Toward Real-Time Image Annotation Using Marginalized Coupled Dictionary Learning

Seyed Mahdi Roostaiyan · Mohammad Mehdi Hosseini · Mahya Mohammadi Kashani · S. Hamid Amiri

Abstract In most image retrieval systems, images include various high-level semantics, called tags or annotations. Virtually all the state-of-the-art image annotation methods that handle imbalanced labeling are search-based techniques which are time-consuming. In this paper, a novel coupled dictionary learning approach is proposed to learn a limited number of visual prototypes and their corresponding semantics simultaneously. This approach leads to a real-time image annotation procedure. Another contribution of this paper is that utilizes a marginalized loss function instead of the squared loss function that is inappropriate for image annotation with imbalanced labels. We have employed a marginalized loss function in our method to leverage a simple and effective method of prototype updating. Meanwhile, we have introduced ℓ_1 regularization on semantic prototypes to preserve the sparse and imbalanced nature of labels in learned semantic prototypes. Finally, comprehensive experimental results on various datasets demonstrate the efficiency of the pro-

posed method for image annotation tasks in terms of accuracy and time. The reference implementation is publicly available on github.

Authors contributed equally on this research.

Seyed Mahdi Roostaiyan
Department of Computer Engineering, Sharif University of Technology, Azadi Ave., Theran, Iran
E-mail: mahdiroostaiyan@ce.sharif.edu

Mohammad Mehdi Hosseini
Department of Computer Engineering, Sharif University of Technology, Azadi Ave., Theran, Iran
E-mail: mohammadmehdi.hosseini@du.edu

Mahya Mohammadi Kashani
Department of Computer Engineering, Shahid Rajaee Teacher Training University, Lavizan, Theran, Iran
E-mail: mahya.mkashani@sru.ac.ir

S. Hamid Amiri
Department of Computer Engineering, Shahid Rajaee Teacher Training University, Lavizan, Theran, Iran
E-mail: s.hamidamiri@sru.ac.ir

1 Introduction

Image annotation deals with the problem in which each instance is represented by a single example that is associated with multiple labels. The main challenges of the image annotation problem, which distinguishes it from standard multi-label problems, are "class-imbalance" (extreme variations in the frequency of different labels), "incomplete-labeling" (many images are not annotated with all the relevant labels of the vocabulary) [37], and "diverse-labeling" (predicted labels must be qialified representative of the image and diverse from each other, to reduce redundancy) [35]. Since the early approaches of image annotation (e.g., generative-based models [25]) did not consider these challenges, they have low performance in the annotation task.

Similarity-based strategy [8, 37] is arguably the most intuitive solution that annotates a given image based on its nearest neighbors, which is an effective approach for image annotation tasks concerning the aforementioned challenges. Considering the potential weakness of this strategy, which ignores correlations between labels, various approaches such as metric learning [1, 37] and sparse multi-view multi-label learning [44] focused on both the visual contents of images and their corresponding labels, simultaneously. Metric learning [37] aims to learn an improved similarity measure to enhance the efficacy of nearest-neighbor based approaches. However, it is a time-consuming task to compare a
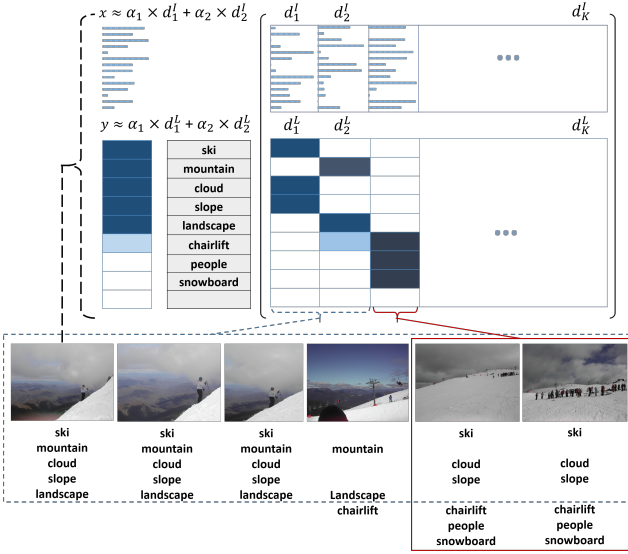
Fig. 1 Abstract of data summarization using MCDL. Example images have been taken from IAPRTC-12 dataset. There are shared visual and semantic contents in these images that can be summarized into three representatives.

query image with all images in the dataset and select the most similar images for annotating a query image. Making a lot of comparisons leads to an uneficient query time in image annotation, so moving toward a real-time image annotation system is a demanded issue in the real-world applications.

On the other hand, in large scale datasets, there are some images with similar information in their visual contents and semantic labels. Figure 1 shows samples from IAPRTC-12 dataset that are visually and semantically similar. Therefore, it is essential to reduce the redundancy of annotated datasets without missing the original information in the labeled images. To reduce redundancy of the dataset, achieve a real-time annotation mechanism, and presrve the generalization of the annotation method, we follow this strategy in which labeled images could be replaced by a set of representatives, called prototypes in this paper. Figure 1 illustrates the idea of data summarization based on the prototype learning. As this figure shows, the primary purpose of the proposed method, called *Marginalized Coupled Dictionary Learning* (MCDL), is to factorize images and their corresponding labels as a weighted sum of learned prototypes. To achieve this goal, MCDL method learns visual (image) prototypes and their corresponding semantic (label) counterparts simultaneously.

In this paper, we have suggested a joint optimization problem to minimize the reconstruction and hinge losses w.r.t the visual and semantic dictionaries. The discrimination term utilized in MCDL can be regarded

as a modification of $\ell_1 - norm$ Support Vector Machine [9, 47], which was first presented for feature selection in high dimensional feature spaces. The number of prototypes is usually greater than the number of positive samples for each label, which can raise the issue of overfitting. MCDL utilizes ℓ_1 regularization to learn semantic prototypes as sparse as possible. Another inspiration to use ℓ_1 regularization is that label vectors are sparse by nature, and each visual prototype could correspond to a few labels.

In the literature, similar strategies have been suggested for multi-label classification [19, 32], and image annotation [15] to incorporate label discrimination in the dictionary learning stage. Most of these methods utilize squared loss function [15, 43] for both image and label modalities, aiming to reduce coding residual w.r.t training samples. When using the squared loss function for tags that are naturally imbalanced with many zero entries, label reconstructions are biased to zero (Figure 3-a) due to the symmetric property of the squared loss function. This will decrease decision margin and lead to less generalization for the annotation step. To tackle these issues, we have suggested a marginalized loss function with ℓ_1 regularization on semantic prototypes. Taking advantage of the marginalized hinge loss function and ℓ_1 regularization, MCDL could obtain labeled prototypes with admissible generalization in the test stage. To sum up, the main contributions of this paper are as follows:

- A coupled dictionary learning strategy is proposed to factorize labeled images into visual prototypes and their corresponding semantic vectors.
- MCDL employs the hinge loss for semantic modality, which imposes loss values just for false positives and false negatives. While conventional coupled dictionary learning approaches, such as Discriminative K-SVD (D-KSVD) [43], Label Consistent K-SVD (LC-KSVD) [14], and Multi-label Dictionary Learning (MLDL) [15], employ the squared loss defined over binary labels (D-KSVD) or codes (LC-KSVD and MLDL) for label discrimination which impose unnecessary reconstruction loss for true positives and true negatives.
- Each visual prototype can be associated with a few semantic tags. We employ ℓ_1 regularization to impose this sparsity prior knowledge about semantic prototypes. This avoids the overfitting originating from high dimensional space, especially when a few positive samples are available for a label.

The rest of this paper is organized as follows. In Section 3, the related works are reviewed. Section 2 introduces the definitions and notations used in this paper. The proposed MCDL method and details of the annotation

Table 1 Symbols and notations used in this paper.

N	The number of training samples
M	Dimensionality of feature vectors
T	The number of labels (tags)
$\mathbf{X} \in \mathbb{R}^{M \times N}$	Images matrix
$\mathbf{Y} \in \mathbb{R}^{T \times N}$	Labels matrix
K	Num of learned prototypes
$\mathcal{N}^i_+$	Num of annotations for i^{th} sample
N^+_t	Num of positive samples for t^{th} tag
$\mathbf{D}^I \in \mathbb{R}^{M \times K}$	Visual dictionary
$\mathbf{D}^L \in \mathbb{R}^{T \times K}$	Semantic dictionary
$\mathbf{D}^C \in \mathbb{R}^{(M+T) \times K}$	Coupled dictionary
$\mathbf{A} \in \mathbb{R}^{K \times N}$	Coefficients matrix
$d^C_k \in \mathbb{R}^{(M+T)}$	k^{th} column of coupled dictionary
$d^I_k \in \mathbb{R}^{(M)}$	k^{th} column of visual dictionary
$d^L_k \in \mathbb{R}^{(T)}$	k^{th} column of semantic dictionary
$\mathbf{d}^L_t \in \mathbb{R}^{1 \times K}$	t^{th} row of semantic dictionary

strategy are described in Sections 4 and 5. Experimental settings and results are then presented in Section 6. Finally, we conclude our work in Section 7.

2 Notations

Table 1 summarizes the notations used in this paper. Suppose that there are N training samples illustrated by $\mathcal{X} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$, where the ith image consists of two modalities: 1- the visual modality $x^i \in \mathbb{R}^M$ (M is the dimensionality of feature vector), 2- the semantic modality $y^i \in \{0,1\}^T$ (T is the number of distinct labels). A non-zero entry of y^i means that the given image has been annotated by the associated label. The number of annotations for i^{th} sample is denoted by $\mathcal{N}^i_+$. The number of annotated samples with t^{th} label is also denoted by N^+_t. Moreover, by concatenating different training vectors, we define $\mathbf{X} = [x^1, \ldots, x^N] \in \mathbb{R}^{M \times N}$ and $\mathbf{Y} = [y^1, \ldots, y^N] \in \mathbb{R}^{T \times N}$, which respectively denote image and label data matrices.

