# Topic driven multimodal similarity learning with multi-view voted convolutional features

**DOI:**
10.1016/j.patcog.2017.02.035

**Citation for published version (APA):**
Gao, X., Mu, T., Goulermas, J. Y., & Wang, M. (2018). Topic driven multimodal similarity learning with multi-view voted convolutional features. *Pattern Recognition*. https://doi.org/10.1016/j.patcog.2017.02.035

**Published in:**
Pattern Recognition

# Topic Driven Multimodal Similarity Learning with Multi-view Voted Convolutional Features

Xinjian Gao[1], Tingting Mu[2], John Y. Goulermas[3], Meng Wang[1]

## Abstract

Similarity (and distance metric) learning plays a very important role in many artificial intelligence tasks aiming at quantizing the relevance between objects. We address the challenge of learning complex relation patterns from data objects exhibiting heterogeneous properties, and develop an effective multi-view multimodal similarity learning model with much improved learning performance and model interpretability. The proposed method firstly computes multi-view convolutional features to achieve improved object representation, then analyzes the similarities between objects by operating over multiple hidden relation types (modalities), and finally fine-tunes the entire model variables via back-propagating a ranking loss to the convolutional layers. We develop a topic-driven initialization scheme, so that each learned relation type can be interpreted as a representative of semantic topics of the objects. To improve model interpretability and generalization, sparsity is imposed over these hidden relations. The proposed method is evaluated by solving the image retrieval task using challenging image datasets, and is compared with seven state-of-the-art algorithms in the field. Experimental results demonstrate significant performance improvement of the proposed method over the competing ones.

*Keywords:* Convolutional auto-encoder, representation learning, multi-view learning, multimodal similarity learning.

[1] X. Gao and Prof. M. Wang are with the School of Computer and Information, Hefei University of Technology, China.

[2] Dr. T. Mu is with the School of Computer Science, University of Manchester, Manchester, M1 7DN, United Kingdom.

[3] Dr. J. Y. Goulermas is with the Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, United Kingdom.

## 1. Introduction

Learning robust measures that can accurately characterize and quantize the relevance between objects plays an important role in many artificial intelligence tasks [1, 2]. For instance, automatic image annotation, retrieval and classification [3, 4, 5], intelligent recommendation systems [6], knowledge graph completion [7], image sentence mapping [8], etc. Conventional similarity (or distance metric) learning approaches usually quantize the relevance between objects via a distance measure, such as Mahalanobis distance [9] or a kernel function [10]. Such measures are usually parameterized on a set of variables, such as the covariance matrix or various kernel parameters. However, the expressive power of such approaches can be insufficient when processing complex relations and data patterns, and therefore, it is necessary to seek alternative strategies for constructing more robust similarity models.

In addition to improving the similarity learning model, it is also important to improve the quality of the input information that is fed into the model. In many real-world applications, raw data is usually collected from domains or sensors possibly including redundant information and noise. Therefore, representation learning becomes an essential stage with the benefit of eliminating redundancy, denoising, and reducing data dimensions. Typical example algorithms for representation learning, include deep neural networks, such as auto-encoders (AE) [11], convolutional neural networks (CNN) [12, 13], and their mixture convolutional auto-encoders (CAE) [14, 15].

With the goals of designing a more robust similarity model and improving the quality of the raw data information, we develop a multi-layered algorithm that sequentially achieves representation learning and similarity learning. A CNN is used as the basic template to decompose the objects of interests into local patterns. Heterogeneous neighboring structures between objects characterized by multiple feature views are globally preserved over multiple modalities. The whole algorithm design aims to cope more effectively with complex data patterns and improve learning performance.

From the input point of view, the analyzed objects can exhibit heterogeneous properties when dealing with a complex problem. Therefore, it is beneficial to characterize the objects from multiple perceptions, corresponding to multi-view feature representations [16]. A simple example in image retrieval is as follows. Given the query of a bear, in order to retrieve pictures including both brown and polar bears, shape view is preferred over the color view, whilst to retrieve bears in different poses, colour information becomes more important than shape. Success of capturing complementary information across different views has the potential of improving the learning performance [17]. Therefore, we incorporate a multi-view mechanism in the training of the CNN in order to improve representation learning.

From the model point of view, objects are not necessarily connected under a fixed type of relation examined with a single distance/kernel function. There exist multiple types of high-level relations based on image appearance and semantics. Again, we use an image retrieval task as the example. Given the query image of an apple, humans are able to infer multiple types of relations, such as the apple company, the apple logo, apple juice or different colors or shapes of apple fruits. This thus, motivates the machine to use multimodal similarity measures to model different high-level relation types. Example algorithms of such types, include transfer distance metric learning [18] and multiple kernel similarity learning [19] that utilize different kernel functions or base distance metrics to represent different relations. However, intermediate results of these algorithms, such as the meaning of the learned relation types and their controlling parameters, can be difficult to interpret. To improve the interpretability of the similarity learning model (i.e., avoiding treating it along with the learned parameters as a black box), and meanwhile maintain robust model expressive power, we propose a topic-driven multimodal similarity learning algorithm. This firstly employs a clustering algorithm to explore hidden data topics and subsequently encodes the resulting topics as different relation types to construct multimodal similarities.

To summarize, the goal of this work is to design a powerful learning system with layered architecture that possesses both representation and similarity learning functions. The system output along with the learned parameters offer: (1) low-dimensional data representation that manages reduced redundancy and noise and is refined by considering complementary information from multiple low-level feature views, and (2) multimodal similarities with each modality closely related to a latent data topic. Specifically, the input layer corresponds to the raw representation of the object pair $(\text{obj}_i, \text{obj}_j)$, e.g., the image pixels or low-level features. The hidden layers are divided into two components. The first component achieves representation learning, and is constructed by taking advantage of the CAE and the multi-view local voting [17] techniques. The second component translates a set of hidden relations $\{\mathrm{rel}_t\}_{t=1}^{c}$ into a set of hidden neurons. The numerical operation defined over each neuron quantizes the confidence level of whether the corresponding hidden relation exists between the two objects. It is controlled by both the relation embedding that interactively characterizes the property of the corresponding hidden relation, and the projection vector that has the potential to increase expressive power. The relation embeddings $\{\mathrm{rel}_t\}_{t=1}^{c}$ are initialized by cluster centers, (e.g., obtained by simple k-means clustering), so that inherent topic structure within the data can be encoded and drive the multimodal similarity learning. The output of the system exhibits an accumulation of the validities of all the hidden relations. To improve the model regularization and interpretability, it is reasonable to assume that only a few of the learned hidden relation types contribute to an existing relationship between objects, instead of all the types. Therefore, sparsity is enforced in the hidden relation layer. Finally, to boost the model performance, a fine-tuning procedure of the whole system is conducted to propagate the changes of the similarity model backward to the CNN network.

Overall, the preservation of the multi-view data structure and the construction of the multiple hidden relation operations allow the proposed system to be able to examine the relationships between objects under multiple feature views and also enable the discovery of diversified relations with multimodality. The remaining paper is organized as follows. Section 2 briefly introduces related previous work. Section 3 describes in detail the proposed algorithm. Experimental results and comparative analyses are included in Section 4, while Section 5 concludes the presentation.

## 2. Related Works

### *2.1. Representation Learning*

Deep learning techniques have been successful in learning numerical representations to characterize objects. AEs are a commonly used deep learning architecture for unsupervised representation learning [20]. Their advantages have been shown in various existing works [14, 15]. A conventional AE usually focuses on learning the global data distribution, but ignores the useful local structure information and therefore, it may not be suitable to process data with highly structured local information, such as imagery. Therefore, CAEs [21] take advantage of the CNN mechanism and capture better local data structure. In a CAE, the encoder and decoder layers of the AE are replaced with convolutional layers. The learned representations offer more balanced characterization between the global and local structures. Alternatively, supervised representation learning is usually capable of offering better performance than the unsupervised one, e.g., by training a CNN to solve an image classification task [22], supported by a sufficient amount of labeled training instances.

Here, we review some successful applications of deep representation learning. In speech emotion recognition, CNNs are applied to learn local invariant features [23]. Different scales of kernels are learned, with which the entire spectrogram fragment is convolved to form a series of feature maps. In synthetic aperture radar image classification, a CAE is applied to generate high-level features, using convolutional kernels initialized by Gabor filters [24]. Another example is to learn the 3D shape feature descriptor using a neural network [25]. The Fisher criterion is used to train the hidden layers, so that the learned features can be discriminative and insensitive to geometric structure variations. Recently, CNNs have been successfully used to analyze complex events in untrimmed videos [26]. After characterizing video keyframes with a CNN descriptor trained with labeled examples, a novel prioritization procedure and algorithm are developed to correspondingly order the keyframes and exploit the obtained order information for event analysis.

