FLORE
Repository istituzionale dell'Università degli Studi di Firenze

Automatic image annotation via label transfer in the semantic space

Automatic Image Annotation via Label Transfer in the Semantic Space

Tiberio Uricchio[a], Lamberto Ballan[b,*], Lorenzo Seidenari[a], Alberto Del Bimbo[a]

[a]*Media Integration and Communication Center (MICC), Università degli Studi di Firenze, Viale Morgagni 65, 50134 Firenze, Italy*
[b]*Department of Mathematics "Tullio Levi-Civita", Università degli Studi di Padova, Via Trieste 63, 35121 Padova, Italy*

Abstract

Automatic image annotation is among the fundamental problems in computer vision and pattern recognition, and it is becoming increasingly important in order to develop algorithms that are able to search and browse large-scale image collections. In this paper, we propose a label propagation framework based on Kernel Canonical Correlation Analysis (KCCA), which builds a latent *semantic space* where correlation of visual and textual features are well preserved into a semantic embedding. The proposed approach is robust and can work either when the training set is well annotated by experts, as well as when it is noisy such as in the case of user-generated tags in social media. We report extensive results on four popular datasets. Our results show that our KCCA-based framework can be applied to several state-of-the-art label transfer methods to obtain significant improvements. Our approach works even with the noisy tags of social users, provided that appropriate denoising is performed. Experiments on a large scale setting show that our method can provide some benefits even when the semantic space is estimated on a subset of training images.

Keywords: Automatic image annotation, Image tagging, Label transfer, Canonical correlation, Semantic space

1. Introduction

A lot of modern applications require image annotation to search, access and navigate the huge amount of visual data stored in personal collections or shared online. Whenever you want to retrieve photos from a particular concert, recall that pleasant summer day in which you napped on your comfortable hammock or look up a person, it is automatic image annotation that enables a plethora of useful applications. The exponential growth of media on sharing platforms, such as Flickr or Facebook, has led to the availability of a huge quantity of images that are enjoyed by millions of people. In such a huge sea of data, it is indispensable to teach computers to correctly label the visual content and help us search and browse image collections.

In this paper, we tackle the challenging task of automatic image annotation. Given an image, we want to assign a set of relevant labels by taking into account image appearance and eventually some prior knowledge on the joint distribution of visual features and labels. Due to its importance, this is a very active subject of research [1, 2, 3, 4, 5, 6, 7, 8]. Previous work typically use images and associated labels to build classifiers and then assign relevant labels to novel images. The early works usually rely on images labeled by domain experts [9, 2, 3, 10, 11], while recently several approaches use weak labels such as user-generated tags in social networks [12, 13, 14] or query terms in search engines [15, 16].

Despite the source of the labeling, non-parametric models which rely on a nearest-neighbor based voting scheme have received a lot of attention for automatic image annotation [17, 10, 18, 19, 20]. The main reason is that these methods have the ability to adapt to complex patterns as more training data become available. To annotate a new image, they apply a common strategy: first, they retrieve similar images in the training set, and second, they rank labels according to their frequency in the retrieval set. Automatic image annotation is thus achieved by transferring the most frequent labels in the neighborhood to the test image. This is essentially a lazy learning paradigm in which the image-to-label association is delayed at test time. In contrast, discriminative models such as support vector machines [21, 22, 23, 24] or fully supervised end-to-end deep networks [8], require to define in advance the vocabulary of labels. This is particularly problematic in a large-scale scenario, such as images on social networks, in which you may have thousands of labels that may also change or increase over time.

Several issues may arise in a nearest-neighbor approach. The set of retrieved images may contain many incorrect labels, mostly because of the so-called *semantic gap* [25]. This happens because visual features may not be powerful enough in abstracting the visual content of the image. Thus the proposed algorithms tend to retrieve just the images whose features are very close in the visual space, but the semantic content is not well preserved. Researchers tried to cope with this issue by improving visual features. To this end, the most significant improvement has been the shift from handcrafting features to end-to-end feature learning, leading to current state-of-the art convolutional neural network representations [26, 27, 28]. Nearest neighbors methods may also suffer when images are not paired with enough label information, leading to a poor statistical quality of the retrieved neighborhood. This is mostly due to the fact

---

*Corresponding author. A major part of this work has been done while the author was on an EU Marie Curie Fellowship at Stanford University and Univ. of Florence.

Email addresses: tiberio.uricchio@unifi.it (Tiberio Uricchio), lamberto.ballan@unipd.it (Lamberto Ballan), lorenzo.seidenari@unifi.it (Lorenzo Seidenari), alberto.delbimbo@unifi.it (Alberto Del Bimbo)
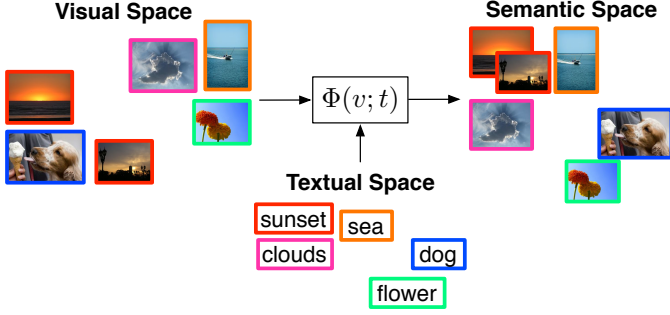
Figure 1: Labels associated to the images can be used to re-arrange the visual features and induce the semantics not caught by the original features. For instance, the sunset images with the red border should be closer to images of clouds and sea, according to the text space. A projection $\Phi(v;t)$ is learned to satisfy correlations in visual and textual space.

that label frequencies are usually unbalanced. Modern methods address this issue by introducing label penalties and metric learning [10, 18, 7].

The image representation can be improved also by shifting to a completely different perspective, namely moving towards a multimodal representation. A way of bridging the semantic gap might be by designing representations that account not just for the image pixels, but also for its textual representation. Here we follow this approach by constructing a framework in which the correlation between visual features and labels is maximized. To this end, we present an automatic image annotation approach that relies on Kernel Canonical Correlation Analysis (KCCA) [29]. Our approach strives to create a semantic embedding through the connection of visual and textual modalities. This embedding lives in a latent space that we refer to as *semantic space*. Images are mapped to this space by jointly considering the visual similarity between images in the original visual space, and label similarities. The projected images are then used to annotate new images by using a nearest-neighbor technique or other standard classifiers. Figure 1 illustrates our pipeline. The main take-home message is that, as illustrated in the figure, the neighborhood of each image will contain more images associated with the same label (e.g. "sunset") in the semantic space than in the original visual space (see for example the images with the red border).

1.1. Main Contributions

(1) The key contribution of our work is to improve image representations using a simple multimodal embedding based on KCCA. This approach has several advantages over parametric supervised learning. First, by combining a visual and textual view of the data, we reduce the semantic gap. Thus we can obtain higher similarities for images which are also semantically similar, according to their textual representation. Second, we are free from predetermining the vocabulary of labels. This makes the approach well suited for nearest neighbor methods, which for the specific task of image annotation are more robust to label noise. A slight disadvantage of our method is its inherent batch nature. Although, as shown in our experimental results, learning the semantic projection is also possible on a subset of the training data.

(2) Previous works that learn multimodal representations from language and imagery exist [30], including prior uses of CCA and KCCA [29, 31, 32, 33]. However, we are the first to propose a framework that combines the two modalities into a joint semantic space which is better exploitable by state-of-the-art nearest neighbor models. Interestingly enough, in our framework the textual information is only needed at training time, thus allowing to predict labels also for unlabeled images.

(3) We provide extensive experimental validations. Our approach is tested on medium and large scale datasets, i.e. IAPR-TC12 [34], ESP-GAME [35], MIRFlickr-25k [36] and NUS-WIDE [37]. We show that our framework is able to leverage recently developed CNN features in order to improve the performance even further. Additionally, we introduce a tag denoising step that allows KCCA to effectively learn the semantic projections also from user-generated tags, which are available at no cost in a social media scenario. The scalability of the method is also validated with subsampling experiments.

This paper builds on our previous contribution on cross-modal image representations [38] and improves in many ways. We report new experimental evaluations covering the large dataset NUS-WIDE. Validate our pipeline with modern convolutional neural network based features. Extend our original approach with a new text filtering method that allows the semantic space to be computed from noisy and sparse tags, such as that from social media. Report new insights on several key aspects such as performance and scalability of our approach when subsampling the training set.

2. Related Work

2.1. Automatic Image Annotation: Ideas and Main Trends

Automatic image annotation is a long standing area of research in computer vision, multimedia and information retrieval [14]. Early works often used mixture models to define a joint distribution over image features and labels [1, 39, 3]. In these models, training images are used as non-parametric density estimators over the co-occurrence of labels and images. Other popular probabilistic methods employed topic models, such as pLSA or LDA, to represent the joint distribution of visual and textual features [40, 2, 41]. They are generative models, thus they maximize the generative data likelihood. They are usually expensive or require simplifying assumptions that can be suboptimal for predictive performance. Discriminative models such as support vector machines (SVM) and logistic regression have also been used extensively [22, 23, 24, 42]. In these works, each label is considered separately and a specific model is trained on a per-label basis. In testing, they are used to predict whether a new image should be labeled with the corresponding label. While they are very effective, a major drawback is that they require to define in advance the vocabulary of labels. Thus, these approaches do not handle well large-scale scenarios in which you may have thousands of labels and the vocabulary may shift over time.

Despite their simplicity, a class of approaches that has gained a lot of attention is that of nearest-neighbor based methods [17, 10, 7, 20]. Their underlying intuition is that similar images are likely to share common labels. Many of these methods start by retrieving a set of visually similar images and then they implement a label transfer procedure to propagate the most common training labels to the test image. The most recent works usually implement also a refinement procedure, such as metric learning [10, 7] or graph learning [43, 44, 45, 46], in order to differently weight rare and common labels or to capture the semantic correlation between labels. They are usually computationally intensive and do not model the intermodal correlation between visual features and labels. In contrast, we introduce a framework in which textual and visual data are mapped to a common semantic space in which labels can be transferred more effectively.

