remote sensing

MDPI

Article

Remote Sensing and Texture Image Classification Network Based on Deep Learning Integrated with Binary Coding and Sinkhorn Distance

Chu He [1,2,*,†], Qingyi Zhang [1,†], Tao Qu [3], Dingwen Wang [3] and Mingsheng Liao [2]

[1] School of Electronic Information, Wuhan University, Wuhan 430072, China; zhqy@whu.edu.cn
[2] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; liao@whu.edu.cn
[3] School of Computer Science, Wuhan University, Wuhan 430072, China; qutaowhu@whu.edu.cn (T.Q.); wangdw@whu.edu.cn (D.W.)
* Correspondence: chuhe@whu.edu.cn; Tel.: +86-27-6875-4367
† These authors contributed equally to this work.

Received: 7 November 2019; Accepted: 29 November 2019; Published: 3 December 2019

check for updates

Abstract: In the past two decades, traditional hand-crafted feature based methods and deep feature based methods have successively played the most important role in image classification. In some cases, hand-crafted features still provide better performance than deep features. This paper proposes an innovative network based on deep learning integrated with binary coding and Sinkhorn distance (DBSNet) for remote sensing and texture image classification. The statistical texture features of the image extracted by uniform local binary pattern (ULBP) are introduced as a supplement for deep features extracted by ResNet-50 to enhance the discriminability of features. After the feature fusion, both diversity and redundancy of the features have increased, thus we propose the Sinkhorn loss where an entropy regularization term plays a key role in removing redundant information and training the model quickly and efficiently. Image classification experiments are performed on two texture datasets and five remote sensing datasets. The results show that the statistical texture features of the image extracted by ULBP complement the deep features, and the new Sinkhorn loss performs better than the commonly used softmax loss. The performance of the proposed algorithm DBSNet ranks in the top three on the remote sensing datasets compared with other state-of-the-art algorithms.

Keywords: image classification; deep features; hand-crafted features; Sinkhorn loss

1. Introduction

1.1. Background

Image classification has always been an important basic problem in computer vision, and it is also the basis of other high-level visual tasks such as image detection, image segmentation, object tracking, and behavior analysis [1]. To propose an effective method to extract features which can represent the characteristics of the image is always critical in image classification [2]. The methods of extracting features can be divided into hand-crafted feature based methods and deep feature based methods. Before the rise of feature learning, people mostly used hand-crafted feature based methods to extract the essential features of the image such as edge, corner, texture, and other information [3]. For example, Laplacian of Gaussian (LoG) operator [4] and Difference of Gaussian (DoG) operator [5] are designed for detecting blobs in the image, scale invariant feature transform (SIFT) [6] is independent of the size and rotation of the object, local binary pattern (LBP) [7] has rotation invariance and gray invariance, features from accelerated segment test (FAST) operator [8] has high computational performance and high repeatability, bag of visual words model [9] pays more attention to the statistical information of features, Fisher vector [10] expresses an image with a gradient vector of likelihood functions, etc. A few of the most successful methods of texture description are the LBP and its variants such as uniform local binary pattern (ULBP) [11], COVariance and LBP Difference (COV-LBPD) [12], median robust extended LBP (MRELBP) [13], fast LBP histograms from three orthogonal planes (fast LBP-TOP) [14]. The ULBP proposed by Ahonen et al. reduces the number of binary patterns of LBP and is robust for high frequency noise. Hong et al. proposed the LBP difference (LBPD) descriptor and the COV-LBPD descriptor. The LBPD characterizes the extent to which one LBP varies from the average local structure of an image region of interest, and the COV-LBPD is able to capture the intrinsic correlation between the LBPD and other features in a compact manner. The MRELBP descriptor proposed by Liu et al. was computed by comparing image medians over a novel sampling scheme, which can capture both microstructure and macrostructure texture information and has attractive properties of strong discriminativeness, grayscale and rotation invariance, and computational efficiency. Hong et al. proposed the fast LBP-TOP descriptor which fastens the computational efficiency of LBP-TOP on spatial-temporal information and introduced the concept of tensor unfolding to accelerate the implementation process from three-dimensional space to two-dimensional space.