Instead of following the classical CNN training through either an unsupervised or a supervised scheme as in the previous works [21, 26], we develop an effective stage-wise CNN training strategy utilizing a mixture of unsupervised and supervised schemes to solve better the targeted similarity learning task. Built upon the CAE network, we keep improving the pre-trained CNN by AE, via following another unsupervised training scheme that considers a multi-view learning strategy. To boost the model performance, a fine-tuning of the model variables is conducted to improve the learned representation, and the CNN that is initialized by its unsupervised training output, is connected to another multimodal similarity model.

### 2.2. Multi-view Learning

The development of contemporary sensor and computer modelling techniques has enabled data information, that may characterize heterogeneous properties of the studied objects, to be collected from various domains, feature collectors and extractors. These are often referred to as multi-view representations of the objects and lead to multi-view learning tasks in machine learning. This facilitates complex data analysis in a multitude of areas, such as video surveillance, multimedia, and image classification. Each representation (view) has different physical meaning and concrete statistical properties. It has the potential to improve the performance of a given task by enabling effective collaboration between multiple views; for instance, a view can be enhanced by its complementary views.

We review some recent advances on multi-view feature representation learning. In image classification, multiple discriminative dictionaries are jointly learned with redundancy among dictionaries from different reduced views [27]. Aiming at reconstructing incomplete views through multi-view learning, [28] assumes that all the views are generated by a shared subspace and proposes to estimate the incomplete views by learning the shared subspace from the complete views. Another example, is a unified multi-view approach for robust classifier training [29], which utilizes different types of weakly labeled multi-view data collected from a broad range of relevant tasks. To simultaneously reduce data dimensionality and infer label information, [30] solves a multi-view multi-label learning task. For each view, a weakly labeled learning problem is modelled, and an optimal classifier is learned from a set of pseudo-label vectors generated by using the classifiers trained from the other views. Recently, a multi-view approached is developed to improve semantic annotation of video street view, where the 2D and 3D features are combined to reduce the amount of used training data and introduce computational efficiency [31].

Multi-view approaches have been shown to be effective in the improvement of unsupervised and semi-supervised learning. For instance, unsupervised alignment hashing is developed based on regularized kernel matrix factorization in [32]. It seeks a compact representation to uncover hidden semantics, preserve joint probability data distribution, and learn low-dimensional embeddings by multi-view weight learning. Another unsupervised example, is [33] which adaptively selects multi-view embeddings or individual features to improve clustering. Working with a small amount of labeled data, [34] develops a semi-supervised multi-view feature fusion method. This preserves the manifold structure for each feature type during the training phase and relies on a statistical approach to exploit the manifold structure elements. Another semi-supervised example, is [35] which utilizes multimedia information, such as visual features and text features to enhance image classification.

In the presented work, we attempt to improve CNN-based representation learning without seeking support from label information as it is not always available in real-world applications. An effective strategy is to take advantage of the multi-view paradigm. Different from most existing works that utilize the multi-view information in the input layer, we take advantage of such information in the training phase of the CNN. The multiple feature views are encoded as a composite neighbor structure, that is used to formulate the objective function of the optimization. One advantage of embedding the multi-view mechanism within the CNN training phase, is that it can be conveniently integrated with existing CNN training schemes (e.g., unsupervised training by an AE and supervised training using label information) through pre-training and fine-tuning to achieve improved system regularization and robustness.

### *2.3. Multimodal Similarity Learning*

Multimodal similarity (or distance metric) learning constitute a type of algorithms that can measure similarity or distance between objects from different aspects. Typically, different modalities can be modelled as different kernel functions or distance metrics. Such techniques have been shown effective in many real-world applications. For instance, in image ranking, different modalities of the image connections can be modelled by different kernel functions, and the overall similarity function can be formulated as a weighted sum of these kernel functions [19]. In person re-identification from camera networks, different cameras are usually affected by different types of noise. Therefore, the work [36] designs multiple Mahalanobis distance metrics to cope with the multiple noise modalities, and these metrics are related and handled for overfitting by enforcing joint regularization. In image retrieval, the multimodality mechanism is used to realize feature fusion, where different distance metrics are optimized in different feature spaces to find the optimal combination of diverse feature types [37]. Multimodality is also considered in transfer learning to overcome the lack of information in a target task [18]. Given multiple related source tasks, objects can be connected in multiple ways, and these connections are modelled as multiple base metrics, whereas the overall metric as their weighted sum.

In the recent years, multimodal design has been adopted in deep learning. For instance, [38] employs different neural networks to model different input modalities that achieve multimodal gesture segmentation and recognition, where the input modalities include the observed skeleton joint information, depth and RGB images. Another example, is multimodal recurrent neural network developed to improve image captioning [39]. This work introduces a multimodal layer to compute the activation from multiple input modalities, such as word embeddings and image features. For image retrieval, [40] develops a multimodal similarity learning model, which encodes the different similarity modalities with different nonlinear transformations of the input features modelled by different neural networks. So far, most existing multimodal learning algorithms are designed as a black box, of which the intermediate results, (e.g., the learned parameters for controlling the modalities and hidden layer output) are usually hard to interpret. In this work, one of our objectives is to develop a model with much improved interpretability, so that its intermediate results can be conveniently observed and analyzed by the user, while maintaining the expressive power of the model.

## 3. Proposed Method

The proposed similarity learning system consists of two components. The first corresponds to the unsupervised feature representation learning. The improved representation is used as the input to the second component of supervised multimodal similarity learning. The two components are first trained separately, and then the whole system is fine-tuned by back-propagating a supervised ranking loss. The overall system architecture is illustrated in Figure 1.

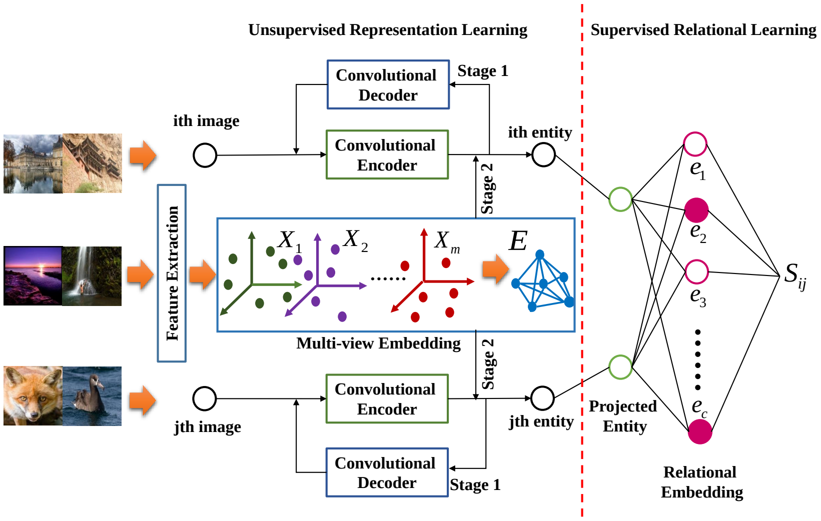

Figure 1: System architecture of the proposed similarity learning algorithm.

### 3.1. *Unsupervised Feature Representation Learning*

Given a collection of $n$ objects $\{\mathbf{x}_i\}_{i=1}^n$, each is characterized by its raw features, that is, the image pixels. We employ a CNN network to compute the high-level representation from these raw features. For each object, we assume that its input features are arranged into a $l \times l$ 2D matrix, also denoted by $\mathbf{x}_i$. To take advantage of the local distribution and also to reduce the amount of parameters to be learned, we compute the output representation by convolutional means [15]. Convolving a $k_c \times k_c$ kernel $\boldsymbol{E}$ with the input matrix, the dimension of the output convolutional feature map $\boldsymbol{h}$ is $(l - k_c + 1) \times (l - k_c + 1)$. Assuming that $m$ kernels are used in the convolutional layer and letting the subscript $j$ index the $j$-th map, the output averaged over different maps can be expressed as

$$\boldsymbol{h}_i = \frac{1}{m}\sum_{j=1}^{m} \text{sigmoid}(\text{convn}(\boldsymbol{x}_i, \boldsymbol{E}_j) + \boldsymbol{b}_j), \tag{1}$$

where convn$(\cdot,\cdot)$ denotes the convolutional operation, and the matrix $\boldsymbol{b}_j$ of size $(l - k_c + 1) \times (l - k_c + 1)$ denotes the bias parameter. Training of the network relies on an appropriate choice of the objective function to optimize the network weights (e.g., $\{\boldsymbol{E}_j, \boldsymbol{b}_j\}_{j=1}^{m}$). Here, we provide the equation for one convolutional layer, and more layers can be added accordingly.