2.2. Towards More Powerful Visual Representations

The most recent breakthrough in computer vision came from end-to-end feature learning through convolutional neural networks. In their seminal paper, Krizhevsky *et al.* [26] demonstrated unprecedented improvement in large-scale image classification on ImageNet [47] using CNNs. These networks are composed of a hierarchy of layers, alternating convolutions and subsampling. They require high quality supervision with minimal noise in labeling. Since then, many researchers have applied deep learning to other visual recognition tasks such as object detection and image parsing [48]. Deeper architectures have been recently proposed, showing further gain in image classification accuracy (e.g. [27]).

Another interesting property of these architectures is that they have the ability to learn representations that can be transferred and used in many other tasks, such as attribute prediction and image retrieval [49]. Convolutional neural networks (CNNs) have been also recently applied to automatic image annotation [8], showing significant improvement in terms of precision and recall. On top of these powerful features, a number of recent works have used more advanced encoding schemes in order to improve feature generalization. For instance, VLAD encoding is applied in [50] to pool multi-scale CNN features computed over different windows, while Fisher Vector encoding applied to dense multi-scale CNN activations is used in [51]. This has been also improved in [52] by applying Fisher Vector to sparse boxes, selected by objectness or random selection. However, all these approaches only focus on the visual modality.

2.3. Cross-media and Multimodal Representations

A number of approaches have been developed for learning multimodal representations from images and labels [1, 3, 12, 30, 53, 54]. In particular, we highlight that previous use of CCA and its variants exists, particularly for the task of cross-modal image retrieval [29, 31, 32, 33, 55, 56] and multi-view learning [57, 58]. This class of methods is often used to learn multi-view embeddings in a unimodal setting. For example, Yang *et al.* [57] use CCA to learn a common representation from two views in the image space. A more general approach is presented in [58] where a latent representation of samples is learned from multiple views. Their framework can be applied also to combine visual features or imagery captured in different conditions.

Hardoon *et al.* were the first to apply KCCA to image retrieval with textual query [29]. Successively, Rasiwasia *et al.* [31] proposed to employ LDA and CCA to perform cross-modal retrieval on text and images obtaining improved results on single modalities. In [32], a method to learn importance of textual object is proposed. They show that features such as word frequency, relative and absolute label rank are helpful to evaluate importance of textual information. Multi-modal learning has been applied to improve ranking in image retrieval fusing visual features and click features in [56]. A three-way CCA is proposed in [33] to address the limited expressiveness of CCA. They show that adding a third view representing categories or clustered labels can improve retrieval performance. Murthy *et al.* [59] propose to combine CNN features and word embeddings using CCA, but their approach is only tested on small scale datasets using expert labels. Embeddings carry many advantages, nonetheless learning such coupled representation may be extremely computationally expensive. Recently, there have been some attempts at making such approaches scalable [53, 60]. These on-line methods have usually low memory footprint, and scale very well to large dataset. Nonetheless, they are not designed to tackle multi-label image annotation and they are not able to learn from noisy examples such as tags extracted from social media.

Differently from prior work, we tackle the specific problem of multi-label image annotation. For this task, only visual features are available at test time. Thus, our approach exploits labels only at training time. To this end, we learn a re-organization of the visual space to that of a semantic space where images that share similar labels are closer. Moreover, when combined to a nearest-neighbors scheme, our approach can predict labels that were not available at training time, when the projections have been learned.

3. Approach

Our key intuition is that the semantic gap of visual features can be reduced by constructing a semantic space that comprises the fusion of visual and textual information. To this end, we learn a transformation that embeds textual and visual features into a common multimodal representation. The transformation is learned using KCCA [29]. This algorithm strives to provide a common representation for two views of the same data. Similarly to [29, 32], we use KCCA to connect visual and textual modalities into a common *semantic space*, but differently from them, which focus on cross-modal retrieval, our framework is designed to effectively tackle the particular problem of image annotation. Moreover, we are able to construct the semantic space even exploiting noisy labels, such as the user tags. Advanced nearest neighbors methods are then used to perform label transfer. An overview of the approach is shown in Fig. 2.
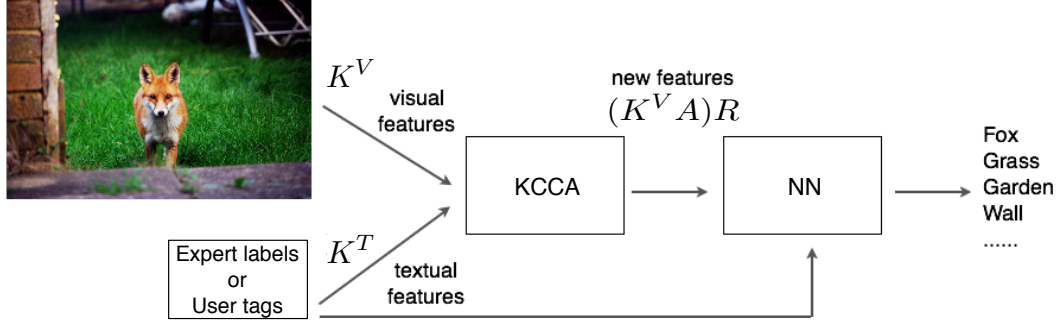
Figure 2: Overview of our approach. Image and textual features are projected onto a common *semantic space* in which nearest-neighbor voting is used to perform label transfer.

Throughout the paper, we use the term *labels* when we refer to generic textual information. We explicitly use the terminology *expert labels* and *user tags* when we refer only to the expert provided labels or the tags provided by users in social network, respectively. We now proceed in detailing the visual and textual representation, how KCCA is used to build the semantic space, and finally we describe our label transfer procedures.

3.1. Visual Features

We use a deep convolutional neural network pre-trained on ImageNet [47] with the VGG-Net architecture presented in [27] (using 16 layers)[1]. We use the activations of the last fully connected layer as image features. Such representation proved to be good for several visual recognition and classification tasks [49, 48].

Given an image I_i, we first warp it to 224×224 in order to fit the network architecture and subtract the training images mean. We use this normalized image to extract the activations of the first fully connected layer. Let $\phi^V(I_i)$ be the extracted feature of I_i. We use the ArcCosine kernel:

$$K_n^V(\phi^V(I_i), \phi^V(I_j)) = \frac{1}{\pi}\|\phi^V(I_i)\|^n \|\phi^V(I_j)\|^n J_n(\theta) \quad (1)$$

where J_n is defined according to the selected order of the kernel. Following [61], we set $n = 2$ which gives us:

$$J_2(\theta) = 3\sin\theta\cos\theta + (\pi - \theta)(1 + 2\cos^2\theta) \quad (2)$$

where θ is the angle between the inputs $\phi^V(I_i)$, $\phi^V(I_j)$. This kernel provides a representation that is better suited to neural networks activations and gives better results. We also tried other kernels such as linear and radial basis function, obtaining a slightly inferior performance ($\sim$1%).

3.2. Textual Features

Depending on how labels are generated, *i.e.* expert labels or user-generated tags, we should use different approaches. While expert labels can be trusted, user-generated tags are noisy and require a more robust representation.

3.2.1. Expert Labels

For expert labels, we use simple binary indicator vectors as textual features. Let D be the vocabulary size, *i.e.* the number of labels used for annotation. We map each label set of a particular image I_i to a D-dimensional feature vector $\phi^T(I_i) = [w_1^i, \cdots, w_D^i]$, where w_k is 0 or 1 if that image has been annotated with the corresponding k-th label l_k. This results in a highly sparse representation. Then we use a linear kernel which corresponds to counting the number of labels in common between two images:

$$K^T(\phi^T(I_i), \phi^T(I_j)) = \sum_{k=1}^{D} w_k^i w_k^j. \quad (3)$$

The basic idea is that we are considering the co-occurrences of labels in order to measure the similarity between two images. Nonetheless, this representation models each label independently from the others. It has been shown in previous works that exploiting semantic relations by weighting each label differently can improve performance [13, 62]. Therefore, we explore two textual kernels that consider semantic relations between labels: an ontology-based textual kernel with bag-of-words [63] and one that exploits the more recent continuous word vector representation [64]. For the bag-of-words semantic kernel, the idea is to weight each label in a linear kernel by using a similarity matrix $S \in \mathbb{R}^{D \times D}$ as:

$$K^T(\phi^T(I_i), \phi^T(I_j)) = \phi^T(I_i) S \phi^T(I_j)^{\mathsf{T}}. \quad (4)$$

We set the elements of S as the Lin similarity [65] between each label, using WordNet. This measure has been used successfully in several works to suggest similar labels (see [14]). Regarding the continuous word vector kernel, Mikolov *et al.* [64] recently showed that it is possible to learn a word representation from a large scale corpus in an unsupervised way. The learned word vector features were proved to model semantics in form of regularities in several applications [53, 59]. Given the learned representation of a label w_k as $\zeta(w_k) \in \mathbb{R}^P$, we represent the set of labels of an image I_i using average pooling

$$\phi^T(I_i) = \frac{1}{N} \sum_k^D w_k^i \cdot \zeta(l_k^i). \quad (5)$$

---

[1]In our preliminary experiments we found that this configuration gives the best results on all our datasets, although other networks gave similar results.