In our formulation, the coupled dictionary is depicted by $\mathbf{D}^C = \left[\mathbf{D}^{I^\top}, \mathbf{D}^{L^\top} \right]^\top \in \mathbb{R}^{(M+T) \times K}$, where $M \ll K < N$ is the number of prototypes. This dictionary is composed of two sub-dictionaries $\mathbf{D}^I \in \mathbb{R}^{M \times K}$ and $\mathbf{D}^L \in \mathbb{R}^{T \times K}$ for visual and semantic modalities respectively. The coefficients matrix is also shown by $\mathbf{A} = [\alpha^1, \ldots, \alpha^N] \in \mathbb{R}^{K \times N}$, where α^i is the sparse representation of ith training sample. The kth column of $\mathbf{D}^C$ is also called a coupled prototype denoted as d^C_k. Each coupled prototype consists of two sub-prototypes depicted by d^I_k and d^L_k for visual and label modalities. In this paper, $\mathbf{d}^L_{r,k}$ denotes the rth row and k^{th} column of the semantic dictionary. To denote the rth row of this matrix we have used $\mathbf{d}^L_r = \mathbf{D}^L_{r,.}$.

3 Literature Review

Due to the challenges discussed in the previous section, many approaches for handling these challenges belong to the search-based methods [4,14,35], by this assumption that the more visual similarity between two images, the more common labels among them. In 2PKNN [35,37], the authors proposed a two-pass version of the k-nearest neighbor technique for image annotation. To annotate an image, this method firstly retrieves the most similar images for each label, then computes an image-to-label similarity score, as well as utilizing a metric learning strategy for improving the image-to-image similarity measure.

Needing to compute the similarity of a query image to all images in the dataset, search-based methods are inherently time-consuming which emphasizes the importance of introducing scalable methods. Regardless of mete-data-based approaches [27,31], different methods have been suggested for scalable image annotation, which can be categorized into three main groups, including prototype-based [28], dimensionality-reduction-based [11,21], and transform-based methods [15,44]. The prototype-based approaches cluster samples and then choose one or a few samples or their representatives in each cluster [28]. Dimensionality-reduction-based approaches, such as product quantization [11] and hashing [21], focus on encoding high-dimensional feature spaces densely to achieve speed-up in search-based methods as well as reducing the memory costs. Our proposed approach belongs to the third group of scalable methods, transform-based approaches [15], that treat image annotation as a multi-label problem. In these approaches, both visual and semantic modalities are incorporated into the learning procedure for transforming input data into another space with higher levels of discrimination. One of the successful techniques in this category is sparse representation whose objective is to represent each pattern just using the linear combination of a few numbers of prototypes. Traditional sparse representation approaches can be considered as unsupervised methods that either ignore label information [23] or learn prototypes for each label separately. In recent years, many researchers have focused on embedding label information into the prototype learning procedure, generally known as discriminative [19] or coupled [30] dictionary learning, extensively applied for multi-label classification problems [32,39].

Discriminative sparse models have many applications in image classification, super-resolution [45], fault-diagnosis, etc. class-specific and shared discriminative dictionary learning (CASDDL) method [46] aims to classify the steel sheets based on the Fisher discrimination method.

They strive to extract the discriminative features for each class separately (inter-class information), along with a shared sub-dictionary which is common between all the classes for extracting the intra-class information. Li et al. [17] offered a weighted regularization approach to tackle the noisy images. They separate the coarse and fine structures of the noisy images by discriminative sparse methods. Class-oriented discriminative dictionary learning (CODLL) [20] is another discriminative-based method that not only maximizes the discrimination power of the dictionary atoms, but also considers the discrimination of the coefficients. They limited the atoms to make a group that describes a specific class and simultaneously restricted the coefficients to reconstruct data utilizing the class-related group of atoms. Structured discriminant analysis dictionary learning (SDADL) [5] aims to learn a structured discriminant analysis dictionary. This structured dictionary consists of class-specific sub-dictionaries. SDADL also introduces a classification loss term to learn and an optimal linear classifier. To continue, we introduce three different sparse-based discriminative methods with more details.

Motivated by the success of coupled dictionary learning for classification problems, similar techniques, such as semantic label embedding dictionary (SLED) [2], MLDL [15], and MSFS [44] have been employed for annotation. SLED uses $||\mathbf{X} - \mathbf{D}\mathbf{A}||^2_F + ||\mathbf{A}||_1 + \Omega(\mathbf{A})$, where $||\mathbf{X} - \mathbf{D}\mathbf{A}||^2_F$ strives to transform the visual training data into a new space, describable with the minimum atoms of matrix $\mathbf{A}$. The sparsity condition is controlled by the second term, i.e. $||\mathbf{A}||_1$. Using this formulation they extract the semantic similarities by the Fisher criterion. Fisher, i.e. $\Omega(\mathbf{A})$, aims to maximize the discrimination of each group of data, and simultaneously minimize the inter-group discrimination. MLDL is an extended version that extracts both the visual and semantic similarities in sparse space. This methos utilizes $||\mathbf{P}^\top\mathbf{X} - \mathbf{D}\mathbf{A}||^2_F + ||\mathbf{Q} - \mathbf{W}\mathbf{A}||^2_F + ||\mathbf{A}||_1$ formula which is somehow similar to our approach with some differences. The second term, $||\mathbf{Q} - \mathbf{W}\mathbf{A}||^2_F$, is where varies from our method. Here, the algorithm represents the $\mathbf{Q}$ matrix containing the semantic information. In fact, $\mathbf{Q} \in \mathbb{R}^{N \times K}$ is a binary matrix (N the number of train samples, and K the number of prototypes). This matrix measures the semantical correspondence of any prototype to the training data. The drawback of $\mathbf{Q}$ is that it is a binary relation, so cannot represent the similarity rate of the data. It assigns 1 if the prototype and the training sample share the same label set, while it could be partially true for many couples. In our method, we learn the semantic similarity of any prototype, while here it is prior knowledge. Moreover,

it uses F-norm which is not appropriate for the label loss function and we utilized the hinge loss function instead (we explain the reasons later). Besides, MSFS concentrates on sparse coding for feature extraction by $||\mathbf{Y} - \mathbf{V}\mathbf{B}||^2_F + \Gamma(\mathbf{V}) + ||\mathbf{X}\mathbf{W} - V||^2_F + \Omega(\mathbf{W}) + \lambda(\mathbf{W})$. The initial term focuses on dictionary learning for semantic representation, through minimizing the distance of $\mathbf{V}\mathbf{B}$ and $\mathbf{Y}$, where $\mathbf{B}$ is the dictionary and $\mathbf{V}$ is the coefficients matrix. Similar to MLDL it exploits Frobenius-norm for semantic similarity extraction. One more point, in the third term of the objective function it finds a W matrix that its multiplication in $\mathbf{X}$ (training data) reconstructs $\mathbf{V}$. In fact, MSFS through $\mathbf{W}$ estimates the $\mathbf{V}$ coefficients that its multiplication in $\mathbf{B}$ provides the estimated labels.

4 Marginalized Coupled Dictionary Learning

In this section, the proposed approach (MCDL) is discussed in detail. We present the objective function and learning algorithm of MCDL in Sections 4.1 and 4.2 respectively. Then, two main steps of the learning algorithm, including marginalized coupled sparse coding 4.3 and visual and semantic dictionary update 4.4 are discussed.

4.1 Objective Function

In this section, we have presented the objective function of the proposed method (MCDL) in detail. This method aims to marginalize scores for positive and negative labels. This means that the negative labels with small reconstruction (less than a margin but not zero) do not need to be penalized. Similarly, positive labels whose reconstructions are above a certain margin will not be penalized. Furthermore, to learn sparse semantic prototypes associated with the visual prototypes, MCDL imposes ℓ_1 regularization on the semantic dictionary. Considering these objectives, the empirical cost function for MCDL has been suggested as below:

$$\underset{\mathbf{D}^I, \mathbf{D}^L, \mathbf{A}}{\text{minimize}} \sum_{i=1}^{N} \left(\frac{\mathcal{N}_+^i}{\lambda} \| x^i - \mathbf{D}^I \alpha^i \|_2^2 + \sum_{t=1}^{T} \ell(y_t^i, \mathbf{d}_t^L \alpha^i) \right)$$
$$+ \sum_{t=1}^{T} \beta_1 \| \mathbf{d}_t^L \|_1 \qquad s.t \| \alpha^i \|_1 \leq \beta_0, \ \alpha_k^i \geq 0,$$
$$\| d_k^I \|_2 \leq 1, \ 0 \leq \mathbf{d}_{t,k}^L \leq v, \ \forall i, t, k,$$
$$\tag{1}$$

where the first term is the reconstruction term for visual vectors, the second term is the hinge loss func-
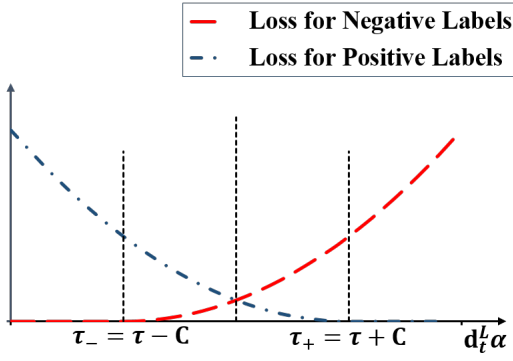
Fig. 2 The squared hinge loss functions for positive and negative labels.

tion defined over label vectors and the last term is ℓ_1 regularization for the semantic dictionary. Note that x^i, $i \in 1, ..., N$ is the ith training vector and $y_t^i \in \{0, 1\}$ is the tth label of it. In the objective function of (1), α^i, $i \in 1, ..., N$ is the shared representation (coupled) over visual and semantic vectors, and $\beta_0 = 1$ is ℓ_1 upper-bound for this representation. We suppose that each visual feature vector is ℓ_2-normalized, and $\beta_0 = 1$. This will decrease the number of tuning parameters and will increase the generalization of such representation. Moreover, the number of positive labels for each sample ($\mathcal{N}_+^i$) is employed as a regularizer between visual and semantic terms for each training sample, as well as the tuning parameter of λ.

