remote sensing

MDPI

Article

Exploring Weighted Dual Graph Regularized Non-Negative Matrix Tri-Factorization Based Collaborative Filtering Framework for Multi-Label Annotation of Remote Sensing Images

Juli Zhang [1,*], Junyi Zhang [2], Tao Dai [2] and Zhanzhuang He [1]

[1] Xi'an Microelectronics Technology Institute, Xi'an 710068, China; hzz771@163.com
[2] School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China; zhangjunyi0806@stu.xjtu.edu.cn (J.Z.); zzj.ddt@stu.xjtu.edu.cn (T.D.)
* Correspondence: juli2320@sina.com; Tel.: +86-29-8860 9000-8203

Received: 14 March 2019; Accepted: 12 April 2019; Published: 16 April 2019

Abstract: Manually annotating remote sensing images is laborious work, especially on large-scale datasets. To improve the efficiency of this work, we propose an automatic annotation method for remote sensing images. The proposed method formulates the multi-label annotation task as a recommended problem, based on non-negative matrix tri-factorization (NMTF). The labels of remote sensing images can be recommended directly by recovering the image–label matrix. To learn more efficient latent feature matrices, two graph regularization terms are added to NMTF that explore the affiliated relationships on the image graph and label graph simultaneously. In order to reduce the gap between semantic concepts and visual content, both low-level visual features and high-level semantic features are exploited to construct the image graph. Meanwhile, label co-occurrence information is used to build the label graph, which discovers the semantic meaning to enhance the label prediction for unlabeled images. By employing the information from images and labels, the proposed method can efficiently deal with the sparsity and cold-start problem brought by limited image–label pairs. Experimental results on the UCMerced and Corel5k datasets show that our model outperforms most baseline algorithms for multi-label annotation of remote sensing images and performs efficiently on large-scale unlabeled datasets.

Keywords: multi-label annotation; remote sensing imagery; collaborative filtering; non-negative matrix tri-factorization

1. Introduction

Remote sensing image annotation is important in a wide range of remote sensing applications, such as environmental monitoring [1], remote sensing retrieval [2], and land use and land cover issues [3]. By assigning one or more predefined semantic labels to an image, image annotation provides mapping from images to semantic concepts. With the continuous development of modern satellite technology, many terabytes of images are delivered by satellite sensors every day. It is tedious and labor-intensive to manually annotate so many images. Meanwhile, rapidly developing remote sensing techniques are improving image resolution, which means that satellite images can provide more detailed geometrical information. This has completely changed the perspective of the traditional remote sensing image annotation task. On the one hand, these images contain much more semantic information, which increase storage cost and makes it difficult to annotate vast volumes of remote sensing images manually. On the other hand, annotating one satellite image with a single label does not fit images with complex semantic concepts. Providing one image with more than one label (multi-label)

can help to describe the image in more detail at the semantic level, which is useful in image retrieval and understanding. This is also a trend for remote sensing image applications. Therefore, an effective and efficient automatic multi-label annotation method for remote sensing images is urgently needed by the remote sensing community.

Most existing remote sensing image annotation methods are based on the visual content of images [4–6]. These methods always first extract low-level visual features and then associate these features with high-level semantic concepts. Because these methods only utilize content information, they intrinsically have a limited ability to deal with the semantic gap [7,8], which happens because visual contents may not be powerful enough to abstract the semantic content of images. Recently, the use of predefined semantic concepts to perform image annotation has received increasing interest [9–12]. A topic-model method is described in [9], which treats an image as a collection of visual words. However, visual words often carry limited semantic information, and excavating efficient visual words for remote sensing images is very time-consuming. Multiple labels for an image provide more semantic information, which indirectly describe the relationships among labels. To make full use of more semantic (multi-label) information, a direct way is to annotate the images with label co-occurrence in the whole dataset. This is similar to collaborative filtering (CF). Given an image with one or more existing labels, additional labels for the image can be predicted by exploiting the correlation between labels. We refer to this method as collaborative image annotation. However, there are three challenging issues with this method: Scalability, sparsity, and cold-start problems. For the scalability problem, using the traditional CF method to search the k most similar neighbors is time-consuming on large datasets. For the sparsity problem, it is hard to execute CF efficiently with very little labeled information, which in turn reduces the annotation performance. Compared with scalability and sparsity, cold-start is a more serious problem in collaborative image annotation, in which label recommendations are required for images with no observed labels. This is commonly found in some recommending systems, in which newly registered users are called cold-start users. However, for image annotation, more cold-start images mean there are many unlabeled images in the dataset. The pure CF framework cannot work well under this cold-start setting, since no preference information from labels or images is available for label recommendations.

Due to its high efficiency, the non-negative matrix factorization (NMF) [13] based model has become one of the most popular collaborative filtering approaches [14–16] in recommending systems. This model maps both users and items to the same latent low-rank feature space, and then predicts the unknown ratings by the product of these learned features. There are two advantages to using this model. First, due to the sparsity of the rating matrix, the dimensionality of the learned latent feature space can be set small, without impairing accuracy. Second, the storage complexity is low, and the related issues are easy to solve in real applications. Recently, non-negative matrix factorization-based image annotation approaches [17–20] have become particularly attractive and achieved good performance. In [19], an NMF-based image annotation framework was proposed to discover the latent space of data. It uses multi-view graph regularized NMF to factorize data into a set of non-negative basis and coefficients, except in extracting multiple features. In [20], a semi-supervised framework based on graph embedding and multi-view NMF was proposed for automatic image annotation, which suffers a high data dimension problem. However, there is a basic assumption in NMF that user vectors and item vectors exist in a common space. This limitation makes NMF not quite fit for annotating images directly without extra image and tag information. Non-negative matrix tri-factorization (NMTF) [21] is a good choice in addressing this problem. It maps images and tags in two different dimensional spaces. Additionally, NMTF-based methods have a better performance than NMF-based methods in collaborative filtering to some extent, which was proved in [22,23]. Consequently, the same as the other CF methods, NMTF-based methods also suffer the sparsity and cold-start problems.

To solve the problems mentioned above in remote sensing image annotation, we propose a novel method based on a collaborative filtering framework. This method formulates the multi-label problem as a multi-label recommendation problem. We refer to this method as weighted dual graph regularized

non-negative matrix tri-factorization (WDG-NMTF). To our knowledge, this is the first work to employ NMTF to solve the remote sensing image multi-label problem. The overview of the proposed method is illustrated in Figure 1. To illustrate the method from the perspective of collaborative filtering, first we construct an image–label matrix as a user-item rating matrix, in which each row and column denotes one image, and one class label, respectively. This matrix displays the semantic relationship between images and labels. Since a pure CF approach can always be regarded as a content-free model that does not directly look into the visual content of images, this will be no help for alleviating the sparsity and cold-start problems. Therefore, we combine the NMTF-based CF framework with image and label information simultaneously. To fully utilize the contents provided in a dataset, we then construct an image graph and a label graph to build the relationships for images and labels. The same as the other image annotation method, a semantic gap is also a challenging issue. To bridge the semantic gap between semantic concepts and visual contents, we build the image graph by employing both low-level visual content features and high-level semantic features. This not only helps to solve the cold-start problem, but also increases the overall accuracy. To further improve the annotation performance, the extracted image graph and label graph are both used to regularize NMTF. Finally, labels are recommended from the recovered image–label matrix.

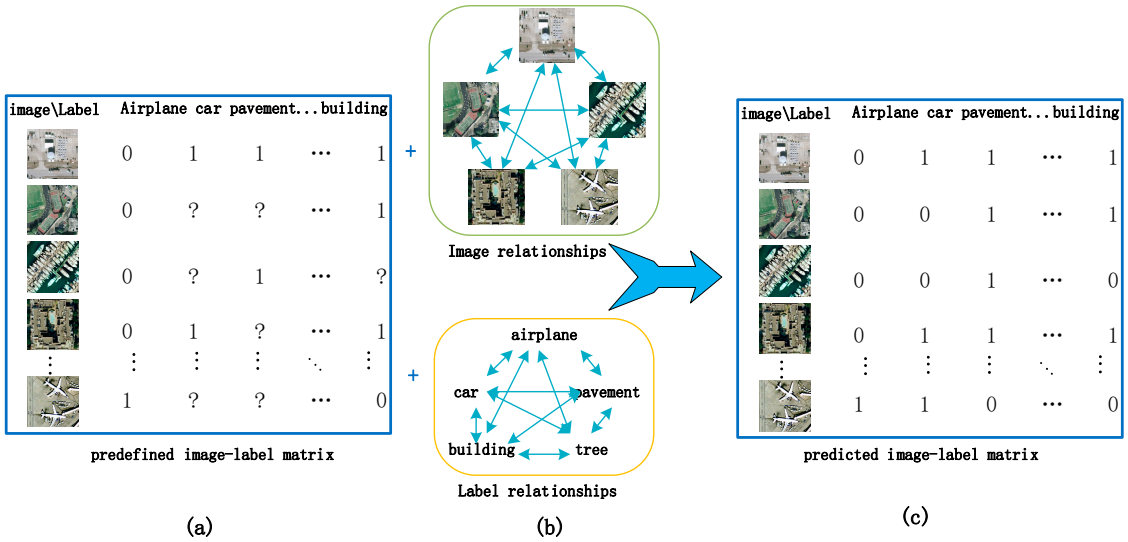


Figure 1. Illustration of the proposed model: (**a**) image–label matrix; (**b**) image graph and label graph; (**c**) predicted image–label matrix.

The main contributions of the proposed method are as follows:

1. We propose a novel NMTF-based automatic remote sensing image annotation method that formulates the image annotation issue as an image recommendation problem.
2. We extended the NMTF by using both the image graph and label graph to enhance the annotation performance.
3. We build the image graph using both high-level semantic features and low-level visual features of images to reduce the semantic gap, which can fully excavate image information contained in a dataset.
4. Thorough experimental studies on two image datasets are carried out to validate the performance of the proposed method.

The remainder of this paper is organized as follows: We first discuss the related literature on automatic image annotation and NMF-based collaborative filtering in Section 2. Then, we describe the proposed WDG-NMTF method in Section 3. In Section 4, we discuss the optimization process and the convergence of the proposed method, and give the annotation procedure via WDG-NMTF.

The experimental results and discussion are given in Sections 5 and 6, respectively. Finally, we conclude this paper in Section 7.

2. Related Works

In this section, we discuss prior research related to our work, including automatic image annotation and NMF-based collaborative filtering.

2.1. Automatic Image Annotation

The aim of image annotation is to provide images with relevant labels. There are three main types of image annotation techniques: Generative models [9,12,24–27], discriminative models [28,29], and the nearest neighbor method [30].

Generative models usually use latent topics to represent image–label relationships, such as variants of latent Dirichlet allocation (LDA) [24], author topic model [9,12], constrained non-negative matrix factorization [25,26], and tag completion [27]. In [24], the authors used the LDA model to annotate satellite images. In [9] and [12], the authors used the label information in an author-topic model and genre information in an author-genre topic model to improve image annotation performance. A probabilistic method in [25] was proposed to fill the missing tags of images via collaborative filtering. In [26], a constrained NMF method, incorporating label information, was introduced to solve a multi-label learning problem. Similar to ours, an image-tag completion algorithm was employed in [27] to solve the missing tags of the images. However, they used linear learning, while we used an NMF-based CF framework. Moreover, they addressed the image retrieval problem, while we solved the multi-label remote sensing image annotation problem.

Discriminative models usually learn individual classifiers based on low-level visual features for each label, such as a support vector machine (SVM) [28], boosting [29,31], and multi-instance multi-label learning (MIML) [32]. These methods are very different from our method. SVM is well known and used widely in binary classifiers. Multi-label SVM usually uses the one-against-rest scheme to realize multiclass classification. This can be achieved by combining the predictions resulting from multiple binary classifiers. Although SVM-based multi-label methods have obtained large margins and good performance, they need many labelled samples to train the classifiers and therefore cannot efficiently deal with multi-label classification with missing labels. Boosting-based methods also need different kinds of heterogeneous features to boost the annotation performance. MIML is a framework for supervised classification. These methods all need large-scale labeled samples to train related models, then induce the labels for the unlabeled data. This procedure is very time-consuming. In contrast, we directly annotate the unknown labels of images without an extra classifier.

Nearest neighbor–based models use the visual similarities among the nearest neighbors to predict the unknown labels of images [30,33]. These methods depend heavily on the metric of the local distributions of samples, which can lead to poor decision boundaries and affect the annotation performance in turn.

