Automatic Image Annotation via Label Transfer in the Semantic Space

Tiberio Uricchio, Lamberto Ballan, Lorenzo Seidenari, and Alberto Del Bimbo, Member, IEEE

Abstract—While most automatic image annotation methods rely solely on visual features, we consider integrating additional information into an unified embedding comprised of visual and textual information. We propose an approach based on Kernel Canonical Correlation Analysis, which builds a latent semantic space where correlation of visual and textual features are well preserved into a semantic embedding. Images in the semantic space have reduced semantic gap and thus they are likely to give better annotation performance. 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 evaluate our framework on four popular datasets. Our results show that our KCCA-based approach can be applied to several state-of-the-art label transfer methods to obtain significant improvements. In particular, nearest neighbor methods for label transfer get the most benefit and enable our approach to scale on never seen labels at training time. 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.

Index Terms—Automatic image annotation, tag assignment, image tagging, canonical correlation, nearest neighbors.

I. 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], [9]. 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 [2], [4], [10], [11], while recently several approaches use weak labels such as user-generated tags in social networks [12], [13], [14], [15], [16] or query terms in search engines [17], [18].

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 image annotation [11], [19], [20], [21], [22], [23]. The main reason is that these methods have the ability to adapt to complex patterns of data 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 to the test image. In contrast, discriminative models such as support vector machines [24], [25], [26] or fully supervised end-to-end deep networks [9], require to define in advance the vocabulary of labels. Thus they do not scale well in a large-scale scenario in which you may have thousands of labels that may also change or increase over time.

However, 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 [27]. This happens since 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 [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 that label frequencies are usually unbalanced. Modern methods address this issue by introducing label penalties and metric learning [8], [11], [20].

However, 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) [31], [32]. 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 referred 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, images associated with the same label (e.g. “sunset”) will be closer in the semantic space than in the original visual space (see for example the images with the red border).

A. 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 classical 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 also 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, including prior uses of CCA and KCCA [33], [34], [35], [36]. 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 neighbors models. Interestingly enough, in our approach textual information is only needed at training time, thus allowing to predict labels also for unlabeled images. In this paper, we build on our previous contribution on cross-modal image representation [37] and improve in many ways. First, the coverage of related work and the description of our method have been significantly extended. Second, we provide more insight on the semantic space learning and present a unified description of different label transfer algorithms.

3) We provide extensive experimental validations. The method is tested on medium and large scale datasets, i.e. IAPR-TC12 [38], ESP-GAME [39], MIRFlickr-25k [40] and NUS-WIDE [41]. 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.

II. RELATED WORK

A. Automatic Image Annotation: Ideas and Main Trends

Automatic image annotation is a long standing area of research in computer vision, multimedia and information retrieval [16]. Early works often used mixture models to define a joint distribution over image features and labels [1], [4], [42], [43]. 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 [2], [44], [45]. 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) have also been used extensively [24], [25], [26]. 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 [8], [11], [19], [46]. Their underlying intuition is that similar images are likely to share common labels. All 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 [8], [11] or graph learning [47], [48], [49], [50], in order to differently weight rare and common labels or to capture the semantic correlation between labels. However, 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.

B. 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. [28] demonstrated unprecedented improvement in large-scale image classification on ImageNet [51] 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 [32]. Deeper architectures have been recently proposed, showing further gain in image classification accuracy (e.g. [29]).

Another interesting property of these architectures is that they have the ability to learn powerful representations that can be transferred and reused in many other tasks, such as attribute prediction and image retrieval [53]. Convolutional neural networks have been also recently applied to automatic image annotation [9], [54], [55], showing significant improvement in terms of precision and recall. On top of these powerful features, recent works proposed to learn embeddings in order to improve feature generalization. VLAD encoding is used in [56] to pool multi-scale CNN features computed on different windows, while Yoo et al. [57] proposed to apply Fisher Vector encoding to dense multi-scale CNN activations. This has been also improved in [58] by applying Fisher Vector to sparse boxes, selected by objectness or random selection. However, all these approaches only focus on the visual modality.