In the optimization problem of (1), v is the upper bound for semantic dictionary elements to avoid noisy prototypes (discussed in Section 6.5). The estimation of y_t^i, denoted by $\mathbf{d}_t^L \alpha^i$, is achieved by multiplying sparse representation into the tth row of the semantic dictionary. The positiveness constraint is also imposed on the coefficients to increase the generalization of learned prototypes. Additionally, the ℓ_1 regularization in the last term aims to learn the most sparse semantic dictionary with annotation power (the second term). Furthermore, $\ell(., .) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is the squared hinge loss function which is defined as below:

$$\ell(y_t^i, \mathbf{d}_t^L \alpha^i) = [\max(0, \, \mathbf{C} - (2y_t^i - 1)(\mathbf{d}_t^L \alpha^i - \tau))]^2, \quad (2)$$

where τ is a constant threshold value and $2\mathbf{C}$ is the desired gap between score values for positive and negative samples of a label. Figure 2 shows the loss function of Equation (2). As can be seen, the loss will be zero for a label if computed score satisfies the margin values. For violated labels, the loss will be computed using squared loss based on its distance to the equivalent margin.

4.2 Learning Algorithm

Algorithm 1 presents the procedure of the suggested optimization technique to solve the optimization problem of (1). This algorithm consists of three main stages:
1) **Data Normalization**: in this stage, feature vectors are normalized to have unit $\ell_2 - norm$. This is useful to trade-off between the visual and semantic modalities (the first and second terms of Equation (1)) for each sample using a regularizer (λ) and its labels counts ($\mathcal{N}_+^i$). Such normalization can also result in more consistent sparse coding in the train and test stage.
2) **Initialization**: this stage has two steps. In step 2.1, visual prototypes are initialized by solving the first term of (1), in which weights are ignored. In the first stage of Step 2.2, the sparse representations of training samples are calculated over the visual dictionary obtained by solving the optimization problem of the Equation of Step 2.1 of Algorithm 1. In other words, visual prototypes are ℓ_2-normalized, and the semantic dictionary elements must be lower than or equal to v.
3) **Optimization**: The problem of Equation (1) is not jointly convex w.r.t dictionaries ($\mathbf{D}^I$ and $\mathbf{D}^L$) and sparse coefficients $\mathbf{A} = [\alpha^1, ..., \alpha^N]$. However, it is convex w.r.t each of these parameters set when the other one is fixed. Thus, this optimization problem is decomposed into two convex problems including sparse coding and dictionary update. These two steps are applied alternatively to solve the original problem (Step 3 of Algorithm 1). These steps are discussed in the following sections.

4.3 Marginalized Coupled Sparse Coding

In this section, proposed Marginalized Coupled Sparse Coding (MCSC) (Step 3.1 of Algorithm 1) is presented to solve the sparse coding problem of Equation (1) when the dictionaries are fixed (Section 4.3). Indeed, MCSC is a fast method based on LARS (Least Angle Regression) technique (also known as LARS-Lasso) [6, 23].

When the dictionaries ($\mathbf{D}^I$ and $\mathbf{D}^L$) in the optimization problem of Equation (1) are fixed, the problem can be rewritten w.r.t each training sample (α^i) individually as below:

$$\underset{\alpha^i}{\text{minimize}} \; f(\alpha^i) = \| x^i - \mathbf{D}^I \alpha^i \|_2^2 + \frac{\lambda}{\mathcal{N}_+^i} \sum_{t=1}^{T} \xi_t^{i\,2}$$
$$s.t \; \| \alpha^i \|_1 \leq \beta_0, \; \alpha_k^i \geq 0, \quad\quad (3)$$
$$\xi_t^i \geq \mathbf{C} - (2y_t^i - 1)(\mathbf{d}_t^L \alpha^i - \tau), \; \xi_t^i \geq 0,$$

where ξ_t^i is the slack variable which measures margin violation for t^{th} label of the ith training data. According

---

Algorithm 1: Marginalized Coupled Dictionary Learning

Input : Set of training images $\mathbf{X} = [x^1, \ldots, x^N]$ and their corresponding label vectors $\mathbf{Y} = [y^1, \ldots, y^N]$. Discriminative regularization λ. The number of prototypes K. The number of repeats R.

Output: Learned prototypes $\mathbf{D}^I$ and $\mathbf{D}^L$.

All necessary notations have been introduced in Section 2.

1. Data Normalization

- $x^i \leftarrow \frac{x^i}{||x^i||_2}, \forall i \in \{1, \ldots, n\}.$

2. Initialization

2.1. Visual Dictionary Initialization
- Initialize $\mathbf{D}^I$ by K using k-means clustering .
- Solve below optimization problem by alternate updating of $\mathbf{A}$ and $\mathbf{D}^I$:
$$\underset{\mathbf{D}^I, \mathbf{A}}{\text{minimize}} \sum_{i=1}^N \| x^i - \mathbf{D}^I \alpha^i \|_2^2$$
$$s.t \| \alpha^i \|_1 \leq \beta_0, \alpha_k^i \geq 0, \| d_k^I \|_2 \leq 1, \forall i, k.$$

2.2. Semantic Dictionary Initialization
- Calculate the sparse representations $\mathbf{A} = [\alpha^1, \ldots, \alpha^N]$ based on the visual dictionary.
- $d_k^L \leftarrow \min(v, \frac{\sum_{i=1}^N \alpha_k^i y^i}{\sum_{i=1}^N (\alpha_k^i)^2}), \forall k \in \{1, \ldots, K\}.$

3. Optimization

Solve Equation (7) by alternate updating of $\mathbf{A}$ and $\mathbf{D}^C$ while the other is assumed to be fixed:

for $r \leftarrow 1$ **to** R **do**

3.1. Update representations $\mathbf{A} = [\alpha^1, \ldots, \alpha^N]$ using proposed MCSC technique ($\forall i \in \{1, \ldots, N\}$):
for $s \leftarrow 1$ **to** S **do**
- Calculate $\tilde{y}_t^i[s-1]$ as an estimate for label scores using (6).
- Update $\hat{\alpha}_t^i[s]$ by solving (7).

$\alpha^i \leftarrow \hat{\alpha}_t^i[S].$

3.2. Update each column of $\mathbf{D}^I$ and $\mathbf{D}^L$ using Algorithm (2).

---

to the constraints in Equation (3), we have:

$$\xi_t^i = \max(0, \mathbf{C} - (2y_t^i - 1)(\mathbf{d}_t^L \alpha^i - \tau)). \tag{4}$$

The problem of Equation (3), which is equivalent to the first and second terms of Equation (1), is a constrained quadratic optimization problem that can be solved using quadratic programming techniques. It is noticeable that this problem must be solved for all training samples in each iteration of the whole optimization that is time-consuming. This motivates us to investigate a simpler and faster iterative coupled sparse coding based on the LARS, called MCSC, to obtain an approximate solution for the problem of Equation (3). Lasso is an effective and fast method to solve traditional sparse coding problems. To describe MCSC, suppose that $\hat{\alpha}^i[s-1]$

is the sparse coefficient vector obtained at the previous iteration of MCSC, one can provide a new approximate ($\hat{\alpha}^i[s]$) by solving (see Lemma 1 in A):

$$\hat{\alpha}^i[s] \triangleq \underset{\alpha^i}{\text{argmin}} \, g(\alpha^i) = \| x^i - \mathbf{D}^I \alpha^i \|_2^2$$
$$+ \frac{\lambda}{\mathcal{N}_+^i} \sum_{t=1}^T (\tilde{y}_t^i[s-1] - \mathbf{d}_t^L \alpha^i)^2 \tag{5}$$
$$s.t \| \alpha \|_1 \leq \beta_0, \alpha_k^i \geq 0,$$

where $\tilde{y}_t^i[s-1]$ ($t \in \{1 \ldots, T\}$) is defined based on current approximate of sparse coefficients ($\hat{\alpha}^i[s-1]$) and its corresponding label penalties ($\hat{\xi}_t^i[s-1], \forall t \in \{1 \ldots, T\}$) as below:

$$\tilde{y}_t^i[s-1] = \begin{cases} \mathbf{d}_t^L \hat{\alpha}^i[s-1], & \text{if } \hat{\xi}_t^i[s-1] = 0, \\ \tau + (2y_t^i - 1)\mathbf{C} & \text{if } \hat{\xi}_t^i[s-1] > 0. \end{cases} \tag{6}$$

The optimization problem of (5) can be reformulated as:

$$\underset{\alpha^i}{\text{argmin}} \left\| \begin{bmatrix} x^i \\ \sqrt{\frac{\lambda}{\mathcal{N}_+^i}} \tilde{y}_t^i[s-1] \end{bmatrix} - \begin{bmatrix} \mathbf{D}^I \\ \sqrt{\frac{\lambda}{\mathcal{N}_+^i}} \mathbf{D}^L \end{bmatrix} \alpha^i \right\|_2^2 \tag{7}$$
$$s.t \| \alpha^i \|_1 \leq 1, \alpha_k^i \geq 0,$$

which is equivalent to a sparse coding problem with positiveness constraint on the coefficients vector. This optimization problem can be solved effectively using LARS. Step 3.1 of Algorithm 1 summarizes MCSC algorithm presented to solve Equation (3). In each iteration of MCSC, the optimization problem of (7) is solved using LARS based on the current estimate for sparse coefficients provided at the previous iteration. In this paper, we have repeated this step four times ($S = 4$). Initial approximate is obtained by supposing that $\hat{\xi}_t^i[0] > 0$ for all labels in Equation (6). We have presented Lemma 1 in A to prove the convergence of the MCSC algorithm.

