Leveraging Quadratic Spherical Mutual Information Hashing for Fast Image Retrieval

Nikolaos Passalis and Anastasios Tefas
Computational Intelligence and Deep Learning Group
Department of Informatics, Faculty of Sciences
Aristotle University of Thessaloniki, Thessaloniki 541 24, Greece
Email: {passalis, tefas}@csd.auth.gr

***Abstract*—Several deep supervised hashing techniques have been proposed to allow for querying large image databases. However, it is often overlooked that the process of information retrieval can be modeled using information-theoretic metrics, leading to optimizing various proxies for the problem at hand instead. Contrary to this, we propose a deep supervised hashing algorithm that optimizes the learned codes using an information-theoretic measure, the Quadratic Mutual Information (QMI). The proposed method is adapted to the needs of large-scale hashing and information retrieval leading to a novel information-theoretic measure, the Quadratic Spherical Mutual Information (QSMI), that is inspired by QMI, but leads to significant better retrieval precision. Indeed, the effectiveness of the proposed method is demonstrated under several different scenarios, using different datasets and network architectures, outperforming existing deep supervised image hashing techniques.**

I. Introduction

The vast amount of data available nowadays, combined with the need to provide quick answers to users' queries, led to the development of several *hashing* techniques. Hashing provides a way to represent images using compact codes, which allows for performing fast queries in large image databases. Early hashing methods, e.g., Locality Sensitive Hashing (LSH) [1], focused on extracting generic codes that could, in principle, describe every possible image and information need. However, it was later established that *supervised hashing*, which learns hash codes that are tailored to the task at hand, can significantly improve the retrieval precision. In this way, it is possible to learn even smaller hashing codes, since the extracted code must only encode the information needs for which the users are actually interested in. However, note that the extracted hash codes must also encode part of the semantic relationships between the encoded objects, to allow for providing a meaningful ranking of the retrieved results.

Many supervised hashing methods have been proposed in recent years [2], [3], [4], [5], [6], [7]. However, most of these methods do not employ an information-theoretic modeling of the process of information retrieval. Instead, they directly optimize various proxies for the problem at hand. For example, many methods employ the pairwise distances between the images [8], [9], [10], or are based on sampling *triplets* that must satisfy specific relationships according to the given ground truth [11], [12]. On the other hand, information-theoretic measures, such as entropy and mutual information [13], have been proven to provide robust solutions to many machine learning problems, e.g., classification [13]. However, only a few steps towards using these measures for supervised hashing tasks have been made so far [14].

In this paper, we discuss the connection between an information-theoretic measure, the Mutual Information (MI), and the process of information retrieval. More specifically, we argue that mutual information can naturally model the process of information retrieval, providing a solid framework to develop retrieval-oriented supervised hashing techniques. Even though MI provides a sound formulation for the problem of information retrieval, applying it in real scenarios is usually intractable, since there is no fast way to calculate the actual probability densities, which are involved in the calculation of MI. The great amount of data as well as their high dimensionality further complicate the practical application of such measures.

The main contribution of this paper is the proposal of a deep supervised hashing algorithm that optimizes the learned codes using a variant of an information-theoretic measure, the Quadratic Mutual Information (QMI) [15]. The architecture of the proposed method is shown in Fig. 1. To derive a practical algorithm that can scale to large datasets:

1) We adapt QMI to the needs of supervised hashing by employing a similarity measure that leads to higher precision in retrieval applications. This gives rise to the proposed *Quadratic Spherical Mutual Information* (QSMI). It is also experimentally demonstrated that the proposed QSMI is more robust compared to the classical Gaussian-based Kernel Density Estimation used in QMI [15], while it does not require tuning any hyper-parameters.
2) We propose using a more smooth optimization objective employing a square clamping approach. This allows for significantly improving the stability of the optimization, while reducing the risk of converging to bad local minima.
3) We adapt the proposed approach to work in batch-based setting by employing a method that dynamically estimates the prior probabilities, as they are observed within each batch. In that way, the proposed method can scale to larger datasets.

The proposed method is extensively evaluated using three

image datasets, including the two standard datasets used for evaluating supervised hashing methods, the CIFAR10 [16] and NUS-WIDE [17], and it is demonstrated that it outperforms the existing state-of-the-art techniques.

It is worth noting that MI has been also investigated to aid various aspects of the retrieval process. In [18], [19] MI is employed to provide relevance feedback, while in [20], [14] MI is used to provide updates for online hashing. More specifically, in [14], the Shannon's definition for MI is used, leading to employing a Monte Carlo sampling scheme to approximate the MI, together with a differentiable histogram binning technique. Our approach is different, since instead of approximating the MI through random sampling, we analytically derive computationally tractable solutions for calculating MI through a QMI formulation. To the best of our knowledge, this is the first work that employs a quadratic spherical mutual information loss fully adapted to the needs of deep supervised hashing. The employed formulation is fully differentiable allowing for end-to-end optimization of deep neural networks for any retrieval-related task.