**C. Cross-media and Multimodal Representations**

We note that previous use of CCA and its variants exists, particularly for the task of cross-modal image retrieval [32], [34], [35], [36]. Hardoon et al. were the first to apply KCCA to image retrieval with textual query [32]. Successively, Rasiwasia et al. proposed to employ LDA and CCA to perform cross modal retrieval on text and images [33] obtaining improved results on single modalities. In [34], 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. A three-way CCA is proposed in [35] to address the limited expressiveness of CCA. They show that adding a third view representing categories or clustered labels, can improve retrieval performance. Habibian et al. proposed a data driven approach to discover vocabulary concepts and align them to textual information without any human annotation effort [36]. They show that their approach reaches improved performance versus predefined vocabularies on both captioned images and videos. However, most of these extensions deal with the problem of using unpaired images and labels as training data. In [59], a scalable approximate extension of CCA is proposed where multiple pairings across the two modalities are automatically discovered. Similarly, Kang et al. [60] perform text to image retrieval with an approach based on partial least squares, where unpaired data can be added to improve the common space. In [61], the authors propose to go beyond simple textual annotation and learn localized correlations by taking advantage of a predefined semantic hierarchy. In contrast, we do not consider additional information such as categories, semantic hierarchies or unpaired labels.

Differently from prior work, we tackle the specific problem of image annotation. For this task, only visual features are available at test time. Thus, our approach exploit 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.

**III. 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 [31]. This algorithm strives to provide a common representation for two views of the same data. Similarly to [32], [34], 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 method 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.

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.

A. Visual and Textual Views

We now proceed in detailing the visual and textual representa-
tion, how KCCA is used to build the semantic space, and
finally we describe our label transfer procedures.

1) Visual Features: We use a deep convolutional neural
network pre-trained on ImageNet [51] with the VGG-Net
architecture presented in [29] (using 16 layers). 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 [52, 53].

Given an image $I_i$, we first warp it to $224 \times 224$ 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:

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

where $J_n$ is defined according to the selected order of the
kernel. Following [62], we set $n = 2$ which gives us:

$$J_2(\theta) = 3\sin \theta \cos \theta + (\pi - \theta)(1 + 2\cos^2 \theta)$$  \hspace{1cm} (2)

where $\theta$ 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%).

2) Textual Features: Depending on how labels are gen-
erated, i.e. expert labels or user-generated tags, we should
use different approaches. While expert labels can be naturally
trusted, user-generated tags are very noisy and require a more
robust representation.

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
$\phi^T(I_i) = [w_{i,1}, \ldots, w_{i,D}]$, where $w_k$ is 0 or 1 if that image
has been annotated with the corresponding $k$-th label. 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:

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

The basic idea is that we are considering the cooccurrences
of labels in order to measure the similarity between two images.

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 goal is to obtain a cleaner tag feature-
vector $\hat{\phi}^T(I_i) = [\hat{w}_{i,1}, \ldots, \hat{w}_{i,D}]$ and then compute the final
textual kernel $K^T$. We start from the same representation $\phi^T$
used for the expert labels, but this time using tags. For each
image $I_i$ we consider the $R = 100$ most similar images,
according to the visual kernel $K^V$, and compute the new tag vector:

$$\hat{\phi}^T(I_i) = \frac{\sum_{k=1}^R x_k \phi^T(I_k)}{\sum_{k=1}^R x_k}$$  \hspace{1cm} (4)

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 $\sigma$
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-$\chi^2$
kernel:

$$K^T(\hat{\phi}^T(I_i), \hat{\phi}^T(I_j)) = \exp\left(-\frac{1}{2C} \sum_{k=1}^D \frac{(\hat{w}_{i,k} - \hat{w}_{j,k})^2}{(\hat{w}_{i,k} + \hat{w}_{j,k})}\right)$$  \hspace{1cm} (5)

where $C$ is set to the mean of the $\chi^2$ distances. This kernel
is designed to match histograms and indeed showed good
performance on bag-of-words representations [63].

