MuseHash: Supervised Bayesian Hashing for Multimodal Image Representation

Maria Pegia
mpegia@iti.gr
CERTH-ITI
Thessaloniki, Greece

Björn Þór Jónsson
bjorn@ru.is
Reykjavik University
Reykjavík, Iceland

Anastasia Moumtzidou
moumtzid@iti.gr
CERTH-ITI
Thessaloniki, Greece

Ilias Gialampoukidis
heliasgj@iti.gr
CERTH-ITI
Thessaloniki, Greece

Stefanos Vrochidis
stefanos@iti.gr
CERTH-ITI
Thessaloniki, Greece

Ioannis Kompatsiaris
ikom@iti.gr
CERTH-ITI
Thessaloniki, Greece

ABSTRACT

This paper presents a novel method for supporting multiple modalities in the field of image retrieval, called **Mu**ltimodal Baye**s**ian Supe**r**vised **Hash**ing (**MuseHash**). The method takes into consideration the semantic information of the training data through the use of Bayesian regression to estimate the semantic probabilities and statistical properties in the retrieval process. This method is an extension of the previously proposed Bayesian ridge-based Semantic Preserving Hashing (BiasHash) method. Experimentation on various domain-specific and benchmark datasets demonstrates that MuseHash outperforms six existing state-of-the-art methods in image retrieval performance, regardless of the feature extractor type, code length, and visual or textual descriptors used. This highlights the robustness and adaptability of MuseHash, making it a promising solution for multimodal image retrieval.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Computing methodologies** → *Supervised Learning*; • **Mathematics of computing** → Probability and statistics.

KEYWORDS

Supervised Hashing, Bayesian Ridge Regression, Late fusion, Cross-modal retrieval

ACM Reference Format:
Maria Pegia, Björn Þór Jónsson, Anastasia Moumtzidou, Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2023. MuseHash: Supervised Bayesian Hashing for Multimodal Image Representation. In *ACM ICMR '23: ACM International Conference on Multimedia Retrieval, June 10–15, 2023, Thessaloniki, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

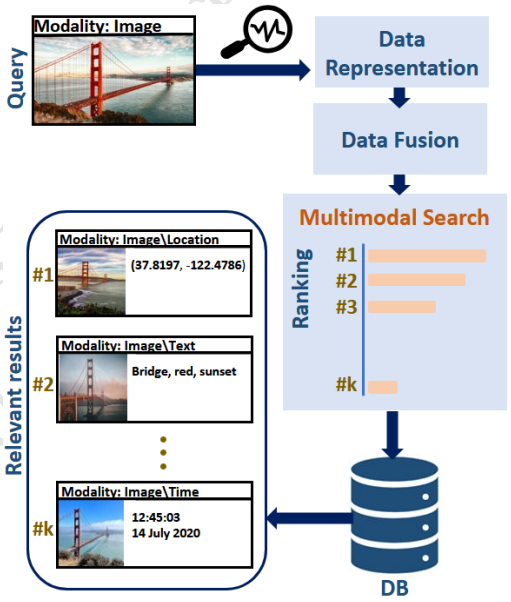

Figure 1: Image retrieval.

1 INTRODUCTION

With the rapid development of the Internet and mobile devices, multimedia data collections have seen explosive growth. Not only are there larger general collections, with a variety of media types, but also a larger variety of specialised collections. Early collection genres included medical images and art collections, which are well studied in the literature [1, 13, 29], and satellite image collections which have been gaining attention [16–18], but more recent and less-studied genres include underwater [26, 31] and aerial

footage [14, 22]. Furthermore, due to (a) media capture devices providing robust metadata and (b) improvements in automatically generated media annotations, the number of available media modalities has also grown. These different modalities can provide semantic correlations, which may be used to support semantically-relevant results of all modalities in response to a unimodal query, such as a visual example or a textual description.

Image retrieval, in particular, has attracted interest among researchers from many fields, including image processing, multimedia retrieval, and computer vision. Modern image retrieval requires a representation for each modality, fusing the different modalities into a common representation, sorting the collection items and returning the items most relevant to the query, as shown in Figure 1. Due to the large volume of the collections and the semantic gap between the digital representation and human perception [28], effective retrieval of multimodal data remains a challenge. Hashing methods, including deep hashing methods, have recently been widely used to represent media modalities for similarity search in multimedia, due to their low memory requirements and efficient comparison. In this paper, we therefore focus on hashing methods, particularly aiming to minimise the aforementioned semantic gap of data from different modalities. Hashing approaches are distinguished into single-view [6, 27] and multi-view approaches [4, 11, 20, 21, 23, 35–37]. The former approaches can only handle one modality, while the latter approaches support two or more modalities. In addition, they can categorised into unsupervised [6, 11, 37] and supervised [4, 15, 20, 21, 23, 27, 35, 36] depending on the method of learning hash functions. In general, supervised methods can take more advantage of the inner relationships of data from annotation and due to that they perform better than unsupervised methods. Therefore we emphasise on supervised methods. Although there is thus a plethora of hashing methods for image representation, there is no method in the literature that can universally combine many modalities for both the collection and query.

Recent works have applied Bayesian frameworks [27, 35] to supervised single-view hashing, by adding both a measure of uncertainty and weight regularisation to their predictions. The BiasHash method proposed in [27] was shown to outperform state-of-the-art single-view approaches. In this paper, we extend the BiasHash approach to a multi-view approach, MuseHash, which applies Bayesian regression per modality, uses the sign of the produced Bayesian feature values for binary codes' generation, and fuses the hash-codes into a complete image representation.

The main contributions of this paper are summarised as follows:

- We propose MuseHash, a novel multimodal Bayesian hashing approach for efficient cross-modal retrieval.
- We validate the effectiveness of our proposed approach using five benchmark datasets from three different genres, including underwater and aerial video footage. To the best of our knowledge, this is the first time these two genres have been explored in the image retrieval context.
- In an ablation study, we explore the influence of various parameters, such as training size and visual feature descriptors, on the performance of the MuseHash approach.

The remainder of this paper is structured as follows. Section 2 examines the relevant state of the art, while Section 3 technically describes the proposed MuseHash approach. Experimental results are presented in Section 4 and the paper concludes with a brief summary in Section 5.

2 RELATED WORK

Various multimodal hashing methods have been proposed for image retrieval. In this section, we discuss state-of-the-art unsupervised and supervised methods from the literature.

Unsupervised hashing methods usually learn hash functions from data distribution in order to preserve the structures of training data. SelfTaught Hashing (STH) [37] finds the optimal l-bit binary codes for all documents in the given corpus via unsupervised learning, and then trains l classifiers via supervised learning to predict the l-bit code for any query.