The rest of the paper is structured as follows. The proposed method is presented in detail in Section II, while the experimental evaluation is provided in Section III. Finally, Section IV concludes the paper. Additional experiments and details regarding the used experimental setup are also provided in the supplementary material.

II. Proposed Method

a) Information Retrieval and Mutual Information: Let $\mathscr{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ be a collection of N images, where $\mathbf{y}_i \in \mathbb{R}^n$ is the representation of the i-th image extracted using an appropriate feature extractor, e.g., a deep neural network. Also, note that even though the term "information need" is used to described a textual query in traditional Information Retrieval [21], in this work we use this term to describe any possible need that arises from a user's query, which is not necessarily limited to textual queries. Therefore, in this paper we focus on the case of content-based image retrieval (CBIR) [22], [23], where each information need is expected by an image query. However, this is without loss of generality, since the proposed method can be used for other types of retrieval as well, e.g., text-based retrieval [24]. Note that the information needs that an image actually fulfills depend on both its content and the needs of the users, since, depending on the application, the interests of the users are usually focused on a specific area. For example, an image of a man entering a bank represents different information needs for a forensics database used by the police to identify suspects and for a generic web search engine. The problem of information retrieval can be then defined as follows: *Given an information need q retrieve the images of the collection $\mathscr{Y}$ that fulfill this information need and rank them according to their relevance to the given information need*. Since this work focuses on content-based information retrieval, the information need q is expressed through a *query image*, which is usually not part of the collection $\mathscr{Y}$.

To be able to measure how well an information retrieval system works, a ground truth set that contains a set of information needs and the corresponding images that fulfill these information needs is usually employed. Let M be the number of information needs $\mathscr{Q} = \{q_1, q_2, ..., q_M\}$. Then, for each information need q_i, a set of images $\mathscr{Q}_i = \{\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, ..., \mathbf{y}_{N_i}^{(i)}\}$, where $\mathbf{y}_j^{(i)} \in \mathbb{R}^n$ is the representation of the j-th image that fulfills the i-th information need, is given. Please note that the notation $\mathbf{y}_j$ is used to refer to the representation of the j-th image (out of N images) of the dataset, while the notation $\mathbf{y}_j^{(i)}$ is used to refer to the representation of the j-th image (out of N_i images) that fulfills the information need q_i. We use this overloaded notation to simplify the presentation of the proposed approach. Also, note that since all images contained in $\mathscr{Q}_i$ fulfill the same information need, they can be all used as queries to express the information need q_i. However, there are also other images, which are usually not known beforehand, which also express the same information need and they can be also used to query the database. The distribution of the images that fulfill the i-th information need can be modeled using the *conditional probability density* function $p(\mathbf{y}|q_i)$.

Let $\mathcal{Y}$ be a random vector that represents the images and $\mathcal{Q}$ be a random variable that represents the information needs. The Shannon's entropy of information needs, which expresses the uncertainty regarding the information need that a randomly sampled image fulfills, is defined as [13]:

$$H(\mathcal{Q}) = -\sum_q P(q) log(P(q)), \tag{1}$$

where $P(q)$ is the prior probability of the information need q, i.e., the probability that a random image of the collection fulfills the information need q. Note that above definition implicitly assumes that the information needs are mutually exclusive, i.e., $\sum_q P(q) = 1$, or equivalently, that each image satisfies only one information need. This is without loss of generality, since it is straightforward to extend this definition to the general case, where each image can satisfy multiple information needs, e.g., by measuring the entropy of each information need separately:

$$H(\mathcal{Q}) = -\sum_q \Big(P(q) log(P(q)) + (1 - P(q)) log(1 - P(q)) \Big). \tag{2}$$

To simplify the presentation of the proposed method, we assume that the information needs are mutually exclusive. Nonetheless, the proposed approach can be still used with minimal modifications, as we also experimentally demonstrate in Section III, even when this assumption does not hold.

When the query vector is known, then the uncertainty of the information need that it fulfills can be expressed by the conditional entropy:

$$H(\mathcal{Q}|\mathcal{Y}) = -\int_{\mathbf{y}} p(\mathbf{y}) \left(\sum_q p(q|\mathbf{y}) log(p(q|\mathbf{y})) \right) d\mathbf{y}. \tag{3}$$

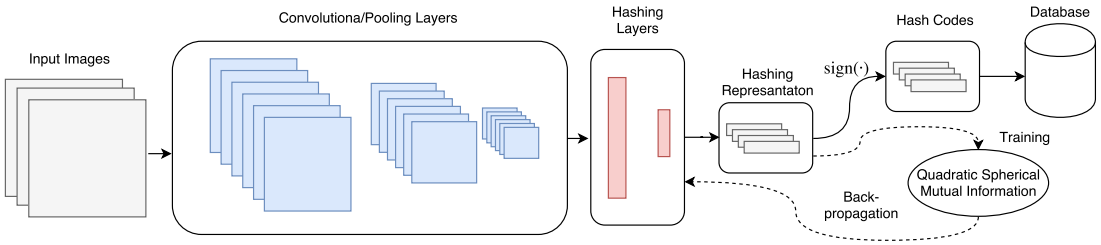

Fig. 1. Pipeline of the proposed method: A deep convolutional neural network (CNN) is used to extract a representation that can be used to directly obtain a binary hash code. The network is optimized using the proposed Quadratic Spherical Mutual Information loss that is adapted towards the needs of image hashing.