### *3.1.1. Training by Multi-view Voted Neighbor Preservation*

The deep embedding method [41] is commonly used to pre-train a neural network by uncovering a similarity structure between objects computed using the original features $\mathbf{x}_i$ to improve the learning performance and prevent the algorithm from getting stuck in local minima or plateaus [42]. The method pushes the intermediate representations of the objects, which are learned at the hidden layer of a neural network, to move towards each other or apart from each other depending on whether the two objects are neighbors or not. Motivated by these, we employ a similar approach to learn the high-level feature representation $\boldsymbol{h}_i$. However, differently from the traditional task, we attempt to generate a representation capable of taking into account heterogeneous properties of the objects under different view perceptions. Thus, we propose to construct a composite similarity structure by seeking augmented neighborhood relations through the examination under different view representations. Network weights are trained to generate high-level features that can preserve optimally such a similarity structure.

To construct different views, different feature extraction methods or different information resources can be used. For instance, given an image retrieval task, the low-level feature extraction methods, such as image color histogram, color correlogram, edge direction histogram, wavelet texture, block-wise color moments, bag of words based on the scale-invariant feature transform (SIFT) descriptions, etc., can characterize different properties of the images. We use $\{\mathbf{X}_s\}_{s=1}^{a}$ to denote the different view matrices, whose rows correspond to the different feature vectors of the objects. We then minimize the following objective function based on a penalized distance error sum

$$\min_{\{\boldsymbol{E}_j, \boldsymbol{b}_j\}_{j=1}^{m}} r_{ij} \left( \left\{ \boldsymbol{x}_i^{(s)} \right\}_{s=1}^{a}, \left\{ \boldsymbol{x}_j^{(s)} \right\}_{s=1}^{a} \right) \|\boldsymbol{\Phi}_i(\boldsymbol{h}_i) - \boldsymbol{\Phi}_j(\boldsymbol{h}_j)\|_2^2 \,. \tag{2}$$

For each object, $\boldsymbol{\Phi}_i$ is the vector version of the feature matrix $\boldsymbol{h}_i$, which is generated by simply arranging the elements of $\boldsymbol{h}_i$ in a single row vector. The penalty weight $r_{ij}$

quantizes the similarity/neighborhood information between two objects. It is computed by taking into account all the feature representations offered by the $a$ views through a confidence-driven voting scheme developed in our previous work [17][4], and is a function of $\{\boldsymbol{x}_i^{(s)}\}_{s=1}^{a}$ and $\{\boldsymbol{x}_j^{(s)}\}_{s=1}^{a}$. The voting scheme basically assumes that, when there are more views agreeing on the existence of the neighborhood relation between two objects, this object pair is considered to be more reliable and thus, it is awarded a higher weight. Specifically, the penalty weight $r_{ij}$ can be computed by

$$r_{ij} = \sum_{\alpha=1}^{m} \frac{\alpha}{m} r_{ij}^{(\alpha)}, \tag{3}$$

with

$$r_{ij}^{(\alpha)} = \begin{cases} 0, & \text{if the } I_{ij}^{\alpha} = \emptyset, \\ \frac{1}{\alpha}\sum_{s \in I_{ij}^{\alpha}} \mathbf{P}_r^{(s)}, & \text{otherwise,} \end{cases} \tag{4}$$

where $\alpha$ represents the number of the views that are expected to agree with each other. The set $I_{ij}^{(\alpha)}$ records the indices of the $\alpha$ views agreeing that the $i$-th object and the $j$-th object are neighbors through the distance comparison using the features of these corresponding views. The matrix $\mathbf{P}_r^{(s)}$ stores local proximity information for the $s$-th view. Its nonzero element stores the similarity value between two neighboring objects computed under the $s$-th feature view, whereas, its zero elements indicate not neighboring object pairs. By minimizing Eq. (2), the network weights are able to offer high-level features that preserve a composite neighborhood structure computed from multiple views.

### *3.1.2. Auto-encoder based Pre-training*

The above training procedure encodes an augmented neighborhood structure combining multiple views within the computed high-level feature representation. Instead of random initialization in training, we start from a pre-trained solution that captures the global data distribution, aiming at improving the balance between the local and global

[4] An alternative weighting scheme used in [43, 44], is to set the weights to the retrieval performance (e.g., the precision score), obtained using each corresponding feature view. This approach is supervised, requiring knowledge on the label information of the training objects. Given that our goal in this stage is to perform an unsupervised training of the CNN, we adopt the unsupervised weighting scheme in [17].

patterns and facilitate generalization. This can be achieved by implementing an AE-based pre-training procedure. A decoding layer is added consisting of $m$ convolutional kernels, which takes the output of the CNN as its input. This is given as

$$\tilde{\boldsymbol{x}}_i = \frac{1}{m}\sum_{j=1}^{m} \text{sigmoid}(\text{convn}(\boldsymbol{h}_i, \boldsymbol{D}_j) + \boldsymbol{c}_j), \tag{5}$$

where the $l \times l$ matrix $\tilde{\boldsymbol{x}}_i$ represents the decoded representation of the $i$-th object, the $k_c \times k_c$ matrix $\boldsymbol{D}_j$ and the $l \times l$ matrix $\boldsymbol{c}_j$ denote the decoder convolutional kernel and bias, respectively, for the $j$-th map. By treating the CNN network described in Eq. (1), as an encoder that is connected to the above decoder, the network parameters $\{\boldsymbol{E}_j, \boldsymbol{b}_j\}_{j=1}^m$ and $\{\boldsymbol{D}_j, \boldsymbol{c}_j\}_{j=1}^m$ can be optimized together by minimizing a reconstruction error, such as

$$\min_{\{\boldsymbol{E}_j, \boldsymbol{b}_j, \boldsymbol{D}_j, \boldsymbol{c}_j\}_{j=1}^m} \sum_i \| \tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i \|_F^2 \ . \tag{6}$$

This corresponds to the stage 1 training of the representation learning component shown in Figure 1. The obtained solution of $\{\boldsymbol{E}_j, \boldsymbol{b}_j\}_{j=1}^m$ is then used as an initialization, and then a further training of the weights $\{\boldsymbol{E}_j, \boldsymbol{b}_j\}_{j=1}^m$ but based on the different objective function of Eq. (2) is conducted. This corresponds to the stage 2 training of the representation learning shown in Figure 1. This two-stage training reflects a learning procedure that shifts its focus from the preservation of the global pattern to local. The entire representation learning procedure can also be viewed as a CAE network with its hidden layer training enhanced by a weighted distance error minimization in order to better reflect a multi-view voted local neighbor structure.

### *3.2. Supervised Multimodal Similarity Learning*

Operating on the high-level feature representation $\{\boldsymbol{\Phi}_i\}_{i=1}^n$ of the objects, the subsequent step is to build a score function $s_{ij} = f(\boldsymbol{\Phi}_i, \boldsymbol{\Phi}_j)$ that measures the similarity between two objects. A major motivation of designing this is to reflect the validities of multiple hidden relations. This is important, for instance, for an image retrieval task because the searched images can be related to the query image under different relation types (examples are mentioned in Section 1). In order to model such multimodal similarity, we introduce a set of hidden neurons, each representing one hidden relationship between objects.

We consider a total of $c$ hidden relation neurons in the model. Unlike conventional similarity learning algorithms that lack interpretability, we attempt to relate different hidden neurons with different clusters of the training objects, so that the resulting relation modalities can correspond to hidden topics amongst the objects. To achieve this, one convenient way is to parameterize each neuron with a row embedding vector $\boldsymbol{e}_t$ ($t = 1, 2, \dots c$) in the same space as the high-level convolutional features $\boldsymbol{\Phi}_i$. This allows the building of links between neurons and object clusters through a straightforward initialization scheme using the cluster center vector (explained at the end of this subsection). The sought task is to examine whether the objects $i$ and $j$ are linked via a hidden relation type $t \in \{1, 2, \dots, c\}$. A score function can be designed for a given relation triplet $(\mathrm{obj}_i, \mathrm{rel}_t, \mathrm{obj}_j)$, so that a high score indicates the existence of $\mathrm{rel}_t$ between $\mathrm{obj}_i$ and $\mathrm{obj}_j$, whereas a low score otherwise. Inspired by the success of a bilinear similarity score used in deep text matching in answer selection [45], we adopt a rank-1 formulation $\boldsymbol{e}_t^T \boldsymbol{e}_t$ to parameterize the bilinear similarity, such that

$$s_{ij}^{(t)} = \boldsymbol{\Phi}_i \left( \boldsymbol{e}_t^T \boldsymbol{e}_t \right) \boldsymbol{\Phi}_j^T = \left( \boldsymbol{\Phi}_i \boldsymbol{e}_t^T \right) \left( \boldsymbol{\Phi}_j \boldsymbol{e}_t^T \right). \tag{7}$$

This formulation first evaluates separately the similarities between the head object and the relation and also between the tail object and the relation via the inner product function. Finally, it derives a composite quantity based on the multiplication of the two similarity values.