Recently, several studies proposed methods to solve remote sensing multi-label annotation problems and the performance showed promise [3,10,11,34,35]. The land cover problem was solved in [3] by inferring the complex relationships between acquired satellite images and spectral profiles of different surface materials. In [10], the authors solved the automatic semantic annotation problem for high-resolution images by proposing a unified annotation framework. The framework combines discriminative high-level feature learning and weakly supervised feature transferring. It uses deep learning to learn the high-level features and transfers the learned features to perform annotation. In [11], a hierarchical semantic multi-instance multi-label learning framework was presented for remote sensing image annotation tasks. It represents the ambiguities between image contents and semantic labels, and then builds the semantic relationships contained in the image. It utilizes the prior knowledge of images in a Gaussian process to improve the performance. A multi-label classification approach, based on low-rank representation, was proposed in [34], which uses a low-rank representation in feature

space to compute the low-rank constrained coefficient matrix. Then it defines a feature-based graph and captures the global relationships between images. After that, a semantic graph is constructed, and finally, it combines feature graph and semantic graph to train a multi-label classifier. Our method is similar to this method, due to the graph-based model, but this method uses low-rank representations for images and induces the labels for unlabeled images, by using both image graph and feature graph. We used the NMTF method, with image and label graph, to recommend labels for images.

2.2. NMF-Based Collaborative Filtering

Owing to their high accuracy and scalability, NMF-based CF models have been attracting more attention in many areas [15,16,22,36]. In addition, some extra techniques have been added to the NMF-based models to improve the performance of the algorithm. The representative works include the maximum-margin MF-based model [37], Singularly Valuable Decomposition (SVD) model [38], weighted NMF model [14,22,39], and graph-based methods [2,40–44]. Moreover, the idea of NMF-based CF has been introduced to solve real application in many domains, e.g., content-based image retrieval [5], image clustering [45], and social recommendation systems [14,16,22]. Because two-factor NMF often gives rather poor low-rank matrix approximation [46], one more factor can improve the scalability of the original factorization. As an extension of NMF, NMTF was first proposed in [21] to co-cluster rows and columns simultaneously, demonstrating its usefulness in co-cluster applications. Due to its encouraging empirical results, NMTF has been widely used in data clustering [47,48], collaborative recommendation systems [22], and social network community detection [49]. These works achieved promising performance by factorizing the original matrix into three matrices. This is because NMTF can integrate different forms of structure constraints on the factors to achieve more interpretable results.

Generally speaking, NMF-based collaborative filtering builds a semi-supervised training process. This process can be optimized by a global loss function between the known ratings and corresponding entries in the learned matrices. During the training process, constraints on the standard NMF in dealing with real application problems have proven to be essential to the performance [14,26,42,45]. However, these methods only focus on one side of information in the factorization process. We propose weighted non-negative matrix tri-factorization combined with dual graphs of images and labels to utilize more information. This will help us find more interpretable low-rank approximations. Furthermore, to reduce the semantic gap, we utilize both low-level visual features and high-level semantic features of images, which can improve the performance of the proposed method as well.

3. Methodology

3.1. Problem Formulation

Before going any further, let us first introduce the problem formulation and notations used in this paper. The remote sensing image annotation problem is formally defined as follows: Suppose there are c labels and m remote sensing images. Our goal was to automatically annotate x unlabeled remote sensing images according to the provided labeled images. Each image was abstracted by a data point, which was annotated with a number of class labels. We consider the non-negative data matrix $G \in \mathbb{R}_{+}^{m \times c}$ as the image–label matrix, whose rows are images and columns are labels. To be specific, for the j^{th} label, we define $g_{ij} = 1$ to indicate that the i^{th} image is related to the j^{th} label and $g_{ij} = 0.001$ otherwise.

3.2. Image Annotation via Non-Negative Matrix Trifactorization-Based Model

NMTF was first introduced in [21] to solve the clustering problem. In this model, the original matrix was factorized into 3 matrices with orthogonal constraints, and achieved promising performance in clustering. Motivated by this, we solved the remote sensing image multi-label annotation problem by using the NMTF-based CF framework. In this framework, we formulated the image annotation as a CF recommendation problem. Image labels can be recommended by the reconstruction of an

image–label matrix $G \in \mathbb{R}_{+}^{m \times c}$ that can be seen as the "user-item" rating matrix in a CF recommending system. The recommendation process has 2 steps. Factorizing the original image–label matrix into 3 matrices is the first step. After that, the new image–label matrix can be recovered by the product of the learned 3 matrices; and learning the 3 matrices is critical for the whole method. We formulated the learning procedure by solving the following optimization problem:

$$\underset{A,V,B \geq 0}{\operatorname{argmin}} J(A,V,B) = \frac{1}{2}\|G - AVB\|_F^2 \tag{1}$$

where $A \in \mathbb{R}_{+}^{m \times k}$ is the learned image feature matrix and $k < \min(m,c)$ denotes the number of latent image features; $B \in \mathbb{R}_{+}^{n \times c}$ is the learned label feature matrix, where $n < \min(m,c)$ denotes the number of latent label features. $k = n$ is a special case. Additionally, $V \in \mathbb{R}_{+}^{k \times n}$ is a condensed view of G.

In this process, we use NMTF to discover 2 low-rank latent feature matrices: Image feature matrix A and label feature matrix B. We describe the NMTF-based CF method for multi-label image annotation in Figure 2. However, as introduced in Section 1, the sparsity and cold-start problems must be solved in the process of matrix factorization. With respect to sparsity, the image–label matrix G is sparse because of the missing label information for some images. Cold-start is a more serious sparsity problem because the unlabeled images provide no label information. Due to these problems, Equation (1) may fail to seek more appropriate latent feature matrices that can perfectly recover the image–label matrix. This indirectly affects the annotation performance. Utilizing more preferable information from images and labels is a natural solution. To make full use of the information from a dataset, we constructed an image graph and a label graph. More specifically, we seek to make 2 image-specific latent feature vectors in matrix A as similar as possible, if the corresponding images have similar label information. We also seek to make the latent label features in matrix B as similar as possible if the corresponding labels are labeled with the same images. With these goals, we employ 2 graph regularization terms to regularize the NMTF procedure by modeling the image and label graphs. These terms are 2 constraints imposed on Equation (1) to force it to learn the most meaningful features for the multi-label annotation problem.

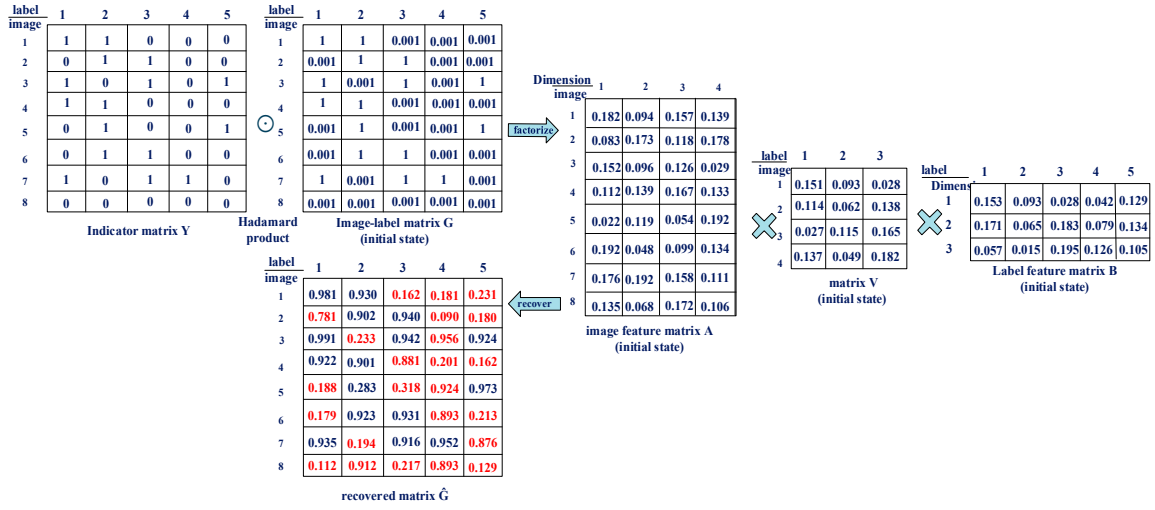


Figure 2. Multi-label image annotation via non-negative matrix tri-factorization (NMTF)-based model.

In the following, we describe how to model the image graph and label graph. After that, we will provide the final objective function for the proposed method.

3.3. Modeling Image Graph

In this subsection, we detail the construction of the image graph. There is no doubt that 2 images embedded in latent space should maintain their relationships. In order to model the relationships

between images, we constructed an image–image graph $G_A = (\nu_A, \varepsilon_A)$, whose vertex set ν_A corresponds to images $\{x_1, x_2, \ldots, x_m\}$. In this graph, each node represents an image in the dataset, and an edge denoted W_{ij} represents the affinity between image x_i and image x_j. It is obvious that W_{ij} is relatively large when x_i and x_j are close, while their new representations, a_i and a_j, in the new space should be close too, which can be formulated as follows:

$$O_1 = \frac{1}{2}\sum_{i,j=1}^{m} \left\|a_i - a_j\right\|^2 W_{ij}^A = Tr(A^T L_A A).$$

where $Tr(.)$ denotes the trace of the matrix. $W^A = \left[W_{ij}^A\right]$ is a symmetric non-negative similarity matrix representing the weights of the edges. $D^A \in \mathbb{R}^{m\times m}$ is a diagonal matrix and $D_{ii}^A = \sum_{j=1}^{m} W_{ij}^A$, and $L_A = D^A - W^A$ is the Laplacian matrix [50] of the image graph. The method for constructing the adjacency matrix W^A is the crucial part of graph regularization, because W^A encodes useful information about images that can be used to improve the factorization performance. In this paper, we considered 2 useful pieces of information about the image: Semantic information and visual content feature information.

3.3.1. Extraction of Semantic Similarity Information

Here, we used the semantic similarity information of images to facilitate getting more interpretable low-dimensional representations. The adjacency matrix of an image graph using semantic similarity is defined as follows:

$$W_{ij}^{A_{ss}} = sim_{SS}(x_i, x_j)$$

In our method, we assumed 2 images are similar if they have similar labels, and we used the number of labels that are co-labeled by 2 images to calculate the semantic similarity between them. We assumed L_{x_i} and L_{x_j} are the label sets that images x_i and x_j, respectively, are labeled with. Then the semantic similarity can be defined as:

$$sim_{SS}(x_i, x_j) = \frac{\left|L_{x_i} \cap L_{x_j}\right|}{\left|L_{x_i} \cup L_{x_j}\right|}.$$

The Laplacian graph by semantic similarity information can be defined as $L^{A_{ss}} = D^{A_{ss}} - W^{A_{ss}}$, where $D_{ii}^{ss} = \sum_j W_{ij}^{ss}$ is the diagonal degree matrix.

3.3.2. Extraction of Visual Content Information

The human visual system (HVS) understands images mainly based on their low-level features, such as color, shape, and edges. Choosing suitable visual features for computer vison tasks heavily affects the performance. In this paper, to compute feature similarity, we used both global and local features to represent each image, which have been used in previous works [17,51] and achieved promising results. Global features used here consist of Gist feature [52] and color histograms of red/green/blue (RGB), while local features include Scale Invariant Feature Transform (SIFT) [53] and robust hue descriptors [54]. Local features are extracted densely from both Harris Laplacian interest points and multiscale grid. We followed the work in [51], using L2 to normalize the GIST features and L1 for other descriptors. These feature descriptors are concatenated to make new descriptors for images, providing preference visual information. Then we used f_i^A to denote the new feature descriptor of the i^{th} image, which characterizes the visual information of one image. Therefore, we can define the feature similarity of visual content information as:

$$W_{ij}^{A_{vs}} = sim_{VS}(f_i^A, f_j^A).$$

where $sim_{VS}(f_i^A, f_j^A)$ measures the similarity of the feature vectors of the i^{th} and j^{th} images.

Due to its clear theoretical meaning and our experimental results in finding the proper function for calculating similarity, we used cosine similarity as the feature similarity measure, which can be defined as follows:

$$sim_{VS}(f_i^A, f_j^A) = \frac{\left\langle f_i^A, f_j^A \right\rangle}{\|f_i^A\| \|f_j^A\|}$$

where $\langle . \rangle$ denotes the inner product of the 2 feature vectors.

The Laplacian matrix graph according to visual content similarity is defined as $L^{A_{vs}} = D^{A_{vs}} - W^{A_{vs}}$, where $D_{ii}^{vs} = \sum_j W_{ij}^{vs}$ is the diagonal degree matrix.

3.3.3. Constructing the Image Similarity Matrix

The original image–label matrix G is a sparse matrix, which leads to a sparse $L^{A_{SS}}$. Therefore, we combine semantic similarity with visual content similarity, which can be defined as:

$$L_A = \sigma L^{A_{ss}} + (1 - \sigma) L^{A_{vs}} \tag{2}$$

where $0 \leq \sigma \leq 1$ weights the importance of image similarity. Specifically, when $\sigma = 0$, it only uses image feature similarity information in the image graph. When $\sigma = 1$, it only uses semantic information in the image graph.

3.4. Modeling Label Graph

Similarly, we constructed an undirected weighted graph $G_B = (\nu_B, \varepsilon_B)$, in which each node represents a label and each edge represents the affinity between 2 labels. We use $\{l_1, l_2, \ldots l_c\}$ to denote the labels. The graph regularization of the label graph is formulated as follows:

$$O_2 = \frac{1}{2} \sum_{i,j=1}^{c} \left\| b_i - b_j \right\|^2 W_{ij}^B = Tr(BL_B B^T)$$

where $W^B = \left[W_{ij}^B \right]$ encodes label information, $D_{ii}^B = \sum_{j=1}^{c} W_{ij}^B$ is a diagonal matrix, and $L_B = D^B - W^B$ is the Laplacian matrix of the label graph.