Mutual information is defined as the amount by which the uncertainty for the information needs is reduced after observing the query:

$$\begin{aligned} I(\mathcal{Q},\mathcal{Y}) &= H(\mathcal{Q}) - H(\mathcal{Q}|\mathcal{Y}) \\ &= \sum_q \int_{\mathbf{y}} p(q,\mathbf{y}) log(\frac{p(q,\mathbf{y})}{P(q)p(\mathbf{y})}) d\mathbf{y} \end{aligned} \tag{4}$$

It is easy to see that MI can be interpreted as the Kullback-Leibler divergence between $p(q,\mathbf{y})$ and $P(q)p(\mathbf{y})$. It is desired to maximize the MI between the representation of the images $\mathcal{Y}$ and the information needs $\mathcal{Q}$, since this ensures that the uncertainty regarding the information need, which a query image expresses, is minimized. Also, note that MI models the intrinsic uncertainty regarding the query vectors, since it employs the conditional probability density between the information needs and the images, instead of just a limited collection of images. In that way, it accounts for all the possible queries that can be used to express each possible information need. On the other hand, it is usually intractable to directly calculate the required probability density $p(\mathbf{y}|q_i)$ and the corresponding integral in Eq. (4), limiting the practical applications of MI. However, as it is demonstrated later, it is possible to estimate the aforementioned probability density and derive a practical algorithm that maximizes the MI between a representation and a set of information needs.

b) Quadratic Mutual Information: When the aim is not to calculate the exact value of MI, but to optimize a distribution that maximizes the MI, then a quadric divergence metric, instead of the Kullback-Leibler divergence, can be used. In that way, the *Quadratic Mutual Information* (QMI) is defined as [15]:

$$I_T(\mathcal{Q},\mathcal{Y}) = \sum_q \int_{\mathbf{y}} \left(p(q,\mathbf{y}) - P(q)p(\mathbf{y})\right)^2 d\mathbf{y}. \tag{5}$$

By expanding Eq. (5), QMI can be expressed as the sum of three *information potentials* as:

$$I_T(\mathcal{Q},\mathcal{Y}) = V_{IN}(\mathcal{Q},\mathcal{Y}) + V_{ALL}(\mathcal{Q},\mathcal{Y}) - 2V_{BTW}(\mathcal{Q},\mathcal{Y}), \tag{6}$$

where

$$V_{IN}(\mathcal{Q},\mathcal{Y}) = \sum_q \int_{\mathbf{y}} p(q,\mathbf{y})^2 d\mathbf{y}, \tag{7}$$

$$V_{ALL}(\mathcal{Q},\mathcal{Y}) = \sum_q \int_{\mathbf{y}} P(q)^2 p(\mathbf{y})^2 d\mathbf{y}, \tag{8}$$

and

$$V_{BTW}(\mathcal{Q},\mathcal{Y}) = \sum_q \int_{\mathbf{y}} p(q,\mathbf{y}) P(q) p(\mathbf{y}) d\mathbf{y}. \tag{9}$$

To calculate these quantities, the probability $P(q)$ and the densities $p(\mathbf{y})$ and $p(q,\mathbf{y})$ must be estimated. The prior probabilities depend only on the distribution of the information needs in the collection of images. Therefore, for the i-th information need: $P(q_i) = \frac{N_i}{N}$, where N_i is the number of images that fulfill the i-th information need. The conditional density of the images that fulfill the i-th information need can be estimated using the Parzen window estimation method [25]:

$$p(\mathbf{y}|q_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} K(\mathbf{y} - \mathbf{y}_j^{(i)}, \sigma^2), \tag{10}$$

where $K(\mathbf{y},\sigma^2)$ is used with slight abuse of notation to refer to $K(\mathbf{y},\sigma^2\mathbf{I})$, i.e., the Gaussian kernel with diagonal covariance matrix $\mathbf{\Sigma} = \sigma^2\mathbf{I}$, which is defined as:

$$K(\mathbf{y},\mathbf{\Sigma}) = \frac{1}{(2\pi)^{n/2}\det(\mathbf{\Sigma})^{1/2}} \exp(-\frac{1}{2}\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y}), \tag{11}$$

where $\mathbf{I}$ denotes the identity matrix. Then, the joint probability density can be estimated as:

$$p(q_i,\mathbf{y}) = p(\mathbf{y}|q_i)P(q_i) = \frac{1}{N} \sum_{j=1}^{N_i} K(\mathbf{y} - \mathbf{y}_j^{(i)}, \sigma^2), \tag{12}$$

while the density of all the images as:

$$p(\mathbf{y}) = \frac{1}{N} \sum_{j=1}^{N} K(\mathbf{y} - \mathbf{y}_j, \sigma^2). \tag{13}$$