Since the rise of feature learning, deep learning methods have become a research hotspot and have broad application prospects and research value in many fields such as speech recognition and image classification [15]. Deep learning architectures mainly include four types: the autoencoder (AE), deep belief networks (DBNs), convolutional neural network (CNN), and recurrent neural network (RNN) [16]. Among these four deep learning architectures, CNN is the most popular and most published to date. For example, neural networks such as GoogLeNet [17], VGGNet [18], and residual neural network (ResNet) [19] have performed well in the field of image classification. GoogLeNet proposes an inception module, VGGNet explores the effects of the depth of deep neural network, and ResNet solves the problem of degradation of deep networks. These deep learning algorithms build the reasonable model by simulating a multi-layer neural network. High-level layers pay more attention to semantic information and less attention to detail information, while low-level layers are more concerned with detailed information and less with semantic information.

The deep learning algorithms have automatic feature learning capabilities for image data relying on large training sets and large models [20], while traditional methods rely primarily on hand-crafted features. Despite the success of deep features, the hand-crafted LBP texture descriptor and its variants have proven to provide competitive performance compared to deep learning methods in recent texture recognition performance evaluation, especially when there are rotations and multiple types of noise [21]. The LBP method introduces the priori information by presetting thresholds, so it can directly extract useful features through a manually designed algorithm, while the acquisition of deep features with excellent performance requires large training sets and large models. Therefore, there are aspects where hand-crafted features and deep features can learn from each other. For example, Courbariaux et al. proposed the BinaryConnect, which means training a DNN with binary weights during the forward and backward propagations, while retaining precision of the stored weights [22]. Hubara et al. introduced a method to train binarized neural networks (BNNs)—neural networks with binary weights and activations at run-time, and the binary weights and activations are used for computing the parameter gradients at train-time [23]. Inspired by the characteristics of hand-crafted features and deep features, this paper mainly studies the complementary performance between deep features and binary coded features and proposes a more effective feature description method.

During the training of the model, the loss function is used to assess the degree to which a specific algorithm models the given data [24]. The better the loss function, the better the performance of the algorithm. The design of the loss function can be guided by two strategies: empirical risk minimization and structural risk minimization. The average loss of the model on the training data set is called empirical risk, and the strategy of minimizing empirical risk is that the model with the least empirical risk is the best model. The related loss functions include center loss [25] and large-margin softmax loss [26], which are typical improved versions of softmax loss. When the size of training set is small or the model is complex, the model with the least empirical risk makes it easy to overfit the data. The strategy of structural risk minimization adds a regularization term based on the empirical risk minimization strategy. The structural risk minimization strategy means that the model with the least structural risk is the optimal model. The commonly used regularization terms include L1-regularization and L2-regularization. Sinkhorn distance [27] is the approximation of Earth mover's distance (EMD) [28] which can be used as a loss function. Different from other distance functions, EMD solves the correlation between two distributions by a distance matrix and a coupling matrix related to the predicted probability distribution and the actual probability distribution. The presetting distance matrix can increase the influence of the inter-class distance on the loss value, thereby improving the performance of the model. However, when EMD is used as loss function, there will be the problem of excessive computational complexity. Thus, as an approximate representation of EMD, the Sinkhorn distance which adds an entropic regularization term based on EMD is introduced as the loss function of the proposed model. The added entropic constraint turns the transport problem between distributions into a strictly convex problem that can be solved with matrix scaling algorithms and avoids the overfitting program.

This paper mainly verifies the performance of the proposed image classification algorithm in texture classification and remote sensing scene classification. Texture classification is an important basic problem in the field of computer vision and pattern recognition as well as the basis of other visual tasks such as image segmentation, object recognition, and scene understanding. However, texture is only the feature of the surface of an object, which cannot fully reflect the essential properties of the object. High-level image features cannot be obtained using only texture features [29]. Remote sensing scene classification is challenging due to several factors, such as large intra-class variations, small inter-class differences, scale changes, and illumination changes [30]. With the rise of remote sensing instruments, a large amount of satellite data has appeared in the field of remote sensing. Therefore, deep learning is gradually introduced into the image classification of remote sensing scenes. There are wild applications receiving more and more attention, such as land cover classification and target detection.

1.2. Problems and Motivation

Firstly, the features used for classification in the deep model are often global features extracted by high-level layers near the end of the model, with very few local features of the image. The global features have proven important in classification tasks but the local features can enhance the discriminability of features and are also helpful for image classification. Besides, when there are rotation and noise in the image, the traditional hand-crafted features have proven to provide competitive performance compared to the deep features. These two types of features have their own characteristics and can be used for reference for each other in some aspects.