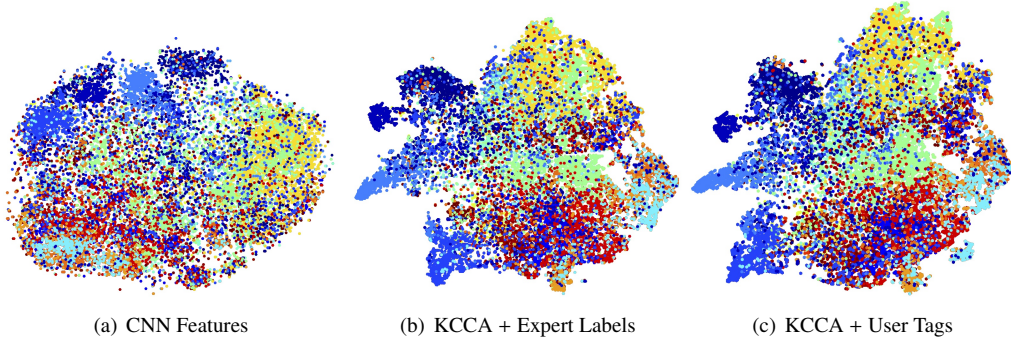(a) CNN Features   (b) KCCA + Expert Labels   (c) KCCA + User Tags

Figure 3: t-SNE visualization of images on MIRFlickr-25K with different features. Each color corresponds to a different label.

Finally, we apply a linear kernel on such representation:

$$K^T(\phi^T(I_i), \phi^T(I_j)) = \phi^T(I_i)\phi^T(I_j)^\intercal. \tag{6}$$

We compare the performance obtained with these three textual representations in Sect. 4.6.

3.2.2. Denoising User-generated Tags

For user-generated tags, we should first reduce the labeling noise. To this end, we perform a "pre-propagation" step based on visual similarity. The purpose of this tag denoising step is two-fold: first, we need to improve the quality of tags of each training image in order to learn a proper embedding; second, we need to cope with the sparsity of user tags. For the first issue, our assumption is that by gathering a neighborhood of visually similar images the more frequent tags will fade out noisy tags in favor of content related ones. Regarding the sparsity issue, images usually are labeled with few tags and in extreme cases they can have no tags at all. For this reason, the visual information is the most reliable information we can exploit.

Thus, we shall obtain a cleaner tag feature-vector $\hat{\phi}^T(I_i) = [\hat{w}_{i,1}, \cdots, \hat{w}_{i,D}]$ and then compute the textual kernel K^T. We start from the representation $\phi^T(I_i) = [w^i_1, \cdots, w^i_D]$, where w_k is 0 or 1 if the image I_i has been annotated with the corresponding tag t_k. For each image I_i we consider the R=100 most similar images, according to the visual kernel K^V (the same precomputed in Eq. 1), and compute the new tag vector:

$$\hat{\phi}^T(I_i) = \frac{\sum_{k=1}^{R} x_k \phi^T(I_k)}{\sum_{k=1}^{R} x_k} \tag{7}$$

where $x_k = \exp(-\frac{\|\phi^V(I_i)-\phi^V(I_k)\|^2}{\sigma})$ is an exponentially decreasing weight computed from image similarities. We set σ to the mean of the distances. This improved tag vector can be seen as an approximation of the probability mass function of tags among its nearest neighbor images. We use the exp-χ^2 kernel:

$$K^T(\hat{\phi}^T(I_i), \hat{\phi}^T(I_j)) = \exp\left(-\frac{1}{2C}\sum_{k=1}^{D}\frac{(\hat{w}_{i,k}-\hat{w}_{j,k})^2}{(\hat{w}_{i,k}+\hat{w}_{j,k})}\right) \tag{8}$$

where C is set to the mean of the χ^2 distances. We demonstrate in section 4.5 that this pre-propagation step is essential to learn the semantic embedding properly, as clearly shown by the results reported in Table 6.

3.3. Kernel Canonical Correlation Analysis

Given two views of the data, such as the ones provided by visual and textual features, we can construct a common multi-modal representation. We first briefly describe CCA and then move to explain the extended KCCA algorithm. CCA seeks to utilize data consisting of paired views to simultaneously find projections from each feature space so that the correlation between the projected representations is maximized.

More formally, given N training pairs of visual and textual features $\{(\phi^V(I_1), \phi^T(I_1)), \ldots, (\phi^V(I_N), \phi^T(I_N))\}$, the goal is to simultaneously find directions z^*_V and z^*_T that maximize the correlation of the projections of ϕ^V onto z^*_V and ϕ^T onto z^*_T. This is expressed as:

$$z^*_V, z^*_T = \arg\max_{z_V, z_T} \frac{\mathrm{E}[\langle\phi^V, z_V\rangle\langle\phi^T, z_T\rangle]}{\sqrt{\mathrm{E}[\langle\phi^V, z_V\rangle^2]\mathrm{E}[\langle\phi^T, z_T\rangle^2]}}$$
$$= \arg\max_{z_V, z_T} \frac{z_V^\intercal C_{vt} z_T}{\sqrt{z_V^\intercal C_{vv} z_V z_T^\intercal C_{tt} z_T}} \tag{9}$$

where $\mathrm{E}[\cdot]$ denotes the empirical expectation, while C_{vv} and C_{tt} respectively denote the auto-covariance matrices for ϕ^V and ϕ^T, and C_{vt} denotes the between-sets covariance matrix.

The CCA algorithm can only model linear relationships. As a result, KCCA has been introduced to allow projecting the data into a higher-dimensional feature space by using the kernel trick [29]. Thus, the problem is now to search for solutions of z^*_V and z^*_T that lie in the span of the N training instances $\phi^V(I_i)$ and $\phi^T(I_i)$:

$$z^*_V = \sum_{i=1}^{N} \alpha_i \phi^V(I_i), \qquad z^*_T = \sum_{i=1}^{N} \beta_i \phi^T(I_i). \tag{10}$$

The objective of KCCA is to identify the weights $\alpha, \beta \in \mathbb{R}^N$ that maximize:

$$\alpha^*, \beta^* = \arg\max_{\alpha, \beta} \frac{\alpha^\intercal K^V K^T \beta}{\sqrt{\alpha^\intercal (K^V)^2 \alpha \beta^\intercal (K^T)^2 \beta}} \tag{11}$$

where K^V and K^T denote the $N \times N$ kernel matrices over a sample of N pairs. As shown by Hardoon *et al.* [29], learning should be regularized in order to avoid trivial solutions. Hence,
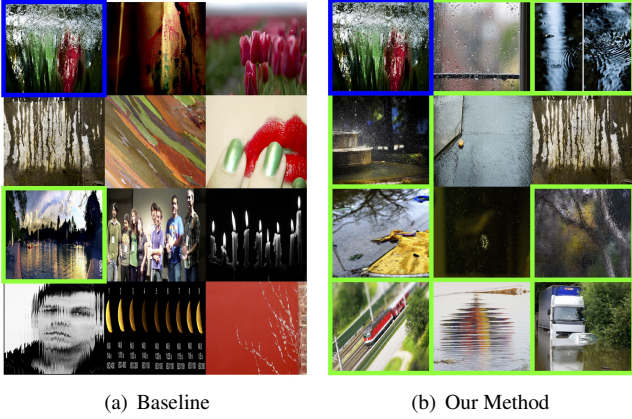
(a) Baseline   (b) Our Method

Figure 4: Nearest neighbors found with baseline representation (a) and with our proposed method (b) for a water image (first highlighted in blue in both figures) from the MIRFlickr-25K dataset. Training images with ground truth label *water* are highlighted with a green border. Nearest neighbors are sorted by decreasing similarity.
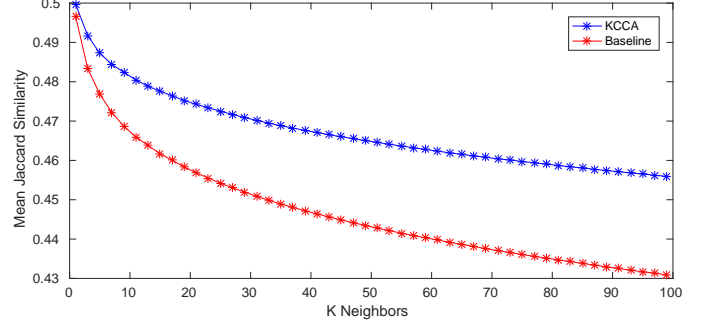

Figure 5: Mean Jaccard similarity between label sets of a test image and the label sets of images in the neighborhood build using visual and KCCA features varying the neighborhood size.

we penalize the norms of the projection vectors and obtain the generalized eigenvalue problem:

$$(K^V + \kappa I)^{-1} K^T (K^T + \kappa I)^{-1} K^V \alpha = \lambda^2 \alpha \qquad (12)$$

where $\kappa \in [0, 1]$. The top M eigenvectors of this problem yield bases $A = [\alpha_1 \dots \alpha_M]$ and $B = [\beta_1 \dots \beta_M]$ that we use to compute the semantic projections of training and test kernels. For each pair (α_j, β_j) of the given bases, the corresponding eigenvalue r_j measures the correlation between projected input pairs. Higher r_j is associated with higher correlation, thus it is convenient to weight more the dimensions of higher energy. According to this principle, we obtain the final features as:

$$\psi(I) = (K^V A) R \qquad (13)$$

where $R = \mathrm{diag}([r_1, \dots, r_M])$. Note that ψ has no dependency on the textual space. Thus, projecting new test images requires only their visual features Φ^V, making our approach suitable for automatic image annotation.

In Figure 3 we show t-SNE embeddings [66] of the CNN features and their projection into the semantic space. These plots qualitatively show that KCCA improves the separation of the classes, both in case of expert labels and user-generated tags. This leads to a more accurate manifold reconstruction and, as our experiments will confirm, a significant improvement in performance.