We assume that if 2 labels are co-labeled by some common images, then they probably will be co-labeled by other images. According to this assumption, we define the adjacency matrix on the label graph as follows:

$$W_{ij}^B = \begin{cases} sim(l_i, l_j), & if\ l_i, l_j \in L(x_i)\ or\ l_i, l_j \in L(x_j) \\ 0, & \text{otherwise.} \end{cases}$$

where $l_i, l_j \in L(x_i)$ means label l_i and l_j appear in the label set of image x_i synchronously, which means 2 labels are related to the same image; and $sim(l_i, l_j)$ denotes the similarity between 2 labels. Let $T = [t_1, \ldots, t_m]^T$ denote an m-vector for the m images in the dataset. $t_i \in \{0, 1\}^c (1 \leq i \leq c)$ is a binary vector for the i^{th} image, in which $t_i(l_i) = 1$ if the i^{th} image belongs to label l_i, and 0 otherwise. According to this, we can calculate the label similarity using cosine similarity, similar to [55]:

$$sim(l_i, l_j) = \cos(t(l_i), t(l_j)) = \frac{\left\langle t(l_i), t(l_j) \right\rangle}{\|t(l_i)\| \|t(l_j)\|}.$$

where $\left\langle t(l_i), t(l_j) \right\rangle$ counts the common images annotated with labels l_i and l_j.

3.5. Objective Function of WDG-NMTF

We integrated the 2 graphs into Equation (1) to regularize the extraction of image features and label features. The objective function can be defined as follows:

$$\underset{\text{s.t.}A\geq 0,V\geq 0,B\geq 0}{L(A,V,B)} = J(A,V,B) + \frac{\alpha}{2}O_1 + \frac{\beta}{2}O_2 = \frac{1}{2}\|G - AVB\|_F^2 + \frac{\alpha}{2}Tr(A^T L_A A) + \frac{\beta}{2}Tr(BL_B B^T) \quad (3)$$

where α and β are additional regularization parameters used to balance the information from the image similarity and label co-occurrence, respectively. When $\alpha = 0$, Equation (3) degenerates to GNMTF, and when $\alpha = \beta = 0$, Equation (3) degenerates to the standard NMTF.

There is often a risk of overfitting when the image–label matrix is very sparse. The cold-start problem is a special example. To avoid this risk and to enhance the robustness of the proposed method, we took into account the Tikhonov regularization terms in the above equation. Moreover, to enhance its scalability, we introduced an indicator matrix and obtained our final objective function as:

$$\underset{s.t.A\geq 0,V\geq 0,B\geq 0}{L(A,V,B)} = \frac{1}{2}\|Y \odot (G - AVB)\|_F^2 + \frac{\alpha}{2}Tr(A^T L_A A) + \frac{\beta}{2}Tr(BL_B B^T) + \frac{\lambda}{2}(\|A\|_F^2 + \|B\|_F^2) \quad (4)$$

where λ is a regularization parameter and Y is an indicator matrix in which each entry can be defined as follows

$$Y_{ij} = \begin{cases} 1, & if\ g_{ij}\ is\ observed; \\ 0, & if\ g_{ij}\ is\ unobserved. \end{cases}$$

4. Optimization Process and Image Annotation

The multi-label image annotation problem can be mathematically formulated by Equation (4), which can be considered as an optimization problem. As we can see, the objective function in Equation (4) is minimized with respect to A, B, and V. It is not a strict convex function about A, B, and V together. We cannot provide a closed-form solution directly. In the following, we optimize the objective function by an iterative multiplicative updating scheme, based on the gradient descent. Then, we provide convergence proof under the iterative updating rules. Finally, we describe the image annotation by the proposed method and analyze the time complexity.

4.1. Optimization Process

In order to optimize a three-factor objective function by the iterative updating scheme, we need to optimize it with respect to one variable by fixing the other two. To achieve this, we rewrite Equation (4) as follows:

$$\begin{aligned} L(A,V,B) = {} & \tfrac{1}{2}Tr(Y \odot GG^T) - Tr(Y \odot GB^T V^T A^T) + \tfrac{1}{2}Tr(Y \odot (AVB)B^T V^T A^T) \\ & + \tfrac{\alpha}{2}Tr(A^T L_A A) + \tfrac{\lambda}{2}Tr(A^T A) + \tfrac{\beta}{2}Tr(BL_B B^T) + \tfrac{\lambda}{2}Tr(B^T B) \end{aligned} \quad (5)$$

Thus, the partial derivatives of Equation (5) with respect to A, V, and B are

$$\begin{aligned} \tfrac{\partial L(A,V,B)}{\partial A} &= -Y \odot GB^T V^T + (Y \odot (AVB))B^T V^T + \alpha L_A A + \lambda A \\ \tfrac{\partial L(A,V,B)}{\partial V} &= -A^T (Y \odot G)B^T + A^T (Y \odot AVB)B^T \\ \tfrac{\partial L(A,V,B)}{\partial B} &= -V^T A^T (Y \odot G) + V^T A^T (Y \odot (AVB)) + \beta L_B B + \lambda B \end{aligned}$$

Because L_A and L_B may take any sign, we decompose them as $L_A = L_A^+ + L_A^-$ and $L_B = L_B^+ + L_B^-$, where $M_{ij}^+ = (|M_{ij}| + M_{ij})/2$ and $M_{ij}^- = (|M_{ij}| - M_{ij})/2$. Then we use the Karush-Kuhn-Tucker (KKT) complementary conditions [56] for the non-negativities of A, V, and B, and get

$$\begin{aligned}
&[-Y \odot GB^T V^T + Y \odot (AVB)B^T V^T + \alpha L_A^+ A + \lambda A - \alpha L_A^- A]_{ij} A_{ij} = 0 \\
&[-A^T (Y \odot G)B^T + A^T (Y \odot (AVB))B^T]_{ij} V_{ij} = 0 \\
&[-V^T A^T (Y \odot G) + V^T A^T (Y \odot AVB) + \beta L_B^+ B + \lambda B - \beta L_B^- B]_{ij} B_{ij} = 0
\end{aligned} \tag{6}$$

According to Equation (6), we can derive the following multiplicative updating rules:

$$A_{ij} \leftarrow A_{ij} \sqrt{\frac{[Y \odot GB^T V^T + \alpha L_A^- A]_{ij}}{[Y \odot (AVB)B^T V^T + \alpha L_A^+ A + \lambda A]_{ij}}} \tag{7}$$

$$V_{ij} \leftarrow V_{ij} \sqrt{\frac{[A^T (Y \odot G)B^T]_{ij}}{[A^T (Y \odot (AVB))B^T]_{ij}}} \tag{8}$$

$$B_{ij} \leftarrow B_{ij} \sqrt{\frac{[V^T A^T (Y \odot G) + \beta L_B^- B]_{ij}}{[V^T A^T (Y \odot AVB) + \beta L_B^+ B + \lambda B]_{ij}}} \tag{9}$$

Regarding the three multiplicative updating rules, we have the following theorem:

Theorem 1. *For $G, A, V, B \geq 0$, the iteration of updating rules in Equations (7)–(9) will lead the object function in Equation (4) converges to a local minimum.*

For better flow of the paper, we provide the proof of this theorem in Appendix A. The proof follows the idea used in [57], which uses an auxiliary function to prove the convergence. Moreover, the multiplicative updating rules stated in Equations (7)—(9) are the special cases of gradients descent with automatic learning steps. Thus, the successive iterations lead the objective function converges to a local minimum. Theorem 1 guarantees that the objective function will find a local minimum under the updating rules.

4.2. Image Annotation

After learning all the parameters and the matrices A, B and V, we will get an approximation of G by the product of the three matrices, which is defined as matrix $\hat{G}$. In this matrix, each value indicates the possibility of the image annotated by the label. To predict the labels of unknown images, we annotate each image with the largest 5 or 10 values of the related column of recovered matrix $\hat{G}$. We summarize the overall procedure of the proposed method in Algorithm 1.

The time complexity of Algorithm 1 is dominated by two parts: The matrix factorization procedure and the image-label matrix reconstruction procedure. The former mainly imposes costs on the multiplicative updating rules in steps 8–13 and the construction of image and label graphs in steps 3–6. We suppose the multiplicative updates stop after t iterations, thus the complexity of multiplicative updates is $O(tR_k(mknc))$, where R_k denotes the number of observed entries in a given training image–label matrix, m is the number of images and c indicates the number of label categories. Moreover, k and n are the dimensions of the learned latent image features and label features, respectively. Because $k, n < \min(m, c)$, the time cost of updates is approximately $O(tR_k mc)$. c is also much smaller than R_k, so it is $O(tR_k m)$. The construction of image graph and label graph spends $O(2m^2)$, and, $O(c^2)$ respectively. Therefore, matrix factorization costs approximately $O(c^2 + 2m^2 + tR_k m)$ in total. The image-label matrix reconstruction mainly implements the product of the three learned matrices, thus its complexity is $O(mkn + mnc)$. Since k and n are much smaller than m and c, c is smaller than m as well, the time cost of reconstruction approximately equals $O(m)$. Therefore, the overall time complexity of Algorithm 1 is $O(c^2 + 2m^2 + tR_k m + m)$.

Algorithm 1. Image annotation of weighted dual graph regularized non-negative matrix tri-factorization (WDG-NMTF).

1. Input:
   - G image–label matrix with missing values as null
   - W_A^{vs} image similarity matrix based on visual content
   - W_B label similarity matrix based on label co-occurrence in the training dataset
   - t step of iterations
   - $N_{\max}$ maximum number of iterations
   - N_{top} number of top annotations
   - k number of latent image features
   - n number of latent label features
   - σ ratio of image semantic similarity
   - m number of images
   - c number of labels
   - ε tolerance of stopping criterion
2. Output: $A \in \mathbb{R}^{m\times k} \geq 0, V \in \mathbb{R}^{k\times n}, B \in \mathbb{R}^{n\times c} \geq 0, \hat{G} \in \mathbb{R}^{m\times k} \geq 0$
3. Construct weight matrix Y using G such that $Y_{ij} = 1$ if $G_{ij} = 0.001$, otherwise $Y_{ij} = 0$
4. Compute image similarity matrix W_A based on the dataset
5. Compute Laplacian matrix L_A based on W_A and W_A^{vs}
6. Compute Laplacian matrix L_B based on W_B
7. Initialize $A_0 \in \mathbb{R}^{m\times k} \geq 0, V_0 \in \mathbb{R}^{k\times n} \geq 0, B_0 \in \mathbb{R}^{n\times c} \geq 0, \alpha, \beta, \lambda, \sigma > 0$
8. Repeat
9. Update A^{t+1} according to Equation (7)
10. Update V^{t+1} according to Equation (8)
11. Update B^{t+1} according to Equation (9)
12. $t \leftarrow t + 1$
13. Until {the stopping criterion $\frac{\|L(A,V,B)^{t+1} - L(A,V,B)^t\|_F}{\|L(A,V,B)^t\|_F} < \varepsilon$ is satisfied or maximum number of iterations $N_{\max}$ is achieved}
14. Take $\hat{G} = AVB$ as the approximation of G
15. Return a tag recommendation list of N_{top} tags with the largest values in $\hat{G}$ for each test image.

5. Experimental Results and Analyses

In this section, we conducted several experiments to evaluate the performance of the proposed WDG-NMTF method on several real-word datasets.

5.1. Dataset Description

To evaluate our proposed model, we chose two image datasets: University of California, Merced (UCMerced) and Corel5k.

The UCMerced dataset has 2100 images in 21 categories, each 256 × 256 pixels in size. The spatial resolution is 30 cm per pixel. The dataset is often used to analyze land cover use. We downloaded the multi-label images from [58], in which each image was relabeled with one or more labels. Then we relabeled the images with class labels. We used 21 land use classes as extra labels for each image. This means the ground truth of each image was added as a class label and the first letter of the class name was capitalized. To construct the image–label matrix G, we set the related column as 1 if the image belonged to this class. This can be used for land use classification as well. The total number of distinct class labels was 17. After adding the class labels, the average number of labels for each image was 4.334. The labels included airplane, bare soil, buildings, cars, chaparral, court, dock, field, grass, mobile home, pavement, sand, sea, ship, tanks, trees, water. Some examples with multi-labels are shown in Figure 3.

Corel5k consists of 5499 annotated images, available at http://lear.inrialpes.fr. In Corel5K, each image is labeled with one to five labels. The average number of annotated labels per image is 3.5.

Table 1 shows statistics of the two datasets. Cardinality in Table 1 measures the average number of labels for each sample, while label density means the average number of labels for samples in the dataset divided by the number of labels.

Figure 3. Examples of images and related labels in the University of California, Merced (UCMerced) dataset. Each image can have several semantic labels, which can be used to construct the image–label matrix. Words in blue are class names of images.