4.4 Dictionary Update

Consider solving the optimization problem of (1) when the sparse coefficients ($\mathbf{A} = [\alpha^1, \ldots, \alpha^N]$) are fixed. In this case, this problem is equivalent to optimize visual and semantic dictionaries separately. We have utilized a randomized coordinate descent algorithm based on warm restart (current parameters) to update prototypes (columns) of both dictionaries in a random sequence, summarized in Algorithm 2. In this section, we have presented the proposed methods to solve these two disjoint dictionary learning problems to optimize semantic and visual prototypes.

Algorithm 2: Dictionary Update

Input : Set of normalized training images
$\mathbf{X} = [x^1, \ldots, x^N]$ and their corresponding
label vectors $\mathbf{Y} = [y^1, \ldots, y^N]$.
Current learned prototypes $\mathbf{D}^I$ and $\mathbf{D}^L$.
Regularization parameter β_1 for $l_1 - norm$.
Sparse representations $\mathbf{A} = [\alpha^1, \ldots, \alpha^N]$.

Output: Updated $\mathbf{D}^I$ and $\mathbf{D}^L$.

Visual and Semantic Dictionary Update
for $j \in \{1, \ldots, K\}$ *at random* **do**

    – Update k^{th} Visual Prototype:

      – $z_k^i = x^i - (D^I \alpha^i - d_k^I \alpha_k^i)$, $\forall i \in \{1, \ldots, N\}$.

      – $\hat{d}_k^I \leftarrow \frac{\sum_{i=1}^N \mathcal{N}_+^i \alpha_k^i z_k^i}{\sum_{i=1}^N \mathcal{N}_+^i (\alpha_k^i)^2}$, $\forall k \in \{1, \ldots, K\}$.

      – $d_k^I \leftarrow \frac{\hat{d}_k^I}{||\hat{d}_k^I||_2}$.

    – Update k^{th} Semantic Prototype:

      for $t \leftarrow 1$ **to** T **do**

        – $q_k^i = D_t^L \alpha^i - \mathbf{d}_{t,k}^L \alpha_k^i$, $\forall i \in \{1, \ldots, N\}$.

        – Update $\mathbf{d}_{t,k}^L$ by solving (11)

4.4.1 Visual Dictionary Update

When keeping the sparse coefficients fixed, the optimization problem of (1) w.r.t visual dictionary is equivalent to solve the below problem:

$$\underset{\mathbf{D}^I}{\text{minimize}} \sum_{i=1}^N \mathcal{N}_+^i \| x^i - \mathbf{D}^I \alpha^i \|_2^2 \quad s.t \| d_k^I \|_2 \leq 1. \quad (8)$$

This objective function is indeed a weighted form of the traditional dictionary learning problem. One of the most used approaches to solve this problem is the block coordinate descent approach, in which prototypes are optimized individually while keeping the others fixed [23, 39]. Taking the gradient of (8) w.r.t d_k^I and setting it equal to zero, we have:

$$\hat{d}_k^I \leftarrow \frac{\sum_{i=1}^N \mathcal{N}_+^i \alpha_k^i z_k^i}{\sum_{i=1}^N \mathcal{N}_+^i (\alpha_k^i)^2}, \quad (9)$$

where $z_k^i = x^i - (D^I \alpha^i - d_k^I \alpha_k^i)$ is residual of the ith input vector w.r.t other prototypes, and $\hat{d}_k^I$ is the optimum of (8) without considering its constraint. This can be shown that solving constrained optimization (8) w.r.t to the kth prototype (column) of the visual dictionary (i.e., d_k^I), when the other prototypes hold fixed, is equivalent to solve unconstrained one, followed by an $\ell_2 - norm$ normalization. It is worth mentioning that $\mathcal{N}_+^i$ (the number of positive labels for i^{th} sample) acts as weights in updating visual prototypes. This means that samples with more labels will have a greater impact on the optimized visual prototypes because they are probably annotated with complete labels. In the marginalized sparse coding problem of (5), these weights play

the role of normalizer to make a balance between visual and semantic loss functions.

4.4.2 Semantic Dictionary Update

If the sparse coefficients are given, the optimization problem of (1) w.r.t the semantic dictionary turns into T independent convex problems (one per each label) as below:

$$\underset{\mathbf{d}_t^L}{\text{minimize}} \sum_{i=1}^N (\xi_t^i)^2 + \beta_1 \| \mathbf{d}_t^L \|_1 \quad s.t \ 0 \leq \mathbf{d}_{t,k}^L \leq v, \forall k,$$
$$\xi_t^i \geq \mathbf{C} - (2y_t^i - 1)(\mathbf{d}_t^L \alpha^i - \tau), \ \xi_t^i \geq 0, \forall i. \quad (10)$$

This problem can be seen as a modification of support vector machine with $\ell_1 - norm$ regularization, where hinge loss is replaced with squared hinge loss. In the proposed semantic dictionary learning approach, $\ell_1 - norm$ regularization can act as a prototype selection for each label, meaning that just a small number of prototypes can be representative for each label. The relation between threshold (τ), margin ($\mathbf{C}$), and semantic elements upper bound (v) are discussed in the next section. The problem of (10) is a quadratic optimization problem that can be solved using quadratic programming approaches, though it needs high computational time and memory. Since this optimization problem should be solved in each iteration of the dictionary learning for all labels, we have proposed a simple and fast approach based on block coordinate descent. Each of these T convex problems of (10) admits separable constraints ($\ell_1 - norm$) in the updated blocks ($\mathbf{d}_{t,k}^L, \forall k \in \{1, \ldots, K\}$). So, the convergence of the proposed coordinate descent based method is guaranteed [40]. To optimize $\mathbf{d}_{t,k}^L$ which is k^{th} element (column) of semantic dictionary for t^{th} label (row) using block coordinate descent when the other variables are fixed, we should solve:

$$\underset{\hat{\mathbf{d}}_{t,k}^L}{\text{minimize}} \sum_{i \in \{i | \alpha_k^i \neq 0\}} (\xi_t^i)^2 + \beta_1 |\hat{\mathbf{d}}_{t,k}^L| + \rho \| \hat{\mathbf{d}}_{t,k}^L - \mathbf{d}_{t,k}^L \|_2^2$$
$$s.t \ 0 \leq \hat{\mathbf{d}}_{t,k}^L \leq v,$$
$$\xi_t^i = \max(0, \mathbf{C} - (2y_t^i - 1)(\hat{\mathbf{d}}_{t,k}^L \alpha_k^i + \mathbf{q}_t^L - \tau)). \quad (11)$$

where $q_k^i = D_t^L \alpha^i - \mathbf{d}_{t,k}^L \alpha_k^i$ is the score (regression) of tth tag of ith label vector using other semantic prototypes and $\hat{\mathbf{d}}_{t,k}^L$ is the new estimate for $\mathbf{d}_{t,k}^L$.

The cost function of (11) is a single variable optimization problem, which can be solved effectively even using a parallel linear search for all tags simultaneously. Since
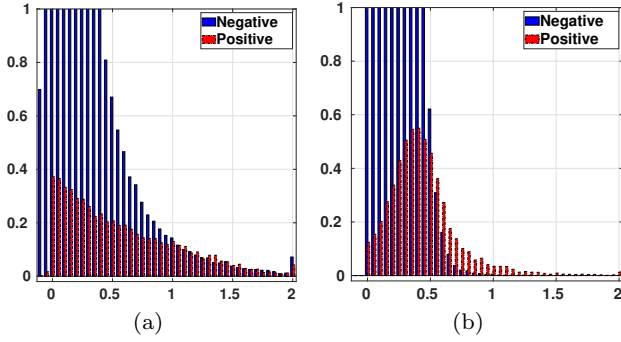
Fig. 3 The distribution of scores (scaled for visualization purpose) for positive and negative labels generated (a) using squared loss function and (b) using MCDL with hinge loss. This result has been provided using IAPRTC-12 dataset and 4000 prototypes.

the optimization problem of (10) is a convex optimization problem with separable regularization, convergence of the proposed coordinate descent method is guaranteed.

Figure 3 illustrates the distribution of scores (for test samples) based on the squared loss function versus the marginalized MCDL approach. As Figure 3-a shows, scores distribution of the positive labels gets biased to zero when the squared loss function is employed. Figure 3-b illustrates the impact of hinge loss on the distribution of scores, where there are less interaction between scores of negative and positive samples.