By substituting these estimations into the definitions of the information potentials, the following quantities are obtained:

$$V_{IN}(\mathcal{Q},\mathcal{Y}) = \frac{1}{N^2} \sum_{k=1}^{M} \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} K(\mathbf{y}_i^{(k)} - \mathbf{y}_j^{(k)}, 2\sigma^2), \tag{14}$$

$$V_{ALL}(\mathcal{Q},\mathcal{Y}) = \frac{1}{N^2} \left(\sum_{k=1}^{M} \left(\frac{N_k}{N} \right)^2 \right) \sum_{i=1}^{N} \sum_{j=1}^{N} K(\mathbf{y}_i - \mathbf{y}_j, 2\sigma^2), \tag{15}$$

and

$$V_{BTW}(\mathcal{Q},\mathcal{Y}) = \frac{1}{N^2}\sum_{k=1}^{M}\left(\frac{N_k}{N}\sum_{i=1}^{N_k}\sum_{j=1}^{N}K(\mathbf{y}_i^{(k)}-\mathbf{y}_j,2\sigma^2)\right) \tag{16}$$

where the following property regarding the convolution between two Gaussian kernels was used:

$$\int_{\mathbf{y}} K(\mathbf{y}-\mathbf{y}_i,\sigma^2)K(\mathbf{y}-\mathbf{y}_j,\sigma^2)d\mathbf{y} = K(\mathbf{y}_i-\mathbf{y}_j,2\sigma^2) \tag{17}$$

The information potential V_{IN} expresses the interactions between the images that fulfill the same information need, the information potential V_{ALL} the interactions between all the images of the collection, while the potential V_{BTW} models the interactions of the images that fulfill a specific information need against all the other images. Therefore, the QMI formulation allows for the fast calculation of MI, since the MI is expressed as a weighted sum over the pairwise interactions of the images of the collection. Using Parzen window estimation with a Gaussian kernel for estimating the probability density leads to the implicit assumption that the similarity between two images is expressed through their Euclidean distance. Thus, the images that fulfill an information need expressed by a query vector $\mathbf{q}$ can be retrieved simply using nearest-neighbor search.

c) Deep Hashing using Quadratic Spherical Mutual Information Optimization: Despite its advantages. QMI still suffers from several limitations: For example, QMI involves the calculation of the pairwise similarity matrix between all the images of a collection. This quickly becomes intractable, as the size of the collection increases. Also, selecting the appropriate width for the Gaussian kernels is not always straightforward, as a non-optimal choice can distort the feature space and slow down the optimization. Furthermore, it was experimentally observed that directly optimizing the QMI is prune to bad local minima, due to the linear behavior of the loss function that fails to distinguish between the pairs of images that cause high error and those which have a smaller overall effect on the learned representation.

To overcome the limitations, we propose the *Quadratic Spherical Mutual Information* (QSMI). Instead of relying on the Euclidean distance between two samples, as in the regular QMI, the angle between two samples is used. In this case, the Gaussian distribution can be replaced with an appropriate circular distribution, e.g., the von Misses distribution [26]. However, such distributions significantly complicate the process of deriving efficient solutions for calculating QMI. Instead of this, the proposed QSMI directly replaces the Gaussian kernel, used for calculating the similarity between two images in the information potentials of Eq. (14), (15), and (16), with the cosine similarity:

$$S_{cos}(\mathbf{y}_1,\mathbf{y_2}) = \frac{1}{2}\left(\frac{\mathbf{y}_1^T\mathbf{y}_2}{\|\mathbf{y}_1\ \|_2\|\mathbf{y}_2\|_2}+1\right), \tag{18}$$

where $\|\cdot\|_2$ is the l^2 norm of a vector. In that way, we maintain the computationally efficient QMI formulation and avoid the need for manually tuning the width parameter of the Gaussian kernel. It is worth noting that the proposed QSMI method is not mathematically equivalent to replacing the Gaussian-based density estimation with cosine-based kernels, e.g., in the density estimation provided in (12). Instead, it is inspired by the QMI formulation, employing the convenient properties of Gaussian kernels to extract an efficient closed-form estimation, which is then substituted by a different kernel. Therefore, QSMI is defined as:

$$I_T^{cos}(\mathcal{Q},\mathcal{Y}) = V_{IN}^{cos}(\mathcal{Q},\mathcal{Y}) + V_{ALL}^{cos}(\mathcal{Q},\mathcal{Y}) - 2V_{BTW}^{cos}(\mathcal{Q},\mathcal{Y}), \tag{19}$$

where

$$V_{IN}^{cos}(\mathcal{Q},\mathcal{Y}) = \frac{1}{N^2}\sum_{k=1}^{M}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}S_{cos}(\mathbf{y}_i^{(k)},\mathbf{y}_j^{(k)}), \tag{20}$$