Secondly, after the feature fusion, not only the diversity of features but also the redundancy of features is increased. When training the model by minimizing the loss function, we would like to remove the redundancy of the fused features and maximize the inter-class distance to improve classification performance of the model. However, the common used softmax loss in deep learning usually has insufficient feature distinguishing ability [31]. The loss function that is more suitable for the algorithm needs to be proposed.

1.3. Contributions and Structure

This paper presents a remote sensing and texture image classification network, which is based on deep learning integrated with binary coding and Sinkhorn distance. The general framework of the proposed algorithm is shown in Figure 1.

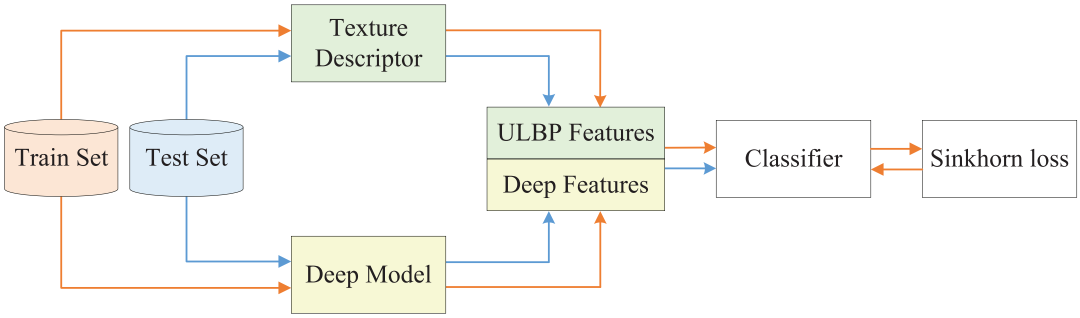


Figure 1. The general framework of the proposed algorithm, deep learning integrated with binary coding and Sinkhorn distance (DBSNet).

The contributions of this paper are summarized as follows.

Firstly, since the deep feature based methods and hand-crafted feature based methods are complementary in some aspects, we combine these two kinds of features to obtain better features to characterize the image. Specifically, the hand-crafted binary coding features extracted by ULBP are introduced to supplement the deep features extracted by representative ResNet-50 in classification, which makes the image features more accurate and comprehensive.

Secondly, a new loss function is proposed, which combines the score function with the Sinkhorn distance to predict the class. Sinkhorn loss analyzes the loss value between distributions from the perspective of doing work and removes the redundancy of the fused features with an entropic regularization term. Since the Sinkhorn distance is bounded from below by the distance between the centers of mass of the two signatures when the ground distance is induced by a norm, we can increase the impact of the inter-class distance on the loss value by presetting the distance matrix to guide the optimization process of the model.

The following sections are arranged as follows: Section 2 introduces the related work; Section 3 shows the proposed image classification algorithm; Section 4 introduces the experiments on texture datasets and remote sensing scene datasets; Section 5 introduces the summary of this paper.

2. Preliminaries

2.1. Deep Feature for Image Classification

Deep learning was successfully applied to image classification, and it is possible to approximate the complex functions of human visual perception by rationally combining several basic modules. The basic modules of the deep model are mainly information extraction module, activation and pooling module, and tricks module. The information extraction module extracts features from the input. The activation and pooling module is mainly devoted to nonlinear transformation and dimensionality reduction. The tricks module can speed up the training procedure and avoid overfitting [32]. The information extraction module of CNN is mainly composed of convolutional layers. Receptive fields are used to describe the area of the input image which can affect the features of the CNN. Figure 2 shows that as the depth of the network deepens, the receptive field of the posterior neurons increases, and the extracted features also change from low-level features such as edge information to mid-level features such as texture information and high-level features such as structural information.

The high-level features are often used for classification in CNN, losing a lot of detailed information, such as edge features and texture features, which may lead to poor performance in image classification requiring more detailed information. In order to improve the discriminative ability of the model, texture features can be used to complement the discriminability of deep features.

Deep models are widely used in texture classification and remote sensing scene classification. ResNet is one of the best deep models for image classification. ResNet was proposed by Kaiming He et al. in 2015. A 152 layer deep neural network was trained with a special network structure and won the championship on the ImageNet competition classification task (top-5 error rate: 3.57%). ResNet overcame the difficulty that the deep network could not be trained, and not only the accuracy of classification was improved, but also the parameter quantity was less than the VGG model. Consequently, ResNet-50 is used as the deep feature extractor in the proposed algorithm.