To improve the expressive power of the model, when computing the similarity induced by inner product, we allow a deviation from the targeted relation embedding by introducing a row projection vector $\boldsymbol{p}_t$ over each hidden neuron. It is embedded in the same space as the relation embedding vector $\boldsymbol{e}_t$ and the convolutional feature vector $\boldsymbol{\Phi}_i$. The similarity between the head object and the targeted relation embedding can be formulated as the inner product between the projected object $\boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t$ and the relation embedding $\boldsymbol{e}_t$. Therefore, the modified similarity formulation becomes

$$s_{ij}^{(t)} = \left( \boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T \right) \left( \boldsymbol{\Phi}_j \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T \right) = \boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T \boldsymbol{e}_t \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{\Phi}_j^T. \tag{8}$$

By comparing Eqs. (7) and (8), we can see that both formulations represent a bilinear operator between two vectors. The first is parameterized over a rank-1 positive semidefinite (PSD) matrix $\mathbf{E}_t = \boldsymbol{e}_t^T \boldsymbol{e}_t$, while the second is parameterized over a matrix that can be factorized as two rank-1 PSD matrices in the form of $\mathbf{P}_t \mathbf{E}_t \mathbf{P}_t$ with $\mathbf{P}_t = \boldsymbol{p}_t^T \boldsymbol{p}_t$.

Finally, a composite similarity is computed by combining these hidden similarities as

$$s_{ij} = \ln \left( 1 + \exp \left( \sum_{t=1}^{c} s_{ij}^{(t)} \right) \right). \tag{9}$$

In the above, the smoothed version of the rectifier activation function, know as the softplus function, is used to enforce sparsity over the hidden similarities $\{s_{ij}^{(t)}\}_{t=1}^{c}$. The reason sparsity is enforced over the learned similarities, is that it is more reasonable to have only several hidden relation types contribute to the final similarity instead of all the types, which in a way has the potential to obtain improved regularization [46].

To optimize the relation embeddings and projection vectors $\{\boldsymbol{e}_t, \boldsymbol{p}_t\}_{t=1}^{c}$, we employ the following optimization problem, that minimizes a ranking loss computed from the stochastic margin error [47] as

$$L\big(\{\mathbf{e}_t\}_{t=1}^{c}, \{\mathbf{p}_t\}_{t=1}^{c}\big) = \sum_{(i,j+) \in I_+} \sum_{(i,j-) \in I_-} \max(s_{ij-} - s_{ij+} - 1, 0). \tag{10}$$

The index set $I_+$ contains the truly related object pairs in the training set, referred to as the positive training pairs, while $I_-$ the truly unrelated object pairs referred to as the negative training pairs. In general, $I_+$ can be set to be the collection of image pairs with each pair containing a query image and one of its correct images to be retrieved, while $I_-$ the collection with each pair including a query image and an image that should not be retrieved given this query. The loss function evaluates the difference between the similarity scores of the negative and positive pairs. The optimization drives the differences (margin errors) to be less than 1 to facilitate the discrimination between the positive and negative pairs. Stochastic gradient descent is employed for the optimization. In each update, the gradients are computed by a positive pair and a negative pair denoted by $(\text{obj}_i, \text{obj}_{j+})$ and $(\text{obj}_i, \text{obj}_{j-})$, respectively, where $(i, j+) \in I_+$ and $(i, j-) \in I_-$. Below, we include the corresponding gradient equations for updating the relation embeddings and projection vectors

$$\boldsymbol{p}_t^{(\text{new})} = \boldsymbol{p}_t - \eta \frac{\partial L}{\partial \boldsymbol{p}_t} = \boldsymbol{p}_t - \eta \left( \frac{\partial s_{ij-}}{\partial \boldsymbol{p}_t} - \frac{\partial s_{ij+}}{\partial \boldsymbol{p}_t} \right), \tag{11}$$

$$\boldsymbol{e}_t^{(\text{new})} = \boldsymbol{e}_t - \eta \frac{\partial L}{\partial \boldsymbol{e}_t} = \boldsymbol{e}_t - \eta \left( \frac{\partial s_{ij-}}{\partial \boldsymbol{e}_t} - \frac{\partial s_{ij+}}{\partial \boldsymbol{e}_t} \right), \tag{12}$$

where $\eta > 0$ controls the learning rate. We further have

$$\frac{\partial s_{ij-}}{\partial s_{ij}^{(t)}} = \text{sigmoid}\left[ (\boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T)(\boldsymbol{\Phi}_{j-} \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T) \right], \tag{13}$$

$$\frac{\partial s_{ij+}}{\partial s_{ij}^{(t)}} = \text{sigmoid}\left[ (\boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T)(\boldsymbol{\Phi}_{j+} \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T) \right], \tag{14}$$

$$\frac{\partial s_{ij-}}{\partial \boldsymbol{e}_t} = [(\boldsymbol{p}_t \boldsymbol{p}_t^T \boldsymbol{\Phi}_i^T)(\boldsymbol{\Phi}_{j-} \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T) + (\boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T)(\boldsymbol{p}_t \boldsymbol{p}_t^T \boldsymbol{\Phi}_{j-}^T)] \frac{\partial s_{ij-}}{\partial s_{ij}^{(t)}}, \tag{15}$$

$$\frac{\partial s_{ij+}}{\partial \boldsymbol{e}_t} = [(\boldsymbol{p}_t \boldsymbol{p}_t^T \boldsymbol{\Phi}_i^T)(\boldsymbol{\Phi}_{j+} \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T) + (\boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T)(\boldsymbol{p}_t \boldsymbol{p}_t^T \boldsymbol{\Phi}_{j+}^T)] \frac{\partial s_{ij+}}{\partial s_{ij}^{(t)}}, \tag{16}$$

$$\begin{aligned} \frac{\partial s_{ij-}}{\partial \boldsymbol{p}_t} = {} & [(\boldsymbol{p}_t^T \boldsymbol{e}_t \boldsymbol{\Phi}_i^T + \boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{e}_t^T)(\boldsymbol{\Phi}_{j-} \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T) + \\ & (\boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T)(\boldsymbol{p}_t^T \boldsymbol{e}_t \boldsymbol{\Phi}_{j-}^T + \boldsymbol{\Phi}_{j-} \boldsymbol{p}_t^T \boldsymbol{e}_t^T)] \frac{\partial s_{ij-}}{\partial s_{ij}^{(t)}}, \end{aligned} \tag{17}$$

$$\begin{aligned} \frac{\partial s_{ij+}}{\partial \boldsymbol{p}_t} = {} & [(\boldsymbol{p}_t^T \boldsymbol{e}_t \boldsymbol{\Phi}_i^T + \boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{e}_t^T)(\boldsymbol{\Phi}_{j+} \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T) + \\ & (\boldsymbol{\Phi}_i \boldsymbol{p}_t^T \boldsymbol{p}_t \boldsymbol{e}_t^T)(\boldsymbol{p}_t^T \boldsymbol{e}_t \boldsymbol{\Phi}_{j+}^T + \boldsymbol{\Phi}_{j+} \boldsymbol{p}_t^T \boldsymbol{e}_t^T)] \frac{\partial s_{ij+}}{\partial s_{ij}^{(t)}}. \end{aligned} \tag{18}$$

It is straightforward to extend the above update to batch training.

To offer interpretability to relation modalities, instead of optimizing the relation embeddings $\{\boldsymbol{e}_t\}_{t=1}^c$ with random initialization, we propose to initialize each embedding with the centre vector $\boldsymbol{c}_t$ of a cluster of the training objects. For an image retrieval task, these clusters can be manually defined according to the training data (e.g., the collection of a real animal, such as bear (or a similar toy, or logo) images for a bear query). This enables the opportunity to include all different types of images to the retrieved image set other than focusing on one type only. One can also derive the clusters by applying a clustering algorithm (e.g., k-means clustering) based on the high-level object representation $\boldsymbol{\Phi}_i$. After identifying $c$ clusters corresponding to a semantic topic structure within the observed data, these clusters (topics) are used to drive the formulation of the similarity measure.