$$V_{ALL}^{cos}(\mathcal{Q},\mathcal{Y}) = \frac{1}{N^2}\left(\sum_{k=1}^{M}\left(\frac{N_k}{N}\right)^2\right)\sum_{i=1}^{N}\sum_{j=1}^{N}S_{cos}(\mathbf{y}_i,\mathbf{y}_j), \tag{21}$$

and

$$V_{BTW}^{cos}(\mathcal{Q},\mathcal{Y}) = \frac{1}{N^2}\sum_{k=1}^{M}\left(\left(\frac{N_k}{N}\right)\sum_{i=1}^{N_k}\sum_{j=1}^{N}S_{cos}(\mathbf{y}_i^{(k)},\mathbf{y}_j)\right). \tag{22}$$

Note that when the information needs are equiprobable, i.e., $P(q)=\frac{1}{M}$, then QSMI can be simplified as:

$$I_T^{cos}(\mathcal{Q},\mathcal{Y}) = V_{IN}^{cos}(\mathcal{Q},\mathcal{Y}) - V_{BTW}^{cos}(\mathcal{Q},\mathcal{Y}). \tag{23}$$

Therefore, when this assumption holds, QSMI can be easily implemented just by defining the similarity matrix $\mathbf{S}\in\mathbb{R}^{N\times N}$, where $[\mathbf{S}]_{ij}=S_{cos}(\mathbf{y}_i,\mathbf{y}_j)$ and the notation $[\mathbf{S}]_{ij}$ is used to refer to the i-th row and j-th column of matrix $\mathbf{S}$. Then, QSMI can be calculated as:

$$I_T^{cos} = \frac{1}{N^2}\mathbf{1}_N^T\left(\mathbf{\Delta}\odot\mathbf{S}-\frac{1}{M}\mathbf{S}\right)\mathbf{1}_N, \tag{24}$$

where the indicator matrix is defined as:

$$[\mathbf{\Delta}]_{ij}\begin{cases}1, & \text{if the } i\text{-th and the } j\text{-th documents fulfill the same information need}\\ 0, & \text{otherwise.}\end{cases} \tag{25}$$

The notation $\mathbf{1}_N\in\mathbb{R}^N$ is used to refer an N-dimensional vector of 1s, while the operator $\odot$ denotes the Hadamard product between two matrices. This formulation also allows for directly handling information needs that are not mutually exclusive. In that case, the values of the indicator matrix are appropriately set to 1, if two images share at least one information need.

Instead of directly optimizing the QSMI, we propose using a "square clamp" around the similarity matrix $\mathbf{S}$, smoothing the optimization surface. That is, instead of directly maximizing $\mathbf{\Delta}\odot\mathbf{S}$ and minimizing $\mathbf{S}$, we proposed maximizing $\mathbf{\Delta}\odot(1-\mathbf{S})\odot(1-\mathbf{S})$ and minimizing $\mathbf{S}\odot\mathbf{S}$. In this way, the values of the similarity matrix are "clamped" around 1 and 0 respectively.

This formulation penalizes the pairs with larger error more heavily than those with smaller error, allowing for discovering more robust solutions. Therefore, the final the loss function is re-derived as:

$$\mathcal{L}_{QSMI} = \frac{1}{N^2}\mathbf{1}_N^T \left(\mathbf{\Delta} \odot (\mathbf{S}-1) \odot (\mathbf{S}-1) - \frac{1}{M}(\mathbf{S} \odot \mathbf{S}) \right) \mathbf{1}_N \tag{26}$$

Indeed, as we also experimentally demonstrate in the ablation study given in Section III, this modification can significantly improve the optimization stability, leading to better solutions.

Note that the complexity for calculating QSMI is quadratic, since calculating V_{IN}^{cos}, V_{ALL}^{cos} and V_{BTW}^{cos} require a quadratic number of similarity calculations, i.e., $O(N^2)$. To allow for scaling to larger datasets, batch-based optimization is used. That allows for reducing the complexity of QMI from $O(N^2)$ to just $O(N_B^2)$ for one optimization step, where N_B is the used batch size that typically ranges from 64 to 256. Therefore, the total complexity for completing one training epoch is reduced from $O(N^2)$ to $O(NN_B)$. However, this implies that each batch will contain images only from a subsample of the available information needs. This in turn means that the observed in-batch prior probability $P(q)$ will not match the collection-level prior, leading to underestimating the influence of the potential V_{ALL} to the optimization. To account for this discrepancy, we propose a simple heuristic to estimate the in-batch prior, i.e., the value of M in Eq. (26): M is estimated as

$$M = N_B^2/(\mathbf{1}_{N_B}^T \mathbf{\Delta} \mathbf{1}_{N_B}), \tag{27}$$

where N_B is the batch size. To understand the motivation behind this, consider that if the whole collection was used for the optimization, then the number of 1s in $\mathbf{\Delta}$ would be: $M(\frac{N}{M})^2 = \mathbf{1}_N^T \mathbf{\Delta} \mathbf{1}_N$. Solving this equation for M yields the value used for approximating M. Note that the value of M is not constant and depends on the distribution of the samples in each batch. It was experimentally verified that this heuristic indeed improves the performance of the proposed method over using a constant value for M.