Unsupervised Deep Hashing with Pseudo Labels (UDHPL) [11] generates the pseudo-labels through the Bayes' rule and maximizes the correlation between the projection vectors of pseudo-labels and deep features. Lightweight Augmented Graph Network Hashing (LAGNH) [6] uses a lightweight deep learning network with the assistance of the auxiliary semantics, which significantly reduces the number of parameters of the network and accelerates the training process.

Supervised methods, on the other hand, learn hash functions using supervised information, and generally outperform unsupervised methods. Self-Supervised Adversarial Hashing Network (SSAH) [15] incorporates a self-supervised semantic network coupled with multi-label information, and carries out adversarial learning to maximize the semantic relevance and feature distribution consistency between different modalities. Fast Cross-Modal Hashing (FCMH) [35] introduces an auxiliary variable to approximate the binary code so that the binary code can be optimized by minimizing the quantization error.

There are also supervised methods that adapt adversarial learning or transfer knowledge. Generalized Semantic Preserving Hashing (GSPH) [23] learns the optimum hash codes for the two modalities simultaneously, and then learns the hash functions to map from the features to the hash codes. Matrix Tri-Factorization Hashing Framework (MTHF) [21] aims to transfer knowledge from single-modal source domain to cross-modal target domain for promoting cross-modal retrieval.

Some supervised methods use deep learning networks (like, CNN) for feature learning and hash function learning. Deep Cauchy Hashing (DCH) [4] inserts Cauchy cross-entropy loss and a Cauchy quantization loss based on Cauchy distribution into the network for learning compact and concentrated hash code for image retrieval. Label-Attended Hashing (LAH) [36] separately generates the image representation and label co-occurrence embeddings, and also learns hash functions following a Cauchy distribution approach.

Other supervised methods use similarity matrix for semantic information. Semantic Preserving Hashing (SePH) [20] generates one unified hash code for all observed views of any instance by transforming the given semantic affinities of training data into a probability distribution. This is done with the use of kernel logistic regression for minimizing their Kullback-Leibler divergence [9]. A recent supervised hashing method, Bayesian Ridge-based Semantic Preserving Hashing (BiasHash) [27], has been proposed for image

retrieval. It inserts a Bayesian framework for learning hash functions and considers that the choice of the training data does not affect the performance of the method.

The state-of-the-art hashing approaches for image representation are limited to handling only two modalities due to limited datasets with more modalities and inability to handle complex relationships. This failure to address multimodal data and queries inspired us to propose a Bayesian-based supervised framework for multimodal retrieval. Our approach combines multiple modalities and outperforms existing state-of-the-art supervised methods. Additionally, our method can be adapted to handle various types of data, such as underwater or aerial visual information. This makes our approach more versatile and applicable to a wider range of data.

3 METHODOLOGY

Figure 2 illustrates an overview of the proposed MuseHash framework. The framework consists of an offline training phase, where the hash functions are first learned and then applied to the retrieval collection, and an online querying phase, where the learned hash functions are used for query images. The structure of the feature extraction and the learnt hash functions are shared between the phases, as represented with orange color in the figure.

In the offline training phase, MuseHash uses the training labels to create an affinity matrix, find semantic probabilities, and map them to Hamming space. It also extracts features from each image in the training set and uses them to learn hash functions for each modality (like visual or textual) through Bayesian ridge regression. These hash functions are then applied to all images in the retrieval set, creating hash codes that are stored in a database for later use.

In the online quering phase, MuseHash extracts the features of each existing modality of the query image and computes hash codes using the respective learnt hash functions. These hash codes are then fused into a single feature vector, which is used to query the database of hash codes. Finally, the results are ranked and the top k relevant results are returned.

In the remainder of this section, we present the proposed method in detail. We start by defining the notation used in the presentation (Section 3.1). Since MuseHash extends the BiasHash approach to multiple modalities, we then review the BiasHash method for the single-modality case (Section 3.2), before describing the MuseHash approach to the multi-modality case (Section 3.3).

3.1 Notation

Let $\mathcal{I}$ be the training set of size $|\mathcal{I}| = n$, with I_i its i-th instance. We define $L \in \{0,1\}^{n \times l}$ as the ground truth labels (typically semantically derived) of the training set, used for the supervised learning process. We define $X^m \in \mathbb{R}^{n \times d_m}$ as the feature vectors from the m-th modality, where each vector is of dimensionality d_m. Each training set instance I_i thus consists of an M-tuple of feature vectors $(X^1, X^2, \ldots, X^M)$ and the ground truth label vector $L_{i,.}$.

Let $A \in [0,1]^{n \times n}$ be the affinity matrix and $H \in \{0,1\}^{n \times d_c}$ the learnt hash codes of the training set, where d_c is the number of bits in the hash codes. Each instance of $H_{i,.}$ then corresponds to the projection of each training set instance I_i. We denote with U_M the set of learnt hash functions for the M modalities, and $u_m^k \in \mathbb{R}^{d_m}$ the learnt hash function of m-th modality and k-th bit, for $1 \le k \le d_c$. Let $c^m \in \{0,1\}^{d_c}$ be the hash code of the m-th modality and c_k^m its k-th bit, for $1 \le k \le d_c$. In BiasHash, the hash code c^1 of the single modality X^1 is stored directly in the collection, while for MuseHash the hash codes $(c^1, c^2, \ldots, c^M)$ for all M modalities are fused into a single feature vector, as described below. In BiasHash, the hash code H_q of a given query q can be computed by $H_q = sign(u_1 \bullet X^1)$, where the operator $\bullet$ is the inner product between vectors. while MuseHash uses the same equation for each modality.

3.2 BiasHash

As the proposed MuseHash method builds on BiasHash, as mentioned above, we describe the BiasHash method here, and then outline the differences in Section 3.3. BiasHash is a single-modality hashing approach ($M = 1$, in the notation above), based on applying Bayesian regression for projecting the semantic relationships of data into Hamming space.