3.4. Label Transfer

The constructed semantic space assures that similar images, in visual space or in textual space, have also similar features. This property is especially useful for the class of nearest-neighbor methods, since they rely on the intuition that similar images share common labels. We show examples of this property in Figure 4. We compare the neighbors retrieved for the same query using the baseline visual features and the semantic space features from our method. The query, depicted in a blue box, is

an image of water where green and red lights produce a fascinating visual effect. The other images are the most similar images retrieved by one of the two settings. We put a box in green on images that have the correct label "water" associated. We see that neighbors retrieved in the baseline space share some visual similarity: they mostly have green and red colors, some line or dotted patterns that mimic the query image. However only one image is really about water. Our method, instead, successfully retrieves 8 of 11 images with the label water, even if they are quite dissimilar in the visual space. Indeed, it is impossible with the images in Figure 4(a) to obtain a meaningful neighborhood since the correct label "water" is not frequent enough to be relevant in the final labels rank.

A quantitative characterization of this behavior can be seen comparing the sets of labels of images in the neighborhood of a test image with the correct labels of the image itself. We run an experiment on NUS-WIDE, measuring this similarity using Jaccard distance. Specifically, for each image $\hat{x}$ of the test set, we retrieve the K most similar images $\{x_1, x_2, \dots, x_K\}$ using the visual features and then compute the mean Jaccard similarity between their sets of labels as:

$$\frac{1}{K} \sum_{i=1}^{K} J(\hat{\mathcal{Y}}, \mathcal{Y}_i) = \frac{1}{K} \sum_{i=1}^{K} \frac{|\hat{\mathcal{Y}} \cap \mathcal{Y}_i|}{|\hat{\mathcal{Y}}| + |\mathcal{Y}_i| - |\hat{\mathcal{Y}} \cap \mathcal{Y}_i|},$$

where $\hat{\mathcal{Y}}$ and $\mathcal{Y}_i$ are, respectively, the set of labels of $\hat{x}$ and x_i. We compute this measure for each test image and average them in a final similarity index as reported in Fig. 5.

The higher Jaccard similarity yielded by KCCA features with respect to baseline visual features, shows that the neighbors retrieved using KCCA have a label distribution which is closer to the one of the query.

Following this key idea, we have used four nearest-neighbor voting algorithms in our semantic space in order to automatically annotate images. Nevertheless, we expect that other general class of learning algorithms may take advantage of the semantic space. To this end, we also consider the off-the-shelf SVM classifier. Given an image and a vocabulary of labels, each algorithm performs automatic image annotation by applying a particular *relevance function* [14], as defined in the following.

3.4.1. Nearest-Neighbor Voting

The most straightforward approach is to project the test image onto the semantic space, and then identify its K nearest-neighbors. Here we rank the vocabulary labels according to the their frequency in the retrieval set. Thus, the relevance function is defined as:

$$f_{KNN}(I, t) := k_t \qquad (14)$$

where k_t is the number of images labeled as t in the neighborhood of I.

3.4.2. Tag Relevance

Li *et al.* [18] proposed a relevance measure based on the consideration that if several people label visually similar images using the same labels, then these labels are more likely to reflect objective aspects of the visual content. Following this idea it can be assumed that, given a query image, the more frequently the tag occurs in the neighbor set, the more relevant it might be. However, some frequently occurring labels are unlikely to be relevant to the majority of images. To account for this fact, the proposed tag relevance measurement takes into account both the number of images with tag t in the visual neighborhood of I (namely k_t) and in the entire collection:

$$f_{TagVote}(I, t) := k_t - K \frac{n_t}{|S|} \qquad (15)$$

where n_t is the number of images labeled with t in the entire collection S and K is the number of neighbors retrieved.

3.4.3. TagProp

Guillaumin *et al.* [10] proposed an image annotation algorithm in which the main idea is to learn a weighted nearest neighbor model, to automatically find the optimal metric that maximizes the likelihood of a probabilistic model. The method can learn rank-based or distance-based weights:

$$f_{TagProp}(I, t) := \sum_{j}^{K} \pi_j \cdot \mathcal{I}(I_j, t) \qquad (16)$$

where K is the number of neighbors retrieved, $\mathcal{I}$ is the indicator function that returns 1 if I_j is labeled with t, and 0 otherwise; π_j is a learned weight that accounts for the importance of the j-th neighbor I_j. In addition the model can be extended with a logistic per-tag model to promote rare labels and suppress the frequent ones.

3.4.4. 2PKNN

Verma and Jawahar [7] formulated the problem as a probabilistic framework and proposed a two-phase approach: given a test image, a first phase is employed to construct a balanced neighborhood. Then, a second phase uses image distances to perform the actual estimation of the tag relevance. Given a test image I and a vocabulary of D labels, the first phase collects a set of neighborhoods $\mathcal{N}(I)$ composed of the nearest M training images annotated with each t in D. On the second phase, the

Table 1: Datasets Statistics.

Dataset	Images	Labels	Tags	Expert Labels	User Tags
IAPR-TC12	19,627	291	-	✓	-
ESP-GAME	20,770	268	-	✓	-
MIRFlickr-25k	25,000	18	1,386	✓	✓
NUS-WIDE	269,648	81	5,018	✓	✓

balanced neighborhood is used to estimate the tag relevance of t to I:

$$f_{2PKNN}(I, t) := \sum_{I_j \in \mathcal{N}(I)} \exp(-d(I, I_j)) \cdot \mathcal{I}(I_j, t) \qquad (17)$$

where $d(I, I_j)$ is a distance function between image I and I_j. Since the distance function is parametrized with a trainable weight for each dimension, the algorithm presented in [7] also performs metric learning similarly to TagProp (we refer to the complete algorithm as to 2PKNN-ML). We only consider the version without metric learning, since our implementation of 2PKNN-ML performs worse than 2PKNN.

3.4.5. SVM

For each label, a binary linear SVM classifier is trained using the L2-regularized least square regression, similarly to [67]. Independently from the source of labels, be it expert labels or user tags, the images with the label are treated as positive samples while the others as negative samples. To efficiently train our classifier we use stochastic gradient descent (SGD). The relevance function is thus:

$$f_{SVM}(I, t) := b + \langle w_t, \psi(I) \rangle, \qquad (18)$$

where w_t are the weights learned for label t and b is the intercept.

4. Experiments

4.1. Datasets

Automatic image annotation with expert labels has been historically benchmarked with three datasets: Corel5K, ESP-GAME and IAPR-TC12. We follow previous work but discard Corel5K since it is outdated and not available publicly. Note that these datasets have poor quality images and they lack metadata as well as user tags. Thus, we additionally consider two popular datasets collected from Flickr, i.e. MIRFlickr-25k and NUS-WIDE. Dataset statistics are summarized in Table 1.

ESP-GAME. The ESP-GAME dataset [35] was built through an online game. Two players, not communicating with each other, describe images through labels and obtain points when they agree on the same terms. Since the image is the only media the players see, they are pushed to propose visually meaningful labels. Following previous work, we used the same split of [10] consisting of 18,689 images for training and 2,081 for test. There is an average of 4.68 annotated labels per image out of 268 total candidates.

IAPR-TC12. This dataset was introduced in [34] for cross-language information retrieval. It is a collection of $19,627$ images comprised of natural scenes such as sports, people, animals, cities or other contemporary scenes. Like previous work, we used the same setting as in [10]. It consists of $17,665$ training images and $1,962$ testing images. Each image is annotated with an average of 5.7 labels out of 291 candidates.

MIRFlickr-25K. The MIRFlickr-25K dataset [36] has been introduced to evaluate keyword-based image retrieval. It contains $25,000$ images downloaded from Flickr, $12,500$ images for training and the same amount for testing. For each image, the presence of 18 labels are available as expert labels as well as user tags (we consider the same labels as in [67]). They are annotated with an average of respectively 2.78 expert labels and 8.94 user tags. Note that tags corresponding to the expert labels are very scarce in this dataset. Beside tag annotations, EXIF information and other metadata such as GPS are available. While the ground-truth labels are exact, the user tags are weak, noisy and overly personalized. Moreover, not all of them are relevant to the image content. We used the same training and test sets as in previous work [67].

NUS-WIDE. The NUS-WIDE dataset [37] is composed of $269,648$ images retrieved from Flickr. Similarly to MIRFlickr, 81 labels are provided as expert labels as well as user tags. Images are annotated with an average of 2.40 expert labels and 8.48 user tags, respectively. NUS-WIDE is one of the largest datasets of images collected from social media. The sparsity of labels and user tags is one of the main challenges in exploiting this dataset as a training set. Moreover the distribution of labels is unbalanced with few concepts being present in almost 80% of the images: "sky", "clouds", "person" and "water". Following previous work, we discard images without any expert label [8], leaving us with $209,347$ images that we further split into ~125K for training and ~80K for testing, by using the split provided by the authors of the dataset.

4.2. Evaluation Protocol

The performance of automatic image annotation on these datasets has been measured with different metrics. Therefore, for each dataset, we carefully follow previous work protocols. We employ four popular metrics to assess the performance of our algorithm and compare to existing approaches.

Image annotation is usually addressed by predicting a fixed number of labels, n, per image (e.g. $n = 3$, $n = 5$). We compute precision (Prec@n) and recall (Rec@n) by averaging these two metrics over all the labels. Considering that image ground-truth labels may be less or more than n, and we are constrained by this setup to predict n labels, perfect precision and recall can not be obtained. We also report results using Mean Average Precision (MAP), which takes into account all labels for every image, and evaluates the full ranking. First, we rank all test images according to the predicted relevance to compute AP for each label, then we report the mean value of AP over all labels. Finally we report N+ which is often used to denote the number of labels with non-zero recall. N+ is an interesting metric when the set of labels has a moderate to high cardinality, otherwise

it tends to saturate easily not providing adequate information on a method. It has to be noted that each metric evaluates very different properties of each method. Therefore a method hardly dominates over the competition on every metric. Some methods, by design, provide better Recall or Precision than others.

For IAPR-TC12 and ESP-GAME, the standard protocol is to report Prec@5, Rec@5 and N+ [9, 17]. For completeness we report MAP on these two datasets although, as can be seen in Table 2, few previous work also report this metric.