d) Deep Supervised Hashing using QSMI: The proposed QSMI is used to train a deep neural network to extract short binary hash codes, as shown in Fig. 1. Let $\mathbf{x}$ be the raw representation of an image (e.g., the pixels of an image) and let $\mathbf{y} = f_{\mathbf{W}}(\mathbf{x}) \in \mathbb{R}^n$ be the output of a neural network $f_{\mathbf{W}}(\cdot)$, where $\mathbf{W}$ denotes the matrix of the parameters of the network and n is the length of the hash code. Apart from learning a representation that minimizes the J_{QSMI} loss, the network must generate an output that can be easily translated into a binary hash code. Several techniques have been proposed to this end, e.g., using the *tanh* function [27]. In this work, the output of the network is required to be close to two possible values, either 1 or -1. Therefore, the used hashing regularizer is defined, following the recent deep supervised hashing approaches [28], as:

$$\mathcal{L}_{hash} = \sum_{i=1}^{N} \| |\mathbf{y}_i| - \mathbf{1}_n \|_1, \tag{28}$$

where $|\cdot|$ denotes the absolute value operator and $||\cdot||_1$ denotes the l^1 norm. The final loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{QSMI} + \alpha \mathcal{L}_{hash}, \tag{29}$$

where α is the weight of the hashing regularizer. The network $f_{\mathbf{W}}(\cdot)$ can be then trained using gradient descent, i.e., $\Delta\mathbf{W} = -\frac{\partial J}{\partial \mathbf{W}}$. After training the network, the hash codes can be readily obtained using the $sign(\mathbf{y})$ function.

III. Experimental Evaluation

A. Datasets and Hyper-parameters

Three datasets were used for evaluating the proposed method. The first one, the Fashion MNIST dataset is composed of 60,000 training images and 10,000 test images [29]. The whole training set was used to train the networks and build the database, while the test set was used to query the database and evaluate the performance of the methods. The CIFAR-10 dataset was also used for evaluating the proposed method [16], while again the whole training set was used to train the networks and build the database. The test set was used to query the database and evaluate the performance of the methods. Finally, the third dataset used for the evaluation is the NUS-WIDE, which is a large-scale dataset that contains 269,648 images that belong to 81 different concepts [17]. The images were resized to 224×224 pixels before feeding them to the network. Following [30], only images that belong to the 21 most frequent concepts, i.e., 195,834 images, were used for training/evaluating the methods. Each image might belong to multiple different concepts, i.e., the information needs are not mutually exclusive. For evaluating the methods, two images were considered relevant if they share at least one common concept, which is the standard protocol used for this dataset [30]. Similarly to the other two datasets, the whole training set (193,734 randomly sampled images) was used to train the networks and build the database, while 2,100 randomly sampled queries (100 from each category) were employed to evaluate the methods.

The hyper-parameters employed for the conducted experiments are summarized in Table I. We selected the best parameters for the two other evaluated methods, i.e., DSH and DPSH, by performing line search for each parameter. The Adam optimizer [31], with the default hyper-parameters, was used for the optimization. The learning rate was set to $\eta = 0.001$, while batches of 128 samples were used. The experiments were repeated 5 times and the mean value of each of the evaluated metrics is reported, except otherwise stated. All the datasets were preprocessed to have zero mean and unit variance, according to the statistics of the dataset used for the training.

For the experiments conducted on the Fashion MNIST dataset a relatively simple Convolutional Neural Network (CNN) architecture was employed, as shown in Table II. For the CIFAR10 dataset, a DenseNet-BC-190 (growth rate 40 and compression rate 2) [32], which was pretrained on the CIFAR dataset, was used. For the NUS-WIDE dataset, a DenseNet-201 (growth rate 32 and compression rate 2), which was

TABLE I
PARAMETERS USED FOR THE CONDUCTED EXPERIMENTS

Parameter	Method	F. MNIST	CIFAR10	NUS-WIDE
Epochs	all	50	5	50
α	DSH	10^{-5}	10^{-5}	10^{-5}
α	QSMIH	10^{-2}	10^{-2}	10^{-1}
η	DPSH	5*	3	5

(3 for 36-48 bits)

TABLE II
NETWORK ARCHITECTURE USED FOR THE FASHION MNIST DATASET

Layer	Kernel Size	Filters / Neurons	Activation
Convolution	5×5	32	ReLU [34]
Max Pooling	2×2	-	-
Convolution	5×5	64	ReLU [34]
Max Pooling	2×2	-	-
Dense	-	equal to code length	-

TABLE IV
FASHION MNIST EVALUATION (THE MAP FOR DIFFERENT HASH CODE LENGTHS IS REPORTED)