4.5 Discussion and Parameters Setting

In the previous sections, we suggested a joint optimization problem to learn visual and semantic dictionaries in a coupled manner. The objective function of (10) can be regarded as a modification of $\ell_1 - norm$ Support Vector Machine [9, 47] which was first presented for feature selection in high dimensional feature spaces. In fact, each row of the semantic dictionary is indeed a regression vector over sparse coefficients to predict the associated labels (refer to [14]). The number of prototypes is usually greater than the number of positive samples for each label, which can raise the issue of overfitting. MCDL utilizes ℓ_1 regularization to learn semantic prototypes as sparse as possible. Another inspiration to use ℓ_1 regularization is that label vectors are sparse by nature, and each visual prototype could correspond to a few labels. Moreover, we are interested in non-negative semantic prototypes and sparse coefficients for the same reasons. This constraint has been extensively applied in non-negative matrix factorization techniques [23, 26] for a wide range of applications and can produce more localized features.

As mentioned in Section 4.1, $\hat{y}_t^i = \mathbf{d}_t^L \alpha^i \in \mathbb{R}$ (Figure 2) is the score value of tth label for the ith training data. Due to the constraints of (1) on sparse coefficients ($\| \alpha \|_1 \leq 1$) and semantic dictionary elements ($0 \leq \mathbf{d}_{t,k}^L \leq v, \forall t, k$), the upper bound value for scores is $0 \leq \hat{y}_t^i \leq v, \forall i, t$. Suppose that $v = 1$ and $\tau = \mathbf{C} = 0.5$. In this case, to obtain zero hinge loss for a positive label, all used prototypes in sparse representation should be exactly 1 for this label, which is impossible in practice. In other words, chosen values for threshold and margin values impact on appropriate value of v. On the contrary, greater values of v can lead to noisy prototypes which can be controlled using tuning parameter of β_1 to some extent. Therefore, in this paper, we have set $\mathbf{C} \in \{0.25, 0.5\}$, where $\tau = 0.25 + \frac{\mathbf{C}}{2}$, and $v = 5$.

Another point is that upper bound for squared hinge loss in the first term of Equation (10) is $N_t^+(\tau + \mathbf{C})^2$ (when $\mathbf{d}_t^L = \vec{0}$). The second term of this equation guarantees that increasing the value of a semantic dictionary element will be equivalent to decreasing hinge loss at a meaningful level. So, the $\ell_1 - norm$ of semantic dictionary rows remains correlated with the number of positive samples for the associated label.

Finally, for tuning of discriminative and ℓ_1 regularizations, we have selected: $\lambda = \frac{1}{T}\eta^2$, where $\eta \in \{0.1, 0.25, 0.5, 1, 2, 5, 10\}$) and $\beta_1 \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.75, 1\}$ respectively. Best parameters are chosen by validation technique for each dictionary size. The maximum number of iterations for Step 3 of Algorithm 1 is 15. To speed up, just two iterations are applied in the validation step.

5 Annotation Strategy

Figure 4 shows the annotation strategy of a query example. In the first step, the shared sparse representation (which is denoted by $\alpha \in \mathbb{R}^K$) is obtained for the query image, which means $\hat{x} = \mathbf{D}^I \alpha, \; s.t \; ||\alpha||_1 \leq 1$, where x is the normalized visual modality. Then, the scoring vector, which shows the similarity of a given image to different labels, is computed as $\mathbf{D}^L \alpha \in \mathbb{R}^T$. Finally, we will have $\hat{y}_t = sign(\mathbf{d}_t^L \alpha - \tau_{optimal})$, where $\hat{y}_t$ is the prediction for t^{th} label and $\tau_{optimal}$ is the optimal threshold for labels prediction. This threshold is computed based on the best F_1 measure on training samples for each dataset.

6 Experimental Results

Datasets. To assess the performance of the proposed method two popular image annotation datasets, IAPRTC-
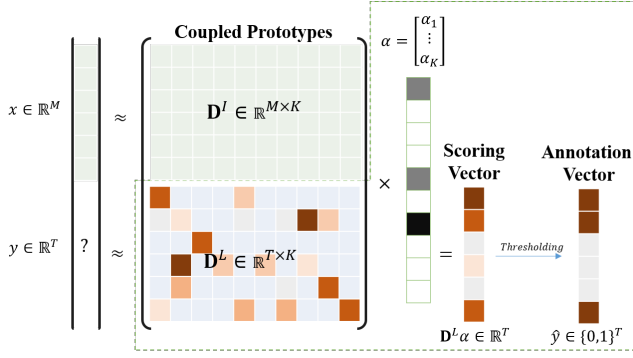
Fig. 4 The annotation strategy for a query image. The image is annotated with labels that their reconstructions (scores) over semantic dictionary are greater than a given threshold.

Table 2 The number of images (total, train, and test sets) and tags for the datasets.

Datasets	Image	Train	Test	Tags
IAPRTC-12	19627	17665	1962	291
ESP-GAME	20770	18689	2081	268
FLICKR-60K	59083	41359	17724	295
FLICKR-125K	124840	87388	37452	568

12 [7] and ESP-GAME [38] are used. Moreover, we utilize another dataset containing one million images from Flickr platform[1], which could be considered as a more challenging and qualified real-world dataset for this task. We extracted two different subsets from Flickr, called FLICKR-60K and FLICKR-125K, using the following procedure. Similar to [13], FLICKR-60K was obtained by listing 295 tags which occurred in at least 500 images in the first 100,000 images of the original dataset. We have also removed images with less than two tags, resulting in a dataset with 59083 images. FLICKR-125K was obtained from the first 200,000 images in a way similar to FLICKR-60K. We then split them into train and test sets with a 70-30 ratio. General statistical information of all datasets has been presented in Table 2.

Features. We have employed CNN models that are trained through ImageNet dataset for object recognition, including VggNet [29], ResNet [10], DenseNet [12], and EfficientNet [33] which result in the feature vectors by the dimensionality of 4048, 2048, 2208, and 2560 respectively. To extract these feature vectors, the output of the layer before the last layer is utilized. Random or regular cropping is a common scheme for data augmentation in both training and testing stages [10, 12, 16]. In the training stage, random cropping is widely used to maintain desired image size depending
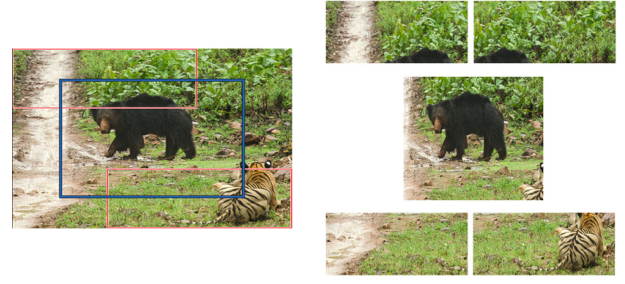
[1] https://www.flickr.com

Fig. 5 The scheme employed to divide an input image before feature extraction using deep models. Five extracted regions have been shown.

on the network configuration [10, 12, 16]. As a result, cropping techniques such as regular [16] or multi-scale cropping [10] are used in the testing stage to consider all spatial information and prevent downsampling or center cropping to provide network input size. Experiments provided in [12] show that the 10-crop strategy (first presented by Krizhevsky et al. [16]) outperforms single-crop at test time.

In the standard 10-crop technique, five patches of the same size (the four corners and the center) are extracted as well as their horizontal flips, which results in ten crops. Finally, the predictions of the network are averaged over ten crops in the test stage. We follow the same strategy and obtain five crops (flipping is ignored) for feature extraction of each training and test image. To segment an input image, we crop the central part of the image included in $\frac{2}{3}$ of the whole area, as well as four crops from the corners with the width of $\frac{2}{3}$ and the height of $\frac{1}{3}$ (see Figure 5). Each segment is then passed to the network and all five extracted feature vectors are averaged to make the final feature vector. Finally, similar to [41], we apply PCA to reduce the dimensionality of the feature vectors to 200.

6.2 Analysis of the Dictionary Size

Figures 6 and 7 illustrate the impact of dictionary size on precision, recall, and F_1 measures for the proposed MCDL algorithm applied to three different datasets. Starting from the lowest dictionary size for IAPRTC-12, which is a small value of 100, F_1 measure for all three features has an admissible value of over 30 percent. This indicates that the learned prototypes using MCDL are comprehensive candidates for training images. As we will discuss in the next section, this effectiveness originates from the marginalized loss function and $\ell_1 - norm$. Although the increased dictionary size for IAPRTC-12 has improved the annotation measures, the increase rate is slower by exceeding the dictionary
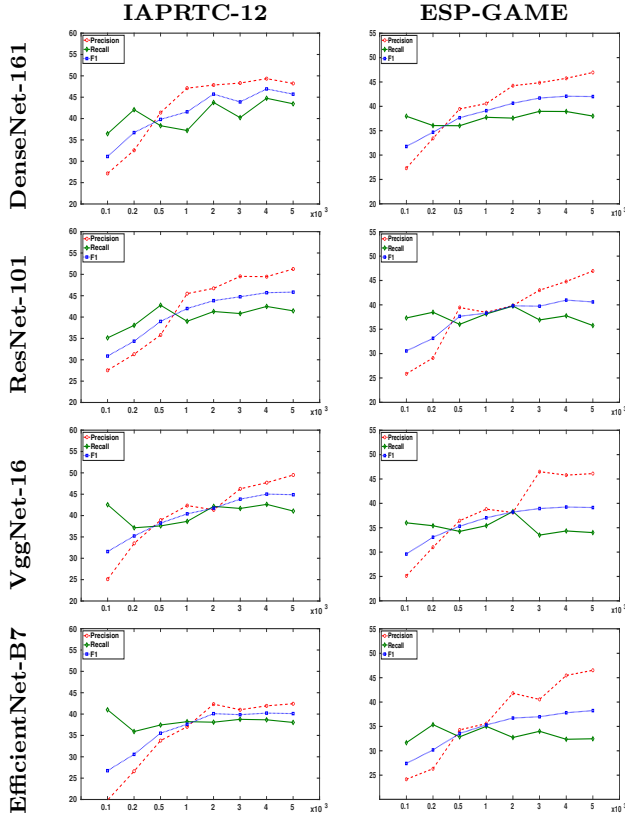
Fig. 6 Precision, Recall, and F_1 for IAPRTC-12 and ESP-GAME w.r.t the different number of prototypes. On each diagram, the x-axis shows the number of prototypes (in kilo) and the y-axis demonstrates the measures (in percent).