For MIRFlickr, considering that annotated labels are used to perform image retrieval, the few existing works report only the MAP [67]. We also report Prec@5 and Rec@5. Considering the low cardinality of the tag vocabulary (18), N+ is not reported for this dataset.

For NUS-WIDE, performances are usually reported either as MAP or precision and recall. Since NUS-WIDE has a lower average number of labels per image than IAPR-TC12 and ESP-GAME, we report results with $n = 3$ labels, as in [8, 13].

4.3. Implementation Details and Baselines

In order to avoid degeneracy with non-invertible Gram matrices and to increase computational efficiency, we approximate the Gram matrices using the Partial Gram-Schmidt Orthogonalization (PGSO) algorithm provided by Hardoon *et al.* [29]. In all the experiments we have empirically fixed $\kappa = 0.5$ (see Eq. 12) since it gave the best performance in early experiments on IAPR-TC12. We use approximate kernel matrices given by the PGSO algorithm, where we consider at most $4,096$ dimensions (i.e. the dimension of the semantic space). Thus the dimensionality of $\psi(I)$ in Eq. 13 is $4,096$. In this case, the distance between two images is defined as the cosine distance between ψ features.

Since our approach is based on semantic space built from visual data and the available labels, we consider as baselines the label transfer methods trained on the bare visual features. The distance between two images I_q and I_i is defined as $d(I_q, I_i) = 1 - K^V(I_q, I_i)$, where K^V is the visual kernel described in Eq. 1, normalized with values in $[0, 1]$.

The number of nearest neighbors K and the C of SVM were fixed by performing a 3-fold cross-validation on the training set for each dataset.

4.4. Experiment 1: Performance with Expert Labels

As a first experiment we analyze the performance of our method when the semantic space is built from expert labels. In Tables 2, 3 and 4 we report the performance of the state of the art, the five methods ran in the visual feature space and in the semantic space, respectively. Our best result is superior to the state of the art on NUS-WIDE and MIRFlickr-25K while it is comparable to more tailored methods on IAPR-TC12 and ESP-GAME.

Table 2 shows the performance of the state of the art methods, the baselines and our approach on IAPR-TC12 and ESP-GAME. We first note that the majority of previous works report results with 15 handcrafted features (HC) [10] while we use the more recent VGG16 CNN activations, the same as [59]. By

Table 2: Results of our method compared to the state of the art on IAPR-TC12 and ESP-GAME, using **expert labels**.

Method	Visual Feat	IAPR-TC12				ESP-GAME			
		MAP	Prec@5	Rec@5	N+	MAP	Prec@5	Rec@5	N+
State of the art:									
MBRM [39]	HC	-	24	23	223	-	-	-	-
JEC-15 [17]	HC	-	29	19	211	-	-	-	-
TagProp [10]	HC	**40**	46	35	266	**28**	39	27	239
GS [5]	HC	-	32	29	252	-	-	-	-
RF-opt [68]	HC	-	44	31	253	-	-	-	-
2PKNN-ML [7]	HC	-	**54**	**37**	**278**	-	53	27	252
KSVM-VT [24]	HC	-	47	29	268	-	**55**	25	**259**
SKL-CRM [69]	HC	-	47	32	274	-	41	26	248
CCA-KNN [59]	VGG16	-	41	34	273	-	44	**32**	254
RLR [42]	Alexnet	-	46	**41**	277	-	-	-	-
Baselines:									
NNvot	VGG16	36	39	29	239	28	31	28	232
TagRel	VGG16	35	34	35	262	30	29	31	240
TagProp	VGG16	38	40	32	257	32	34	32	241
2PKNN	VGG16	**41**	**41**	**39**	**276**	**36**	**43**	**36**	**257**
SVM	VGG16	34	31	29	221	31	29	30	224
Our Approach:									
KCCA + NNvot	VGG16	40	44	34	250	34	38	34	240
KCCA + TagRel	VGG16	40	41	37	259	35	33	37	249
KCCA + TagProp	VGG16	41	44	34	257	37	38	36	247
KCCA + 2PKNN	VGG16	**43**	**49**	**38**	**278**	**39**	**45**	**39**	**260**
KCCA + SVM	VGG16	41	44	35	252	37	38	37	251

Table 3: Results of our method compared to the state of the art on the dataset MIRFlickr-25K, using **expert labels**.

Methods	Visual Feat	MIRFlickr-25K		
		MAP	Prec@5	Rec@5
State of the art:				
TagProp [67]	HC	46.5	-	-
SVM [67]	HC	52.3	-	-
Autoencoder [30]	HC	60.0	-	-
DBM [30]	HC	60.9	-	-
MKL [54]	HC	**62.3**	-	-
Baselines:				
NNvot	VGG16	69.9	44.7	69.2
TagRel	VGG16	68.9	41.5	72.1
TagProp	VGG16	70.8	**45.5**	70.1
2PKNN	VGG16	66.5	46.4	70.9
SVM	VGG16	**72.7**	38.8	**72.4**
Our Approach:				
KCCA + NNvot	VGG16	72.9	46.1	73.1
KCCA + TagRel	VGG16	70.7	45.2	72.6
KCCA + TagProp	VGG16	73.0	44.6	74.1
KCCA + 2PKNN	VGG16	67.7	**47.3**	74.6
KCCA + SVM	VGG16	**73.0**	38.9	**75.0**

Table 4: Results on the NUS-WIDE dataset using **expert labels**.

Methods	Visual Feat	NUS-WIDE		
		MAP	Prec@3	Rec@3
State of the art:				
CNN + SoftMax [8]	RGB	-	31.7	31.2
CNN + WARP [8]	RGB	-	31.7	35.6
CNN + NNvot [13]	BLVC	44.0	**44.4**	30.8
CNN + logistic [13]	BLVC	**45.8**	40.9	**43.1**
MIE Ranking [70]	BLVC	-	37.9	38.9
MIE Full Model [70]	BLVC	-	37.8	40.2
Baselines:				
NNvot	VGG16	49.3	39.6	44.0
TagRel	VGG16	49.2	32.1	50.3
TagProp	VGG16	**50.9**	**41.3**	44.6
2PKNN	VGG16	48.0	39.7	52.2
SVM	VGG16	50.2	34.6	**60.6**
Our Approach:				
KCCA + NNvot	VGG16	51.7	40.2	50.5
KCCA + TagRel	VGG16	51.4	34.4	**57.2**
KCCA + TagProp	VGG16	**52.2**	45.2	49.2
KCCA + 2PKNN	VGG16	50.7	**53.0**	47.0
KCCA + SVM	VGG16	51.8	43.3	48.4

exploiting this feature, simple nearest neighbor methods like NNvot and TagRel reach a higher Prec@5 and Rec@5 compared to the similar JEC-15 [17] which uses a combination of HC features. Our baseline TagProp has a slight inferior performance to that reported in [10], probably due to the lower number of learnable parameters, having only one single feature versus 15. Comparing our approach versus the baselines, we ob-

serve that all metrics consistently report higher values when label transfer is applied in the semantic space. This suggests that classes in the semantic space are easier to separate. We reach our best result on IAPR-TC12 and ESP-GAME with KCCA + 2PKNN, still inferior to 2PKNN-ML [7] that is additionally applying metric learning.

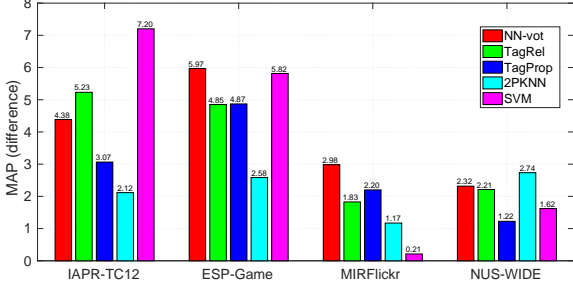Table 3 shows our results on the MIRFlickr-25k dataset.

Figure 6: MAP difference of the four methods trained with KCCA on ESPGame, IAPR-TC12, MIRFlickr-25k and NUS-WIDE. KCCA is trained using **expert labels**.
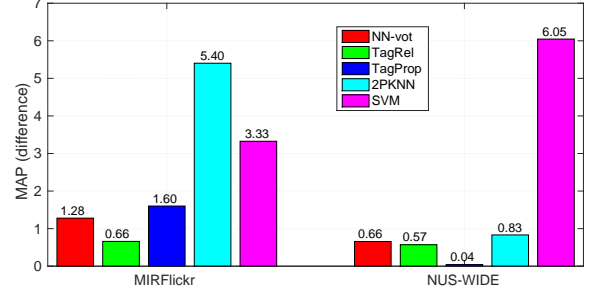

Figure 7: MAP relative difference of the four methods trained with KCCA on ESP-Game, IAPR-TC12, MIRFlickr-25k and NUS-WIDE. KCCA is trained with **user tags**.

Again, we first note that by simply switching from HC features to VGG16, a large boost of MAP is obtained. Focusing on TagProp and SVM baselines, which are directly comparable with previous work [67], MAP increases from 52.3 to 72.7 and from 46.5 to 70.8, respectively. This is consistent with recent literature that suggests CNN activations are way more powerful than handcrafted features. We also report the experimental results of [30], obtained using autoencoders and multimodal Deep Boltzmann Machines, and [54] (semi-supervised multimodal kernel learning), which are the previous state-of-the-art results on this dataset. Applying our KCCA-based framework to the five methods results in a generalized improvement of all metrics, especially on the four nearest neighbor schemes. The best MAP is obtained by KCCA + SVM that reaches a score of 73.0, higher than the best baseline. Interestingly, KCCA + NNvot and KCCA + TagProp reach a score of 72.9, that is higher than the best baseline SVM. We can observe that our semantic space improves both Rec@5 and Prec@5, specifically an average increase of 3.1 for Rec@5 and of 2.1 of Prec@5 can be measured for all 5 baseline methods.