The BiasHash computes the affinity matrix of training set using the ground truth labels for items I_i and I_j by:

$$A_{i,j} = \frac{< L_{i,.}, L_{j,.} >}{||L_{i,.}||||L_{j,.}||} \tag{1}$$

where $< ., . >$ is the dot-product of two vectors. Then the probabilities in semantic space $\mathcal{P}$ are computed by:

$$p_{i,j} = \frac{A_{i,j}}{\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} A_{i,j}} \tag{2}$$

The corresponding semantic probabilities $q_{i,j}$ of instances in Hamming space $\mathcal{Q}$ can be derived using the work of van der Maaten and Hinton [33] for t-distribution and the work for Kullback-Leibler divergence [9] to measure the differences between $\mathcal{Q}$ and $\mathcal{P}$. Due to its NP-hardness [24], the problem is relaxed to:

$$\begin{aligned} \Psi &= \min_{\hat{H} \in \mathbb{R}^{n \times d_c}} \sum_{s=1}^{n} \sum_{t=1, s \neq t}^{n} p_{s,t} log \frac{p_{s,t}}{q_{s,t}} + \frac{a}{C} |||\hat{H}| - I||_2^2 \\ \text{with} \quad q_{s,t} &= \frac{(1 + ||\hat{H}_{s,.} - \hat{H}_{t,.}||_2^2)^{-1}}{\sum_{k=1}^{n} \sum_{m=1, m \neq k}^{n} (1 + ||\hat{H}_{k,.} - \hat{H}_{m,.}||_2^2)^{-1}} \end{aligned} \tag{3}$$

where Ψ is the minimization problem, a is a model parameter for weighting quantization loss and $C = n \times d_c$ is a normalization factor for the hash code length and the training set size. BiasHash uses gradient descent [30] to find $\hat{H}$ for that purpose and computes a Hamming space matrix $H = sign(\hat{H})$. Next, BiasHash uses Bayesian regression [25, 32] to learn the hash function u_1 that projects the visual feature X^1 to hash code c^1. The hash code is stored in the database, while the learn hash functions are stored outside the database.

For a given query q, BiasHash computes its hash code H_q and sorts the retrieval set based on the Hamming distance between the bits of H_q and H_i, ., in ascending order using:

$$h(H_q, H_i, .) = bit\_count(H_q \oplus H_i, .) \tag{4}$$

where $\oplus$ denotes the XOR operation between the bits of H_q and H_i, ., and bit_count counts the number of 1s in the binary XOR result. Finally, BiasHash returns the top k elements from the ordered retrieval set.

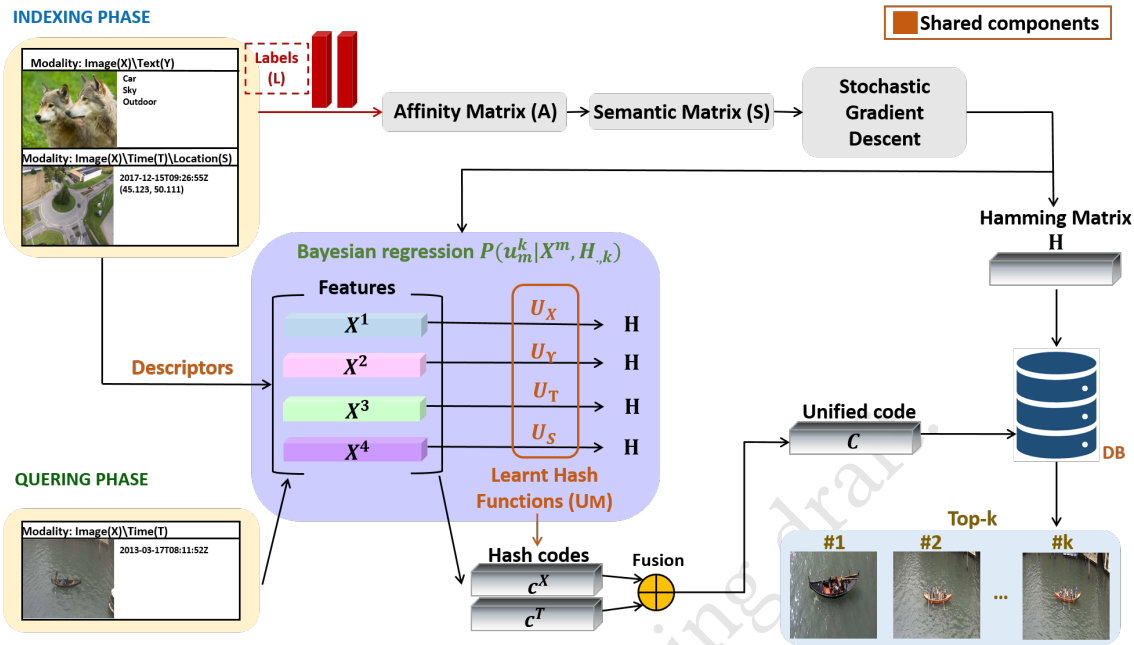

Figure 2: Overview of the proposed multimodal indexing and retrieval method.

Table 1: Five benchmark datasets used in experiments.

Dataset	Ground Truth Labels	Modalities				Collection Size	Retrieval Set Size	Training Set Size	Test Set Size
		Image	Text	Time	Location				
AU-AIR	8	✓	✗	✓	✓	32283	32183	2000	100
MarDCT	24	✓	✗	✓	✗	6743	6043	1064	350
SeaDronesSee	6	✓	✗	✓	✓	5630	5249	2113	381
MIRFlickr25K	24	✓	✓	✗	✗	18357	15357	3000	2000
NUS-WIDE	10	✓	✓	✗	✗	186577	184477	4000	2100

3.3 MuseHash

In the proposed MuseHash framework, each image in the training set contains an M-tuple of different feature vectors, along with the ground truth labels L. Inspired by the work of [19], and as already mentioned in Section 2, MuseHash is a Bayesian-based supervised method for image retrieval, which can be applied for any number of modalities. Since queries are often posed using a single modality, however, the proposed scheme is designed to handle unimodal queries.

For each available modality m, a feature vector X^m is used. For each feature vector, the corresponding hash function u_m is learnt using Bayesian regression using the same approach as in BiasHash. For each image in the retrieval set, the corresponding hash codes are then fused into a single unified hash code:

$$C_i = f(H_{i,1}, H_{i,2}, \ldots, H_{i,M}) = \sum_{s=1}^{s=M} \sum_{t=1}^{t=M} H_{i,t} \oplus H_{i,s} \qquad (5)$$

where $\oplus$ denotes the XOR operation between the hash codes through all binary codes. To increase flexibility during query processing, the hash codes for each modality are stored in the database, while the combined hash code for all modalities is computed on the fly.

For a given query, the method again computes the hash code of each existing modality c^m using the corresponding hash function u_M, and fuses the resulting hash codes into a single feature vector C_q. The retrieval set is then sorted based on the Euclidean distance between C_q and C_i, in ascending order, and returns the top k items of the ordered retrieval set.