B. 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 $z^V_T$ and $z^T_T$ that maximize
the correlation of the projections of $\phi^V$ onto $z^V_T$ and $\phi^T$ onto
$z^T_T$. This is expressed as:

$$z^V_T, z^T_T = \arg\max_{z^V_T, z^T_T} \frac{E[(\phi^T, z^V_T) \langle \phi^T, z^T_T \rangle]}{\sqrt{E[(\phi^V, z^V_T)^2]E[(\phi^T, z^T_T)^2]}}$$

$$= \arg\max_{z^V_T, z^T_T} \frac{z^V_T C_{vt} z^T_T}{\sqrt{z^V_T C_{vv} z^V_T z^T_T C_{tt} z^T_T}}$$  \hspace{1cm} (6)

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

However, the CCA algorithm can only model linear relationships.
KCCA has been introduced to allow projecting the data
into a higher-dimensional feature space by using the kernel
trick [31]. Thus, the problem is now to search for solutions of $z^V_T$ and $z^T_T$ that lie in the span of the $N$ training instances
$\phi^V(I_i)$ and $\phi^T(I_i)$:

$$z^V_T = \sum_{i=1}^N \alpha_i \phi^V(I_i),$$

$$z^T_T = \sum_{i=1}^N \beta_i \phi^T(I_i).$$  \hspace{1cm} (7)
The objective of KCCA is thus to identify the weights $\alpha, \beta \in \mathbb{R}^N$ that maximize:

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

where $K^V$ and $K^T$ denote the $N \times N$ kernel matrices over a sample of $N$ pairs. As shown by Hardoon [31], learning should be regularized in order to avoid trivial solutions. Hence, we penalize the norms of the projection vectors and obtain the generalized eigenvalue problem:

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

where $\kappa \in [0, 1]$. The top $M$ eigenvectors of this problem yield bases $A = [\alpha_1 \ldots \alpha_M]$ and $B = [\beta_1 \ldots \beta_M]$ that we use to compute the semantic projections of training and test kernels. For each pair $(\alpha_j, \beta_j)$ of the given bases, the corresponding eigenvalue $r_j$ measures the correlation between projected input pairs. Higher $r_j$ are 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:

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

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

In Figure 3 we show t-SNE embeddings [64] 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.

C. Label Transfer

The constructed semantic space assure 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 (NN) 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.

Following this key idea, we have investigated and applied four NN-voting algorithms to our semantic space in order to automatically annotate images. Nevertheless, we expect that other general class of learning algorithms may take advantage of our 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 [16], as defined in the following.

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 their frequency in the retrieval set. Thus, the relevance function is defined as:

$$f_{KN} \left( I, t \right) := k_t$$

(11)

where $k_t$ is the number of images labeled as $t$ in the neighborhood of $I$. If $K$ is fixed to a very small number (e.g. $K = 2$) this approach is similar to the simple algorithm proposed by Makadia et al. [19].

2) Tag Relevance: Li et al. [20] 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: $S$.

$$f_{TagVote} \left( I, t \right) := k_t - k_t \frac{n_t}{|S|}$$

(12)

where $n_t$ is the number of images labeled with $t$ in the entire collection $S$.

3) TagProp: Guillaumin et al. [11] 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 maximize the likelihood of a probabilistic model. The method can learn rank-based or distance-based weights:

$$f_{TagProp} \left( I, t \right) := \sum_{j=1}^{k} \pi_j \cdot \mathbb{I}(I_j, t)$$

(13)

where $\mathbb{I}$ is the indicator function that returns 1 if $I_j$ is labeled with $t$, and 0 otherwise; $\pi_j$ 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 hamper the frequent ones.