Table 1. Associated properties of multi-label datasets used in the experiments.

Name	Instances	Labels	Cardinality	Density	Train	Test
UCMerced	2100	38	4.334	0.114	1800	300
Corel5k	5000	374	3.522	0.009	4500	499

5.2. Evaluation Metrics

Multi-label image annotation performance is usually evaluated by comparing the label sets predicted for the test set with the human-produced ground truth.

To permit a quantitative evaluation of the effectiveness of the proposed method, the performance of multi-label remote sensing image annotation in this paper is evaluated in terms of the following five metrics: Precision, recall, F1 score, Hamming loss, and mean average precision. F1 score, one of the most popular evaluation measures, is calculated as follows:

$$F_1 - score(l_i) = 2\frac{\text{Precision}(l_i) \times \text{Recall}(l_i)}{\text{Precision}(l_i) + \text{Recall}(l_i)}$$

$$precision(li) = \frac{N_{correct}}{N_{labeled}}, recall(li) = \frac{N_{correct}}{N_{all}}$$

where $N_{correct}$ is the number of images that are correctly annotated, $N_{labeled}$ denotes the number of correct images within ground-truth annotations, and N_{all} is the number of images automatically labeled.

From the above equations, we can see that F1 score is the harmonic average of precision and recall, and will be more reliable.

To compare some multi-label learning–based methods for image annotation, we also considered two other common measures: Hamming loss and mean average precision. Hamming loss checks the difference between the predicted label set and the ground-truth label set. The smaller the value of Hamming loss, the better the performance will be. It can be defined as follows:

$$Hloss = 1 - \frac{1}{cq}\sum_{i=1}^{q}\sum_{j=1}^{c} 1_{y'^{j}_{i}=y^{j}_{i}}$$

where c is the number of labels, q is the number of images that are labeled in the testing dataset, and y'^{j}_{i} and y^{j}_{i} denote the predicted label set and ground-truth label set for the i^{th} image, respectively.

The mean average precision (mAP) was also an important metric in measuring the whole quality, by computing the average precisions for each label [2].

For fair comparison, we selected the top 5 and top 10 relevant labels for annotation. It is obvious that these metrics evaluate the performance of multi-label remote sensing image annotation from different perspectives. It is difficult for one method to perform well across all of the metrics.

5.3. Comparison with Other Approaches

To evaluate the performance of our method, we compared it with several multi-label annotation approaches. We summarize these methods as follows:

- Multi-label least squares (MLLS) [59] takes advantage of the annotation information by extracting the common subspace shared by the annotations. It uses the graphic information while exploring the linear annotation information.
- Bi-Relation graph (BG) [60] constructs a data graph and a label graph to solve the image annotation problem.
- Multi-label ReliefF (MRF) and multi-label F-statistic (MF) [61] extend the common ReliefF and F-statistic tackling feature selection problem and use the 1-NN method to evaluate the multi-label classification problem for the selected features and reports the best result. We only use MRF to compare with our method here.
- Multiview-based multi-label propagation (MMP) [62] explores the consistencies among different views by requiring them to generate the same annotation result, and imposes the similarity constraints to capture the correlations among different labels.
- Ensemble classifier chain (ECC) [63] solves the incomplete label assignment problem by learning a model from the training image examples.
- In random k-label sets (RakEL) [64] for multi-label classification, k is a parameter that specifies the size of the subset. We use it here for the multi-label annotation problem.
- Multi-label classification framework based on the neighborhood rough sets (MLNRS) [30] uses neighborhood rough sets for automatic image annotation to consider the uncertainty of the mapping from the visual feature space to the semantic concept space.
- Multi-label classification based on low rank representation (MLC-LRR) [34] first computes the low-rank constrained coefficient matrix by utilizing low-rank representation of images, then defines a feature-based graph and captures the global relationship between images.

For fairness, we performed parameter tuning in advance and used the best setting to compare with other methods. Table 2 lists the parameters used in these experiments.

To evaluate the effectiveness of the proposed method, we conducted the experiments with 5 and 10 annotations, and reported the precision, recall, and F1 for 5 and 10 labels; we also calculated the Hamming loss and mean average precision measures as an overall evaluation. To quantify the robustness of the method, we also implemented the experiments on different ratios of the training dataset.

Table 2. Parameters required in the method and their settings.

Parameter Notation	Description	Parameter Setting
α	Weight of image graph regularization term	100
β	Weight of label graph regularization term	50
σ	Weight of image similarity between low-level and high-level features	0.5
λ	Tikhonov regularization parameter	10
k	Dimension of latent image features	50
n	Dimension of latent label features	30

In the experiments, we randomly selected 20%, 50%, and 80% of entries in matrix G as the training set, and the rest as the test set. Additionally, we set the related parameters of the approaches being compared according to their original suggested parameters or codes and evaluated the annotation performance with the same measures on both datasets. We repeated 10 independent runs of each experiment, then got an average value for each metric. All experiments were carried out on a workstation with an Intel Core i7-7600 3.4 GHz CPU and 32 GB memory. The experimental results are shown in the following tables: Tables 3–5 show the results on UCMerced, and Tables 6–8 show the results on Corel5k. We analyzed the performance of the results on the different datasets.

5.3.1. Performance on UCMerced Dataset and Analysis

The following tables show the experimental results under the three metrics of the top 5 and top 10 annotations on different training data sizes. The best result for each metric is shown in bold font.

Tables 3–5 exhibit the precision (P), recall (R), and F1 score of the top 5(@5), and top 10 (@10) annotations, respectively. We also report the Hamming loss (Hloss) and mean average precision (mAP) metrics results for the proposed framework and the compared approaches. It is obvious that the proposed method has better performance in most cases. More detailed analyses are described as follows.

Table 3. Experimental results on UCMerced with 20% training dataset.

Method	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
MLLS	0.3313	0.2422	0.2798	0.3229	0.2515	0.2827	0.0402	0.3156
BG	0.3071	0.2687	0.2866	0.2916	0.2809	0.2861	0.0369	0.3338
MRF	0.3282	0.2809	0.3027	0.3114	0.3032	0.3072	0.0267	0.3024
MMP	0.3296	0.2863	0.3035	0.3019	0.3044	0.3031	0.0264	0.3328
ECC	0.3339	0.2935	0.3123	0.3012	0.3313	0.3155	0.0323	0.3687
RAkEL	0.3303	0.2824	0.3044	0.3083	0.3076	0.2979	0.0295	0.3527
MLNRS	0.3222	0.3071	0.3145	0.3156	0.3372	0.3348	0.0262	0.3693
MLC-LRR	0.3503	0.3049	0.3260	0.3421	0.3129	0.3268	0.0256	0.3561
WDG-NMTF	**0.3878**	**0.3265**	**0.3545**	**0.3731**	**0.3523**	**0.3629**	**0.0227**	**0.3869**

Table 4. Experimental results on UCMerced with 50% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
MLLS	0.3612	0.2879	0.3204	0.3421	0.3043	0.3221	0.0348	0.3562
BG	0.3771	0.2886	0.3270	0.3628	0.3121	0.3355	0.0303	0.3781
MRF	0.3626	0.3211	0.3405	0.3517	0.3323	0.3417	0.0253	0.3395
MMP	0.3753	0.3252	0.3484	0.3553	0.3424	0.3487	0.0261	0.3527
ECC	0.3978	0.3521	0.3735	0.3738	0.3812	0.3774	0.0265	0.3789
RAkEL	0.4012	0.3438	0.3702	0.3819	0.3623	0.3718	0.0258	0.3841
MLNRS	0.3923	0.3452	0.3672	0.3901	0.3646	0.3769	0.0232	0.3992
MLC-LRR	0.3941	**0.3845**	0.3892	0.3829	**0.4088**	0.3954	0.0237	0.4081
WDG-NMTF	**0.4251**	0.3808	**0.4017**	**0.4129**	0.4012	**0.4069**	**0.0213**	**0.4296**

Table 5. Experimental results on UCMerced with 80% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
MLLS	0.3982	0.3243	0.3574	0.3823	0.3435	0.3618	0.0315	0.3785
BG	0.4234	0.3216	0.3655	0.4077	0.3452	0.3738	0.0272	0.3954
MRF	0.4027	0.3532	0.3763	0.3879	0.3762	0.3819	0.0231	0.3679
MMP	0.4166	0.3675	0.3905	0.3987	0.3895	0.3940	0.0219	0.3895
ECC	0.4249	0.3875	0.4053	0.4123	0.3932	0.4025	0.0246	0.3923
RAkEL	0.4354	0.3889	0.4108	0.4201	0.3941	0.4050	0.0223	0.4065
MLNRS	0.4238	0.3790	0.4001	0.4021	0.3856	0.3936	0.0221	0.4177
MLC-LRR	0.4358	**0.4187**	0.4270	0.4213	**0.4296**	0.4254	0.0209	0.4268
WDG-NMTF	**0.4516**	0.4108	**0.4303**	**0.4329**	0.4268	**0.4298**	**0.0201**	**0.4453**

First, we performed a comparison among MLLS, BG, MRF, MMP, and the proposed WDG-NMTF method when the training dataset size was 20%. WDG-NMTF completely and significantly outperformed them in all metrics, i.e., an average of 7.13% and 8.45% for MLLS and MRF in mAP, and about 7.47% and 6.79% for F1@5 compared to MLLS and BG on this dataset. This is better than the four methods under the Hamming loss metric, which quantifies the whole performance of the method. Due to the limitations in dealing with a sparse training dataset, we found that MLLS is obviously worse than the other methods in terms of metrics. BG works better than MMLS under all metrics, which indicates that the bi-relational graph of feature and label is more important than only exploring a single graph of subspace. MRF performs better than BG because of the fine-grained image features extracted by the method. MMP displays a clear performance gain over MRF, due to the construction of multi-view image-feature graph, and inter-graph, simultaneously. However, it fully ignores the inter-feature relationships, which have important visual content information. Therefore, it biases the multi-view features, not the feature inter-relationships, which demonstrates the intrinsic relationships of images. For the top 10 annotations, we find from these results that all methods have a slight performance degradation in precision, but a slight improvement in recall. Due to the harmonics of recall and precision, some methods have an improvement in F1 score, i.e., MLLS, MLNRS, MLC-LRR, and WDG-NMTF. This indicates that these methods can get more accurate labels with 10 annotations. Moreover, some methods have a slight degradation, i.e., BG, MMP, and RAkEL, which means that these methods can provide stable results for the top five annotations. It is hard to improve when increasing the number of annotated labels.

Second, we compared the performance among ECC, RAkEL, MLNRS, MLC-LRR, and WDG-NMTF. It is obvious that MLNRS achieves the worst values in precision@5, recall@5, and F1@5, since it relied heavily on the similarity among neighboring images, but it had a relatively better performance in Hamming loss and mAP, due to the uncertainty of mapping from the visual feature space to semantic concept space. ECC and RAkEL achieved competitive results in all metrics, and they outperformed MLNRS in precision, recall, and F1 score. ECC used a complexity classifier chains to get better performance without considering the semantic gap between visual content and semantic concepts. This method had the most computation cost among these methods. RAkEL randomly selected a number of small subsets to train a corresponding classifier and predict related labels. This method also suffers computational efficiency and predictive performance problems. MLC-LRR achieved the second best results next to ours, because it utilized the feature graph and label graph of the images. However, when the training dataset was sparse, the feature and semantic information, extracted from the training dataset, was limited, which affects the accuracy of the method. Therefore, there is an assumption that if the ratio of the labeled images increases, the performance of this method will improve.

Third, to evaluate the robustness of the proposed method more thoroughly, we further compared the performance of all methods with 50% and 80% of the training data. Tables 4 and 5 show the related results under all metrics for these approaches. We can see that when the training data size increased to 50%, the results of all methods improved in all metrics. For example, MLLS has improvements of 2.99% in precision, and 4.57% in recall, compared to 20% in the labeled images. It also has 4.06% and 3.94%

improvements in F1@5, and F1@10, respectively. The mAP and Hamming loss measures also improved. MLC-LRR improved most in recall with 5 and 10 annotations, performing better than our method. MLNRS performed best in Hamming loss among the compared methods. This is because enough neighborhood information can boost the performance. Our method performs best in precision and F1 with the top 5 and top 10 annotations, and also has the best mean average precision and Hamming loss values. When the training data size increase to 80%, the performance of all methods is almost stable. The proposed method achieved a higher mean average precision and lower Hamming loss values than the compared methods. In a word, it yields superior annotation performance compared with the other methods and performs best on most evaluation metrics. The success of our method is ascribed to exploring more useful information from images and labels.

5.3.2. Performance on Corel5k Dataset and Analysis

The Corel5k dataset is larger than the UCMerced database. Tables 6–8 show the precision, recall, Hamming loss, F1 score, and mean average precision measures for the Corel5k dataset, and the best result for each metric is shown in bold font.