We report in Table 4 the results of the comparison on the large-scale NUS-WIDE dataset. Previous works used BLVC (Caffe reference model) features (e.g. [13]) while we use VGG16, but this does not provide significant differences in performance. Moreover Gong *et al.* [8] attempted to train the network from scratch, obtaining an inferior performance with respect to pretrained features on ImageNet [13, 8]. A higher score of Rec@3 is observed in all our experiments with respect to the state of the art. This suggests that our approach is able to work with unbalanced distribution of labels, and improves recall of rare labels. KCCA + TagProp is the overall best method on this dataset, even superior to SVM that is commonly recognized as better than kNN-based methods for classification.

In summary, our framework is always able to improve performance in all datasets with every metric. This is an important result since each particular metric captures different properties. On smaller datasets, such as IAPR-TC12 and ESP-GAME, metric learning based approaches [7, 10] take more advantage from using 15 different but weaker features then a single, stronger one, as we do. Although on larger and more challenging datasets, such as MIRFlickr and NUS-WIDE, this effect is largely moderated. Finally, Figure 6 shows the difference of MAP between the semantic space and their baseline, for all the five meth-

ods. We highlight that the improvement is generally higher on IAPR-TC12 and ESP-GAME, where fewer training examples are available. In particular, SVM has the largest gain followed by the simpler NNvot and TagRel. This might be because these methods suffer on rare concepts due to sample insufficiency.

4.5. Experiment 2: Performance with User Tags

We now turn our attention to the more difficult setting of noisy user tags. Instead of using expert labels, we rely on user tags as training labels and repeat the same experiments of Section 4.4. Only MIRFlickr-25k and NUS-WIDE provide user tags, therefore we report results on these datasets.

Table 5 shows the performance of the state of the art, the baselines and our approach on MIRFlickr-25k and NUS-WIDE. As previously noted, changing the features from HC to VGG16 has a strong positive impact. Comparing the methods ran in the semantic space to the baselines ran on the bare visual feature, we observe that every metric is generally improved. Fisher-Boxes [52] uses improved features with the same TagProp algorithm, as our baseline. Since our TagProp MAP is higher than FisherBoxes, this suggests that VGG16 features alone are more powerful than the combinations of VGG128 boxes. SVM is inferior to nearest neighbor techniques in terms of MAP while having comparable precision and recall. Consistently to expert labels results, 2PKNN performs poorly on NUS-WIDE. In the first phase few images per label are selected, thus reducing its power to address the high visual variability of images with frequent labels. We also note that all scores are lower than those reported with expert labels in Table 3 and Table 4. In particular SVM MAP is the most hampered. This is expected given the noise in user tags, and was also noted in previous work [67].

In Figure 7 we report the relative MAP difference of the five methods with our technique and the baselines. We observe that largest gains are obtained with 2PKNN and SVM. We believe this is due to the fact that 2PKNN and SVM have numerous learning parameters that are likely to generate complex boundaries with label noise. In contrast, the other three schemes have few or no parameters at all. This suggests that features in the semantic space have also some robustness to tag noise.

We believe that such robustness is partially due to the denoising algorithm. To confirm this, we perform an ablation study on MIRFlickr-25k with the same settings as before, except that we omit the pre-propagation step. We report in Table

Table 5: Results on the MIRFlickr-25k and NUS-WIDE datasets using **user tags**.

Methods	Visual Feat	MIRFlickr-25k			NUS-WIDE		
		MAP	Prec@5	Rec@5	MAP	Prec@5	Rec@5
State of the art:							
SVM v [67]	HC	35.4	-	-	-	-	-
SVM v+t [67]	HC	37.9	-	-	-	-	-
TagProp [67]	HC	38.4	-	-	-	-	-
FisherBoxes [52]	VGG128	**54.8**	-	-	**39.7**	-	-
Baselines:							
NNVot	VGG16	**59.3**	34.2	67.1	**43.1**	30.1	46.3
TagRel	VGG16	59.2	34.8	**68.0**	42.5	27.9	49.7
TagProp	VGG16	58.1	33.5	66.0	42.8	28.4	**50.2**
2PKNN	VGG16	51.4	35.9	67.1	41.2	**37.5**	43.7
SVM	VGG16	43.8	**40.0**	50.8	35.5	30.4	45.2
Our Approach:							
KCCA + NNvot	VGG16	**60.6**	35.4	**68.8**	**43.7**	36.3	48.0
KCCA + TagRel	VGG16	59.8	37.2	68.5	43.5	29.0	**55.1**
KCCA + TagProp	VGG16	59.7	33.6	67.4	42.9	29.3	51.3
KCCA + 2PKNN	VGG16	56.8	**42.9**	65.4	42.0	**56.9**	34.0
KCCA + SVM	VGG16	47.1	37.5	56.5	41.6	37.9	47.6

Table 6: Ablation study on the denoising method. Results are in terms of MAP.

Methods	MIRFlickr-25k		
	Baseline	KCCA - NoPreProp	KCCA
NNvot [17]	59.3	56.2	**60.6**
TagRel [18]	59.2	54.5	**59.8**
TagProp [10]	58.1	54.9	**59.7**
2PKNN [7]	51.4	42.9	**56.8**
SVM [10]	43.8	41.3	**47.1**

6 the MAP of three different cases: (i) the baseline methods (Baseline); (ii) our approach without the pre-propagation step (KCCA - NoPreProp); (iii) our full approach (KCCA). We observe that avoiding the denoising step leads to an inferior MAP, even less than the baseline case. This confirms that, in presence of excessive sparsity like that in MIRFlickr-25k, KCCA alone is unable to improve the visual features.

4.6. Experiment 3: Performance with different Textual Features

In this section, we compare the performance of the three proposed textual kernels, defined in Section 3.2.1, on expert labels: a bag-of-words linear kernel (*Linear*), a semantic ontology-based kernel (*Ontology*) and a continuous word vector kernel (*Word2Vec*). Here we perform an experiment with the same settings as experiment 1 (Section 4.4), but the Linear kernel is swapped with the Ontology or Word2Vec kernels. For the Ontology kernel we use WordNet as the underlying ontology while for Word2Vec we employ the pre-trained word vectors on news article. In Table 7, we report results on the two largest datasets MIRFlickr-25k and NUS-WIDE, but similar results were obtained on ESP-Game and IAPR-TC12.

We observe that all methods have better performance than
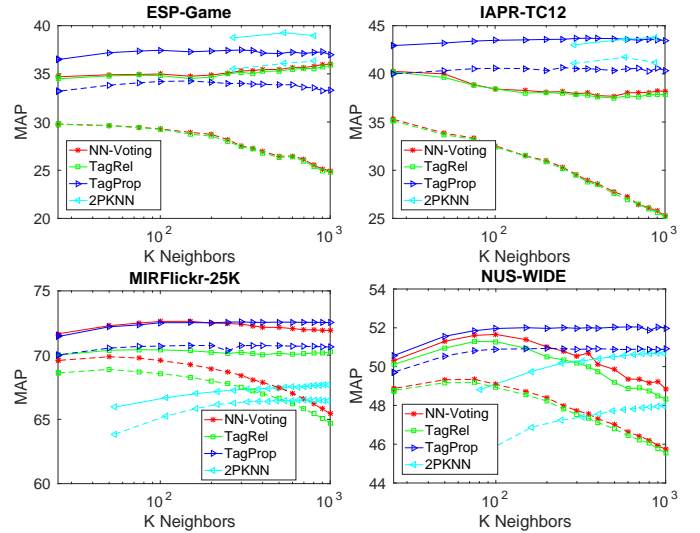
Figure 8: MAP of NN-voting, TagRel and TagProp trained with KCCA on ESPGame, IAPR-TC12, MIRFlickr-25k and NUS-WIDE varying the number of nearest neighbors. KCCA is trained with **expert labels**. Dashed lines represent baseline methods.

the baseline when using our approach, regardless of the textual kernel. Some combinations of kernels and methods favor one metric over the others, although the Linear Kernel has almost always the best MAP. Nevertheless, these slight differences in performance do not suggest a superiority of a kernel over the others. We believe that further studies on how to integrate label relations in KCCA are required, leaving the problem of choosing a better textual kernel for KCCA open.

Table 7: Results of our method with the Linear, Ontology and Word2Vec textual kernels on MIRFlickr-25k and NUS-WIDE, using **expert labels**.

| Method	Textual Kernel	MIRFlickr-25k			NUS-WIDE		
		MAP	Prec@5	Rec@5	MAP	Prec@5	Rec@5
Baselines:							
NNvot	-	69.9	44.7	69.2	49.3	39.6	44.0
TagRel	-	68.9	41.5	72.1	49.2	32.1	50.3
TagProp	-	70.8	45.5	70.1	**50.9**	**41.3**	44.6
2PKNN	-	66.5	**46.4**	70.9	48.0	39.7	52.2
SVM	-	**72.7**	38.8	**72.4**	50.2	34.6	**60.6**
Our Approach:							
KCCA + NNvot	Linear	**72.9**	46.1	73.1	**51.7**	40.2	**50.5**
KCCA + NNvot	Ontology	72.5	46.6	72.3	51.2	**46.7**	46.3
KCCA + NNvot	Word2Vec	72.3	**46.9**	**73.4**	50.6	40.8	50.1
KCCA + TagRel	Linear	70.7	45.2	72.6	**51.4**	34.4	**57.2**
KCCA + TagRel	Ontology	70.6	**47.4**	73.9	49.5	**35.9**	54.3
KCCA + TagRel	Word2Vec	**70.9**	47.2	**74.2**	49.8	34.9	57.0
KCCA + TagProp	Linear	**73.0**	44.6	**74.1**	52.2	45.2	49.2
KCCA + TagProp	Ontology	72.7	44.6	73.7	51.7	45.2	48.1
KCCA + TagProp	Word2Vec	72.9	**45.3**	73.8	51.6	40.9	**50.6**
KCCA + 2PKNN	Linear	**67.7**	47.3	74.6	**50.7**	53.0	47.0
KCCA + 2PKNN	Ontology	65.7	44.1	**76.1**	49.2	46.3	51.1
KCCA + 2PKNN	Word2Vec	66.2	44.2	75.7	48.9	47.3	**51.4**
KCCA + SVM	Linear	**73.0**	38.9	**75.0**	51.8	43.3	**48.4**
KCCA + SVM	Ontology	71.4	39.3	73.0	51.4	**44.7**	46.7
KCCA + SVM	Word2Vec	71.8	**39.5**	74.1	50.2	42.7	47.7
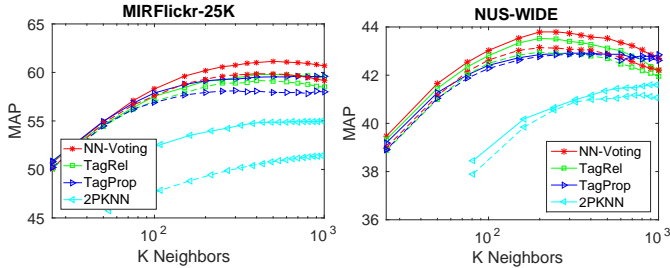
Figure 9: MAP evaluation for NN-voting, TagRel and TagProp trained with KCCA on MIRFlickr-25k and NUS-WIDE varying the number of nearest neighbors. KCCA is trained with **user tags**. Dashed lines represent baseline methods.