4) 2PKNN: Verma and Jawahar [8] formulated the problem as a probabilistic framework and proposed a two-phases 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 neighborhoods $\mathcal{N}(I)$ composed of the nearest $M$ training images annotated with each $t$ in $D$. On the second phase, the balanced neighborhood is used to estimate the tag relevance of $t$ to $I$:

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

(14)

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 [8] 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.

5) SVM: For each label, a binary linear SVM classifier is trained using the L2-regularized least square regression, similarly to [65]. 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) as the optimizer [66]. The relevance function is thus:

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

(15)

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

IV. EXPERIMENTS

A. 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 nowadays 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 I while detailed information are reported below.

**TABLE I**

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

**ESP-GAME.** The ESP-GAME dataset [39] 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 [11] 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 [38] for cross-language information retrieval. It is a collection of 19,805 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 [11]. 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 candidate.

**MIRFlickr-25K.** The MIRFlickr-25K dataset [40] has been introduced to evaluate keyword-based image retrieval. It contains 25,000 images downloaded from Flickr. For each image, the presence of 18 labels are available as expert labels as well as user tags. They are annotated with an average of respectively 2.78 and 0.26 labels. Note that tags 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 [65].

**NUS-WIDE.** The NUS-WIDE dataset [41] 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 and 1.04 labels, respectively. NUS-WIDE is one of the largest dataset of images collected from social media. The scarcity 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 [9], 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.

### B. Evaluation Measures

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 label. 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 evaluate 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+ [10], [19]. For completeness we report MAP on these two datasets although, as can be seen in Table II, few previous work report also this metric.

For MIRFlickr, considering that annotated labels are used to perform image retrieval, the few existing works report only the MAP [65]. 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 ESPGAME, we report results with $n = 3$ labels, as in [9], [15].

### C. 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. [31]. In all the experiments we have empirically fixed $\kappa = 0.5$ (see Eq. 2) 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. 10 is 4,096. The distance between two images is defined as the cosine distance between $\psi$ 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 normalized visual kernel described in Eq. 1.

### D. Experiment 1: Measuring 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 [I], [II], [III] and [IV] we report the performance of the state

\begin{table}[h]
\centering
\caption{RESULTS OF OUR METHOD COMPARED TO THE STATE OF THE ART ON THE DATASET MIRFlickr-25K, USING EXPERT LABELS.}
\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Methods} & \textbf{VisualFeat} & \textbf{MAP} & \textbf{Prec@5} & \textbf{Rec@5} \\
\hline
\textbf{State of the art:} & & & & \\
SVM [65] & HC & 52.3 & - & - \\
TagProp [65] & HC & 46.5 & - & - \\
\hline
\textbf{Baselines:} & & & & \\
NNvot & VGG16 & 69.9 & 44.7 & 69.2 \\
TagRel & VGG16 & 68.7 & 41.5 & 72.1 \\
TagProp & VGG16 & 70.7 & 45.5 & 70.1 \\
2PKNN & VGG16 & 64.3 & 41.7 & 72.2 \\
SVM & VGG16 & 72.7 & 38.8 & 72.4 \\
\hline
\textbf{Our Approach:} & & & & \\
KCCA + NNvot & VGG16 & 72.9 & 46.1 & 73.1 \\
KCCA + TagRel & VGG16 & 70.4 & 45.2 & 72.6 \\
KCCA + TagProp & VGG16 & 72.9 & 44.6 & 74.1 \\
KCCA + 2PKNN & VGG16 & 66.7 & 47.8 & 75.7 \\
KCCA + SVM & VGG16 & 73.0 & 38.9 & 75.0 \\
\hline
\end{tabular}
\end{table}
TABLE II
RESULTS OF OUR METHOD COMPARED TO THE STATE OF THE ART ON THE DATASETS IAPR-TC12 AND ESP-GAME, USING EXPERT LABELS.