Table 6. Experimental results on Corel5k with 20% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
MLLS	0.2632	0.3081	0.2839	0.2519	0.3102	0.2729	0.0328	0.3124
BG	0.2819	0.3212	0.3003	0.2765	0.3308	0.3012	0.0257	0.3325
MRF	0.2632	0.3527	0.3014	0.2517	0.3627	0.2971	0.0268	0.3329
MMP	0.3013	0.3986	0.3431	0.2944	0.4003	0.3392	0.0271	0.3568
ECC	0.2981	0.3602	0.3262	0.2911	0.3687	0.3253	0.0227	0.3428
RAkEL	0.2943	0.3529	0.3209	0.2894	0.3576	0.3199	0.0232	0.3387
MLNRS	0.3258	0.3783	0.3502	0.3167	0.3812	0.3459	0.0225	0.3652
MLC-LRR	0.3379	0.3983	0.3656	0.3218	0.4043	0.3583	0.0223	0.3791
WDG-NMTF	**0.3627**	**0.4211**	**0.3897**	**0.3566**	**0.4253**	**0.3879**	**0.0217**	**0.3942**

Table 7. Experimental results on Corel5k with 50% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
MLLS	0.3044	0.3563	0.3283	0.2912	0.3629	0.3231	0.0288	0.3454
BG	0.3319	0.3678	0.3489	0.3211	0.3872	0.3510	0.0246	0.3588
MRF	0.3046	0.4123	0.3503	0.2987	0.4172	0.3481	0.0252	0.3673
MMP	0.3435	0.4426	0.3863	03357	0.4451	0.3827	0.0243	0.3865
ECC	0.3467	0.4328	0.3849	0.3342	0.4416	0.3804	0.0236	0.3728
RAkEL	0.3248	0.4234	0.3676	0.3068	0.4301	0.3581	0.0212	0.3679
MLNRS	0.3712	0.4423	0.4036	0.3589	0.4576	0.4022	0.0203	0.3927
MLC-LRR	0.3928	0.4763	0.4305	0.3877	0.4834	0.4302	0.0211	0.4309
WDG-NMTF	**0.4075**	**0.4861**	**0.4433**	**0.3942**	**0.4983**	**0.4401**	**0.0201**	**0.4316**

Table 8. Experimental results on Corel5k with 80% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
MLLS	0.3632	0.3877	0.3750	0.3521	0.3986	0.3739	0.0243	0.3688
BG	0.3725	0.4076	0.3892	0.3634	0.4132	0.3872	0.0221	0.3731
MRF	0.3472	0.4478	0.3911	0.3324	0.4512	0.3827	0.0228	0.3831
MMP	0.3769	0.4825	0.4232	0.3631	0.4986	0.4201	0.0235	0.4026
ECC	0.3953	0.4667	0.4280	0.3931	0.4783	0.4315	0.0213	0.4025
RAkEL	0.3658	0.4561	0.4059	0.3612	0.4642	0.4062	0.0198	0.4164
MLNRS	0.4216	0.4672	0.4432	0.4145	0.4769	0.4435	0.0197	0.4329
MLC-LRR	**0.4412**	0.5013	0.4693	**0.4305**	0.5094	0.4666	**0.0191**	0.4562
WDG-NMTF	0.4321	**0.5217**	**0.4726**	0.4201	**0.5293**	**0.4684**	0.0192	**0.4579**

As we can see from Tables 6–8, the values of all approaches increased monotonically with increased training data size in this dataset. Although there are fewer labels for each image, the results under these metrics are similar to those on the UCMerced dataset, compared with other methods. However, it achieved better improvement on Corel5k. We believe this is due to the different cardinality of the datasets: UCMerced has an average of 4.3 labels per image, while Corel5k has 3.5. When we annotate 5 or 10 labels per image, the results can easily cover all the right labels for each test image on Corel5k. This means that with more labels assigned, recall for each label will increase while precision decreases.

The other compared approaches also achieved significant improvement on this dataset; in particular, MMP and MLC-LRR improved more than the other methods. The multi-view features of images used by MMP can help achieve greater performance when there are relatively fewer labels per image. Our method performed slightly worse in the precision and Hamming loss measures than MLC-LRR when the training data size was set at 80%. However, it achieved the best recall, F1 score, and mAP than the other methods. According to our analysis, the reason is that MLC-LRR depends heavily on the number of training samples. When the data size increases, the performance of MLC-LRR can be improved as well. However, this raises the complexity of computation and undermines scalability.

Our method also yielded the best mean average precision value on this dataset. We believe it is due to the use of the image and label graphs, as well as the high-level and low-level features. All the meaningful information makes the proposed method outperform the other compared methods.

In addition, we find that the F1 score results for the top 5 annotations are almost the same as the results for the top 10 on this dataset. We think the possible reason is the smaller cardinality of this dataset. The top 5 annotations are likely to cover most of the true labels. It hardly improves the recall performance for the top 10 annotations. Therefore, annotation performance is stable with more than five tagged labels.

5.3.3. Comparison of Running Time

In this subsection, we evaluate the running time of the proposed method with other methods described above. We report the time consumption of all methods annotating the top 5 labels on the Core5K dataset. The results are shown in Table 9.

Table 9. Running time on Corel5k.

Method	Training Time (s)	Testing Time (s)	Total Time (s)	Average per Image (s)
MLLS	138.20	42.26	180.46	0.3616
BG	-	165.93	165.93	0.3325
MRF	142.95	23.60	166.55	0.3337
MMP	-	245.18	245.18	0.4913
ECC	528.70	38.17	566.87	1.1360
RAkEL	216.26	31.13	247.39	0.4957
MLNRS	208.39	61.22	269.61	0.5403
MLC-LRR	-	203.58	203.58	0.4079
WDG-NMTF	-	212.08	212.08	0.4250

It can be observed that the main time consumption consists of two parts, training time and testing time. Training time shows the total time to train the 4500 images in the dataset, while testing time shows the total time to label all 499 test images. From Table 9, we can see that the graph-based methods, BG, MMP, MLC_LRR, and WDG-NMTF, do not need the training step. Among these methods, BG costs the least time, due to the use of a page rank algorithm and the unified framework, which replaces the two-graph model for data and labels in the dataset. ECC spends the most total time. This is because ECC needs more classifiers to train. MLNRS, a KNN-based method, spends more time than the others, except ECC. This is due to the vast search range, while finding the k most similar neighbors. Other relatively simple methods, MLLS and MRF, cost less time than MLNRS. However, the graph-based methods, MMP, MLC-LRR, and WDG-NMTF, cost relatively more than the

simple methods. For example, by utilizing the low-rank method with two graphs, MLC-LRR costs the least. WDG-NMTF spends more than MLC-LRR since it reconstructs the image–label matrix, which MLC-LRR does not need. Because extracting multi-view features needs more time, MMP spends more than WDG-NMTF. Although WDG-NMTF does not cost the least amount of time, it achieves better results in the annotation task.

5.3.4. Visible Results and Analysis

In this paper, we focused on the remote sensing image multi-label annotation issue. To demonstrate the effectiveness of the proposed method, we showed the visible results on the UCMerced remote sensing image dataset. In the experiments, we compared WDG-NMTF with eight approaches and displayed the annotating results on three images with the top five annotations. The results are shown in Figure 4.

			
Ground truth labels	buildings, cars, pavement, sand, tanks, water, Storage-tanks	buildings, cars, court, grass, pavement, trees, Tennis-court,	bare-soil, buildings, cars, trees, Medium-residential
MLLS	buildings, court, mobile-home, tanks, trees	buildings, pavement, sand, trees, water	buildings, pavement, sand, tanks, trees,
BG	buildings, field, tanks, trees,Storage-tanks	buildings, field, tanks, trees, Tennis-court	buildings, court, sand, tanks, Medium-residential,
MRF	bare-soil, buildings, cars, tanks, trees	buildings, cars, pavement, water, Medium-residential	bare-soil, buildings, sand, water, Medium-residential,
MMP	buildings, cars, court, trees, Storage-tanks	buildings, pavement, field, tanks water	buildings, bare-soil, cars, mobile-home, trees
ECC	buildings, pavement, tanks, bare-soil, Storage-tanks	buildings, cars, trees, water, Tennis-court	buildings, cars, court, sand, trees
RAkEL	bare-soil, buildings, pavement, building, trees	buildings, pavement, tanks, trees, Tennis-court,	buildings, sand, tanks, trees, Medium-residential,
MLNRS	cars, building, pavement, trees, Storage-tanks	buildings, grass, pavement, sand, Tennis-court	buildings, bare-soil, cars, trees, water
MLC-LRR	cars, pavement, tanks, water, Storage-tanks	buildings, cars, pavement, water, Tennis-court	bare-soil, buildings, sand, trees, Medium-residential
WDG-NMTF	buildings, cars, sand, water, Storage-tanks	buildings, court, grass, field, Tennis-court	bare-soil, buildings, cars, trees, Medium-residential

Figure 4. Predicted labels for example images on UCMerced dataset. Each image can be labeled by several semantic labels. Words in green are the right labels for those images. Black means the word misses the ground truth.

From Figure 4, we can observe that WDG-NMTF has the best annotation results among these methods. Because we only predicted the top 5 labels for each image, we just calculated the total number of correct labels for each image with different methods. Among these methods, WDG-NMTF, MLC-LRR, MLNRS, RAkEL, ECC, MMP, MRF, BG, and MLLS correctly predicted 14, 13, 12, 10, 12, 10, 9, 9, and 8 labels, respectively, out of 19 ground-truth labels in these three images. For each image, each method gives the top five labels according to their characteristics and the results agree with the results in Section 5.3.1. Additionally, MLLS works worst and our method performs best among all the methods.

For each image, from left to right, the total number of incorrect labels is 12, 12, and 14 among all methods. Obviously, the right image has the most wrong labels. We think the reason is the low resolution of this image. Another interesting observation is that some words were frequently predicted by most of the methods: e.g., "building" was predicted by almost every method. This may be because many images have buildings not only in the visual content but also in the semantic meaning. It is helpful for the annotation performance. Especially for our method, we can construct a label graph by calculating the co-occurrence relationships of labels. This also can reduce the semantic gap between low-level features and high-level semantic features.

Additionally, Figure 4 also shows that the proposed method can provide more correct labels in image annotating tasks than the other approaches. These results visibly indicate the promising performance of the proposed method.

5.4. Benifits of Image Graph and Label Graph

In this subsection, we conduct experiments to compare the effectiveness of the image graph and label graph. We use WDG-NMTF-A to denote only using image graph information, and WDG-NMTF-B to denote only using label graph information. WDG-NMTF uses both image graph and label graph information. Similar to previous experiments, we randomly used 20%, 50%, and 80% of the dataset as the training data size. The other parameters were set as $k = 50$, $n = 30$, $\sigma = 0.5$, $\lambda = 10$.

Tables 10–12 show the experimental results on UCMerced and Tables 13–15 show the results on Corel5k.

The following tables show the results on the Corel5k dataset.

Table 10. Experimental results on UCMerced with 20% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
WDG-NMTF-A	0.3235	0.2534	0.2903	0.3142	0.2713	0.2911	0.0254	0.3393
WDG-NMTF-B	0.3088	0.2486	0.2754	0.3063	0.2537	0.2775	0.0262	0.3156
WDG-NMTF	**0.3878**	**0.3265**	**0.3545**	**0.3731**	**0.3523**	**0.3629**	**0.0227**	**0.3869**

Table 11. Experimental results on UCMerced with 50% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
WDG-NMTF-A	0.4173	0.3523	0.3820	0.4013	0.3662	0.3829	0.0236	0.3946
WDG-NMTF-B	0.4056	0.3458	0.3733	0.3902	0.3623	0.3757	0.0243	0.3823
WDG-NMTF	**0.4251**	**0.3808**	**0.4017**	**0.4129**	**0.4012**	**0.4069**	**0.0213**	**0.4296**

Table 12. Experimental results on UCMerced with 80% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
WDG-NMTF-A	0.4483	0.3937	0.4192	0.4362	0.4154	0.4255	0.0221	0.4362
WDG-NMTF-B	0.4385	0.3916	0.4137	0.4251	0.4117	0.4182	0.0224	0.4257
WDG-NMTF	**0.4516**	**0.4108**	**0.4303**	**0.4329**	**0.4268**	**0.4298**	**0.0201**	**0.4453**

Table 13. Experimental results on Corel5k with 20% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
WDG-NMTF-A	0.2951	0.3426	0.3170	0.2887	0.3623	0.3213	0.0223	0.3487
WDG-NMTF-B	0.2854	0.3373	0.3091	0.2776	0.3442	0.3073	0.0242	0.3299
WDG-NMTF	**0.3627**	**0.4211**	**0.3897**	**0.3566**	**0.4253**	**0.3879**	**0.0217**	**0.3942**

Table 14. Experimental results on Corel5k with 50% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
WDG-NMTF-A	0.3732	0.4478	0.4078	0.3602	0.4533	0.4014	0.0215	0.3981
WDG-NMTF-B	0.3659	0.4371	0.3941	0.3536	0.4432	0.3933	0.0226	0.3967
WDG-NMTF	**0.4075**	**0.4861**	**0.4433**	**0.3942**	**0.4983**	**0.4401**	**0.0201**	**0.4316**

Table 15. Experimental results on Corel5k with 80% training dataset.