### *3.3. Model Fine-tuning*

As previously described, the proposed model includes two components. The former is the unsupervised feature representation learning that generates high-level convolutional features by optimizing the CNN network weights. Its training includes the AE-based pre-training followed by the proposed multi-view training. The later is the multimodal similarity component, taking the learned convolutional features as the input. It learns a similarity function by optimizing the model parameters $\{\mathbf{e}_t, \mathbf{p}_t\}_{t=1}^{c}$ in a supervised manner. After training these two components separately, we connect them and seek a better solution by performing a supervised fine-tuning for the entire model. Specifically, the entire model is initialized with the separately trained CNN network weights and similarity model parameters. Then, all the model variables carry on being optimized via the minimization of the ranking loss in Eq. (10). This procedure propagates the changes of the sparse hidden similarities backward to the CNN model.

## 4. Experimental Results and Analysis

We conduct various experiments to verify and analyze the performance of the proposed algorithm through the image retrieval task, and compare it with seven existing algorithms including online multiple kernel similarity learning (OMKS) [19], the conventional approach of iterative quantization (ITQ) [48], mutliview alignment hashing (MAH) [32], deep regularized similarity comparison hashing (DRSCH) [49], deep semantic ranking hashing (DSRH) [50], kernel based supervised hashing (KSH-CNN) [51] and neighborhood discriminant hashing (NDH) [52]. Some of these methods focus on multi-view or deep representation learning, some on multimodal similarity learning, while some are specialized in image retrieval.

Two benchmark image datasets are used for evaluation. One is the CIFAR-10 [53], containing 60,000 colour images belonging to 10 object classes, such as airplane, truck, bird, cat, deer, horse, etc., with each class containing 6,000 images. The other is NUS-WIDE [54], which is a large collection of Flickr web images containing 269,648 images belonging to 81 concepts, such as garden, street, tower, dancing, tree, etc. For CIFAR-10, four different feature extraction methods are employed, including the 900-

D local binary pattern (LBP) [55], the 256-D colour histogram (CH) [56], the 324-D histogram of gradient (HoG) [57] and the 1024-D wavelet texture (WT) [58]. These are the different views to compute the neighbor weights through multi-view voting as explained in Section 3.1.1. To generate the LBP features, the input image is divided into $3 \times 3$ cells, discarding the remaining pixels. To generate the binary numbers for preparing the LBP features, the center pixel is compared with its 8 neighboring pixels in each cell. To generate the CH features, after transforming the RGB image into HSV image, 16 bins for the hue space, 4 bins for the saturation space and 4 bins for the value space are used; this leads to features of pixel counts in $16 \times 4 \times 4$ = 256 bins. To generate the HoG feature, the input image is divided into 8 cells. Gradients from 36 angles within 180 degrees are computed in each cell and 9 bins are set for each angle. This leads to features of gradient counts in $9 \times$ 36=324 bins. To generate the WT features, the $20 \times 20$ Gaussian filter is used on the input image. For NUS-WIDE, six groups of readily extracted features[5] are used as the different feature views, including 64-D CH,144-D color correlogram (CORR) [59], 73-D edge direction histogram (EDH) [60], 128-D WT, 225-D block-wise color moments (CM) [61] and the 500-D bag-of-word model based on SIFT descriptions [62]. In the experiments, for CIFAR-10, 1,000 images per class are randomly selected as query images, 1,000 as training images, while all the remaining ones as testing images. For NUS-WIDE, 2,000 randomly selected images are used as queries, 5,000 as training images and 260,000 as the testing ones to evaluate the retrieval performance.

For the proposed method, to compute the multi-view convolutional features, the $5 \times 5$ kernel size and $m = 5$ maps are used in both encoding and decoding layers. For the auto-encoder based pre-training, all the parameters are randomly initialized. To implement the topic-driven multimodal similarity learning, the $k$-means clustering is first applied to the training images to obtain a preview of the topic structure of the images; then, the resulting cluster centers are used to initialize the relation embeddings so that different types of hidden relations can potentially be linked to different image topics. Different numbers of hidden relation types $k \in \{5, 10, 15, 20\}$, corresponding

[5] These features can be downloaded from http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

to different numbers of hidden relation neurons, are tested. To optimize the network parameters, stochastic gradient descent based on batch training is applied, where the batch size is set to 50 and the learning rate to $\eta = 0.1$. For the competing methods, we use the same parameters as in their corresponding published works. The retrieval performance is assessed by the precision of the top 500 retrieved images (500AP) and the mean average precision (mAP).

### *4.1. Performance Comparison and Analysis*

The two main components of the proposed algorithm are referred to as multi-view convolutional (MVC) feature descriptor and multimodal similarity (MS). To quantitatively evaluate the effectiveness of the proposed design, we compare with various alternative design options. Firstly, we examine a baseline retrieval system referred to as CAE-E that employs the convolutional winner-take-all (CONV-WTA) AE [15] to generate the convolutional features, and computes the similarity score using Euclidean distances. We test this baseline system with different types of input, such as the raw image pixels, different types of features extracted by different methods, and the multi-view features that include all the extracted features in one concatenated vector. Performance of CAE-E with different types of input features is reported in Table 1 (right side of $\rightarrow$). We also compare with the original performance of these features without using CAE to de-noise the data (left side of $\rightarrow$). The improved performance (from the left to right) shows the effectiveness of CAE-based representation learning in this baseline system. It can also be seen from Table 1 (comparison between the last row and the previous ones) that the simple multi-view processing of combining different types of features in a single vector does not improve the retrieval performance and fails to take advantage of the multiple feature views.

To compare the superiority of the proposed MVC features over CAE, the same retrieval operation as used above, which is the Euclidean distance based similarity comparison, is applied using MVC features as the input. The resulting system is referred to as as MVC-E, and its performance is reported in the second row of Table 2. For comparison purposes, we also display the best performance achieved by CAE-E in the first row of of Table 2. It can be seen that the proposed MVC features prove to have

| **Effectiveness of CAE** | NUS-WIDE | NUS-WIDE | CIFAR-10 | CIFAR-10 |
|---|---|---|---|---|
| | 500AP (%) | mAP (%) | 500AP (%) | mAP (%) |
| Raw Pixels | 0.05 → 0.12 | 0.05 → 0.10 | 0.11→ 0.21 | 0.11 → 0.23 |
| CH Features | 0.12 → 0.25 | 0.13 → 0.24 | 0.11→ 0.15 | 0.13 → 0.16 |
| CORR Features | 0.14 → 0.22 | 0.14 → 0.23 | N/A | N/A |
| EDH Features | 0.18 → **0.30** | 0.19 → 0.23 | N/A | N/A |
| WT Features | 0.16 → 0.25 | 0.16 → 0.23 | 0.14 → 0.20 | 0.16 → 0.20 |
| CM Features | 0.16 → **0.30** | 0.15 → **0.29** | N/A | N/A |
| SIFT Features | 0.15 → 0.21 | 0.16 → 0.20 | N/A | N/A |
| LBP Features | N/A | N/A | 0.15 → **0.26** | 0.16 → **0.24** |
| HoG Features | N/A | N/A | 0.12 → 0.18 | 0.08 → 0.15 |
| All Features | 0.18 → 0.24 | 0.16 → 0.23 | 0.13 → 0.20 | 0.16 → 0.20 |

Table 1: Performance comparison of the baseline system using auto-encoder based representation learning.

significantly better retrieval performance than the CAE features.

We further examine the proposed system with the two components of MVC and MS trained separately without fine-tuning under different settings of $k$; we refer to this as MVC-MS. It can be seen from Table 2 that similarly good performance has been achieved using $k \geq 10$. Compared to MVC-E, MVC-MS replaces the simple Euclidean distance based comparison by the more sophisticated MS learning, and leads to significantly better performance.