| Method | Visual Feat | IAPR-TC12 | ESP-GAME |
|--------|-------------|-----------|----------|
| **State of the art:** | | | |
| MBRM [43] | HC | 24 | 23 | 223 |
| JEC-15 [19] | HC | 29 | 19 | 211 |
| TagProp [11] | HC | 40 | 35 | 266 |
| GS [6] | HC | 32 | 29 | 252 |
| RF [67] | HC | 44 | 31 | 253 |
| 2PKNN-ML [8] | HC | 54 | 37 | 278 |
| KSVM-VT [26] | HC | 47 | 29 | 268 |
| SKL-CRM [68] | HC | 47 | 32 | 274 |
| CCA-KNN [54] | VGG16 | 41 | 34 | 273 |

**Baselines:**
- NNvot VGG16: 36, 39, 29, 239
- TagRel VGG16: 44.0, 44.4, 30.8
- TagProp VGG16: 50.9, 41.3, 44.6
- SVM VGG16: 50.2, 34.6, 60.6

**Our Approach:**
- KCCA + NNvot VGG16: 50.7, 40.2, 50.5
- KCCA + TagRel VGG16: 50.3, 34.4, 57.2
- KCCA + TagProp VGG16: 51.6, 41.0, 50.2
- KCCA + 2PKNN VGG16: 35.2, 37.2, 43.8
- KCCA + SVM VGG16: 51.1, 43.3, 48.4

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 III 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) [11] while we use the more recent VGG16 CNN activations, the same as [54]. By 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 [19] which use a combination of HC features. Our baseline TagProp has a slight inferior performance to that reported in [11], probably due to the lower number of learnable parameters, having only one single feature versus 15. Comparing our approach versus the baselines, we observe that all metrics consistently report higher values with our approach on every configuration. Interestingly, all our methods except KCCA + 2PKNN show better performance. 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 [8] that is additionally applying metric learning.

Table IV shows our results on the MIRFlickr-25k dataset. 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 [65], MAP increases from 52.3 to 72.7 and from 46.5 to 70.7, respectively. This is consistent with recent literature that suggests CNN activations are way more powerful than handcrafted features. Applying our technique to the five methods results in a generalized improvement of all metrics, especially on the four nearest neighbors 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 IV the results of the comparison on the large-scale NUS-WIDE dataset. Previous work used BLVC (Caffe reference model) features [20] while we use the VGG16. From our preliminary experiments and as also reported in previous work [13], no outstanding differences were found. Moreover Gong et al. [9] attempted to learn directly from RGB images, obtaining an inferior performance with respect to pre-trained features on ImageNet [9], [15]. With the only exception of 2PKNN, our technique gives significant improvement over the state of the art. A higher score of Rec@5 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 approach 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 [8], [11] take more advantage from using 15 different but weaker features than 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 5 shows the difference of MAP between the semantic space and their baseline, for all the five methods. 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.

E. Experiment 2: Measuring 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 IV-D. Only MIRFlickr-25k and NUS-WIDE provide user tags, therefore we report results on these datasets.

Table V shows the performance of the state of the art, the baselines and our approach on MIRFlickr-25k. 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 improved. Our method is thus robust to noise and permits to obtain increased performance.

Regarding NUS-WIDE, the results are reported in Table VI. We compare our approach with the baselines and FisherBoxes [58]. FisherBoxes [58] 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. Similarly to MIRFlickr-25k, 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 this dataset due to the high unbalanced distribution of images across labels. We also
with any number of neighbors selected. This again confirms (Figure 8). Nevertheless, a gain is observed for each method three considered nearest neighbor methods have systematically seen from both figures, the KCCA variants (solid lines) of the this parameter, we only perform this evaluation on NNvot, our approach and the baselines. Since SVM does not use an evaluation of how $K$ (user tags)

Fig. 6. 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.