Methods	P@5	R@5	F1@5	P@10	R@10	F1@10	Hloss	mAP
WDG-NMTF-A	0.4154	0.4976	0.4527	0.4127	0.5003	0.4522	0.0203	0.4381
WDG-NMTF-B	0.3982	0.4837	0.4368	0.3812	0.4896	0.4286	0.0212	0.4227
WDG-NMTF	**0.4321**	**0.5217**	**0.4726**	**0.4201**	**0.5293**	**0.4684**	**0.0192**	**0.4579**

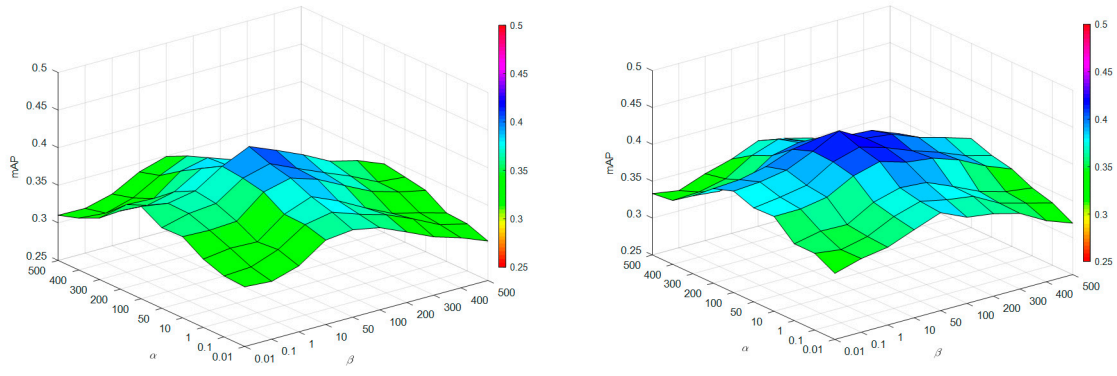
We can find from these tables that when the image–label matrix is sparse, only the using image graph performs better than only using label graph. This is the case for both datasets. According to our analysis, the reason is that when the image–label matrix is sparse, it provides little useful information. However, the visual content of images is stable, which can alleviate the sparsity. With the increased training dataset size, the image similarity from semantic information is enhanced. This is helpful to the performance of the overall algorithm.

Meanwhile, we also show that the proposed method can effectively annotate more labels for images by using both image and label graphs. This is a result of fully exploiting the relationships between low-level features and high-level semantics.

5.5. Sensitivity of Parameters and Analyses

5.5.1. Impact of Regularization Parameters α and β

To find the best combination of α and β, we implemented a set of experiments on the two datasets, and show the results of mean average precision, Hamming loss, and F1 score in Figure 5. In these experiments, we set the parameters as $\sigma = 0.5$, $\lambda = 10$, and dimensionality parameters as $k = 50, n = 30$. Meanwhile, we calculated the metrics with the top five annotations.



(**a**) (**b**)

Figure 5. *Cont.*

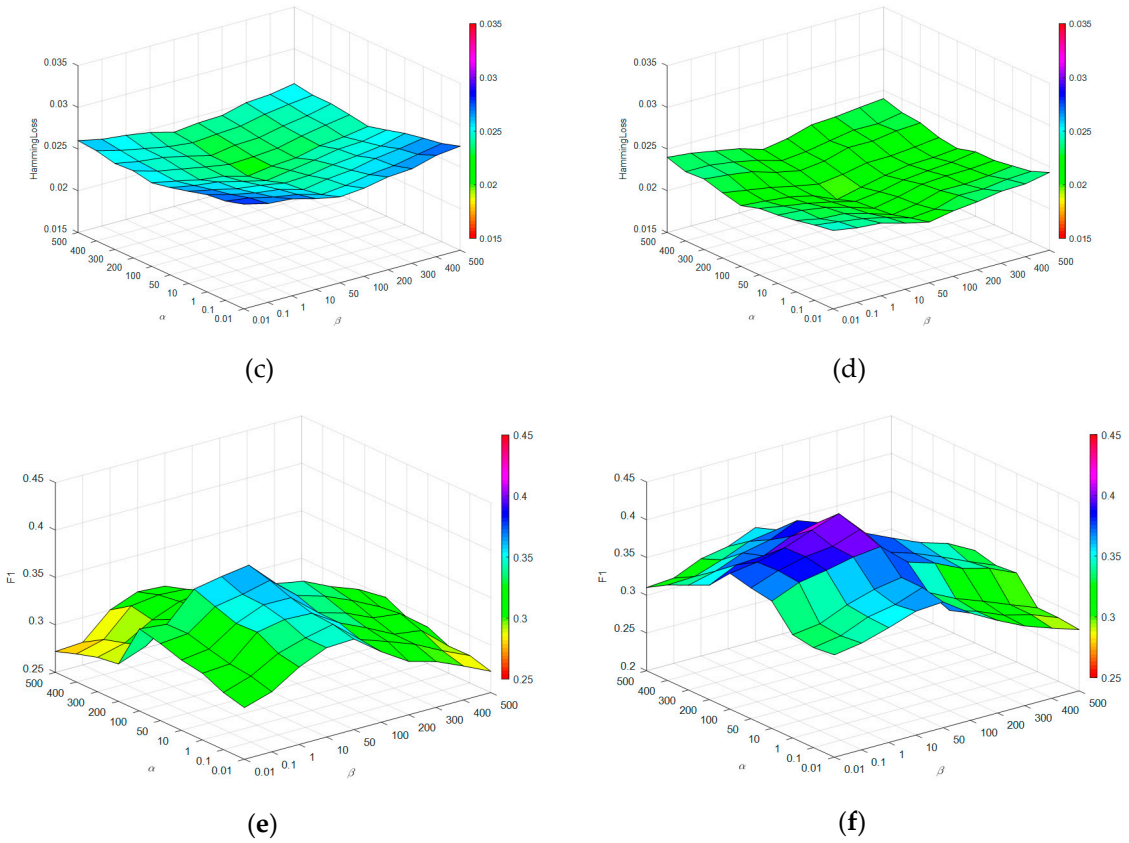


Figure 5. Impact of regularization parameters α and β: (**a**) mean average precision on UCMerced dataset; (**b**) mean average precision on Corel5k dataset; (**c**) Hamming loss on UCMerced dataset; (**d**) Hamming loss on Corel5k dataset; (**e**) F1 score on UCMerced dataset; (**f**) F1 score on Corel5k dataset.

From these figures, we can observe that when α and β are both too small, the performances are not satisfactory under the three metrics. With increasing α and β, the performance improves on both datasets. This proves the importance of the image and label graph. Moreover, optimal results can be achieved when $\alpha = 100$ and $\beta = 50$ on the two datasets, although their cardinalities are different. These results demonstrate that the image graph is more important than the label graph in our method, which agrees with Section 5.4.

5.5.2. Impact of Image Similarity Parameter σ

To evaluate the impact of the two image similarities with different weights, we fixed the parameters as α= 100, β= 50, $\lambda = 10$, and set the feature dimensions as $k = 50$ and $n = 30$. In the implementing procedure, we changed the parameter $\sigma \in [0, 1]$ under different training dataset sizes. Then, we obtained the F1 score results with the top five annotations on the two datasets, which are shown in Figure 6a,b.

From Figure 6a,b, we can observe that when the training data size is 20% on both datasets, more weight ($\sigma < 0.5$) on image visual similarity can improve the performance significantly, but when it increases to 80%, $\sigma > 0.5$ achieves better values than $\sigma < 0.5$. According to our analysis, the reason is that, as the training data increases, the semantic information can provide enough useful meaning for the method, which can improve the performance of the algorithm. However, when the image-label matrix is sparse, less information can be obtained from the semantic similarity. It is worth noting that visual similarity can compensate for useful information of this limitation. Another interesting phenomenon is that our method can attain better performance with $\sigma = 0.5$ on the two datasets, when the ratio of training data increases to 50%. Therefore, $\sigma = 0.5$ is a compromise for the two datasets, and can provide a promising result as well.

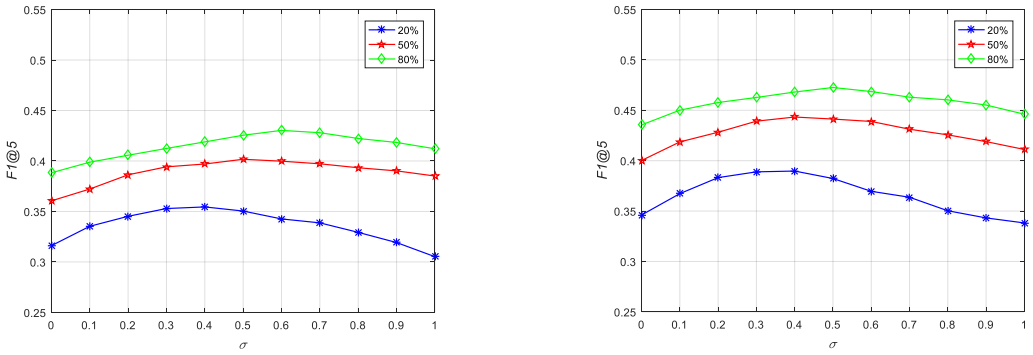


Figure 6. Impact of σ: (**a**) F1 score with different weights of σ on UCMerced; (**b**) F1 score with different weights of σ on Corel5k.

5.5.3. Impact of Number of Latent Features

There are two other important parameters in our method: The number of latent image features k and the number of latent label features n. Usually, they are different with good adaptive ability for moderate representations. To study how k and n affect the performance of our method simultaneously, we conducted a set of experiments on 20% training datasets. k varies from 5 to 60 and n varies from 5 to 35 on UCMerced, while on Corel5k, the range is from 5 to 60. We fixed the other parameters as $\alpha = 100, \beta = 50, \lambda = 10, \sigma = 0.5$ and results are shown in Figure 7a,b.

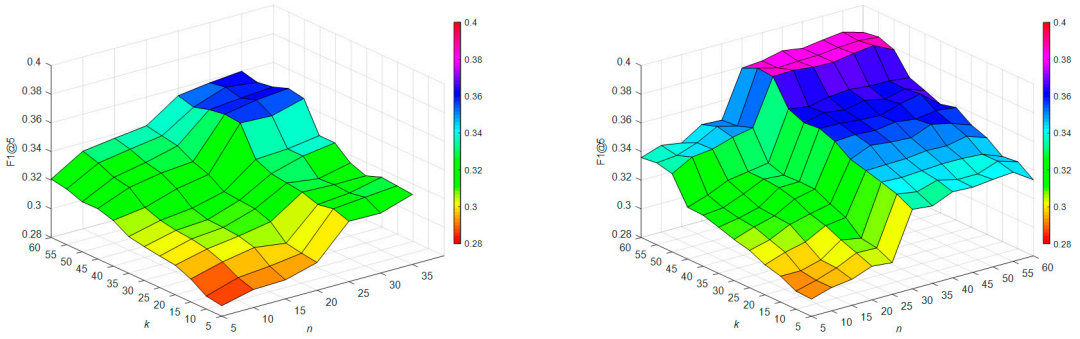


Figure 7. Impact of the number of feature dimensions: (**a**) F1@5 with different image feature dimensions k and label feature dimensions n on UCMerced; (**b**) F1@5 with different image feature dimensions k and label feature dimensions n on Corel5k.

It can be seen from Figure 7a,b that the values of F1 score change along with the change of feature dimensionalities. In a certain range, the higher the dimensionality of latent features, the better performance can be achieved. We believe the reason for this is that with more latent features, more information can be represented by the low-rank matrices. However, when k and n increase to a certain range, the performance is relatively stable. This is because the existing latent features can represent the useful information well. The empirical results show that the ranges are different on the two datasets. On UCMerced, the range is $k > 40, n > 25$, while on Corel5k it is $k > 50, n > 30$. There is a fact that large latent features will significantly increase the computation cost. For the overall consideration, we chose relatively smaller values for k and n to maintain acceptable performance. Thus, we take $k = 50, n = 30$ to get a trade-off for both datasets.

5.5.4. Impact of Regularization Parameter λ

To evaluate the contribution of λ, we built a set of experiments on both datasets under the F1 score measure, when the training dataset size is 20%, and set the other parameters as $\alpha = 100$, $\beta = 50, k = 50, n = 30, \sigma = 0.5$. The results are displayed in Figure 8a,b. From these figures, we can see that when the value of λ is 1 or larger, the F1 score is better. This means larger λ can achieve better F1 scores. Additionally, when $10 < \lambda < 50$, the proposed method can get the best performance on both datasets. However, too-large λ can result in much computation cost. Considering this, we take $\lambda = 10$ in our experiments as a trade-off.