4.7. Experiment 4: Varying the Size of Neighborhood

Nearest neighbor methods proved to be well performing on all settings we considered. Although they are simple and do not require much training, they still depend on choosing the right number *K* of nearest neighbors. Thus, we conduct an evaluation of how *K* affect the performance for both our approach and the baselines. Since SVM does not use neighbors, we only perform this evaluation on NNvot, TagRel, TagProp and 2PKNN.

We report in Figures 8 and 9 the MAP scores when using the expert labels and the user tags, respectively. As can be seen from both figures, the KCCA variant of the nearest neighbor methods (solid lines) have systematically better MAP than baselines, for any number of neighbors used. As expected, MAP scores are lower when using user tags (Figure 9). Nevertheless, a gain is observed for each method with any number of

neighbors selected. This again confirms that features in the semantic space are better re-arranged, since images with similar semantics are closer in this space.

4.8. Experiment 5: Scaling by Subsampling the Training Set

One key issue with KCCA is that it can be onerous to scale the training over millions of images. The most expensive effort is carried out in the training phase where the projection vectors are estimated. At test time, the computational cost is negligible since it is only given by the multiplication of the features with the estimated projection vectors.

As also noted by Hardoon *et al.* [29], big training sets with large kernel matrices can lead to computational problems. Two main issues arise: *i*) high computational cost to compute the generalized eigenvalues problem, and *ii*) the memory footprint of handling large kernel matrices.

For the first issue, we compute only a reduced number of dimensions in the semantic space by using partial Gram-Schmidt orthogonalization (PGSO), i.e. we solve the generalized eigenvalues with an incomplete Cholesky factorization of the kernel matrices. This is a reasonable approximation because the projection is built up as the span of a subset of independent projections, and it reconstructs a limited amount of energy.

For the second issue, the memory footprint increases quadratically with the number of training images. In this section we explore the possibility of using a subsample of the training set to manage also this problem. To this end, we randomly select a subset of size *M* from the original training set used to train KCCA, and obtain the projections. Then we use them to project

Image	Exp Labels	NNVot		TagRel		TagProp	
		Baseline	**Our**	**Baseline**	**Our**	**Baseline**	**Our**
	desert mountain range salt sky	beach cloud **mountain** sea **sky**	cloud **desert** landscape **mountain** **sky**	beach cloud sea **sky** wave	**desert** lake landscape **mountain** **salt**	beach cloud **mountain** sea **sky**	**desert** man **mountain** **salt** **sky**
	hammock man woman	**man** room table wall **woman**	front house **man** wall **woman**	**man** room table wall **woman**	front **hammock** **man** wall **woman**	bottle **man** people table **woman**	front **hammock** **man** wall **woman**
	cap flag hair man polo portrait shirt	boy girl hat **man** sky	**cap** front **man** sky woman	boy **cap** girl **hair** hat	boy **cap** hat **man** **shirt**	boy **cap** child sky sweater	**cap** **man** **polo** **portrait** **shirt**
	man sky statue view	building front people **sky** tower	front **man** **sky** **statue** tree	building column **sky** **statue** tower	base building **sky** square **statue**	column front **man** **sky** **statue**	front **man** **sky** **statue** **view**

Figure 10: Qualitative results of the baseline methods and our proposed representation on IAPR-TC12. Labels ordered according to their relevance scores.
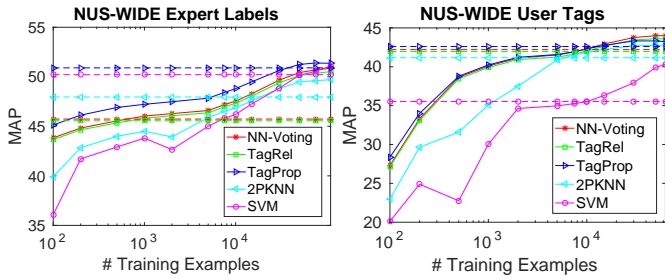

Figure 11: **Training KCCA with a subset of data.** MAP of the five methods trained with KCCA on NUS-WIDE varying the number of images used for training the projections, with expert labels (on the left) and user tags (on the right). Dashed lines represent baseline methods.

the full training set and test the methods in this approximate semantic space. We run the experiment only on NUS-WIDE since it has the highest number of images. The whole experiment is repeated with five different splits in the two settings of using expert labels or user tags. Note that this setting is different from the one used in Sect. 4.4 and Sect. 4.5 for NUS-WIDE, where we used the split provided by the authors of the dataset.

Figure 11 shows the MAP scores obtained with a subset of the training data. We report results by increasing M from 100 to the full training set size (with exponential steps). Using more training data, we expect the quality of the projections to be improved. Either with expert labels or user tags, more the training data, the better the projections obtained. We note that

a minimum quantity of data is required to obtain a performance higher than the baseline; this corresponds to the point in the figure in which the corresponding dashed and solid lines intersect each other. The specific subset of training data depends on the method and on the quality of the annotations. When expert labels are available, NNvot and TagRel obtains an improvement even with a very small amount of training images. In contrast, TagProp requires more data to gain MAP because of its rank learning phase. This means that our approach can provide some improvements even when very few labeled images are available, but more data may be needed with advanced nearest neighbors schemes. Considering the scenario of user tags, the three methods show similar performance with similar numbers of training images. This suggests that differently from expert labels, the noise in user tags is responsible for the hampered performance and more data is needed to reliably estimate good projections.

We evaluate the additional computational cost of our approach, by timing the run of KCCA on NUS-WIDE on our sub-sampling experiment. It can be noted from Fig. 12 that the overall computation is dominated by the visual kernel computation. Since we approximated the kernel matrices with GSD to a fixed rank value, the running time required to compute the KCCA projections can only increase up to a fixed maximum value, independently from the number of samples.
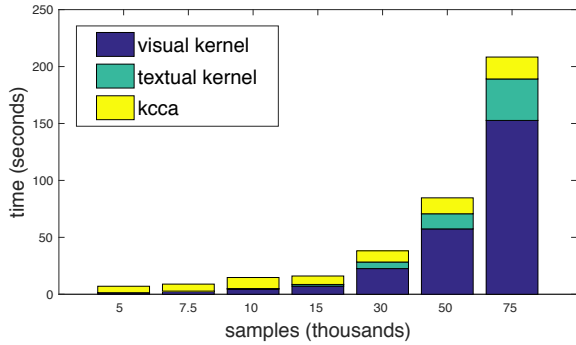
Figure 12: Timing of our approach varying the number of samples employed for learning KCCA. We report separately the time for visual kernel, textual kernel and KCCA computation. The time is dominated by the visual kernel computation.

4.9. Qualitative Analysis

Figure 10 shows four examples of annotations produced by our method on the IAPR-TC12 dataset. It can be seen that Tag-Prop and TagRel perform better for both baseline representation and the proposed semantic space. Thanks to the integration of labels into the semantic space, our technique allows nearest neighbor methods to distinguish between visually similar but semantically different images. Look for instance at the first example: a salt desert. Baseline approaches wrongly predict that this might be a "beach" image, since the salt visually resembles sand. Differently, our semantic space dismisses beach images and allows NN methods to find samples with "desert" and "salt", thus obtaining a correct image labeling.

Moreover, our method can also deal with information that was missing in the visual space. A good example is given by the second picture shown in Figure 10. This image depicts two people and an "hammock". Since the label "hammock" is not in the 1K concepts used to train the VGG16 network, similar hammock images are difficult to be retrieved for the baseline methods. In contrast, our method has integrated this missing information into the semantic space, allowing TagRel and Tag-Prop to find semantically similar images and predict the presence of the hammock correctly.

The third and fourth images demonstrate that our technique is able to bring closer images with fine-grained labels. For instance, the third image is a close-up of a person wearing several well visible clothing. Baseline methods correctly found easy concepts like "man", "cap" or "hair", while label transfer methods operating in the semantic space can also predict more specific labels such as "shirt", "polo" and "portrait". Finally, the fourth image depicts a statue portrayed from below, in contrast with the blue sky. This image is correctly annotated with the difficult labels "man" and "view" only by TagProp when trained on the semantic space.