Fig. 8. MAP evaluation for all methods 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.

note that all scores are lower than those reported with expert labels in Table III and Table IV. In particular SVM MAP is the most hampered. This is expected given the noise in user tags, and was also noted in previous work [65].

In Figure 6 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, which are methods that do not have a denoising component and are not designed to be resilient to noise. This suggests that features in the semantic space have also some robustness to tag noise.

F. Experiment 3: Measuring Performance by Varying the Size of Neighborhood

Nearest neighbors methods proved to be well performing on all settings we considered. Although they are simple and do not require any 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, and 2PKNN does not have a direct way to control this parameter, we only perform this evaluation on NNvot, TagRel and TagProp.

We report in Figures 7 and 8 the MAP scores when using the expert labels and the user tags, respectively. As can be seen from both figures, the KCCA variants (solid lines) of the three considered nearest neighbor methods have systematically better MAP than baselines, for any amount of neighbors used. As expected, MAP scores are lower when using user tags (Figure 8). 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 semantic are closer in this space.

G. Experiment 4: 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. [51], 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 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. For the sake of simplicity, we only consider NN-Vot, TagRel and TagProp. The whole experiment is repeated with five different splits in the two settings of using expert labels or user tags.

Figure 10 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, the more training data, the better the projection obtained. We note that a minimum quantity of data is required to obtain a performance higher than the baseline; this corresponds to
**Fig. 7.** MAP of the four methods 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.

| Image       | Exp Labels | NNVot     | TagRel   | TagProp |
|-------------|------------|-----------|----------|---------|
| desert      | beach      | cloud     | beach    | desert  |
| mountain    | cloud      | desert    | cloud    | lake    |
| range       | mountain   | landscape | sea      | landscape |
| salt        | sea        | mountain  | sky      | mountain |
| sky         | sky        | sky       | wave     | salt    |

**Fig. 9.** Qualitative results of the baseline methods and our proposed representation on IAPR-TC12. Labels ordered according to their relevance scores.

**H. Qualitative Analysis**

Figure 9 shows four examples of annotations produced by our method on the IAPR-TC12 dataset. It can be seen that TagProp 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 9. 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 TagProp 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.

V. CONCLUSION

This paper presents a novel automatic image annotation approach 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. This representation allows nearest-neighbor approaches to retrieve better neighbors that in turns enable better image annotation performance. Our method does not require the textual information at test time, and it is therefore suitable for label prediction on unannotated images.

We additionally propose a label denoising algorithm that allows our approach to exploit user tags in place of expert labels. This scenario is of extreme interest given the abundant amount of images with user tags that can be extracted from social media. Experiments on large scale settings show consistent improvements over standard approaches that rely only on visual features. Moreover, all best performing image annotation methods are able to exploit the proposed embedding.

Finally, we show that semantic projections can be learned also with a subset of the training set, making it feasible to apply our algorithm to large-scale realistic datasets.

Acknowledgments. This research was supported by “THE SOCIAL MUSEUM AND SMART TOURISM”, MIUR project no. CTN01_00034_23154_SMST. L. Ballan was supported by the EC’s FP7 under the grant agreement No. 623930 (Marie Curie IOF).

