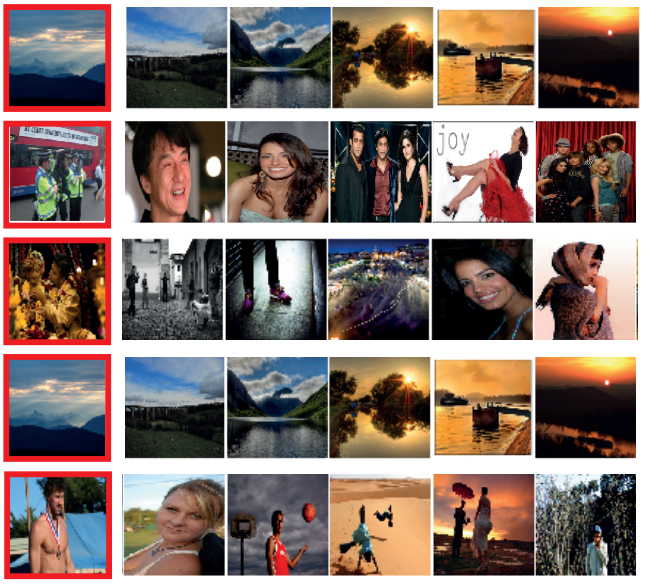

Fig. 6. Cross-modal (Bof–Tags) retrieval on the NUS dataset. Shown are top five matches different image queries (marked in red), ranked according to Tags similarity using 64-bit MM-NN hash.

TABLE 6
Cross-modal retrieval experiment on the Wiki dataset using 32-bit hashes (L2 with 32 $\tanh$ units) and Euclidean embeddings from [31] (marked with *).

		Image–Text	Text–Image	Avg
CM-SSH		22.2	18.4	20.3
CM*		24.9	19.6	22.3
SM*		22.5	22.3	22.4
SCM*		27.7	22.6	25.2
MM-NN	**L1**	37.8	24.7	31.2
	L2	**57.5**	**27.4**	**42.4**
CM-NN	**L1**	32.6	23.2	25.5
	L2	48.5	25.8	37.1

Acknowledgment

J. Masci is supported by the ArcelorMittal project "Development of a new generic framework for 2D-mapping sensors data processing". A. M. Bronstein is supported by the ERC Starting Grant no. 335491. M. M. Bronstein is supported by the ERC Starting Grant no. 307047.

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. FOCS*, 2006.
[2] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proc. ICML*, 2004.
[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
[4] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

Fig. 7. Cross-modal (Image–Text) retrieval on the Wiki dataset. Shown are top five matches different image queries (marked in red), ranked according to text similarity using 32-bit MM-NN hash.

[5] I. Borg and P. J. F. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

[6] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *Proc. CVPR*, 2010.

[7] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. NUS-WIDE: A real-world web image database from national university of Singapore. In *Proc. CIVR*, 2009.

[8] R. R. Coifman and S. Lafon. Diffusion maps. *App. Comp. Harmonic Analysis*, 21(1):5–30, 2006.

[9] Davis et al. Information-theoretic metric learning. In *Proc. ICML*, 2007.

[10] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proc. VLDB*, 1999.

[11] Y. Gong, S. Kumar, V. Verma, and S. Lazebnik. Angular quantization-based binary codes for fast similarity search. In *Proc. NIPS*, 2012.

[12] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Proc. CVPR*, 2011.

[13] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, 2006.

[14] R. A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*, volume 4. Prentice Hall, 2002.

[15] S. Korman and S. Avidan. Coherency sensitive hashing. In *Proc. ICCV*, 2011.

[16] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[17] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Proc. NIPS*, 2009.

[18] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Proc. CVPR*, pages 2130–2137, 2009.

[19] Q. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Ng. On optimization methods for deep learning. In *Proc. ICML*, 2011.

[20] Y. LeCun. Une procédure d'apprentissage pour réseau à seuil asymétrique. *Proceedings of Cognitiva 85, Paris*, pages 599–604, 1985.

[21] D. Lee, M. Hofmann, F. Steinke, Y. Altun, N. D. Cahill, and B. Scholkopf. Learning similarity measure for multi-modal 3d image registration. In *Proc. CVPR*, 2009.

[22] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *Proc. CVPR*, 2012.

[23] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *Proc. ICML*, 2011.

[24] J. Masci, D. Migliore, M. M. Bronstein, and J. Schmidhuber. Descriptor learning for omnidirectional image matching. Technical Report arXiv:1112.6291, 2011.

[25] B. McFee and G. R. G. Lanckriet. Partial order embedding with multiple kernels. In *Proc. ICML*, 2009.

[26] B. McFee and G. R. G. Lanckriet. Learning multi-modal similarity. *JMLR*, 12:491–523, 2011.

[27] S. Mika, G. Ratsch, J. Weston, B. Schoelkopf, and K. R. Mueller. Fisher discriminant analysis with kernels. In *Proc. Neural Networks for Signal Processing*, 1999.

[28] M. Norouzi and D. Fleet. Minimal loss hashing for compact binary codes. In *Proc. ICML*, 2011.

[29] M. Norouzi, D. Fleet, and R. Salakhutdinov. Hamming distance metric learning. In *Proc. NIPS*, 2012.

[30] G.-J. Qi, C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. Huang. Towards cross-category knowledge propagation for learning visual concepts. In *Proc. CVPR*, 2011.

[31] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G.R.G. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proc. ICM*, 2010.

[32] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323, 2000.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[34] R. Salakhutdinov and G. Hinton. Semantic hashing. *Int. J. of Approximate Reasoning*, 50(7):969–978, 2009.

[35] J. Schmidhuber and D. Prelinger. Discovering predictable classifications. *Neural Computation*, 5(4):625–635, 1993.

[36] B. Schoelkopf, A. Smola, and K. R. Mueller. Kernel principal component analysis. *Artificial Neural Networks*, pages 583–588, 1997.

[37] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. CVPR*, 2003.

[38] A. Sharma, A. Kumar, H. Daume, and D. W Jacobs. Generalized multiview analysis: A discriminative latent space. In *Proc. CVPR*, 2012.

[39] C. Shen, J. Kim, L. Wang, and A. Hengel. Positive semidefinite metric learning with boosting. In *Proc. NIPS*, 2009.

[40] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua. LDA-Hash: Improved matching with smaller descriptors. *PAMI*, 34(1):66–78, 2012.

[41] G. W. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invariance through imitation. In *Proc. CVPR*, 2011.

[42] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI*, 30(11):1958–1970, 2008.

[43] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Proc. CVPR*, 2008.

[44] J. Wang, S. Kumar, and S.-F. Chang. Sequential projection learning for hashing with compact codes. In *Proc. ICML*, 2010.

[45] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.

[46] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proc. NIPS*, 2008.

[47] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

[48] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, 2010.

[49] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Proc. NIPS*, 2002.