4 EXPERIMENTS

In this section, we first describe the datasets used for evaluation and the experimental setup. We then show detailed experimental results for a variety of modalities and hash code lengths, explore different visual and textual descriptors and discuss the training and testing time of the methods.

4.1 Datasets

We use for our experiments datasets of different types. Specifically, one aerial dataset (AU-AIR [3]), two underwater datasets

(MarDCT [2], SeaDronesSee [34]) and two traditional benchmark datasets from the literature (MIRFlickr25K [12], NUS-WIDE [5]).

AU-AIR The AU-AIR dataset is a collection of 8 video clips recorded for aerial traffic surveillance at a specific intersection in Aarhus, Denmark. The footage was taken on windless days to eliminate any disturbance caused by wind. The video clips capture various lighting conditions such as sunny, partly sunny, and cloudy weather, due to the time of the day and the weather conditions. The videos have been recorded at 30 frames per second (fps) and have a resolution of 1920x1080 pixels. To prevent the redundant occurrence of frames, the dataset consists of five frames per second, resulting in 32,823 frames in total. These frames were extracted from the raw videos.

MarDCT The MarDCT (Maritime Detection Classification and Tracking) benchmark is a dataset that was acquired through the use of the ARGOS system. The ARGOS system has been operating in Venice, Italy since 2006 and includes 14 survey cells that cover the entire Grand Canal of Venice. The system is capable of automatically extracting up to 2,000 snapshots of boats per day for each cell. The dataset is particularly challenging due to the high variability of boats navigating in Venice, which makes recognition and classification tasks difficult even for humans. The data sets were generated by using an automatic acquisition procedure that extracts snapshots of boats by using the detection and tracking functionality of the ARGOS system.

SeaDronesSee The SeaDronesSee is a large-scale data set of people in open water captured using various UAVs and cameras. The dataset contains videos and images of swimming probands, and is particularly useful for Search and Rescue (SAR) missions. The RGB footage in the dataset has high resolution, ranging from 3840×2160px to 5456×3632px, to enable detection and tracking of objects from a large distance. The data is carefully annotated with ground-truth bounding box labels for objects of interest, including swimmers, floaters (swimmers with life jackets), life jackets, swimmers on boats not wearing life jackets, floaters on boats wearing life jackets, and boats.

MIRFlickr25K The MIRFlickr25K dataset is a collection of 25,000 images and their associated textual tags, sourced from Flickr. The images are manually annotated with 24 unique labels. To prepare the dataset for use, the textual tags that appear less than 20 times are removed, and any instances that do not have both textual tags and manually annotated labels are also removed.

NUS-WIDE The NUS-WIDE database is a collection of 269,648 images and their associated tags, sourced from the web. The images are manually annotated with one or more of 81 concepts. For the purpose of this study, the ten most frequent concepts are selected and the corresponding 186,577 images are kept for the experimental analysis.

The dataset size, number of labels and available modalities are summarized in Table 1. The symbols "✓" and "✗" denote the existence or not of the specific modality in the dataset, respectively. The dataset was split randomly into testing and retrieval set. From the retrieval set we choose randomly some elements based on Table 1 to form the training set.

We use the notation V, A, T and S for visual, textual annotations, temporal and spatial modality, respectively. The notation $Q \rightarrow DB$ corresponds to the query modality (Q) and the database modality (DB). The letter Q represents a generic query modality and is part of the set of modalities represented by V, A, T, S, M, where M represents all existing modalities combined.

In our experiments, the four modalities were encoded into feature vectors as follows. For the visual modality (V) we use a 2048-D vector from the fc-7 layer of ResNet50 [10] pre-trained network on ImageNet. For text annotations (A), we used the BERT[1] model for extracting a 1024-D vector. Each location corresponds to a 2-D vector (S) with values (altitude, longitude). Finally, each datetime is represented as a 203-D vector (T), where the first four coordinates of the temporal feature belongs to the 4 digits of the year, the next 12 digits to the one-hot-encoding for month, the next 31 digits to the one-hot encoding for day, the next 24 to the one-hot-encoding for hours, the next 60 to the one-hot encoding for minutes, the next 60 to the one-hot-encoding for seconds. while the final 12 digits contain a binary encoding of microseconds. In the ablation study, we consider alternative feature vectors for the visual and textual modalities.

4.2 Experimental Setup

In our experiments, we use cosine similarity between semantic labels calculation of the affinity matrix A. We compute hash codes of bit length d_c = 16, 32, 64, 128. We assign a from Eq. 3 to 0.01.

For each dataset, we evaluated the impact of training set size on the performance of our proposed framework. We performed experiments for training set sizes ranging from 1,000 to 7,000 and calculated the mean average precision (mAP, defined below) for each modality. Our results showed that the mAP increased until a certain value and then it decreased. Based on these observations, we selected the optimal training set size which corresponded to the highest mAP.

We use mean Average Retrieval (mAP) to measure the retrieval performance of methods:

$$mAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{m_i}\sum_{j=1}^{m_i} precision(R_{j,i}) \tag{6}$$

where Q is the query set and m_i is the number of its ground-truth relevant instances [8, 38] in the retrieval set. Furthermore, the term $R_{i,j}$ corresponds to the subset of its ranked retrieval result from the top one to the j-th ground-truth relevant one, and precision($R_{j,i}$) measures the precision value in $R_{j,i}$.

We compare our approach with two state-of-the-art multimodal hashing methods SSAH[2], FCMH[3], and two multimodal deep hashing methods, LAH[4] and DCH[5]. In addition, we compare results on the visual modality with two further unimodal approaches, SePH and BiasHash.

[1]https://github.com/hanxiao/bert-as-service
[2]https://github.com/lelan-li/SSAH
[3]https://github.com/yxinwang/FCMH-TCyb2021?fbclid=IwAR1ZxxoUvI3ny9Y1fjFVPSTSBBWhJmqeWsz3S4sFMdLjuUvFfBEo1J2f2dU
[4]https://github.com/IDSM-AI/LAH
[5]https://github.com/thulab/DeepHash

Table 2: Multimodal scenario results for AU-AIR with different code lengths and query modalities.