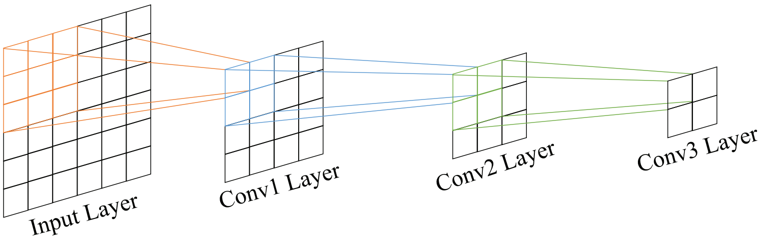


Figure 2. The relationship between receptive field and network depth.

ResNet deepens the network without reducing the accuracy of classification by residual learning. Based on the existing design ideas (batch normalization, small convolution kernel, and fully convolution network), the residual module is introduced. Each residual module contains two paths, one of which performs two or three convolution operations on the input feature to obtain the residual of the feature; the other path is the direct path of the input feature. The outputs of these two paths are finally added together to be the output of the residual module. There is an example of residual module shown in Figure 3. The first 1 × 1 convolution in the module is used to reduce the dimension (from 256 to 64), and the second 1 × 1 convolution is used to upgrade the dimension (from 64 to 256). Consequently, the number of input and output channels of the intermediate 3 × 3 convolution is small (from 64 to 64), and the parameters to be learned can be significantly reduced.

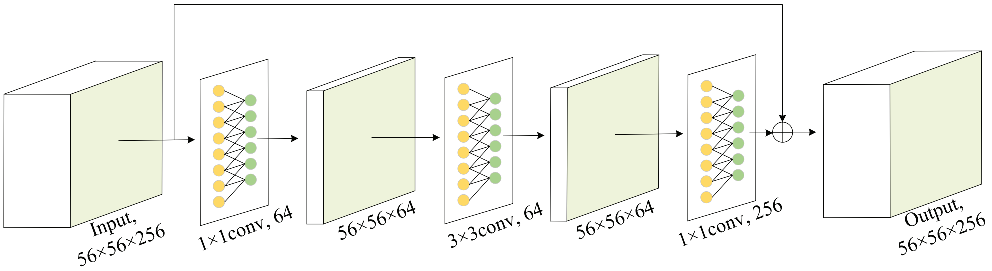


Figure 3. Residual module in ResNet-50.

Figure 4 shows the framework of ResNet-50, in which the residual modules are repeated 3 times, 4 times, 6 times, and 3 times respectively. The deep features of the image are extracted using the fine-tuned ResNet-50.

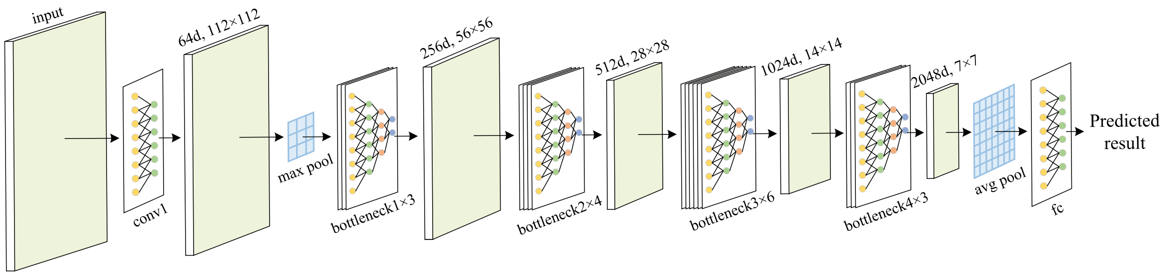


Figure 4. The framework of ResNet-50.

2.2. Binary Coded Feature for Image Classification

In the field of texture recognition, LBP is one of the most commonly used texture description methods, which was firstly proposed by T. Ojala et al. in 1994. It was originally developed as a method of describing texture images and later improved for image feature analysis. LBP has significant advantages such as gray invariance and rotation invariance [29]. LBP has been extensively researched in many fields and has demonstrated outstanding performance in several comparative studies [33,34]. The LBP descriptor works by thresholding the values of its neighborhood pixels, while the threshold is set as the value of the center pixel. The LBP descriptor is capable of detecting local primitives, including flat regions, edges, corners, curves, and edge ends, and it was later extended to obtain multi-scale, rotational invariant, and uniform representations and has been successfully applied to other tasks, such as object detection, face recognition [11], and remote sensing scene classification. The framework of traditional LBP can be presented by Figure 5, which can be divided into three steps. Firstly, the binary relationship between each pixel in the image and its local neighborhood is calculated in grayscale. Then, the binary relationship is weighted into an LBP code according to certain rules. Finally, the histogram sequence obtained by statistics in the LBP image is described as image features.