Next, we examine the effectiveness of the sparsity design in MS and the unsupervised training of MVC features under the dimension setting of $k = 15$. Table 2 reports the performance of MVC-MS, but replacing Eq. (8) with the simpler setting of Eq. (7); this is referred to as MVC-MS-Eq.(7). The results show that Eq. (8) is a more effective score formulation than Eq. (7), offering thus a better retrieval performance. We also replace the proposed MVC features with the features extracted using the deep learning network Caffe, based on supervised training using the ImageNet data [22] and report its performance in Table 2; this is referred to as Caffe-MS. Although Caffe is trained in a supervised way, it is trained to solve a different task of image classification

| **Methods** | NUS-WIDE | NUS-WIDE | CIFAR-10 | CIFAR-10 |
|---|---|---|---|---|
| | 500AP (%) | mAP (%) | 500AP (%) | mAP (%) |
| CAE-E | 0.30 | 0.29 | 0.26 | 0.24 |
| MVC-E | 0.37 | 0.39 | 0.29 | 0.32 |
| MVC-MS ($k = 5$) | 0.65 | 0.66 | 0.67 | **0.65** |
| MVC-MS ($k = 10$) | 0.66 | 0.66 | 0.64 | **0.65** |
| MVC-MS ($k = 15$) | **0.67** | **0.67** | **0.68** | **0.65** |
| MVC-MS ($k = 20$) | **0.67** | 0.66 | 0.67 | **0.65** |
| MVC-MS-Eq.(7) ($k = 15$) | 0.65 | 0.63 | 0.61 | 0.62 |
| MVC-MS-sig ($k = 5$) | 0.61 | 0.62 | 0.55 | 0.56 |
| MVC-MS-sig ($k = 10$) | 0.62 | 0.61 | 0.56 | 0.56 |
| MVC-MS-sig ($k = 15$) | 0.63 | 0.62 | 0.56 | 0.58 |
| MVC-MS-sig ($k = 20$) | 0.62 | 0.61 | 0.59 | 0.60 |
| Caffe-MS ($k = 15$) | 0.62 | 0.64 | 0.63 | **0.65** |
| MVC-MS-fn ($k = 15$) | **0.68** | **0.68** | **0.70** | **0.67** |
| OMKS [19] | 0.60 | 0.62 | 0.58 | 0.55 |
| ITQ [48] | 0.28 | 0.28 | 0.22 | 0.25 |
| MAH [32] | 0.35 | 0.32 | 0.38 | 0.40 |
| DRSCH [49] | 0.63 | 0.64 | 0.65 | 0.63 |
| DSRH [50] | 0.62 | 0.63 | 0.64 | 0.63 |
| KSH-CNN [51] | 0.62 | 0.62 | 0.52 | 0.47 |
| NDH [52] | 0.30 | 0.32 | 0.26 | 0.32 |

Table 2: Comparison of the retrieval performance for different methods and datasets. The best and second best performance of the proposed method under different settings are highlighted in bold. The best performance of the existing methods is underlined!!.

other than image retrieval, and offers lower retrieval performance than the proposed MVC features as seen in Table 2. To demonstrate the effectiveness of the sparsity design over the hidden relation neurons, we evaluate the system by replacing the softplus activation function in the relational learning layer with the standard sigmoid activation function; this modified system is referred to as MVC-MS-sig. It can be seen from Table 2 that the softplus function (see MVC-MS performance) offers significantly better results than the sigmoid (see MVC-MS-sig performance). By setting the mean value of all the computed hidden similarities as the threshold to classify the active and in-

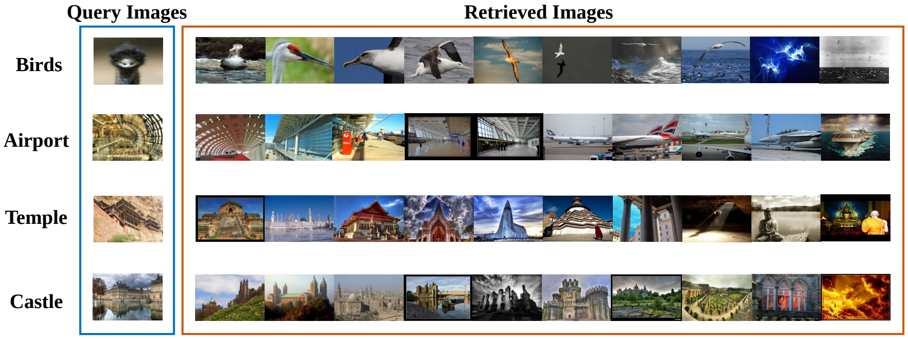

Figure 2: Example images retrieved by the proposed method are displayed in rows, given different NUS-WIDE query images belonging to different concepts.

active relation neurons and compute the percentage of the active neurons, we observe that softplus outputs $35\%$, $33\%$, $30\%$ and $30\%$ active relation neurons, given a total of $k = 5, 10, 15, 20$ neurons, respectively. In contrast, the sigmoid function outputs a higher percentage of active neurons, e.g., values of $45\%$ and $44\%$ for varying $k$. This shows that the use of softplus to enforce sparsity not only provides higher retrieval performance, but also returns a more highlighted picture of the active relation types that can potentially lead to improved model interpretability.

To further boost the performance, we fine-tune the whole system using the separately trained MVC and MS as its initialization. The corresponding performance is reported in Table 2 and is referred to as MVC-MS-fn, and offers good improvement over the separate training. Finally, we compare the performance of seven existing methods in Table 2 with the proposed one. It is observed that both MVC-MS and MVC-MS-fn outperform all the competing methods satisfactorily for both datasets, in terms of not only the precision computed using the top 500 retrieved images (500AP), but also the mean averaged precision (mAP) computed using all the test images.

### *4.2. Example Demonstration*

In addition to the quantitative comparison of the retrieval performance in Section 4.1, we further illustrate and compare some examples of the system output and intermediate results, to contrast the proposed method with some competing ones. First,

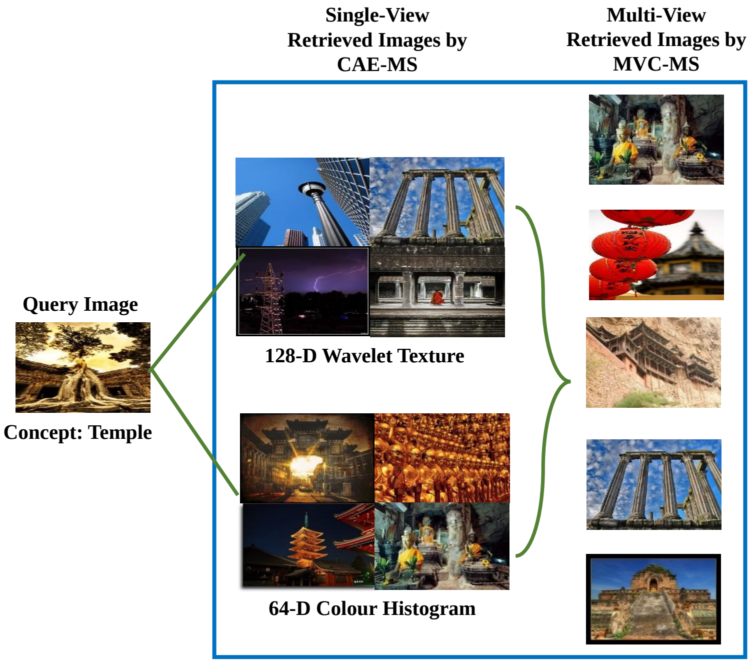

Figure 3: Examples of the retrieved images by CAE-MS with single-view features of 128-D WT and 64-D CH and by the proposed method, using both views given a temple image as the query.

we illustrate four examples of the query image along with their retrieved images by the proposed method in Figure 2, using the NUS-WIDE data. These examples show a good level of diversity among the retrieved images, facilitated by the multi-view and multimodal similarity design.

To demonstrate the advantages of the proposed multi-view training, we replace the proposed MVC features with the CAE features generated by taking two types of single-view features (128-D WT and 64-D CH) as the input. Similarity computation is performed using MS. Two examples are illustrated using the NUS-WIDE dataset, corresponding to two randomly selected query images belonging to the temple and waterfall concepts (Figures 3 and 4). It can be seen that the images retrieved by a single view are similar to the query image under a specific feature type, and this is insufficient to measure the similarities between images. On the contrary, the proposed method takes into account complementary information across views, resulting in more

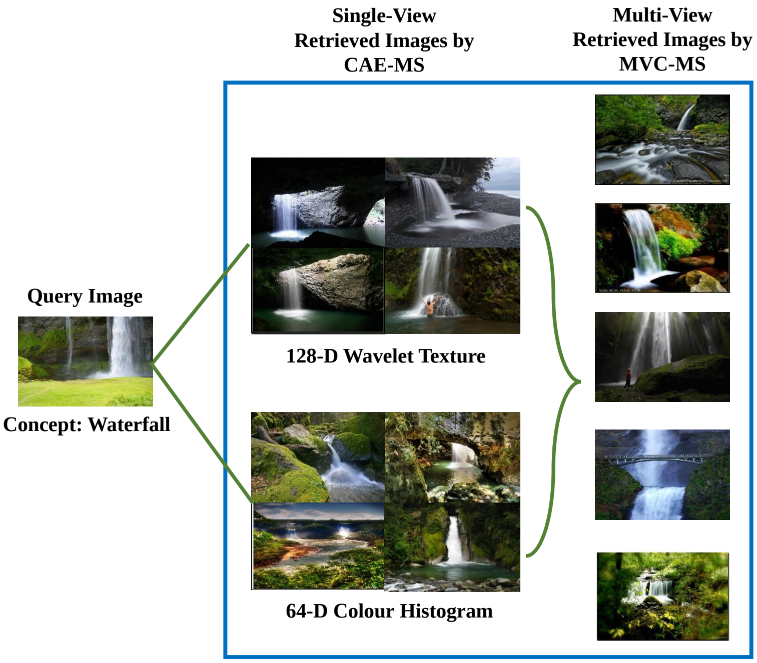
Single-View Retrieved Images by CAE-MS
Multi-View Retrieved Images by MVC-MS
Query Image
Concept: Waterfall
128-D Wavelet Texture
64-D Colour Histogram

Figure 4: Examples of the retrieved images by CAE-MS with single-view features of 128-D WT and 64-D CH and by the proposed method, using both views given a waterfall image as the query.

mixed retrieval output.