Query	Method	16bit	32bit	64bit	128bit
V → M	SSAH [15]	0.8180	0.8195*	0.8179*	0.8102*
	FCMH [35]	**0.8343**	0.8533*	**0.9030**	0.8930
	SePH [20]	0.6901*	0.6923*	0.6921*	0.7011*
	LAH [36]	0.8423*	0.8501*	0.8661	0.8698
	MuseHash	0.8242	**0.8956**	0.8839	**0.8932**
T → M	SSAH [15]	0.8330	0.8390	0.8201*	0.8294*
	FCMH [35]	0.8035*	0.8013*	0.8003*	0.8000*
	SePH [20]	0.7001*	0.7111*	0.7132*	0.7201*
	LAH [36]	0.8200*	0.8231*	0.8245*	0.8311*
	MuseHash	**0.8412**	**0.8681**	**0.8951**	**0.8896**
S → M	SSAH [15]	0.8350*	0.7740*	0.7883*	0.8511*
	FCMH [35]	0.9012*	0.8800*	0.8453*	0.8200*
	SePH [20]	0.7901*	0.8002*	0.8214*	0.8276*
	LAH [36]	0.8410*	0.8496*	0.8545	0.8591
	MuseHash	**0.9626**	**0.8901**	**0.8680**	**0.8609**

We measured runtime of the experiments, which showed that all python implementations performed similarly, while the matlab implementation was slightly slower but would likely perform the same if implemented in python. Additionally, all implementations used similar (and small) amounts of memory. As the running time and memory usage were similar, we focus on evaluating the quality of the results in the remainder of this section.

4.3 Experimental Results

The following experiments were conducted in this study: 1) Multimodal Scenario, where the dataset is described using all available modalities, but queries are made using one modality, such as visual or text (Section 4.3.1); 2) Unimodal Scenario, where both the collection and the queries are made use only the visual features (Section 4.3.2); and 3) Ablation Study, where the effect of different features on retrieval performance is analyzed (Section 4.3.3).

4.3.1 Multimodal Scenario $Q \rightarrow M$. As we mentioned in Section 1, we focus on the case where different modalities may be applicable for different collections, while the queries have a single modality. Tables 2 through 6 show the mAP for the proposed MuseHash approach, compared to the four state-of-the-art multimodal methods in AU-AIR, MarDCT, SeaDronesSee, MIRFlickr25K, and NUS-WIDE dataset, respectively. The symbol $*$ indicates that the mAP value the MuseHash method is statistically significant compared to the corresponding method, using a t-test [7] to measure the significance between the results from relatively large sample of queries (Test Set Size in Table 1).

Overall, the tables show that MuseHash outperforms all the baselines in most cases for different hash code lengths (length = 16, 32, 64, 128 bit) in all five benchmark datasets. We observe that as the hash code length increases, the performance of the MuseHash mainly improves, reflecting its capability of enhance more semantic information into longer hash codes. In addition, MuseHash gives

Table 3: Multimodal scenario results for MarDCT with different code lengths and query modalities.

Query	Method	16bit	32bit	64bit	128bit
V → M	SSAH [15]	0.6578*	0.6045*	0.6046*	0.6043*
	FCMH [35]	0.7170*	0.7023*	0.6812*	0.6701*
	SePH [20]	0.6441*	0.6452*	0.6500*	0.6523*
	LAH [36]	0.7011*	0.7056*	0.7200*	0.7146
	MuseHash	**0.7729**	**0.7624**	**0.7404**	**0.7148**
T → M	SSAH [15]	0.6278	0.6299	0.6079	0.6041
	FCMH [35]	0.6018*	0.6073*	0.6134*	0.6101*
	SePH [20]	0.6001*	0.6014*	0.6075*	0.6198*
	LAH [36]	0.6211*	0.6291*	0.6301*	0.6354*
	MuseHash	**0.6500**	**0.6523**	**0.6555**	**0.6589**

Table 4: Multimodal scenario results for SeaDronesSee with different code lengths and query modalities.

Query	Method	16bit	32bit	64bit	128bit
V → M	SSAH [15]	0.8221*	0.8211*	0.8256*	0.8300*
	FCMH [35]	0.8300*	0.8323*	0.8389*	0.8401
	SePH [20]	0.8201*	0.8258*	0.8265*	0.8279*
	LAH [36]	0.8231*	0.8299*	0.8345*	0.8401*
	MuseHash	**0.8311**	**0.8346**	**0.8401**	**0.8521**
T → M	SSAH [15]	0.7901*	0.7989*	0.8011*	0.8031*
	FCMH [35]	0.7801*	0.7889*	0.8023*	0.8067*
	SePH [20]	0.7721*	0.7733*	0.7801*	0.7856*
	LAH [36]	0.7801*	0.7811*	0.7830*	0.7840*
	MuseHash	**0.8361**	**0.8385**	**0.8468**	**0.8475**
S → M	SSAH [15]	0.8430*	0.8445*	0.8489*	0.8493*
	FCMH [35]	0.8312*	0.8323*	0.8441*	0.8450*
	SePH [20]	0.8301*	0.8338*	0.8359*	0.8401*
	LAH [36]	0.8269*	0.8271*	0.8291*	0.8330*
	MuseHash	**0.8584**	**0.8601**	**0.8634**	**0.8690**

Table 5: Multimodal scenario results for MIRFlickr25K with different code lengths and query modalities.

Query	Method	16bit	32bit	64bit	128bit
V → M	SSAH [15]	**0.8464**	**0.8426**	**0.8432**	0.8429*
	FCMH [35]	0.7024*	0.7035*	0.7010*	0.7013*
	SePH [20]	0.6612*	0.6643*	0.6701*	0.6711*
	LAH [36]	0.7001	0.7023	0.7001	0.6931
	MuseHash	0.8201	0.8225	0.8228	**0.8457**
A → M	SSAH [15]	0.5526*	0.5913*	0.5253*	0.5178*
	FCMH [35]	0.7012*	0.7023*	0.7018*	0.6953*
	SePH [20]	0.6801*	0.6899*	0.6932*	0.6989*
	LAH [36]	0.5012*	0.5001*	0.5021*	0.5120*
	MuseHash	**0.7071**	**0.7121**	**0.7128**	**0.7177**

Table 6: Multimodal scenario results for NUS-WIDE with different code lengths and query modalities.

Query	Method	16bit	32bit	64bit	128bit
V → M	SSAH [15]	0.8091*	0.8101*	0.8156*	0.8200*
	FCMH [35]	0.9110*	0.9233*	0.9340*	0.9401
	SePH [20]	0.8030*	0.8058*	0.8101*	0.8189*
	LAH [36]	0.8001*	0.8099*	0.8145*	0.8201*
	MuseHash	**0.9255**	**0.9278**	**0.9300**	**0.9345**
A → M	SSAH [15]	0.6630*	0.6645*	0.6689*	0.7023*
	FCMH [35]	0.7112*	0.7143*	0.7189*	0.7201*
	SePH [20]	0.6811*	0.6878*	0.6901*	0.6944*
	LAH [36]	0.7260*	0.7301*	0.7340*	0.7401*
	MuseHash	**0.7471**	**0.7511**	**0.7549**	**0.76001**

better results for visual and spatial queries in comparison to other types. This happens due to better quality information of those modalities. The time and text information of the used datasets contain huge time intervals or sentences with noise.