Fig. 7 Precision, Recall, and F_1 for FLICKR-60K and FLICKR-125K w.r.t the different number of prototypes. On each diagram, the x-axis shows the number of prototypes (in kilo) and the y-axis demonstrates the measures (in percent).

size of 3000. The best results are achieved through the dictionary size of 4000 in IAPRTC-12 dataset. For ESP-GAME also the trend is almost incremental. However, the best results are in the dictionary size of 4000. The overall initial F_1 for FLICKR-60K is around 20 percent and the best dictionary size for all feature types is 12000, except for EfficientNet-B7 which is a=8000. By increasing the dictionary size to above 12000 and 8000, the F_1 measure has decreased for this dataset. A plausible reason for such decreases in F_1 measure is the overfitting phenomenon for semantic dictionary, which is common when the number of parameters is increased, and in our case, the can be biased to For FLICKR-125K, F_1 has gradually increased before reaching the best value of around 16000 and 20000 prototypes. As it can be seen, EfficientNet-B7 has poor results versus other networks. Experimentally, if a network provides features which is more linear separable, MCDL can perform better.

Figures 6 and 7 show that MCDL can achieve admissible results with a small number of prototypes. The best dictionary size is about 4000 for IAPRTC-12 and
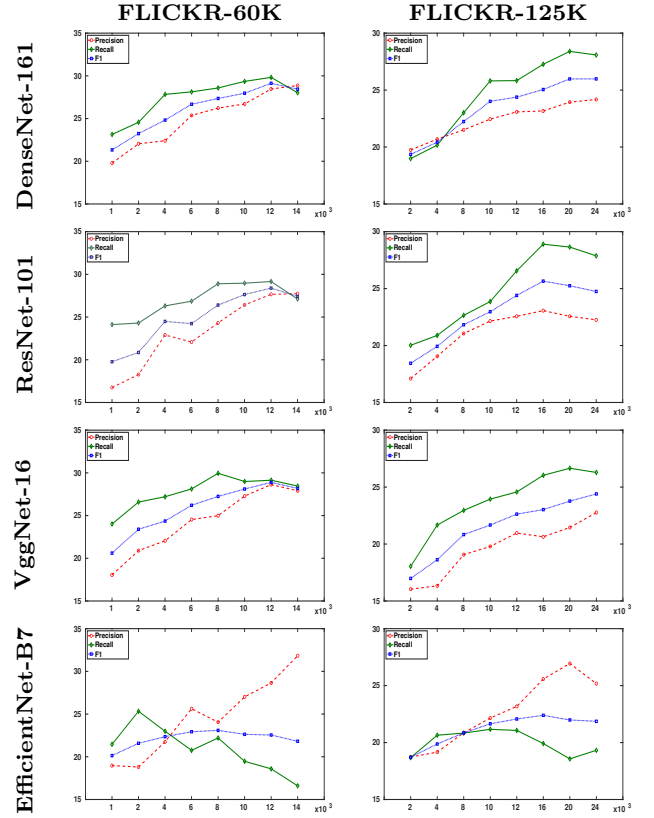
ESP-GAME, and 12000, and 20000 for two FLICKR-60K and 125K. These dictionary sizes are around 20 percent of the training data size, indicating the efficiency of our approach for summarizing large datasets to a limited number of prototypes.

6.3 Analysis of The Objective Function

In this section, the effectiveness of different stages of the proposed method is assessed. As the first baseline, we have eliminated coupled learning in MCDL by examining unsupervised dictionary learning called UDL. In this baseline, visual dictionary is first learned the same as Step 2.1 of Algorithm 1. Then, semantic labels of learned prototypes are obtained using Step 2.2 of Algorithm 1. Furthermore, to study the importance of marginalized loss function and ℓ_1 regularization in MCDL, we have examined another baseline, named Coupled Dictionary Learning (CDL), where hinge loss has been replaced with squared loss function and ℓ_1 regularization is omitted.

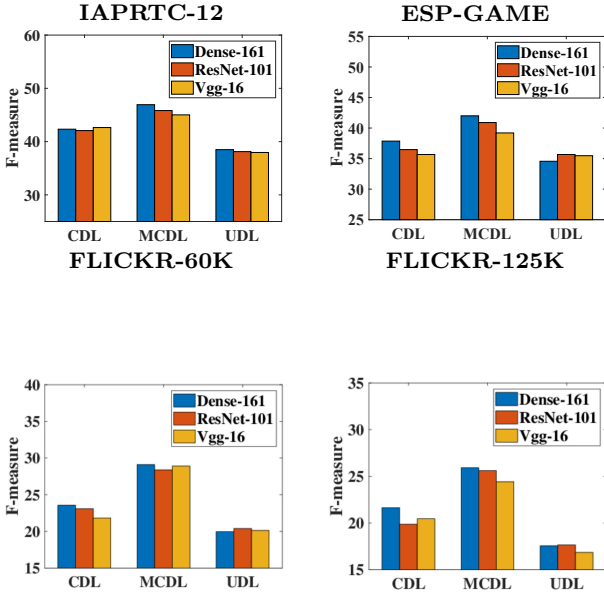Results in Figure 8 are provided based on the best dictionary size mentioned in the previous section. It illus-

Fig. 8 F_1 measure for CDL and UDL versus the proposed MCDL.

Table 3 F_1 comparison between MCDL and 2PKNN (in percent).

DATASET	*FEATURE*	*2PKNN*	*MCDL*
IAPRTC-12	**DenseNet-161**	37.1	46.9
	ResNet-101	37.3	45.8
	VggNet-16	37.6	45.0
	EfficientNet-B7	38.0	40.2
ESP-GAME	**DenseNet-161**	37.9	42.0
	ResNet-101	37.0	40.9
	VggNet-16	36.4	39.2
	EfficientNet-B7	36.4	38.2
FLICKR-60K	**DenseNet-161**	28.9	29.1
	ResNet-101	28.5	28.4
	VggNet-16	26.4	28.9
	EfficientNet-B7	20.5	23.0
FLICKR-125K	**DenseNet-161**	27.1	25.9
	ResNet-101	25.5	25.6
	VggNet-16	23.2	24.4
	EfficientNet-B7	18.5	22.4

trates that CDL method (supervised version of UDL) has better performance with significant differences in large-scale datasets. It is noticeable that the marginalized hinge loss used in MCDL method has raised the generalization of the prototypes significantly. Moreover, CDL method uses squared loss function instead of the marginalized loss function suggested in the MCDL. Although least square loss function is appropriate for visual modality, it yields to biased prototypes for imbalanced zero-one labels, as it can be investigated from Figure 3. As figure shows, most of the scores are concentrated around zero (labels are considered 0 and 5).

6.4 Scalability Analysis and Comparison

This section focuses on scalability analysis based on two important criteria, including performance and runtime. We have considered one of the most popular and state-of-the-art similarity-based approaches, 2PKNN, as the main baseline method. To reach a fair comparison, we have used the same feature vectors mentioned in the previous sections for both 2PKNN and our approach.

Feature Analysis
Table 3 reveals that in IAPRTC-12 dataset, MCDL considerably outperforms the baseline method, 2PKNN, in all three feature types. The improvement is about 9.8 percent for feature type of DenseNet-161. For the sec-

ond dataset, ESP-GAME, the results have also been improved through MCDL and the most considerable improvement is 4.1 percent. On the other hand, in FLICKR-60K, the most egregious progress is related to VggNet-16, where it reaches to 29 percent in MCDL from 26 percent in 2PKNN. Finally, the results for 125K version of FLICKR are improved in ResNet-101 and VggNet-16. The reason for the better performance of MCDL on IAPRTC-12 and ESP-GAME rather than two FLICKR datasets is that there is considerable redundancy in the former datasets, as can be investigated from Figure 6 and 7. Indeed, when we retrieve a fixed number of similar images to annotate a query image, the retrieved images may be highly correlated in their visual contents and semantic labels. However, MCDL tries to reconstruct query images based on various prototypes and thus the redundancy will be reduced.

Annotation Time
Table 4 compares the annotation time of MCDL against the baseline. The experiments are conducted on a PC with an Intel (R) Core (TM) i7-6700 HQ 3.1 GHz CPU, and 16G RAM, in MATLAB environment. Furthermore, the annotation time is averaged over all the test images. While 2PKNN needs a tremendous number of peer to peer comparisons for finding the most similar images per each label, MCDL needs a tiny proportion of 2PKNN time to annotate an input image. For IAPRTC-12 and ESP-GAME with roughly 20000 images, labeling a new image takes over 25 milliseconds using the 2PKNN method. This measure is sharply declined by MCDL to under 1.5 milliseconds. To sum up, the information presented in Tables 3 and 4 implies that not only the scalability is acquired, but also the performance of

Table 4 The average annotation time (in millisecond) for input images in MCDL compared to 2PKNN, using DenseNet-161 feature vector. The third column shows the percentage of the reduction in annotation time using MCDL method.