To demonstrate the advantages of introducing multimodality to compute similarity, we illustrate example images retrieved by MVF-MS using different ways of initializing the relation embeddings $\{\boldsymbol{e}_t\}_{t=1}^{c}$ with the NUS-WIDE data. When the number of neurons is reduced to $c = 1$, the images are retrieved based on a unimodal relation controled by $\boldsymbol{e}_1$. When initializing $\boldsymbol{e}_1$ with different example images selected randomly, the retrieved images can be different. For example, given an ocean image as query, we experiment with initializing $\boldsymbol{e}_1$ with a fish image, a beach image and a scene image separately. We display the three sets of retrieved images in the left part of Figure 5. It can be seen that the images retrieved using a single relation embedding are only similar to the query image in a specific way. When multiple relation neurons ($c > 1$) are used, relevance between objects becomes more diverse benefiting from the multimodal setup. We experiment with $c = 3$, allowing the inclusion of three relation types. Their

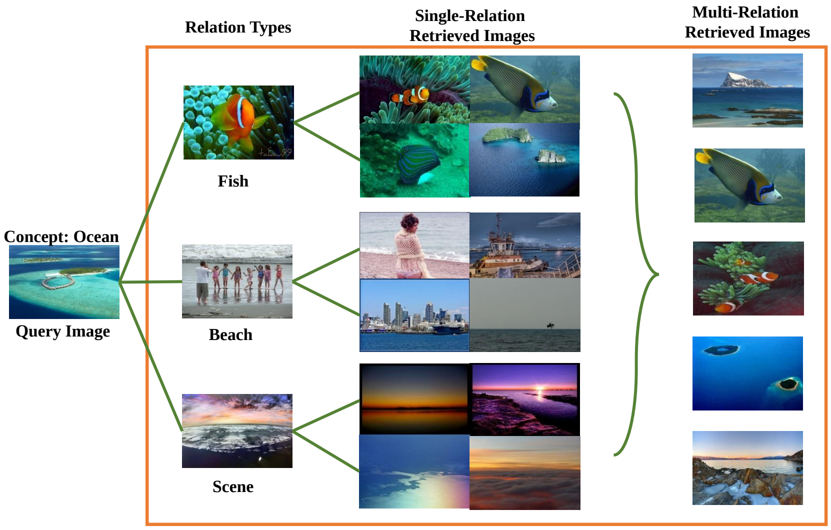
Relation Types
Single-Relation Retrieved Images
Multi-Relation Retrieved Images
Fish
Concept: Ocean
Query Image
Beach
Scene

Figure 5: Examples of the retrieved images by learning under the single relation type and multiple relation types, given an ocean image as the query.

corresponding relation embeddings $\{e_t\}_{t=1}^3$ are initialized by the same three example images (fish, beach, scene) as used to initialize the unimodal relation. The retrieved images are demonstrated in the right part of Figure 5, from which it can be seen that a diverse range of images are retrieved.

In addition to the example of an ocean query as shown in Figure 5, we provide another example for the unimodal and multimodal comparison, following exactly the same way as above, but using a temple image as query. Three example images from the pillar, carving and Buddhism concepts are used to initialize the relation embeddings. The retrieved images are displayed in Figure 6. It can be observed from both Figures 5 and 6 that a unimodal relation exhibits a narrow perspective for measuring the similarities between images. On the contrary, the use of multiple relation neurons controlled by different embeddings, enables to measure more diverse similarity types and leads to improved retrieval performance.

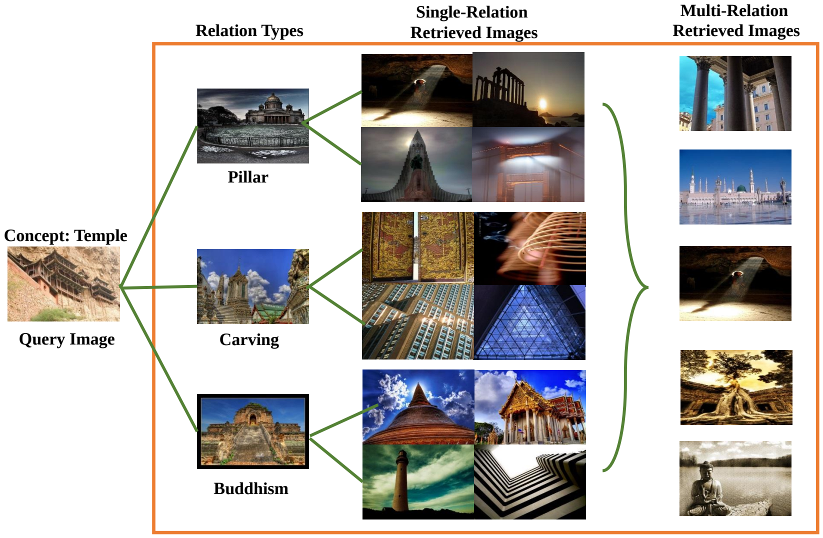
Relation Types
Single-Relation Retrieved Images
Multi-Relation Retrieved Images
Concept: Temple
Query Image
Pillar
Carving
Buddhism

Figure 6: Examples of the retrieved images by learning under the single relation type and multiple types, given a temple image as the query.

## 5. Conclusion

In this work, we have proposed a novel similarity learning model with layered architecture, possessing significantly improved learning performance and model interpretability. By computing topic driven multimodal similarity scores from multi-view voted convolutional features, the model captures not only the individual object characteristics under different view perceptions but also the multimodal nature of the high-level semantics in object interactions.

Overall, the proposed model contains two learning components; an representation learning one and a supervised similarity learning component. The former refines the input data by firstly de-noising and removing redundant information through the AE-based pre-training, and then mining the complementary information across different views, by training the convolutional features to preserve a voted composite neighbor structure. The second component characterizes multiple relation modalities (relation types) with a set of hidden neurons that are parameterized over a set of relation embedding and projection vectors. The model is capable of exploring both straightforward and hidden semantic relations through the initialization control of the relation embeddings. Model interpretability is improved by initializing the relation embeddings with center vectors of the semantic clusters discovered, based on either human expertise knowledge or data-driven approaches (such as clustering algorithms). Sparsity is imposed over the similarities computed for the multiple modalities to further improve model interpretability and enhance generalization.

Evaluated using the CIFAR-10 and NUS-WIDE datasets and compared with seven state-of-the-art algorithms, the proposed algorithm offers the best retrieval performance (around 4-5% improvement over state-of-the-art). Comparative analyses and illustrative examples demonstrate the advantages of the various novel elements in the proposed method, such as the proposed MVC features over the CAE and Caffe features, multi-view over single-view, multimodal over unimodal.

The proposed similarity learning method is fairly generic. In addition to image retrieval, it can be used for other interesting applications, such as image captioning, text matching, etc., and has the potential to analyze similarities between not only image objects, but also text and video objects. This is part of future work direction, which also includes investigation of how to further improve the model design by, for instance, taking into account attention mechanisms to capture salient object parts that contribute to different modalities of the object relevance, and memory mechanisms to improve problem solving by selectively utilizing historical training information.

## Acknowledgements

This work was funded by the visiting Ph.D. studentship of Chinese Scholarship Council.

## References

[1] W. Ma, B. Manjunath, Texture features and learning similarity, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 1996, pp. 425–430.

[2] M. Bicego, V. Murino, M. Pelillo, A. Torsello, Similarity-based pattern recognition, Pattern Recognition 39 (10) (2006) 1813 – 1814.

[3] S. Zhang, M. Yang, T. Cour, K. Yu, Query specific rank fusion for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (4) (2015) 803–815.

[4] J. Wu, H. Shen, Y. D. Li, Z. B. Xiao, M. Y. Lu, C. L. Wang, Learning a hybrid similarity measure for image retrieval, Pattern Recognition 46 (11) (2013) 2927–2939.

[5] A. Ruta, Y. Li, Learning pairwise image similarities for multi-classification using kernel regression trees, Pattern Recognition 45 (4) (2012) 1396–1408.