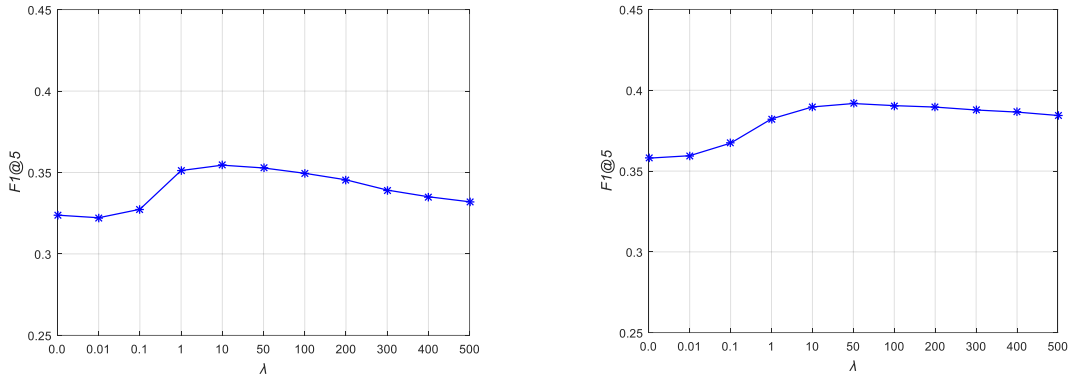


Figure 8. Impact of parameter λ: (**a**) F1 score with different λ on UCMerced; (**b**) F1 score with different λ on Corel5k.

6. Discussion

To assess the efficiency of the proposed method, several qualitative comparison experiments were done on the UCMerced and Corel5k datasets. We utilized five metrics to evaluate the performance of all approaches. Meanwhile, to evaluate the robustness of the proposed method, we deployed these experiments on training datasets of different sizes. The results and analyses are described in Section 5.3.1, Section 5.3.2, and Section 5.3.3.

It can be seen from these results that our method outperforms the baseline methods under most metrics on both datasets, especially when the dataset is sparse. Precision is boosted significantly on the UCMerced dataset, which confirms the advantage of using the image graph and label graph simultaneously. Moreover, recall is improved by a large margin on the Corel5K dataset. This is because this dataset has fewer labels per image and more accurate labels can be covered by the proposed method. Furthermore, recall and precision affect the value of F1 score simultaneously, which is more sensitive for performance. Nevertheless, our method still achieves the best F1 score on both datasets, even more on the different sizes of training data. According to our analysis, when the ratio of training data increases, more information can be obtained from the dataset; e.g., more label information can be obtained from UCMerced dataset. Also, we can obtain more image information from both datasets, therefore F1 improves. Hamming loss and mean average precision metrics quantify the overall performance of the methods. Our method achieves better performance under these two metrics on both datasets, which proves the effectiveness of the proposed method.

In the compared methods, MLC-LRR achieves relatively better performance on both datasets; specifically, when the ratio of labeled images on UCMerced increases to 50% and 80%, it achieves better values in recall than our method. Also, it attains better precision and Hamming loss values on Corel5k than our method when the ratio of labeled images increases to 80%. This proves that there is an advantage to using the feature graph and semantic graph simultaneously. However, the limitations of this method are that it depends heavily on the ratio of labeled images and does not consider the semantic gap between low-level features and high-level features. On the other hand, this method

has a rather more complicated process than our method. Our method overcomes these limitations by introducing both the visual content and semantic similarities to reduce the semantic gap and have a simpler framework. Among these methods, MLLS performs worst on the two datasets due to the limitation of only using label correlation in image annotation. By using a bi-relational semantic graph for images and labels, BG performs better, ignoring the visual content of the images. MRF and MMP achieve rather good results on both datasets by making full use of the image features. MMP achieves many improvements on Corel5k because of the multi-view features of images. Extracting more features can improve the overall performance of the method. However, more features can result in more dimensionality problems. It increases the computation cost and storage burden. ECC uses more classifiers to improve the performance, which increases the computation cost. Computation efficiency is also a challenging issue for the RAkEL approach, although it achieves rather good performance on both datasets. MLNRS gets a better result than the others except MLC-LRR, which provides the advantage of considering the semantic neighborhood rough sets, but it does not consider the visual content of images, and the process of finding the nearest neighbors will be too time-consuming.

Although our method achieved the best performance in the experiments, there are several limitations. First, to solve the sparsity problem of the input matrix, we introduced the weighted matrix, which increased the computation cost of this method. The running time proved that it is not a very quick method. Second, to find the best performance for different datasets, we need to tune the parameters very carefully, which is also time-consuming.

Nevertheless, these experimental results indicate that our method outperformed most of the methods under most of the metrics. It has the advantages of simplicity, high efficiency, low storage cost, scalability, and strong robustness. It is suitable for not only remote sensing multi-label annotation tasks but also the natural multi-label image annotation problem.

7. Conclusions and Future Works

In this paper, we proposed the WDG-NMTF, a novel method to solve the remote sensing multi-label annotation problem. This method seamlessly combines matrix factorization and image annotation, utilizing both the high efficiency and scalability of matrix factorization for image annotation. To improve the performance of the method, we employed both image and label graphs to extend the NMTF. Moreover, to reduce the semantic gap between images and labels, we used both low-level and high-level features of images to make full use of the hidden semantics and image content information in datasets. Generally speaking, using this method will make remote sensing image labeling more efficient and less labor-intensive, especially for huge numbers of unlabeled images. Experimental results on benchmark datasets demonstrated that the proposed method performs better than most state-of-the-art multi-label annotation methods.

In the future, we will investigate how to accelerate the method during the updating process. Moreover, we will reduce the computation cost resulting from the introduction of weighted matrix, which is calculated at each step of iteration. It is worth developing a more efficient multi-label annotation method in terms of both accuracy and computation in the future.

Author Contributions: All authors extensively discussed the contents of this paper and contributed to its preparation. J.Z. (Juli Zhang) conceived and designed the experiments, and wrote the paper; J.Z. (Junyi Zhang) implemented the experiments; T.D. analyzed the data and reviewed the paper; Z.H. supervised the study and reviewed the manuscript.

Funding: This research was funded by China Youth Science Foundation, grant number 61702413, and the Technology Innovation Funds for the Ninth Academy of China Aerospace, grant number 2016JY06.

Acknowledgments: The authors thank everyone who contributed to this work. We gratefully thank Xuehan Tang, Zhuping Wang, Lan Liu and Zhong Ma for their sincere support in our work. We also thank all the anonymous reviewers and editors for their helpful comments and suggestions.

Conflicts of Interest: The authors declare that there are no conflict of interest regarding the publication of this paper.

Appendix A

To prove Theorem 1, we need to show that the object function in Equation (4) is nonincreasing under the steps in Equations (7)–(9) and converges to a local minimum, respectively. We follow a scheme similar to that described in [57]. We begin with the definition of auxiliary function.

Definition 1. *$Z(H, H')$ is an auxiliary function for $F(H)$ if the following conditions are satisfied:*

$$Z(H, H') \geq F(H), \text{and} (H, H) = F(H)$$

The auxiliary function is very useful thanks to the following lemma:

Lemma 1 [57]. *If $Z(H, H')$ is the auxiliary function of $F(H)$, then $F(H)$ is nonincreasing under the following update rule:*

$$H^{t+1} = \underset{H}{\text{argmin}} Z(H, H^t) \tag{A1}$$

where H^t is the t^{th} update iteration of H.

Because $F(H^{t+1}) \leq Z(H^{t+1}, H^t) \leq Z(H^t, H^t) = F(H^t)$, $F(H)$ is monotonically decreasing. Therefore, the key is to find a proper auxiliary function for $F(H)$. To construct the auxiliary functions for these objective functions, we employ the following propositions:

Lemma 2 [65]. *For any matrices $D \in \mathbb{R}_+^{m \times r}, E \in \mathbb{R}_+^{m \times r}, E' \in \mathbb{R}_+^{m \times r}$, we have the following inequality:*

$$Tr(D^T E') \geq \sum_{ij} D_{ij} E_{ij} (1 + \log \frac{E'_{ij}}{E_{ij}})$$

Lemma 3 [21]. *For any non-negative matrices $M \in \mathbb{R}_+^{n \times n}, N \in \mathbb{R}_+^{k \times k}, Q \in \mathbb{R}_+^{n \times k}, Q' \in \mathbb{R}_+^{n \times k}$, where* M *and* N *are symmetric matrices, we have the following inequality:*

$$\sum_{ij} \frac{(MQ'N)_{ij} Q_{ij}^2}{Q'_{ij}} \geq Tr(Q^T M Q N)$$

Lemma 4 [65]. *For any symmetric matrix $O \in \mathbb{R}_+^{r \times r}$, and any matrices $W \in R_+^{m \times r}, W' \in R_+^{m \times r}$, we have the following inequality:*

$$\sum_{ij} \frac{(WO)_{ij} W'^2_{ij}}{W_{ij}} \geq Tr(W'^T W' O)$$

Lemma 5 (quadratic lower bound) [65]. *For $Q \in R_+^{m \times r}, W \in R_+^{m \times r}, W' \in R_+^{m \times r}$, we have the following inequality:*

$$Tr(W'^T W' Q) \geq \sum_{ijl} B_{jl} W_{ij} W_{il} (1 + \log \frac{W'_{ij} W'_{il}}{W_{ij} W_{il}})$$

Due to the alternatively updating rules, we prove that each rule leads the objective function to converge to a local minimum. We first proved the object function in Equation (4) with respect to A being non-increasing under the steps in Equation (7) and converging to a local minimum.

First, we write the objective function in Equation (4) with respect to A as

$$\begin{aligned} F(A) &= \tfrac{1}{2}\|Y \odot (G - AVB)\|_F^2 + \tfrac{\alpha}{2} Tr(A^T L_A A) + \tfrac{\lambda}{2}\|A\|_F^2 \\ &= Tr(-Y \odot GB^T V^T A^T + \tfrac{1}{2} Y \odot (AVB) B^T V^T A^T + \tfrac{1}{2} A^T (\alpha L_A^+ + \lambda I) A - \tfrac{\alpha}{2} A^T L_A^- A) \end{aligned}.$$

Second, we prove that the following function is an auxiliary function of $F(A)$:

$$\begin{aligned} Z(A, A') &= -\textstyle\sum_{ij} (Y \odot GB^T V^T)_{ij} A'_{ij} (1 + \log \frac{A_{ij}}{A'_{ij}}) + \frac{1}{2} \sum_{ij} \frac{[Y \odot (A'VB) B^T V^T]_{ij} A_{ij}^2}{A'_{ij}} \\ &\quad + \tfrac{1}{2} \textstyle\sum_{ij} \frac{[A'(\alpha L_A^+ + \lambda I)]_{ij} A_{ij}^2}{A'_{ij}} - \frac{\alpha}{2} \sum_{ijl} (L_A^-)_{jl} A'_{ij} A'_{il} (1 + \log \frac{A_{ij} A_{il}}{A'_{ij} A_{il}}) \end{aligned}$$

Since $Z(A, A') = F(A)$ is obvious when $A' = A$, we only need to show that $Z(A, A') \geq F(A)$. We can find that: (a) the first term in $Z(A, A')$ is always smaller than the first term in $F(A)$, due to Lemma 2; (b) the second term in $Z(A, A')$ is always bigger than the second term in $F(A)$, due to Lemma 3; (c) the third term in $Z(A, A')$ is always bigger than the third term in $F(A)$, due to Lemma 4; and (d) the fourth term in $Z(A, A')$ is always smaller than the fourth term in $F(A)$, due to Lemma 5. By summing over all the bounds, we get $Z(A, A') \geq F(A)$. Thus, the conditions of Definition 1 are satisfied. $Z(A, A')$ is an auxiliary function of $F(A)$.

Third, we prove $Z(A, A')$ is a convex function in A and it has a local minimum.

To prove it is convex, we calculate the first derivative of $Z(A, A')$

$$\frac{\partial Z(A, A')}{\partial A_{ij}} = -(Y \odot GB^T V^T)_{ij} \frac{A'_{ij}}{A_{ij}} + \frac{[Y \odot (A'VB) B^T V^T]_{ij} A_{ij}}{A'_{ij}} + \frac{[A'(\alpha L_A^+ + \lambda I)]_{ij} A_{ij}}{A'_{ij}} - \alpha \frac{(A' L_A^-)_{ij} A'_{ij}}{A_{ij}}$$

Then we have the Hessian matrix as

$$\frac{\partial Z^2(A, A')}{\partial A_{ij} \partial A_{kl}} = \sigma_{ik} \sigma_{jl} \left[\begin{array}{c} \frac{(Y \odot GB^T V^T)_{ij} A'_{ij}}{A_{ij}^2} + \frac{[Y \odot (A'VB) B^T V^T]_{ij}}{A'_{ij}} \\ + \frac{[A'(\alpha L_A^+ + \lambda I)]_{ij}}{A'_{ij}} + \frac{\alpha (A' L_A^-)_{ij} A'_{ij}}{A_{ij}^2} \end{array} \right] = \sigma_{ik} \sigma_{jl} \Lambda_{ij}$$

It can be seen that the above matrix is a diagonal matrix with positive diagonal elements, where σ_{ik}, σ_{jl} are the functions that equal 1 when $i = k$ or $j = l$ and 0 otherwise. Moreover, $\Lambda_{ij} = \frac{[Y \odot GB^T V^T + \alpha A' L_A^-]_{ij} A'_{ij}}{A_{ij}^2} + \frac{[Y \odot (A'VB) B^T V^T + A'(\alpha L_A^+ + \lambda I)]_{ij}}{A'_{ij}}$. Therefore, $Z(A, A')$ is a convex function. We can find a local minimum of $\min_A Z(A, A')$ by fixing A' and setting the first derivative of $Z(A, A')$ equal to zero. After solving it for A_{ij}, we get the minimum as

$$A_{ij} = A_{ij}' \sqrt{\frac{[Y \odot GB^T V^T + \alpha L_A^- A]_{ij}}{[Y \odot (AVB) B^T V^T + \alpha L_A^+ A + \lambda A]_{ij}}}$$

According to Lemma 1, $A^{t+1} = A$ and $A' = A^t$, and we recover Equation (7). Under this update rule, $F(A)$ decreases monotonically and converges to a local minimum.