	2PKNN	MCDL	Reduction
IAPRTC-12	27.5	1.5	94.5%
ESP-GAME	25.4	1.2	95.2%
FLICKR-60K	57	1.8	96.8%
FLICKR-125K	390	10	97.4%

MCDL is improved.

Performance Analysis

To compare the performance of our approach against the other methods, we provide Table 5. It is necessary to mention that there are three reports for 2PKNN in this table. The first, 2PKNN (SD), is its performance on traditional standard features. The second utilizes the same features with a metric learning algorithm. In 2PKNN (CNN), we fed it our CNN-based features to make a fair comparison by our approach. The initial impression of this table is that CNN-based features could provide better performances in comparison with the standard features (i.e., color, histogram, shape, sift, etc.). A meticulous glance at the precision and recall values reveals that there is a considerable variance between them in almost all the methods. Fortunately, this is not true in our method. The reason is that, against the others, MCDL assigns different number of labels to any input sample based on its scores. The other methods take a fixed number of labels for annotation, which is mainly less than the required. Therefore while the precision improves, the recall does not. This fact originates from the increase of the number of false-negatives. The reason for a good trade-off between precision and recall in MCDL is that our technique utilizes a marginalized approach on scores. Table 5 demonstrates that in both datasets, IAPRTC-12 and ESP-GAME, the proposed MCDL method could achieve the highest F_1 scores, 47 percent for IAPRTC-12 and 42 percent for ESP-GAME. Looking at the precision values of Table 5 depicts that among the standard-feature-based methods MLDL and ML-based 2PKNN can provide better results than our method. This is also true for 2PKNN with CNN features. On the other hand, the recall value for MCDL is significantly higher than all the methods, and this is the reason that our method could pick the best F_1 score.

One more point, in Table 6 we compare our approach and baseline method on FLICKR-60K and FLICKR-125K datasets where it is obvious that those F_1 scores are significantly close to each other. The importance of this subject is clarified when we note that our ap-

Table 5 Precision, Recall, and F_1 comparison of different methods on IAPRTC-12 and ESP-GAME datasets (in percent) using Dense-161 feature .

	Method	IAPRTC-12			ESP-GAME		
		P	R	F_1	P	R	F_1
Standard Feature	*ML [8]*	48	25	33	49	20	28
	σML [8]	46	35	40	39	27	32
	Fast Tag [3]	47	26	34	46	22	30
	KSVM-VT [36]	47	29	36	33	32	33
	MLDL [15]	56	40	47	56	31	40
	2PKNN (SD) [37]	49	32	39	51	23	32
	2PKNN (ML) [37]	54	37	44	53	27	36
	Mvg-NMF [26]	47	40	43	41	33	37
CNN Feature	*MVSAE [42]*	43	38	40	47	28	34
	CCA-KNN [24]	45	38	41	46	36	41
	RPLRF [18]	48	29	36	43	27	34
	AHL [34]	47	35	40	46	23	31
	SEM [22]	41	39	40	38	42	40
	VLAD [4]	46	33	38	44	33	38
	2PKNN (CNN)	51	29	37	50	31	38
	MCDL	**49**	**45**	**47**	**46**	**39**	**42**

Table 6 Precision, Recall and F_1 comparison of different methods on FLICKR-60K and FLICKR-125K datasets (in percent) using Dense-161 feature.

Method	Dataset	Pre	Rec	F_1
2PKNN	FLICKR-60K	34	25	29
	FLICKR-125K	32	24	27
MCDL	FLICKR-60K	28	30	29
	FLICKR-125K	24	28	26

proach reaches this F_1 score by replacing all training images with a few prototypes (20000 prototypes instead of 124840 images of FLICKR-125K). This property leads to a considerable reduction in the annotation time, as presented in Table 4.

Computational Complexity

In the matter of computational complexity, our method outperforms the baseline, 2PKNN. To begin with, we first apply a dimensionality reduction on each input visual vector to convert it to a low-dimension vector of size M. 2PKNN includes the distance computation of the input image with all training samples which have the time complexity of $O(N \times M)$, followed by finding the K_1 most similar training images [35] for each label with time complexity of $O(N \times N \times T)$. Note that if linear algorithms are used to find first K_1 nearest neighbors instead of calling sort in the original algorithm of 2PKNN, it will be of $O(N \times K_1 \times T)$). There-

fore, the total time complexity of 2PKNN is $O(N \times M + N \times N \times T)$.

On the other hand, our method needs to solve a sparse coding using Lasso [6] in the annotation stage. First, we need to compute the gram matrix $\mathbf{D}^{I\top}\mathbf{D}^{I}$ over the leaned dictionary, which can be pre-computed [23] beforehand and so does not impact the time complexity of MCDL. Then in the annotation stage, $\mathbf{D}^{I\top}x$ needs to be computed for a given input image with the time complexity of $O(K \times M)$. Then, a Cholesky-based algorithm with $O(K \times M^2)$ [6] should be done to find the coefficient over $\mathbf{D}^{I}$. Thus, the overall complexity of the MCDL approach in the annotate step is $O(K \times M + K \times M^2)$. Since the number of prototypes is much less than the number of training samples, the time complexity of 2PKNN is higher than MCDL.

Precision-Recall Curves

Figure 9 depicts Precision-Recall curves for our method against 2PKNN method, on four datasets. To obtain these results, we have changed the decision threshold ($\tau_{optimal}$) and the number of assigned labels in the annotation step of MCDL and 2PKNN, respectively. The intial threshold for MCDL is 1 and the initial number of the labels for 2PKNN is also 1. So, the precision and recall do not start from one and zero. By increasing the recall, i.e. increasing the positive labels, the precision goes through an upward trend (especially for IAPRTC-12 and ESP-GAME datasets). Then, it reaches a peak, before dropping to its minimum, where all the labels are considered positive. It stems from the fact that in contrast to the binary classification problems, in image annotation the precision and recall are computed over all the labels. Two noticable points can be concluded from the Figure 9. The initial impression is that the area under the precision-recall curve (AUCPR) of MCDL is larger than 2PKNN on IAPRTC-12 and ESP-GAME datasets, and almost similar on the Flickr datasets. The reason is that in MCDL, we marginalize the scores, therefore by increasing the recall the number of false positives drop, which results in higher precision for MCDL versus 2PKNN. The second impression is that the optimum point of all the datasets occur for the MCDL approach which shows the superiority of this method.

6.5 Discussion

There are some reasons that our work can outperform other existing methods. The first reason originates from prototype learning. Each prototype is a well-defined description of a set of visual features to summarize all
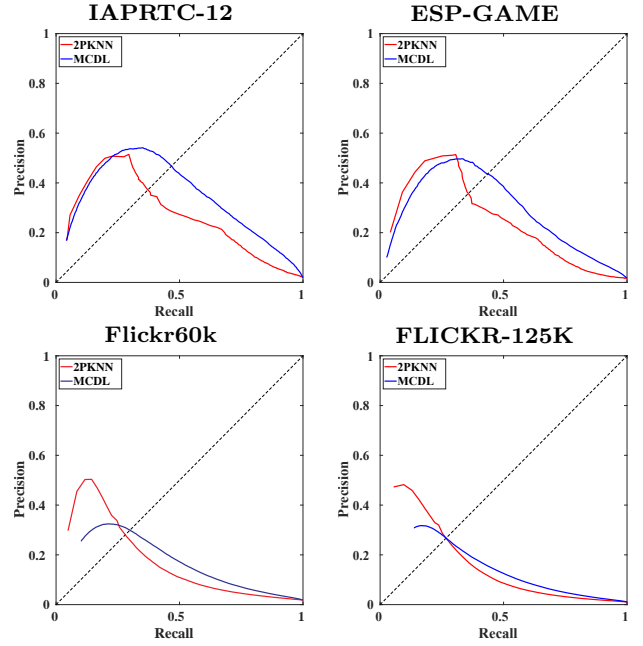

Fig. 9 Precision-Recall Curve for MCDL versus 2PKNN.

features of a dataset. By learning these prototypes, an input image could be well-reconstructed using the minimum prototypes. The second reason is learning semantic features. For any visual prototype, its corresponding semantic part determines the association degree of any label to that specific visual prototype. The next property of our approach is controlling the weight of each visual feature in prototype learning. In fact, the more labels for a visual feature (image in dataset), the less impact on the prototypes. This stems from the fact that an image with more labels consists of less details about its labels. The fourth is about hinge loss function and its marginalized penalty. In contrast to the mean squared error loss function, it ignores the penalties out of two upper and lower bounds for positive and negative labels, respectively. Therefore, the associated labels do not impact the loss value. Lastly, L-1 regularization imposes a constraint on the semantic sparsity, which results in allocating the least and most related labels to the prototypes. All these features together would result in finding the most informative prototypes equipped with the most related labels.

7 Conclusion and Future Work

There is a considerable redundancy in the visual and semantic contents of large-scale image datasets. MCDL provides an efficient strategy to summarize large datasets into a fewer number of prototypes with admissible accuracy. Experimental results show the superiority of the

proposed method in image annotation tasks. we can reach a 90% reduction in the annotation time while the performance is maintained or even improved in comparison to the search-based method. MCDL method provides these benefits: Firstly, it utilizes a two-step optimization algorithm for solving a non-convex objective function which yields informative prototypes. Secondly, a marginalized loss over labels' scores is utilized to increase the generalization of the learned prototypes. Finally, the other annotation methods could leverage the prototypes extracted by MCDL. For future work, it is worth mentioning that visual features can be projected into a low-dimensionality space by embedding a transformation matrix in the optimization process, as suggested in [15]. The instances with the same labels could have more consistent sparse representation through learning such a transformation matrix.

A Lemma 1. Convergence of Equation (5).