5. Conclusion

This paper presents a novel automatic image annotation framework based on KCCA. Our work shows that it is indeed useful to integrate textual and visual information into a semantic space

that is able to preserve correlation with the respective original features. Our method does not require the textual information at test time, and it is therefore suitable for label prediction on unlabeled images. We additionally propose a label denoising algorithm that allows to exploit user tags in place of expert labels. This scenario is of extreme interest given the abundance of images with user tags that can be extracted from social media. Finally, we show that semantic projections can be learned also with a subset of the training set, making it possible to obtain some benefits even on large-scale datasets.

We report extensive experimental results on all the classic automatic image annotation datasets, as well as more recent datasets collected from Flickr. Our experiments show that label transfer in the semantic space allows consistent improvement over standard schemes that rely only on visual features. All the best performing image annotation methods have shown to be able to exploit the proposed embedding. We believe that our framework will provide a strong baseline to compare and better understand future automatic image annotation algorithms.

References

References

[1] V. Lavrenko, R. Manmatha, J. Jeon, A model for learning the semantics of pictures, in: Proc. of NIPS, 2003.

[2] F. Monay, D. Gatica-Perez, PLSA-based image auto-annotation: Constraining the latent space, in: Proc. of ACM MM, 2004.

[3] G. Carneiro, A. B. Chan, P. J. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (2007) 394–410.

[4] T. Mei, Y. Wang, X.-S. Hua, S. Gong, S. Li, Coherent image annotation by learning semantic distance, in: Proc. of CVPR, 2008.

[5] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, D. N. Metexas, Automatic image annotation using group sparsity, in: Proc. of CVPR, 2010.

[6] D. Zhang, M. M. Islam, G. Lu, A review on automatic image annotation techniques, Pattern Recognition 45 (2012) 346–362.

[7] Y. Verma, C. V. Jawahar, Image annotation using metric learning in semantic neighbourhoods, in: Proc. of ECCV, 2012.

[8] Y. Gong, Y. Jia, T. K. Leung, A. Toshev, S. Ioffe, Deep convolutional ranking for multilabel image annotation, in: Proc. of ICLR, 2014.

[9] P. Duygulu, K. Barnard, N. De Freitas, D. A. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: Proc. of ECCV, 2002.

[10] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, in: Proc. of ICCV, 2009.

[11] A.-M. Tousch, S. Herbin, J.-Y. Audibert, Semantic hierarchies for image annotation: A survey, Pattern Recognition 45 (2012) 333–345.

[12] J. McAuley, J. Leskovec, Image labeling on a network: using social-network metadata for image classification, in: Proc. of ECCV, 2012.

[13] J. Johnson, L. Ballan, L. Fei-Fei, Love thy neighbors: Image annotation by exploiting image metadata, in: Proc. of ICCV, 2015.

[14] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. M. Snoek, A. Del Bimbo, Socializing the semantic gap: A comparative survey on image tag assignment, refinement and retrieval, ACM Computing Surveys 49 (2016) 14:1–14:39.

14

[15] X.-J. Wang, L. Zhang, X. Li, W.-Y. Ma, Annotating images by mining image search results, IEEE Trans. on Pattern Analysis and Machine Intelligence 30 (2008) 1919–1932.

[16] L.-J. Li, L. Fei-Fei, OPTIMOL: Automatic online picture collection via incremental model learning, International Journal of Computer Vision 88 (2010) 147–168.

[17] A. Makadia, V. Pavlovic, S. Kumar, A new baseline for image annotation, in: Proc. of ECCV, 2008.

[18] X. Li, C. G. M. Snoek, M. Worring, Learning social tag relevance by neighbor voting, IEEE Transactions on Multimedia 11 (2009) 1310–1322.

[19] A. Znaidia, , H. Le Borgne, C. Hudelot, Tag completion based on belief theory and neighbor voting, in: Proc. of ACM ICMR, 2013.

[20] L. Ballan, M. Bertini, T. Uricchio, A. Del Bimbo, Data-driven approaches for social image and video tagging, Multimedia Tools and Applications 74 (2015) 1443–1468.

[21] X. Qi, Y. Han, Incorporating multiple svms for automatic image annotation, Pattern Recognition 40 (2007) 728–741.

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

[23] H. Sahbi, X. Li, Context dependent svms for interconnected image network annotation, in: Proc. of ACM MM, 2010.

[24] Y. Verma, C. V. Jawahar, Exploring svm for image annotation in presence of confusing labels, in: Proc. of BMVC, 2013.

[25] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (2000) 1349–1380.

[26] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proc. of NIPS, 2012.

[27] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. of ICLR, 2015.

[28] O. Russakovsky et al., ImageNet Large Scale Visual Recognition Challenge, Int'l Journal of Computer Vision 115 (2015) 211–252.

[29] D. R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Computation 16 (2004) 2639–2664.

[30] N. Srivastava, R. Salakhutdinov, Multimodal learning with deep boltzmann machines, in: Proc. of NIPS, 2012.

[31] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, in: Proc. of ACM MM, 2010.

[32] S. J. Hwang, K. Grauman, Learning the relative importance of objects from tagged images for retrieval and cross-modal search, Int'l Journal of Computer Vision 100 (2012) 134–153.

[33] Y. Gong, Q. Ke, M. Isard, S. Lazebnik, A multi-view embedding space for internet images, tags, and their semantics, International Journal of Computer Vision 106 (2014) 210–233.

[34] M. Grubinger, P. Clough, H. Muller, T. Deselaers, The IAPR TC-12 benchmark: a new evaluation resource for visual information systems, in: Proc. of LREC Workshops, 2006.

[35] L. von Ahn, L. Dabbish, Labeling images with a computer game, in: Proc. of CHI, 2004.

[36] M. J. Huiskes, M. S. Lew, The MIR Flickr retrieval evaluation, in: Proc. of ACM MIR, 2008.

[37] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y.-T. Zheng, NUS-WIDE: A Real-World Web Image Database from National University of Singapore, in: Proc. of ACM CIVR, 2009.

[38] L. Ballan, T. Uricchio, L. Seidenari, A. Del Bimbo, A cross-media model for automatic image annotation, in: Proc. of ACM ICMR, 2014.

[39] S. L. Feng, R. Manmatha, V. Lavrenko, Multiple bernoulli relevance models for image and video annotation, in: Proc. of CVPR, 2004.

[40] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, M. I. Jordan, Matching words and pictures, Journal of Machine Learning Research 3 (2003) 1107–1135.

[41] Y. Xiang, X. Zhou, T.-S. Chua, C.-W. Ngo, A revisit of generative model for automatic image annotation using markov random fields, in: Proc. of CVPR, 2009.

[42] H. Izadinia, B. C. Russell, A. Farhadi, M. D. Hoffman, A. Hertzmann, Deep classifiers from image tags in the wild, in: Proc. of ACM MM Workshops, 2015.

[43] J. Liu, M. Li, Q. Liu, H. Lu, S. Ma, Image annotation via graph learning, Pattern Recognition 42 (2009) 218–228.

[44] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, R. Jain, Image annotation by knn-sparse graph-based label propagation over noisily tagged web images, ACM Trans. on Intelligent Systems and Tech. 2 (2011).

[45] X. Zhu, W. Nejdl, M. Georgescu, An adaptive teleportation random walk model for learning social tag relevance, in: SIGIR, 2014.

[46] F. Su, L. Xue, Graph learning on k nearest neighbours for automatic image annotation, in: Proc. of ACM ICMR, 2015.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proc. of CVPR, 2009.

[48] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. of CVPR, 2014.

[49] A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: An astounding baseline for visual recognition, in: Proc. of CVPR Workshops, 2014.

[50] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, in: Proc. of ECCV, 2014.

[51] D. Yoo, S. Park, J.-Y. Lee, I. Kweon, Multi-scale pyramid pooling for deep convolutional representation, in: Proc. of CVPR Workshops, 2015.

[52] T. Uricchio, M. Bertini, L. Seidenari, A. Del Bimbo, Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging, in: Proc. of ICCV Workshops, 2015.

[53] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: A deep visual-semantic embedding model, in: Proc. of NIPS, 2013.

[54] M. Guillaumin, J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: Proc. of CVPR, 2010.

[55] A. Habibian, T. Mensink, C. G. M. Snoek, Discovering semantic vocabularies for cross-media retrieval, in: Proc. of ACM ICMR, 2015.

[56] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Transactions on Cybernetics in press (2017).

[57] X. Yang, W. Liu, D. Tao, J. Cheng, Canonical correlation analysis networks for two-view image recognition, Information Sciences 385 (2017) 338–352.

[58] C. Xu, D. Tao, C. Xu, Multi-view intact space learning, IEEE Trans. on Pattern Analysis and Machine Intelligence 37 (2015) 2531–2544.

[59] V. N. Murthy, S. Maji, R. Manmatha, Automatic image annotation using deep learning representations, in: Proc. of ACM ICMR, 2015.

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

[61] Y. Cho, L. K. Saul, Kernel methods for deep learning, in: Proc. of NIPS, 2009.

[62] H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, G. Mori, Learning structured inference neural networks with label relations, in: Proc. of CVPR, 2016.

[63] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis, Cambridge university press, 2004.

[64] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proc. of NAACL-HLT, 2013.

[65] D. Lin, et al., An information-theoretic definition of similarity., in: Proc. of ICML, 1998, pp. 296–304.

[66] L. van Der Maaten, Accelerating t-SNE using tree-based algorithms, Journal of Machine Learning Research 15 (2014) 3221–3245.

[67] J. Verbeek, M. Guillaumin, T. Mensink, C. Schmid, Image annotation with tagprop on the mirflickr set, in: Proc. of ACM MIR, 2010.

[68] H. Fu, Q. Zhang, G. Qiu, Random forest for image annotation, in: Proc. of ECCV, 2012.

[69] S. Moran, V. Lavrenko, Sparse kernel learning for image annotation, in: Proc. of ACM ICMR, 2014.

[70] Z. Ren, H. Jin, Z. Lin, C. Fang, A. Yuille, Multi-instance visual-semantic embedding, arXiv:1512.06963 (2015).