REFERENCES

[1] J. Jeon, V. Lavrenko, and R. Mannath, “Automatic image annotation and retrieval using cross-media relevance models,” in Proc. of SIGIR, 2003.
[2] F. Monay and D. Gatica-Perez, “PLSA-based image auto-annotation: Constraining the latent space,” in Proc. of ACM MM, 2004.
[3] A. Yavinsky, E. Schofield, and S. Rüfer, “Automated image annotation using global features and robust nonparametric density estimation,” in Proc. of ACM CIVR, 2005.
[4] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 29, no. 3, pp. 394–410, 2007.
[5] T. Mei, Y. Wang, X-S. Hua, S. Gong, and S. Li, “Coherent image annotation by learning semantic distance,” in Proc. of CVPR, 2008.
[6] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. N. Metaxas, “Automatic image annotation using group sparsity,” in Proc. of CVPR, 2010.
[7] F. Wu, Y. Han, Q. Tian, and Y. Zhuang, “Multi-label boosting for image annotation by structural grouping sparsity,” in Proc. of ACM MM, 2010.
[8] Y. Verma and C. V. Jawahar, “Image annotation using metric learning in semantic neighborhoods,” in Proc. of ECCV, 2012.
[9] Y. Gong, Y. Jia, T. K. Leung, A. Toshev, and S. Ioffe, “Deep convolutional ranking for multilabel image annotation,” in Proc. of ICLR, 2014.
[10] P. Duygulu, K. Barnard, N. De Freitas, and D. A. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” in Proc. of ECCV, 2002.
[11] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation,” in Proc. of ICCV, 2009.
[12] J. McAuley and J. Leskovec, “Image labeling on a network: using social-network metadata for image classification,” in Proc. of ECCV, 2012.
[13] Y.-H. Kuo, W.-H. Cheng, H.-T. Lin, and W. H. Hsu, “Unsupervised semantic feature discovery for image object retrieval and tag refinement,” IEEE Transactions on Multimedia, vol. 14, no. 4, pp. 1079–1100, 2012.
[14] M. Chen, A. Zheng, and K. Weinberger, “Fast image tagging,” in Proc. of ICML, 2013.
[15] J. Johnson, L. Ballan, and L. Fei-Fei, “Love thy neighbors: Image annotation by exploiting image metadata,” in Proc. of ICCV, 2015.
[16] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. M. Snoek, and A. Del Bimbo, “Socializing the semantic gap: A comparative survey on image tag assignment, refinement and retrieval,” ACM Computing Surveys, vol. in press, 2016.
[17] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma, “Annotating images by mining image search results,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 30, no. 11, pp. 1919–1932, 2008.
[18] L.-J. Li and L. Fei-Fei, “OPTIMOL: Automatic online picture collection via incremental model learning,” International Journal of Computer Vision, vol. 88, no. 2, pp. 147–168, 2010.
[19] A. Makadia, V. Pavlovic, and S. Kumar, “A new baseline for image annotation,” in Proc. of ECCV, 2008.
[20] X. Li, C. G. M. Snoek, and M. Worring, “Learning social tag relevance by neighbor voting,” IEEE Transactions on Multimedia, vol. 11, no. 7, pp. 1310–1322, 2009.
[21] T. Uricchio, L. Ballan, M. Bertini, and A. Del Bimbo, “An evaluation of nearest-neighbor methods for tag refinement,” in Proc. of IEEE ICME, 2013.
[22] A. Znaidia, H. Le Borgne, and C. Hudelot, “Tag completion based on belief theory and neighbor voting,” in Proc. of ACM ICMR, 2013.
[23] L. Ballan, M. Bertini, T. Uricchio, and A. Del Bimbo, “Data-driven approaches for social image and video tagging,” Multimedia Tools and Applications, vol. 74, no. 4, pp. 1443–1468, 2015.
[24] D. Grangier and S. Bengio, “A discriminative kernel-based approach to rank images from text queries,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 30, no. 8, pp. 1371–1384, 2008.
[25] H. Sahbi and X. Li, “Context dependent svns for interconnected image network annotation,” in Proc. of ACM MM, 2010.
[26] Y. Verma and C. V. Jawahar, “Exploring svm for image annotation in presence of confusing labels,” in Proc. of BMVC, 2013.
[27] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1349–1380, 2000.
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. of NIPS, 2012.
[29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. of ICLR, 2015.
[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” Int’l Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.
[31] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” Neural Computation, vol. 16, no. 12, pp. 2639–2664, 2004.
[32] D. R. Hardoon and J. Shawe-Taylor, “KCCA for different level precision in content-based image retrieval,” in Proc. of IEEE CBMI, 2003.