However, SSAH gives the best results on MIRFlickr25K dataset for visual queries, while FCMH surpasses the compared methods in AU-AIR dataset. Note, however, that MuseHash can handle more than two modalities in contrast to these other state-of-the-art methods. To further explore the performance for the MIRFlickr25K dataset, we employed a 5-fold cross-validation methodology to evaluate the performance of our proposed method. In this case, the results obtained from the two best methods (SSAH and MuseHash) were found to be indistinguishable, indicating that both methods performed similarly.

All methods perform well on AU-AIR and SeaDronesSee datasets, particularly for visual and spatial modalities. MarDCT results are lower and may be due to the quality of visual and temporal modalities captured over long periods. MIRFlickr25K and NUS-WIDE show intermediate results with lower values for textual queries due to noisy sentences. Overall, MuseHash performs consistently well across all datasets.

4.3.2 Unimodal Visual Scenario $V \longrightarrow V$. To study the performance of the multimodal approaches in unimodal situations, we compare all the aforementioned methods with BiasHash and DCH using visual queries over the visual modality, as both methods perform unimodal queries and are designed for image retrieval. The results of those methods over the five datasets are given in Table 7. In this scenario, MuseHash outperforms all six state-of-the-art methods for all datasets.

It is interesting to compare the performance of MuseHash in this unimodal scenario to its performance in the multimodal scenario above, with a visual query modality. MuseHash performs better when using all modalities on the MarDCT, MIRFlickr25K and NUS-WIDE datasets, because the other modalities contain information of high quality in these collections. For the remaining datasets, AU-AIR and SeaDronesSee, the performance is better using only the visual modality to respresent the images, since the temporal modality is weaker as each image represents long time intervals.

Table 7: Unimodal scenario of visual modality with different code lengths.

Dataset	Method	16bit	32bit	64bit	128bit
AU-AIR	SSAH [15]	0.7980*	0.8090*	0.8101*	0.8122*
	FCMH [35]	0.8800*	0.8812*	0.8901*	0.8966*
	SePH [20]	0.7901*	0.7933*	0.7900*	0.8020*
	LAH [36]	0.8001*	0.8123*	0.8130*	0.8223*
	DCH [4]	0.8511*	0.8423*	0.8399*	0.8390*
	BiasHash [27]	0.8142*	0.8256*	0.8139*	0.8230*
	MuseHash	**0.9255**	**0.9300**	**0.9378**	**0.9401**
MarDCT	SSAH [15]	0.6312*	0.6365*	0.6367*	0.6370*
	FCMH [35]	0.6901*	0.6976*	0.6500*	0.6412*
	SePH [20]	0.6411*	0.6401*	0.6510*	0.6553*
	LAH [36]	0.7231*	0.7301*	0.7369*	0.7388*
	DCH [4]	0.7081*	0.7080*	0.7034*	0.7011*
	BiasHash [27]	0.7201*	0.7281*	0.7301*	0.7349*
	MuseHash	**0.7401**	**0.7451**	**0.7461**	**0.7488**
SeaDronesSee	SSAH [15]	0.8201*	0.8293*	0.8301*	0.8322*
	FCMH [35]	0.8201*	0.8234*	0.8254*	0.8366*
	SePH [20]	0.8003*	0.8090*	0.8100*	0.8120*
	LAH [36]	0.8230*	0.8232*	0.8251*	0.8303*
	DCH [4]	0.8399*	0.8411*	0.8456*	0.8490*
	BiasHash [27]	0.8239*	0.8389*	0.8443*	0.8501*
	MuseHash	**0.9388**	**0.9390**	**0.9400**	**0.9411**
MIRFlickr25K	SSAH [15]	0.8001*	0.8101*	0.8112*	0.8211*
	FCMH [35]	0.6882*	0.6800*	0.6771*	0.6770*
	SePH [20]	0.6601*	0.6639*	0.6670*	0.6690*
	LAH [36]	0.6811*	0.6849*	0.6881*	0.6888*
	DCH [4]	0.6831*	0.6852*	0.6833*	0.6859*
	BiasHash [27]	0.6820*	0.6820*	0.6844*	0.6871*
	MuseHash	**0.8089**	**0.8191**	**0.8210**	**0.8223**
NUS-WIDE	SSAH [15]	0.8080*	0.8093*	0.8101*	0.8120*
	FCMH [35]	0.9000*	0.9012*	0.9104*	0.9066*
	SePH [20]	0.8021*	0.8078*	0.8101*	0.8120*
	LAH [36]	0.8001*	0.8032*	0.8090*	0.8123*
	DCH [4]	0.8399*	0.8401*	0.8414*	0.8390*
	BiasHash [27]	0.8042*	0.8056*	0.8150*	0.8186*
	MuseHash	**0.9255**	**0.9261**	**0.9290**	**0.9300**

Figure 3 shows the qualitative results of image retrieval given an image query on AU-AIR. The queries were selected as the most representative from each class. Each row corresponds to a retrieval method. The first column contains the image query, while the following 10 images belong to the top-10 retrieved results of each method, ranking from left (most relevant, red color) to right (less relevant, blue color). The symbols of a green corner, with a check mark and a red corner, with an X indicate that the retrieved image is relevant to the query or not, respectively. MuseHash returns more relevant results to the given query in comparison with the state-of-the art methods.

4.3.3 Ablation Study. In addition, we perform experiments based on different descriptors for visual and textual modality and compare

Figure 3: Retrieval performance of the proposed MuseHash and compared methods on the AU-AIR benchmark dataset.

Table 8: Comparison of visual descriptors in the unimodal scenario $V \rightarrow V$.