To prove the convergence of the update rule suggested in Equation (5), it can be shown easily that $f(\alpha_*^i) \leq g(\hat{\alpha}^i[s]) \leq f(\hat{\alpha}^i[s-1])$, where α_*^i is the optimum of Equation (3). First of all, w have (notice that $(2y_t^i - 1) \in \{-1, 1\}$):

$$g(\hat{\alpha}^i[s-1]) = \| x^i - \mathbf{D}^I \hat{\alpha}^i[s-1] \|_2^2$$

$$+ \frac{\lambda}{\mathcal{N}_+^i} \sum_{t=1}^{T} (\tilde{y}_t^i[s-1] - \mathbf{d}_t^L \hat{\alpha}^i[s-1])^2 = \| x^i - \mathbf{D}^I \hat{\alpha}^i[s-1] \|_2^2$$

$$+ \frac{\lambda}{\mathcal{N}_+^i} \sum_{\{t | \hat{\xi}_t^i[s-1] > 0\}} (\tau + (2y_t^i - 1)\mathbf{C} - \mathbf{d}_t^L \hat{\alpha}^i[s-1])^2$$

$$= \| x^i - \mathbf{D}^I \hat{\alpha}^i[s-1] \|_2^2 + \frac{\lambda}{\mathcal{N}_+^i} \sum_{t=1}^{T} (\hat{\xi}_t^i)^2 = f(\hat{\alpha}^i[s-1]).$$

$$(12)$$

Equation (12) means that $g(\alpha^i) \triangleq \lim_{\alpha^i \to \hat{\alpha}^i[s-1]} f(\alpha^i)$ is almost a smooth approximation of $f(\alpha^i)$ in the current estimate of MCSC. Moreover, since $\hat{\alpha}^i[s]$ is supposed to be the optimum of $g(\alpha^i)$, we have $g(\hat{\alpha}^i[s]) \leq g(\hat{\alpha}^i[s-1])$. Therefore, considering Equation (12), $g(\hat{\alpha}^i[s]) \leq f(\hat{\alpha}^i[s-1])$. Moreover, it is obvious that $f(\alpha^i) \leq g(\alpha^i)$, for all possible α^i, because the squared loss used in $g(\alpha^i)$ is greater than or equal to hinge loss. So, we can now confirm our first proposition and minimizing the primary optimization problem of Equation (3).

References

1. Bragantini, J., Falcão, A., Najman, L.: Rethinking interactive image segmentation: Feature space annotation. arXiv preprint arXiv:2101.04378 (2021)
2. Cao, X., Zhang, H., Guo, X., Liu, S., Meng, D.: Sled: Semantic label embedding dictionary representation for multilabel image annotation. IEEE Transactions on Image Processing **24**(9), 2746–2759 (2015)
3. Chen, M., Zheng, A., Weinberger, K.: Fast image tagging. In: International conference on machine learning, pp. 1274–1282 (2013)
4. Chen, Y., Liu, L., Tao, J., Chen, X., Xia, R., Zhang, Q., Xiong, J., Yang, K., Xie, J.: The image annotation algorithm using convolutional features from intermediate layer of deep learning. Multimedia Tools and Applications pp. 1–25 (2020)
5. Du, H., Zhang, Y., Ma, L., Zhang, F.: Structured discriminant analysis dictionary learning for pattern classification. Knowledge-Based Systems **216**, 106794 (2021)
6. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. The Annals of statistics **32**(2), 407–499 (2004)
7. Grubinger, M., Clough, P., Müller, H., Deselaers, T.: The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In: International workshop ontoImage, vol. 2 (2006)
8. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: 2009 IEEE 12th international conference on computer vision, pp. 309–316. IEEE (2009)
9. Hastie, T., Tibshirani, R., Wainwright, M.: Statistical learning with sparsity: the lasso and generalizations. CRC press (2015)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
11. Heo, J.P., Lin, Z., Yoon, S.E.: Distance encoded product quantization for approximate k-nearest neighbor search in high-dimensional space. IEEE transactions on pattern analysis and machine intelligence (2018)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708 (2017)
13. Huiskes, M.J., Thomee, B., Lew, M.S.: New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative. In: Proceedings of the international conference on Multimedia information retrieval, pp. 527–536. ACM (2010)
14. Jiang, Z., Lin, Z., Davis, L.S.: Learning a discriminative dictionary for sparse coding via label consistent k-svd. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 1697–1704. IEEE (2011)
15. Jing, X.Y., Wu, F., Li, Z., Hu, R., Zhang, D.: Multi-label dictionary learning for image annotation. IEEE Transactions on Image Processing **25**(6), 2712–2725 (2016)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
17. Li, H., Wang, Y., Yang, Z., Wang, R., Li, X., Tao, D.: Discriminative dictionary learning-based multiple component decomposition for detail-preserving noisy image fusion. IEEE Transactions on Instrumentation and Measurement **69**(4), 1082–1102 (2019)
18. Li, X., Shen, B., Liu, B.D., Zhang, Y.J.: Ranking-preserving low-rank factorization for image annotation with missing labels. IEEE Transactions on Multimedia **20**(5), 1169–1178 (2017)
19. Li, Z., Zhang, Z., Qin, J., Zhang, Z., Shao, L.: Discriminative fisher embedding dictionary learning algorithm for object recognition. IEEE transactions on neural networks and learning systems (2019)

20. Ling, J., Chen, Z., Wu, F.: Class-oriented discriminative dictionary learning for image classification. IEEE Transactions on Circuits and Systems for Video Technology **30**(7), 2155–2166 (2019)
21. Luo, Y., Yang, Y., Shen, F., Huang, Z., Zhou, P., Shen, H.T.: Robust discrete code modeling for supervised hashing. Pattern Recognition **75**, 128–135 (2018)
22. Ma, Y., Liu, Y., Xie, Q., Li, L.: Cnn-feature based automatic image annotation method. Multimedia Tools and Applications **78**(3), 3767–3780 (2019)
23. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research **11**(Jan), 19–60 (2010)
24. Murthy, V.N., Maji, S., Manmatha, R.: Automatic image annotation using deep learning representations. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 603–606. ACM (2015)
25. Putthividhy, D., Attias, H.T., Nagarajan, S.S.: Topic regression multi-modal latent dirichlet allocation for image annotation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3408–3415. IEEE (2010)
26. Rad, R., Jamzad, M.: A multi-view-group non-negative matrix factorization approach for automatic image annotation. Multimedia Tools and Applications **77**(13), 17109–17129 (2018)
27. Samih, H., Rady, S., Ismail, M., Gharib, T.: Improving natural language queries search and retrieval through semantic image annotation understanding. International Journal of Intelligent Computing and Information Sciences **20**(2), 67–78 (2021)
28. Shooroki, H.K., Chahooki, M.A.Z.: Selection of effective training instances for scalable automatic image annotation. Multimedia Tools and Applications **76**(7), 9643–9666 (2017)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
30. Song, P., Rodrigues, M.R.: Multimodal image denoising based on coupled dictionary learning. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 515–519. IEEE (2018)
31. Sun, Y., Loparo, K.: Context aware image annotation in active learning. arXiv preprint arXiv:2002.02775 (2020)
32. Sun, Y., Quan, Y., Fu, J.: Sparse coding and dictionary learning with class-specific group sparsity. Neural Computing and Applications **30**(4), 1265–1275 (2018)
33. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, pp. 6105–6114. PMLR (2019)
34. Tang, C., Liu, X., Wang, P., Zhang, C., Li, M., Wang, L.: Adaptive hypergraph embedded semi-supervised multi-label image annotation. IEEE Transactions on Multimedia **21**(11), 2837–2849 (2019)
35. Verma, Y.: Diverse image annotation with missing labels. Pattern Recognition **93**, 470–484 (2019)
36. Verma, Y., Jawahar, C.: Exploring svm for image annotation in presence of confusing labels. In: BMVC, pp. 25–1 (2013)
37. Verma, Y., Jawahar, C.: Image annotation by propagating labels from semantic neighbourhoods. International Journal of Computer Vision **121**(1), 126–148 (2017)
38. Von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 319–326. ACM (2004)
39. Wang, X., Gu, Y.: Cross-label suppression: A discriminative and fast dictionary learning with group regularization. IEEE Transactions on Image Processing **26**(8), 3859–3873 (2017)
40. Wright, S.J.: Coordinate descent algorithms. Mathematical Programming **151**(1), 3–34 (2015)
41. Wu, B., Jia, F., Liu, W., Ghanem, B.: Diverse image annotation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2559–2567 (2017)
42. Yang, Y., Zhang, W., Xie, Y.: Image automatic annotation via multi-view deep representation. Journal of Visual Communication and Image Representation **33**, 368–377 (2015)
43. Zhang, Q., Li, B.: Discriminative k-svd for dictionary learning in face recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2691–2698 (2010)
44. Zhang, Y., Wu, J., Cai, Z., Philip, S.Y.: Multi-view multi-label learning with sparse feature selection for image annotation. IEEE Transactions on Multimedia **22**(11), 2844–2857 (2020)
45. Zhao, F., Si, W., Dou, Z.: Image super-resolution via two stage coupled dictionary learning. Multimedia Tools and Applications **78**(20), 28453–28460 (2019)
46. Zhou, S., Liu, H., Cui, K., Hao, Z.: Jointly class-specific and shared discriminative dictionary learning for classifying surface defects of steel sheet. ISIJ International pp. ISIJINT–2021 (2021)
47. Zhu, J., Rosset, S., Tibshirani, R., Hastie, T.J.: 1-norm support vector machines. In: Advances in neural information processing systems, pp. 49–56 (2004)