Method	12 bits	24 bits	36 bits
DSH	0.761 ± 0.018	0.792 ± 0.012	0.809 ± 0.008
DPSH	0.767 ± 0.023	0.773 ± 0.005	0.774 ± 0.008
QSMIH	$\mathbf{0.842 \pm 0.012}$	$\mathbf{0.857 \pm 0.004}$	$\mathbf{0.858 \pm 0.007}$

TABLE V
CIFAR10 EVALUATION (THE MAP FOR DIFFERENT HASH CODE LENGTHS IS REPORTED)

Method	8 bits	12 bits	24 bits	36 bits	48 bits
DSH*	0.936	0.958	0.967	0.970	0.970
DPSH*	0.776	0.933	**0.971**	**0.971**	**0.971**
QSMIH	**0.962**	**0.970**	**0.971**	**0.971**	**0.971**

Results using our implementation of DSH [28] and DPSH [30].

pretrained on the Imagenet dataset [33], was also employed. The feature representation was extracted from the last average pooling layers of the network. Then, two fully connected layers were used: one with N_H neurons and rectifier activation functions, and one with as many neurons as the desired code length (no activation function was used for the output layer). The size of hidden layer was set to $N_H = 64$ for the CIFAR10 dataset and to $N_H = 2048$ for the NUS-WIDE dataset. To speedup the training process for these two datasets, we back-propagated the gradients only to the last two layers of the network

B. Experimental Evaluation

First, an ablation study was performed using the Fashion MNIST data. The effect of various design choices, i.e., using or not the proposed clamped loss and spherical formulation, is evaluated in Table III. The mean Average Precision (mAP) is averaged over 5 runs, while the code length was set to 48 bits for these experiments. The precision for retrieved documents withing hamming distance 2 from the query is also reported. Several conclusions can be drawn from the results reported in Table III. First, employing the proposed clamped loss, instead of directly optimizing the QMI ($\sigma = 10$), improves the hashing precision, confirming our hypothesis regarding the benefits of using the proposed clamped loss. The value for the width ($\sigma = 10$) was selected after performing cross-validation experiments to ensure a fair comparison between the QMI and QSMI methods. Indeed, when the spherical formulation is used (QSMI method), then the mAP further increase to 86.1% from 72.7% (standard QMI formulation).

The proposed method was compared to two other state-of-the-art techniques, the Deep Supervised Hashing (DSH) method [28] and the Deep Pairwise Supervised Hashing (DPSH) method [30]. The evaluation results are shown in Table IV. The proposed method is abbreviated as "QSMIH" and significantly outperforms the other two state-of-the-art pairwise hashing techniques, highlighting the importance of using theoretically-sound objectives for learning deep supervised hash codes.

The evaluation results for the CIFAR10 dataset are reported in Table V. The proposed method outperforms all the other techniques by a large margin for small code lengths, i.e., 8 and 12 bits. For larger hash codes, the proposed method performs equally well with the DSH and DPSH methods. However, the proposed method is capable of achieving almost the same performance as the DSH and DPSH methods using less than half the bits, highlighting the expressive power of the proposed technique. The proposed QSMIH method was evaluated using the larger-scale NUS-WIDE dataset, as shown in Table VI. Again, the proposed method outperforms the rest of the evaluated methods for any code length. Note that the improvements obtained with the proposed method are larger for this more challenging dataset, compared to CIFAR-10.

TABLE III
ABLATION STUDY USING THE FASHION MNIST DATASET (THE MAP IS REPORTED)

Clamped	Spherical	mAP	precision (< 2bits)
No	No	0.727 ± 0.008	0.674 ± 0.021
Yes	No	0.816 ± 0.009	0.864 ± 0.010
Yes	Yes	$\mathbf{0.861 \pm 0.004}$	$\mathbf{0.876 \pm 0.004}$

TABLE VI
NUS-WIDE EVALUATION (THE MAP FOR DIFFERENT HASH CODE LENGTHS IS REPORTED)

Method	8 bits	12 bits	24 bits	36 bits	48 bits
DSH	0.660	0.659	0.671	0.689	0.694
DPSH	0.735	0.748	0.759	0.758	0.755
QSMIH	**0.746**	**0.753**	**0.766**	**0.764**	**0.763**

TABLE VII
CIFAR10 EVALUATION: COMPARISON WITH OTHER STATE-OF-THE-ART APPROACHES (THE MAP FOR DIFFERENT HASH CODE LENGTHS IS REPORTED)

Method	Source	16 bits	32 bits	64 bits
DNNH	[5]	0.555	0.558	0.623
DSH	[5]	0.689	0.691	0.716
DPSH	[5]	0.646	0.661	0.686
HashNet	[5]	0.703	0.711	0.739
HashGAN	[6]	0.668	0.731	0.749
PGDH	[5]	0.736	0.741	0.762
MIHash	[7]	0.760	**0.776**	0.761
QSMIH	-	**0.762**	**0.776**	**0.780**

For all the evaluated methods, apart from the proposed one, the results are as reported in the corresponding literature (please refer to "Source"). The same setup and neural network architecture are used for the evaluated methods.