Dataset	Method	ResNet50		VGG16	
		16bit	128bit	16bit	128bit
NUS-WIDE	SSAH [15]	0.8080	0.8222	0.7901	0.8101
	FCMH [35]	0.9000	0.9066	0.8901	0.9001
	SePH [20]	0.8021	0.8120	0.7810	0.7911
	LAH [36]	0.8001	0.8123	0.7882	0.8021*
	DCH [4]	0.8399	0.8390	0.8511	0.8423
	BiasHash [27]	0.8042	0.8186	0.8000	0.8171
	MuseHash	**0.9255**	**0.9300**	**0.9211**	**0.9298**
SeaDronesSee	SSAH [15]	0.8201	0.8322	0.8000	0.8200
	FCMH [35]	0.8201	0.8366	0.8100	0.8212
	SePH [20]	0.8003	0.8120	0.7921	0.8033
	LAH [36]	0.8230	0.8303	0.8011	0.8223
	DCH [4]	0.8399	0.8490	0.8211	0.8223
	BiasHash [27]	0.8239	0.8501	0.8252	0.8306
	MuseHash	**0.9388**	**0.9411**	**0.9277**	**0.9301**

our method with the state-of-the-art methods to check how much influence the feature descriptor has to the performance of methods.

Table 8 shows the results for two different visual feature vectors, ResNet50 which was used in the previous experiments and VGG16, a competing state-of-the-art method. For space reasons, we focus on two collections; the same pattern holds for the remaining collections. As the table shows, the performance of the different methods is largely unaffected by the visual feature vector, although VGG16 performs slightly worse. As before, however, MuseHash outperforms all other methods.

Table 9: Comparison of textual descriptors in the unimodal scenario $T \rightarrow T$.

Dataset	Method	BERT		BoW	
		16bit	128bit	16bit	128bit
NUS-WIDE	SSAH [15]	0.6626	0.6278	0.5626	0.5278
	FCMH [35]	0.7112	0.7253	0.6954	0.7011
	SePH [20]	0.6801	0.6680	0.6620	0.6660
	LAH [36]	0.7231	0.7301	0.7001	0.7121
	MuseHash	**0.7471**	**0.7477**	**0.7491**	**0.7492**

Turning to the textual modality, Table 9 shows the results for two different textual feature vectors, BERT which was used in the previous experiments and BoW, and earlier state-of-the-art method. In contrast to the visual features, some methods are affected by the textual descriptor, in particular SSAH which performs much worse using BoW. On the other hand, both SePH and MuseHash perform equally well with both feature vectors, and MuseHash retains the best performance of all methods. Overall, the results in this section demonstrate the effectiveness of MuseHash across a variety of feature vectors.

5 CONCLUSIONS

In this paper, we have transformed the Bayesian Ridge-based Semantic Preserving Hashing to support multimodal queries. This change exploits the inner relations between different modalities. Our experiments show that MuseHash consistently outperforms six state-of-the-art methods in both multimodal and unimodal scenarios across a variety of different domain-specific and benchmark image collections. The high performance is achieved with different visual (VGG16, ResNet50) and textual (BoW or BERT) descriptors

and across a range of code lengths (16bit to 128bit), regardless of feature extractor type. MuseHash is therefore robust and adaptable to multiple modalities and scenarios, surpassing the state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported by the EU's Horizon 2020 research and innovation programme under grant agreements H2020-883302 ISOLA and H2020-101004152 CALLISTO.

REFERENCES

[1] Shubham Agrawai, Aastha Chowdhary, Saurabh Agarwala, Veena Mayya, and Kamath Sowmya S. 2022. Content-based medical image retrieval system for lung diseases using deep CNNs. *International Journal of Information Technology* 14 (2022), 3619–3627. https://doi.org/10.1007/s41870-022-01007-7

[2] Domenico D. Bloisi, Luca Iocchi, Andrea Pennisi, and Luigi Tombolini. 2015. ARGOS-Venice Boat Classification. *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2015). https://doi.org/10.1109/AVSS.2015.7301727

[3] Ilker Bozcan and Erdal Kayacan. 2020. Seadronessee: A maritime benchmark for detecting humans in open water. *IEEE International Conference on Robotics and Automation (ICRA)* (2020). https://doi.org/10.1109/ICRA40945.2020.9196845

[4] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. 2018. Deep Cauchy Hashing for Hamming Space Retrieval. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). https://doi.org/10.1109/CVPR.2018.00134

[5] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. *ACM International Conference on Image and Video Retrieval (ICMR)* (2009), 1–9. https://doi.org/10.1145/1646396.1646452

[6] Hui Cui, Lei Zhu, Jingjing Li, Zhiyong Cheng, and Zheng Zhang. 2021. Two-pronged Strategy: Lightweight Augmented Graph Network Hashing for Scalable Image Retrieval. *MM '21: Proceedings of the 29th ACM International Conference on Multimedia* (Aug. 2021). https://doi.org/10.1145/3474085.3475605

[7] Ben Derrick, Deirdre Toher, and Paul White. 2017. How to compare the means of two samples that include paired observations and independent observations: A companion to Derrick, Russ, Toher and White (2017). *The Quantitative Methods for Psychology* 13, 2 (April 2017), 120–126. https://doi.org/10.20982/tqmp.13.2.p120

[8] Guiguang Ding, Yuchen Guo, and Jile Zhou. 2014. Collective matrix factorization hashing for multimodal data. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2014), 2075–2082. https://doi.org/10.1109/CVPR.2014.267

[9] Tim Erven, Peter Harremos, and John Shawe-Taylor. 2014. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory* 60, 7 (July 2014), 3797 – 3820. https://doi.org/10.1109/TIT.2014.2320500

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), 770–778.

[11] Qinghao Hu, Jiaxiang Wu, Jian Cheng, Lifang Wu, and Hanqing Lu. 2017. Pseudo label based unsupervised deep discriminative hashing for image retrieval. *ACM International Conference on Multimedia* 28, 7 (July 2017), 1584–1590. https://doi.org/10.1109/TIP.2019.2897944

[12] Mark J. Huiskes and Michael Lew. 2008. The MIR flickr retrieval evaluation. *Proceedings of the 1st ACM international conference on Multimedia information retrieval* (Oct. 2008), 39–43. https://doi.org/10.1145/1460096.1460104

[13] Amin Khatami, Morteza Babaie, Abbas Khosravi, H.R. Tizhoosh, and Saeid Nahavandi. 2017. Parallel deep solutions for image retrieval from imbalanced medical imaging archives. *Applied Soft Computing* 63 (Feb. 2017), 197–205. https://doi.org/10.1016/j.asoc.2017.11.024

[14] Margarita Khokhlova, Valérie Gouet-Brunet, Nathalie Abadie, and Liming Chen. 2020. Cross-Year Multi-Modal Image Retrieval Using Siamese Networks. *IEEE International Conference on Image Processing (ICIP)* (Oct. 2020), 15. https://doi.org/10.1109/ICIP40778.2020.9190662