[6] D. Wu, G. Zhang, J. Lu, A fuzzy preference tree-based recommender system for personalized business-to-business e-services, IEEE Transactions on Fuzzy Systems 23 (1) (2015) 29–43.

[7] Y. Lin, Z. Liu, M. Sun, Y. Liu, Learning entity and relation embeddings for knowledge graph completion, in: Association for the Advancement of Artificial Intelligence, AAAI, 2015.

[8] A. Karpathy, A. Joulin, F. Li, Deep fragment embeddings for bidirectional image sentence mapping, in: Advances in Neural Information Processing Systems, NIPS, 2014, pp. 1889–1897.

[9] E. Xing, M. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, in: Advances in Neural Information Processing Systems, NIPS, 2002, pp. 505–512.

[10] D. Grangier, S. Bengio, A discriminative kernel-based approach to rank images from text queries, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (8) (2008) 1371–1384.

[11] G. E. Hinton, R. S. Zemel, Autoencoders, minimum description length, and helmholtz free energy, Advances in Neural Information Processing Systems, NIPS (1994) 3–12.

[12] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[13] C. Barat, C. Ducottet, String representations and distances in deep convolutional neural networks for image classification, Pattern Recognition 54 (2016) 104 – 115.

[14] B. Leng, S. Guo, X. Zhang, Z. Xiong, 3d object retrieval with stacked local convolutional autoencoder, Signal Processing 112 (2015) 119–128.

[15] A. Makhzani, B. J. Frey, Winner-take-all autoencoders, in: Advances in Neural Information Processing Systems, NIPS, 2015, pp. 2791–2799.

[16] P. Merkle, A. Smolic, K. Müller, T. Wiegand, Multi-view video plus depth representation and coding, in: IEEE International Conference on Image Processing, Vol. 1, 2007, pp. I–201.

[17] X. Gao, T. Mu, M. Wang, Local voting based multi-view embedding, Neurocomputing 171 (2016) 901–909.

[18] Y. Luo, T. Liu, D. Tao, C. Xu, Decomposition-based transfer distance metric learning for image classification, IEEE Transactions on Image Processing 23 (9) (2014) 3789–3801.

[19] H. Xia, S. Hoi, R. Jin, P. Zhao, Online multiple kernel similarity learning for visual search, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (3) (2014) 536–549.

[20] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1798–1828.

[21] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional autoencoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks, ICANN, Springer, 2011, pp. 52–59.

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408. 5093.

[23] Q. Mao, M. Dong, Z. Huang, Y. Zhan, Learning salient features for speech emotion recognition using convolutional neural networks, IEEE Transactions on Multimedia 16 (8) (2014) 2203–2213.

[24] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, F. Chen, High-resolution sar image classification via deep convolutional autoencoders, IEEE Geoscience and Remote Sensing Letters 12 (11) (2015) 2351–2355.

[25] J. Xie, Y. Fang, F. Zhu, E. Wong, Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 1275–1283.

[26] X. Chang, Y. Yu, Y. Yang, E. P. Xing, Semantic pooling for complex event analysis in untrimmed videos, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[27] F. Wu, X. Jing, X. You, D. Yue, R. Hu, J. Yang, Multi-view low-rank dictionary learning for image classification, Pattern Recognition 50 (2016) 143–154.

[28] C. Xu, D. Tao, C. Xu, Multi-view learning with incomplete views, IEEE Transactions on Image Processing 24 (12) (2015) 5812–5825.

[29] X. Xu, W. Li, D. Xu, I. W. Tsang, Co-labeling for multi-view weakly labeled learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (6) (2016) 1113–1125.

[30] B. Qian, X. Wang, J. Ye, I. Davidson, A reconstruction error based framework for multi-label and multi-view learning, IEEE Transactions on Knowledge and Data Engineering 27 (3) (2015) 594–607.

[31] X. Wang, G. Yan, H. Wang, J. Fu, J. Hua, J. Wang, Y. Yang, G. Zhang, H. Bao, Semantic annotation for complex video street views based on 2d–3d multi-feature

fusion and aggregated boosting decision forests, Pattern Recognition 62 (2017) 189–201.

[32] L. Liu, M. Yu, L. Shao, Multiview alignment hashing for efficient image search, IEEE Transactions on image processing 24 (3) (2015) 956–966.

[33] Y. Xu, C. Wang, J. Lai, Weighted multi-view clustering with feature selection, Pattern Recognition 53 (2016) 25–35.

[34] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, A. G. Hauptmann, Multi-feature fusion via hierarchical regression for multimedia analysis, IEEE Transactions on Multimedia 15 (3) (2013) 572–581.

[35] Y. Yan, F. Nie, W. Li, C. Gao, Y. Yang, D. Xu, Image classification by cross-media active learning with privileged information, IEEE Transactions on Multimedia 18 (12).

[36] L. Ma, X. Yang, D. Tao, Person re-identification over camera networks using multi-task distance metric learning, IEEE Transactions on Image Processing 23 (8) (2014) 3656–3670.

[37] P. Wu, S. C. Hoi, P. Zhao, C. Miao, Z. Liu, Online multi-modal distance metric learning with application to image retrieval, IEEE Transactions on Knowledge and Data Engineering 28 (2) (2016) 454–467.

[38] D. Wu, L. Pigou, P. J. Kindermans, N. D. H. Le, L. Shao, J. Dambre, J. M. Odobez, Deep dynamic neural networks for multimodal gesture segmentation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (8) (2016) 1583–1597.

[39] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, A. Yuille, Deep captioning with multimodal recurrent neural networks (m-rnn), in: Proceedings of the International Conference on Learning Representations, ICLR, 2015.

[40] P. Wu, S. C. H. Hoi, H. Xia, P. Zhao, D. Wang, C. Miao, Online multimodal deep similarity learning with application to image retrieval, in: Proceedings of the 21st ACM International Conference on Multimedia, 2013, pp. 153–162.

[41] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep learning via semi-supervised embedding, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 639–655.

[42] Y. Bengio, Learning deep architectures for AI, Foundations and trends® in Machine Learning 2 (1) (2009) 1–127.

[43] Y. Yang, D. Xu, F. Nie, J. Luo, Y. Zhuang, Ranking with local regression and global alignment for cross media retrieval, in: Proceedings of the 17th ACM international conference on Multimedia, ACM, 2009, pp. 175–184.

[44] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, Y. Pan, A multimedia retrieval framework based on semi-supervised ranking and relevance feedback, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (4) (2012) 723–742.

[45] L. Yu, K. M. Hermann, P. Blunsom, S. Pulman, Deep learning for answer sentence selection, in: NIPS Deep Learning and Representation Learning Workshop, Montreal, 2014.

[46] X. Glorot, A. Borders, Y. Bengio, Deep sparse rectifier neural networks, in: Aistats, Vol. 15, 2011, p. 275.

[47] A. Bordes, X. Glorot, J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data, Machine Learning 94 (2) (2014) 233–259.

[48] Y. Gong, S. Lazebnik, A. Gordo, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (12) (2013) 2916–2929.

[49] R. Zhang, L. Lin, R. Zhang, W. Zuo, L. Zhang, Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification, IEEE Transactions on Image Processing 24 (12) (2015) 4766–4779.

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

[51] W. Liu, J. Wang, R. Ji, Y. Jiang, S. F. Chang, Supervised hashing with kernels, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2012, pp. 2074–2081.

[52] J. Tang, Z. Li, M. Wang, R. Zhao, Neighborhood discriminant hashing for large-scale image retrieval, IEEE Transactions on Image Processing 24 (9) (2015) 2827–2840.

[53] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Tech. rep., University of Toronto (2009).

[54] T. Chua, J. Tang, R. Hong, H. Li, Nus-wide: a real-world web image database from national university of singapore, in: ACM International Conference on Image and Video Retrieval, ACM, 2009, p. 48.

[55] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[56] T. Tan, J. Kittler, Colour texture analysis using colour histogram, IEE Proceedings-Vision, Image and Signal Processing 141 (6) (1994) 403–412.

[57] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Vol. 1, IEEE, 2005, pp. 886–893.

[58] A. Ahmadian, A. Mostafa, An efficient texture classification algorithm using gabor wavelet, in: IEEE Conference on Engineering in Medicine and Biology Society, EMBS, Vol. 1, IEEE, 2003, pp. 930–933.

[59] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, R. Zabih, Image indexing using color correlograms, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 1997, pp. 762–768.

[60] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (6) (1986) 679–698.

[61] K. Sande, T. Gevers, C. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1582–1596.

[62] P. C. Ng, S. Henikoff, Sift: Predicting amino acid changes that affect protein function, Nucleic acids research 31 (13) (2003) 3812–3814.