With respect to updating rules in Equations (8) and (9), the proofs are analogous to Equation (7).

In summary, we have proved the convergence of Theorem 1.

References

1. Xu, G.; Shen, W.; Wang, X. Applications of wireless sensor networks in marine environment monitoring: a survey. *Sensors* **2014**, *14*, 16932–16954. [CrossRef]
2. Chaudhuri, B.; Demir, B.; Chaudhuri, S.; Bruzzone, L. Multilabel Remote Sensing Image Retrieval Using a Semisupervised Graph-Theoretic Method. *IEEE Trans. Geosci. Remote Sens.* **2017**, 1114–1158. [CrossRef]

3. Karalas, K.; Tsagkatakis, G.; Zervakis, M.; Tsakalides, P. Land Classification Using Remotely Sensed Data: Going Multilabel. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3548–3563. [CrossRef]
4. Li, D. Content-based remote sensing image retrieval. *Proc. SPIE Int. Soc. Opt. Eng.* **2005**, *6044*, 60440Q. [CrossRef]
5. Demir, B.; Bruzzo, L. Kernel-based hashing for content-based image retrval in large remote sensing data archive. In Proceedings of the 2014 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Quebec City, QC, Canada, 13–18 July 2014; pp. 3542–3545.
6. Demir, B.; Bruzzone, L. A Novel Active Learning Method in Relevance Feedback for Content-Based Remote Sensing Image Retrieval. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2323–2334. [CrossRef]
7. Smeulders, A.W.M.; Worring, M.; Santini, S.; Gupta, A.; Jain, R.C. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1349–1380. [CrossRef]
8. Hao, M.; Zhu, J.; Lyu, R.T.; King, I. Bridging the Semantic Gap Between Image Contents and Tags. *IEEE Trans. Multimed.* **2010**, *12*, 462–473. [CrossRef]
9. Luo, W.; Li, H.; Liu, G. Automatic Annotation of Multispectral Satellite Images Using Author–Topic Model. *IEEE Geosci. Remote Sens. Lett.* **2012**, *9*, 634–638. [CrossRef]
10. Yao, X.; Han, J.; Cheng, G.; Qian, X.; Guo, L. Semantic Annotation of High-Resolution Satellite Images via Weakly Supervised Learning. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3660–3671. [CrossRef]
11. Chen, K.; Jian, P.; Zhou, Z.; Guo, J.; Zhang, D. Semantic Annotation of High-Resolution Remote Sensing Images via Gaussian Process Multi-Instance Multilabel Learning. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1285–1289. [CrossRef]
12. Luo, W.; Li, H.; Liu, G.; Zeng, L. Semantic Annotation of Satellite Images Using Author–Genre–Topic Model. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 1356–1368. [CrossRef]
13. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [CrossRef]
14. Gu, Q.; Zhou, J.; Ding, C.H.Q. Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs. In Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010, Columbus, OH, USA, 29 April–1 May 2010; pp. 199–210.
15. Hernando, A.; Ortega, F. A non-negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowl.-Based Syst.* **2016**, *97*, 188–202. [CrossRef]
16. Li, Y.; Wang, D.; He, H.; Jiao, L.; Xue, Y. Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems. *Neurocomputing* **2017**, *249*, 48–63. [CrossRef]
17. Kalayeh, M.M.; Idrees, H.; Shah, M. NMF-KNN: Image Annotation Using Weighted Multi-view Non-negative Matrix Factorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 184–191.
18. Jia, X.; Sun, F.; Li, H.; Cao, Y.; Zhang, X. Image Multi-Label Annotation Based on Supervised Nonnegative Matrix Factorization with New Matching Measurement. *Neurocomputing* **2017**, *219*, 518–525. [CrossRef]
19. Rad, R.; Jamzad, M. Image Annotation using Multi-view Non-negative Matrix Factorization with Different Number of Basis Vectors. *J. Vis. Commun. Image Represent.* **2017**, *46*, 1–12. [CrossRef]
20. Ge, H.; Yan, Z.; Dou, J.; Wang, Z.; Wang, Z. A Semisupervised Framework for Automatic Image Annotation Based on Graph Embedding and Multiview Nonnegative Matrix Factorization. *Math. Probl. Eng.* **2018**, *2018*, 1–11. [CrossRef]
21. Ding, C.H.Q.; Li, T.; Peng, W.; Park, H. Orthogonal nonnegative matrix t-factorizations for clustering. In Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Ming, Philadelphia, PA, USA, 20–23 August 2006; pp. 126–135.
22. Yoo, J.; Choi, S. Weighted Nonnegative Matrix Co-Tri-Factorization for Collaborative Prediction. In Proceedings of the 1st Asian Conference on Machine Learning: Advances in Machine Learning, Nanjing, China, 2–4 November 2009; pp. 396–411.
23. Chen, G.; Wang, F.; Zhang, C. Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Inf. Process. Manag.* **2009**, *45*, 368–379. [CrossRef]
24. Lienou, M.; Maitre, H.; Datcu, M. Semantic Annotation of Satellite Images Using Latent Dirichlet Allocation. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 28–32. [CrossRef]
25. Zhou, N.; Cheung, W.K.; Qiu, G.; Xue, X. A Hybrid Probabilistic Model for Unified Collaborative and Content-Based Image Tagging. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1281–1294. [CrossRef]

26. Liu, H.; Wu, Z.; Cai, D.; Huang, T.S. Constrained Nonnegative Matrix Factorization for Image Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1299–1311. [CrossRef]
27. Wu, L.; Jin, R.; Jain, A.K. Tag Completion for Image Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 716–727. [CrossRef]
28. Carneiro, G.; Chan, A.B.; Moreno, P.J.; Vasconcelos, N. Supervised Learning of Semantic Classes for Image Annotation and Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 394–410. [CrossRef]
29. Tieu, K.; Viola, P. Boosting image retrieval. In Proceedings of the Computer Vision and Pattern Recognition, Hilton Head, SC, USA, 13–15 June 2000; pp. 228–235.
30. Yu, Y.; Pedrycz, W.; Miao, D. Neighborhood rough sets based multi-label classification for automatic image annotation. *Int. J. Approx. Reason.* **2013**, *54*, 1373–1387. [CrossRef]
31. Fei, W.; Han, Y.; Qi, T.; Zhuang, Y. Multi-label boosting for image annotation by structural grouping sparsity. In Proceedings of the 18th ACM International Conference on Multimedea, Firenze, Italy, 25–29 October 2010; pp. 15–24. [CrossRef]
32. Briggs, F.; Fern, X.Z.; Raich, R.; Lou, Q. Instance Annotation for Multi-Instance Multi-Label Learning. *ACM Trans. Knowl. Discov. Data* **2013**, *7*, 1–30. [CrossRef]
33. Zhang, H.; Berg, A.C.; Maire, M.; Malik, J. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2126–2136. [CrossRef]
34. Tan, Q.; Liu, Y.; Chen, X.; Yu, G. Multi-Label Classification Based on Low Rank Representation for Image Annotation. *Remote Sens.* **2017**, *9*, 109. [CrossRef]
35. Karalas, K.; Tsagkatakis, G. Deep learning for multi-label land cover classification. In Proceedings of the SPIE Remote Sensing XXI, Toulouse, France, 21–24 September 2015; Volume 9643, p. 96430Q.
36. Sharifi, Z.; Rezghi, M.; Nasiri, M. A new algorithm for solving data sparsity problem based-on Non negative matrix factorization in recommender systems. In Proceedings of the 4th International Econference on Computer and Knowledge Engineering, Mashhad, Iran, 29–30 October 2014; pp. 56–61. [CrossRef]
37. Tommi, N.; Jaakkola, S. Maximum-Margin Matrix Factorization. *Adv. Nips* **2005**, *37*, 1329–1336.
38. Talwalkar, A.; Kumar, S.; Mohri, M.; Rowley, H. Large-scale SVD and manifold learning. *J. Mach. Learn. Res.* **2013**, *14*, 3129–3152. [CrossRef]
39. Kim, Y.D.; Choi, S. Weighted nonnegative matrix factorization. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 1541–1544. [CrossRef]
40. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1548–1560. [CrossRef]
41. Li, H.; Zhang, J.; Liu, J. Graph-regularized CF with local coordinate for image representation. *J. Vis. Commun. Image Represent.* **2017**, *49*, 392–400. [CrossRef]
42. Shang, F.; Jiao, L.C.; Wang, F. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognit.* **2012**, *45*, 2237–2250. [CrossRef]
43. Ye, J.; Jin, Z. Dual-graph regularized concept factorization for clustering. *Neurocomputing* **2014**, *138*, 120–130. [CrossRef]
44. Yin, M.; Gao, J.; Lin, Z.; Shi, Q.; Guo, Y. Dual Graph Regularized Latent Low-Rank Representation for Subspace Clustering. *IEEE Trans. Image Process.* **2015**, *24*, 4918–4933. [CrossRef]
45. Li, X.; Cui, G.; Dong, Y. Graph Regularized Non-Negative Low-Rank Matrix Factorization for Image Clustering. *IEEE Trans. Syst. Man Cybern.* **2017**, *47*, 3840–3853. [CrossRef]
46. Hua, W.; Nie, F.; Huang, H.; Makedon, F. Fast Nonnegative Matrix Tri-Factorization for Large-Scale Data Co-Clustering. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Spain, 19–22 July 2011.
47. Buono, N.D.; Pio, G. Non-negative Matrix Tri-Factorization for co-clustering: An analysis of the block matrix. *Inf. Sci.* **2015**, *301*, 13–26. [CrossRef]
48. Ma, H.; Zhao, W.; Tan, Q.; Shi, Z. Orthogonal Nonnegative Matrix Tri-factorization for Semi-supervised Document Co-clustering. *Adv. Knowl. Discov. Data Min.* **2010**, *6119*, 189–200. [CrossRef]
49. Pei, Y.; Chakraborty, N.; Sycara, K. Nonnegative matrix tri-factorization with graph regularization for community detection in social networks. In Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.

50. Merris, R. Laplacian matrices of graphs: A survey. *Linear Algebra Its Appl.* **1994**, *197–198*, 143–176. [CrossRef]
51. Guillaumin, M.; Mensink, T.; Verbeek, J.; Schmid, C. TagProp: Discriminative Metric Learning in Nearest Neighbor Models for Image Auto-Annotation. In Proceedings of the 12th IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2010.
52. Oliva, A.; Torralba, A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
53. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
54. Weijer, J.V.D.; Schmid, C. Coloring Local Feature Extraction. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006.
55. Hua, W.; Huang, H.; Ding, C. Multi-label Feature Transform for Image Classifications. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010.
56. Boyd, S.; Vandenberghe, L. Convex Optimization. *IEEE Trans. Autom. Control* **2006**, *51*, 1859. [CrossRef]
57. Lee, D.D.; Seung, H.S. *Algorithms for Non-Negative Matrix Factorization*; NIPS; MIT Press: Cambridge, MA, USA, 2000; pp. 556–562.
58. Grangier, D.; Bengio, S. A Discriminative Kernel-Based Approach to Rank Images from Text Queries. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1371–1384. [CrossRef]
59. Ji, S.; Tang, L.; Yu, S.; Ye, J. A shared-subspace learning framework for multi-label classification. *Acm Trans. Knowl. Discov. Data* **2010**, *4*, 1–29. [CrossRef]
60. Wang, H.; Huang, H.; Ding, C. Image annotation using bi-relational graph of images and semantic labels. In Proceedings of the Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011.
61. Kong, D.; Ding, C.; Huang, H.; Zhao, H. Multi-label ReliefF and F-statistic feature selections for image annotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
62. He, Z.; Chen, C.; Bu, J.; Li, P.; Cai, D. Multi-view based multi-label propagation for image annotation. *Neurocomputing* **2015**, *168*, 853–860. [CrossRef]
63. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier chains for multi-label classification. *Mach. Learn.* **2011**, *85*, 333. [CrossRef]
64. Tsoumakas, G.; Katakis, I.; Vlahavas, I. Random k-Labelsets for Multilabel Classification. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1079–1089. [CrossRef]
65. Yang, Z.; Oja, E. Linear and nonlinear projective nonnegative matrix factorization. *IEEE Trans Neural Netw* **2010**, *21*, 734–749. [CrossRef] [PubMed]