Finally, the proposed method was also compared to other competitive deep supervised methods in Table VII. The proposed method is compared to DNNH [35], DSH [28], DPSH [30], HashNet [36], MIHash [20], [14], HashGAN [6] and PGDH [5] methods using the same evaluation protocol, i.e., 1,000 test images are sampled, 5,000 images are used for training the models and the rest of them are used to form the database. Note that the results for the competitive methods are as reported in the corresponding literature [5], [6], [7], while we also used the same neural network architecture (AlexNet [37]) for the experiments conducted using the proposed method. The proposed method significantly outperforms most of the evaluated methods, including the recently proposed HashGAN [6] and PGDH [5] methods. It also achieves comparable performance with the MIHash approach for 16 and 32 bits. However, for longer hash codes (64 bits), it slighltly further increases the mAP to 0.78 from 0.76 (next best performing method).

IV. CONCLUSIONS

A deep supervised hashing algorithm, adapted to the needs of large-scale hashing, which optimizes the learned codes using an information-theoretic measure, the Quadratic Spherical Mutual Information, was proposed. The proposed method was evaluated using three different datasets and evaluation setups and compared to other state-of-the-art supervised hashing techniques. The proposed method outperformed all the other evaluated methods regardless the size of the used dataset and the training setup, exhibiting a significantly more stable behavior than the rest of the evaluated methods.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449 (OpenDR). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. Annual Symposium on Computational Geometry*, 2004, pp. 253–262.

[2] Y. Chen, Z. Lai, Y. Ding, K. Lin, and W. K. Wong, "Deep supervised hashing with anchor graph," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9796–9804.

[3] C. Deng, E. Yang, T. Liu, and D. Tao, "Two-stream deep hashing with class-specific centers for supervised image search," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[4] X. Luo, P.-F. Zhang, Z. Huang, L. Nie, and X.-S. Xu, "Discrete hashing with multiple supervision," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2962–2975, 2019.

[5] X. Yuan, L. Ren, J. Lu, and J. Zhou, "Relaxation-free deep hashing via policy gradient," in *Proc. European Conf. on Computer Vision*, 2018, pp. 134–150.

[6] Y. Cao, B. Liu, M. Long, and J. Wang, "Hashgan: Deep learning to hash with pair conditional wasserstein gan," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 1287–1296.

[7] Y. Shen, J. Qin, J. Chen, L. Liu, and F. Zhu, "Embarrassingly simple binary representation learning," in *Proc. Int. Conf. in Computer Vision - Compact and Efficient Feature Representation and Learning in Computer Vision*, 2019.

[8] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. AAAI Conf. on Artificial Intelligence*, 2014, pp. 2156–2162.

[9] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.

[10] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. Int. Joint Conf. on Artificial Intelligence*, 2016, pp. 2415–2421.

[11] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. on Image Processing*, vol. 24, no. 12, pp. 4766–4779, 2015.

[12] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 1556–1564.

[13] J. C. Principe, *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[14] F. Cakir, K. He, S. A. Bargal, and S. Sclaroff, "Hashing with mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2424–2437, 2019.

[15] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, 2003.

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

[17] 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. of ACM Conf. on Image and Video Retrieval*, 2009.

[18] M. Almasri, C. Berrut, and J.-P. Chevallet, "A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information," in *Proc. European Conf. on Information Retrieval*, 2016, pp. 709–715.

[19] J. Hu, W. Deng, and J. Guo, "Improving retrieval performance by global analysis," in *Proc. Int. Conf. on Pattern Recognition*, vol. 2, 2006, pp. 703–706.

[20] F. Cakir, K. He, S. Adel Bargal, and S. Sclaroff, "Mihash: Online hashing with mutual information," in *Proc. IEEE Int. Conf. on Computer Vision*, 2017.

[21] M. Sanderson, "Christopher d. manning, prabhakar raghavan, hinrich schütze, introduction to information retrieval, cambridge university press. 2008. isbn-13 978-0-521-86571-5, xxi+ 482 pages." *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[22] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol. 2, no. 1, pp. 1–19, 2006.

[23] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, 2016.

[24] ——, "Learning bag-of-embedded-words representations for textual information retrieval," *Pattern Recognition*, vol. 81, pp. 254–267, 2018.
[25] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.
[26] N. I. Fisher, *Statistical analysis of circular data*. Cambridge University Press, 1995.
[27] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 769–790, 2018.
[28] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 2064–2072.
[29] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[30] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *Proc. Twenty-Fifth Int. Joint Conf. on Artificial Intelligence*, 2016, pp. 1711–1717.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2015.
[32] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.
[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. on Machine Learning*, 2010, pp. 807–814.
[35] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3270–3278.
[36] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. on Computer Vision*, 2017, pp. 5608–5617.
[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.