[15] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. 2018. Self-Supervised Adversarial Hashing Networks for Cross-Modal Retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018), 4242–4251. https://doi.org/10.1109/CVPR.2018.00446

[16] Peng Li, Lirong Han, Xuanwen Tao, Xiaoyu Zhang, Christos Grecos, Antonio Plaza, and Peng Ren. 2020. Hashing Nets for Hashing: A Quantized Deep Learning to Hash Framework for Remote Sensing Image Retrieval. *IEEE Transactions on Geoscience and Remote Sensing* 58, 10 (Oct. 2020), 7331–7345. https://doi.org/10.1109/TGRS.2020.2981997

[17] Peng Li, Xiaoyu Zhang, Xiaobin Zhu, and Peng Ren. 2018. Online Hashing for Scalable Remote Sensing Image Retrieval. *Learning to Understand Remote Sensing Images* 10, 5 (May 2018). https://doi.org/10.3390/rs10050709

[18] Yansheng Li, Yongjun Zhang, Xin Huang, and Jiayi Ma. 2018. Learning Source-Invariant Deep Hashing Convolutional Neural Networks for Cross-Source Remote Sensing Image Retrieval. *IEEE Transactions on Geoscience and Remote Sensing* 56, 11 (June 2018), 6521–6536. https://doi.org/10.1109/TGRS.2018.2839705

[19] Zijia Lin, Guiguang Ding, Jungong Han, and Jianmin Wang. 2016. Cross-View Retrieval via Probability-Based Semantics-Preserving Hashing. *IEEE Transactions on Cybernetics* 47, 12 (Sept. 2016). https://doi.org/10.1109/TCYB.2016.2608906

[20] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. 2015. Semantics-Preserving Hashing for Cross-View Retrieval. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), 3864–3872. https://doi.org/10.1109/CVPR.2015.7299011

[21] Xin Liu, Zhikai Hu, Haibin Ling, and Yiu-ming Cheung. 2019. MTFH: A matrix tri-factorization hashing framework for efficient cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 3 (Sept. 2019), 964–981. https://doi.org/10.1109/TPAMI.2019.2940446

[22] Yishu Liu, Zhengzhuo Han, Conghui Chen, Liwang Ding, and Yingbin Liu. 2020. Eagle-Eyed Multitask CNNs for Aerial Image Retrieval and Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing* 58, 9 (March 2020), 6699–6721. https://doi.org/10.1109/TGRS.2020.2979011

[23] Devraj Mandal, Kunal N. Chaudhury, and Soma Biswas. 2018. Generalizes Samantic Preserving Hashing for N-Label Cross-Modal Retrieval. *IEEE Transactions on Image Processing* 28, 1 (Jan. 2018), 102 – 112. https://doi.org/10.1109/TIP.2018.2863040

[24] Christos H. Papadimitriou. 1981. On the complexity of integer programming. *J. ACM* 28, 4 (Oct. 1981). https://doi.org/10.1145/322276.322287

[25] F. Pedregosa et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* (2011), 2825–2830. https://hal.inria.fr/hal-00650905v2

[26] O. C. S. Ribeiro Pedro, Santos Matheus M., Paulo L. J. Drews, S. C. Botelho, Silvia, M. Longaray, Lucas, Giovanni G. Giacomo, and Marcelo R. Pias. 2018. Underwater Place Recognition in Unknown Environments with Triplet Based Acoustic Image Retrieval. *IEEE International Conference on Machine Learning and Applications (ICMLA)* (Dec. 2018). https://doi.org/10.1109/ICMLA.2018.00084

[27] Maria Pegia, Anastasia Moumtzidou, Ilias Gialampoukidis, Jónsson, Björn Þór, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2022. BiasHash: A Bayesian Hashing Framework for Image Retrieval. *IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP 2022)* (June 2022). https://doi.org/10.1109/IVMSP54334.2022.9816233

[28] Jose Costa Pereira, Nikhil Rasiwasia, Roger Levy, and Nuno Vasconcelos. 2014. On the role of correlation and abstraction in cross-modal multimedia retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (March 2014), 521–535. https://doi.org/10.1109/TPAMI.2013.142

[29] Adnan Qayyum, Syed Muhammad Anwar, Muhammad Awais, and Muhammad Majid. 2016. Medical image retrieval using deep convolutional neural network. *Neurocomputing* 266 (Nov. 2016), 8–20. https://doi.org/10.1016/j.neucom.2017.05.025

[30] Sebastian Ruder. 2017. *An overview of gradient descent optimization algorithms.* Retrieved February 24, 2021 from https://arxiv.org/abs/1609.04747

[31] Patricia Schöntag, David Nakath, Stefan Röhrl, and Kevin Köser. 2022. Towards Cross Domain Transfer Learning for Underwater Correspondence Search. *Image Analysis and Processing – ICIAP 2022* 13233 (May 2022), 461–472. https://doi.org/10.1007/978-3-031-06433-3_39

[32] Michael E. Tipping. 2001. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research* 1, 3 (Jan. 2001), 211–244. https://doi.org/10.1162/15324430152748236

[33] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 85 (Aug. 2008), 2579–2605.

[34] Leon A. Varga, Benjamin Kiefer, Martin Messmer, and Andreas Zell. 2022. Seadronessee: A maritime benchmark for detecting humans in open water. *IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), 2260–2270. https://doi.org/10.48550/arXiv.2105.01922

[35] Yongxin Wang, Zhen-Duo Chen, Luo Xin, Rui Li, and Xin-Shun Xu. 2021. Fast Cross-Modal Hashing With Global and Local Similarity Embedding. *IEEE Transactions on Cybernetics* (2021). https://doi.org/10.1109/TCYB.2021.3059886

[36] Yanzhao Xie, Yu Liu, Yangtao Wang, Lianli Gao, Peng Wang, and Ke Zhou. 2020. Label-attended hashing for multi-label image retrieval. *IEEE Transactions on Cybernetics* (2020). https://doi.org/10.1109/TCYB.2021.3059886

[37] Yi Zhen, Ye Gao, Dit Y. Yeung, Hongyuan Zha, and Xuelong Li. 2016. Spectral Multimodal Hashing and Its Application to Multimedia Retrieval. *IEEE Transactions on Cybernetics* 46, 1 (Jan. 2016), 27–38. https://doi.org/10.1109/TCYB.2015.2392052

[38] Jile Zhou, Guiguang Ding, and Yuchen Guo. 2014. Latent semantic sparse hashing for cross-modal similarity search. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (July 2014), 415–424. https://doi.org/10.1145/2600428.2609610