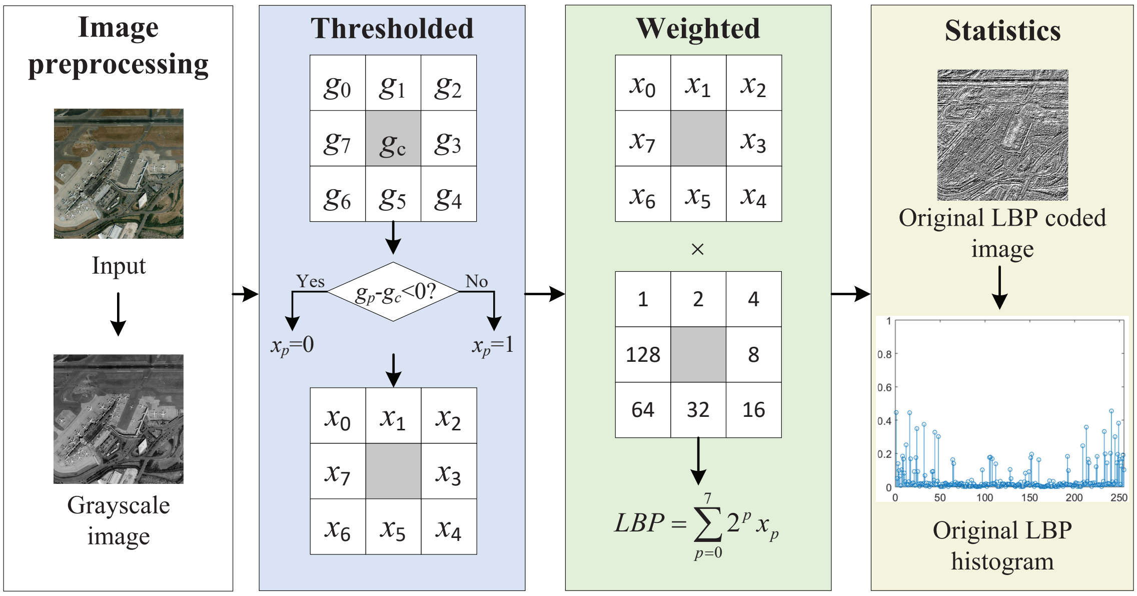


Figure 5. Calculation of the local binary pattern (LBP).

The traditional LBP algorithm can be expressed by Equations (1) and (2):

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \tag{1}$$

where

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}.$$

P is the number of neighborhoods, R represents the distance between the central pixel and the neighborhood pixel, and we set $R = 1$ here. g_c is the grayscale value of the center pixel, and g_p is the grayscale value of the neighborhood pixel. Compared with the value of center pixel, the value of neighborhood pixel is set as 1 when it is greater or 0 when it is less. Then binary encoding is performed in a certain order. For those pixels which are not at the center of the neighborhood pixels, the grayscale values of their neighborhood pixels can be estimated by linear interpolation. Finally, by traversing all LBP pixel values, a histogram is created to represent the texture features of the image. Assuming the image size is $I \times J$, the histogram is expressed as:

$$H(k) = \sum_{i=1}^{I} \sum_{j=1}^{J} f(LBP_{P,R}(i,j),k), \tag{2}$$

where

$$k \in [0, 255],$$

$$f(x,y) = \begin{cases} 1, & x = y \\ 0, & otherwise \end{cases}.$$

As one of the binary coded feature extractors, LBP can effectively deal with illumination changes and is widely used in texture analysis and texture recognition. However, with the increase of the variety of patterns, the computational complexity and the data volume of the traditional LBP method will increase sharply. LBP will also be more sensitive to noise, and the slight fluctuations of the central pixel may cause the coded results to be quite different. In order to solve the problem of excessive binary patterns and make the process of statistic more concise, Ojala et al. proposed a uniform pattern to reduce the dimension of the patterns. Ojala believed that most patterns only contain up to two jumps from 1 to 0 or from 0 to 1 in the natural image. Therefore, Ojala defined the "Uniform Pattern", that is, when a loop binary pattern has a maximum of two jumps from 0 to 1 or from 1 to 0, the pattern is called a uniform pattern, such as 00000000 (0 jump), 10001111 (first jump from 1 to 0, then jump from 0 to 1). Except for the uniform pattern, other patterns are classified as mixed pattern, such as 10010111 (totally 4 jumps). The framework of ULBP is shown in Figure 6. $U(LBP_{P,R})$ can be used to indicate the number of jump in the code, which is calculated as follows:

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \tag{3}$$

Through such an improvement, the number of patterns is reduced from the original 2^P to $P \times (P-1) + 3$, where P represents the number of sampling points in the neighborhood. For 8 sampling points in the 3×3 neighborhood, the number of binary pattern is reduced from 256 to 59. The values of uniform pattern are assigned from 1 to 58 in ascending order, and the mixed pattern is assigned 0. Since the range of grayscale value is 0–58, the ULBP feature image is entirely dark, which makes the feature vector less dimensional and less impacted by high frequency noise.

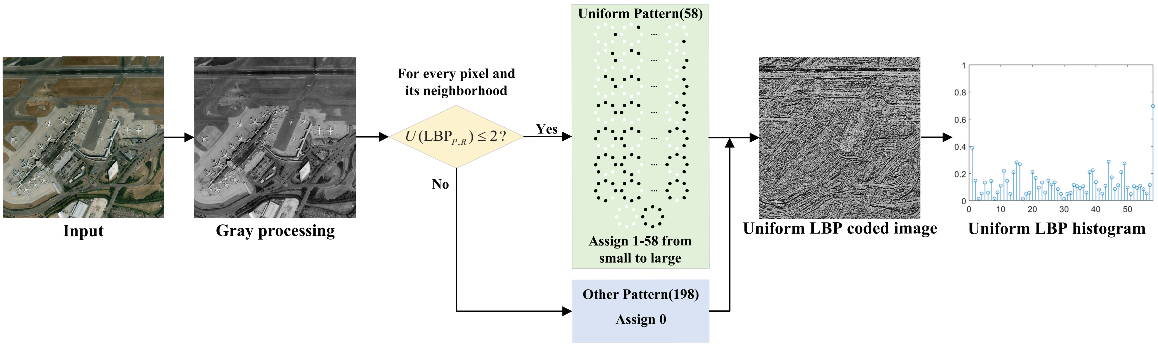


Figure 6. The framework of uniform local binary pattern (ULBP).

3. Methodology

3.1. Sinkhorn Loss

The Sinkhorn loss consists of softmax function and Sinkhorn distance. When the score vector is output from the fully connected layer, we convert it into a probability distribution by the softmax function and then calculate distance between the actual distribution and the predicted distribution using Sinkhorn distance. The approximate solution of optimal transport problem between two distributions can be determined by iterative learning. The advantages of Sinkhorn distance in calculating the distance between two distributions will be introduced next.

Two signatures, P and Q, are defined to represent the predicted distribution and the actual distribution, with m classes respectively. These two signatures can be represented by Equations (4) and (5), where p_i is the label in P, w_{p_i} is the probability of p_i in P, q_j is the label in Q, and w_{q_j} is the probability of q_j in Q. Here we set $p_i = i$ and $q_j = j$ to represent the different labels. The value of w_{p_i} is determined by the output of softmax function. The value of w_{q_i} is determined based on the real class. For a specific sample, if i is the real class of it, we set $w_{q_i} = 1$, otherwise we set $w_{q_i} = 0$.

$$P = \left\{ \left(p_1, w_{p_1}\right), \ldots, \left(p_m, w_{p_m}\right) \right\}, \tag{4}$$

$$Q = \left\{ \left(q_1, w_{q_1}\right), \ldots, \left(q_m, w_{q_m}\right) \right\}. \tag{5}$$

In order to measure the work of transforming one distribution into another, two matrices are introduced: the distance matrix D and the coupling matrix F. Each element d_{ij} in the distance matrix D represents the distance of moving p_i to q_j. Here we set $d_{ij} = 1$ when $i \neq j$ and $d_{ij} = 0$ when $i = j$. Each element f_{ij} in the coupling matrix F indicates the probability quality that needs to be assigned when moving from p_i to q_j. According to the above definition, the total cost $t(P, Q)$ can be calculated by the Frobenius inner product between F and D:

$$t(P,Q) = \langle D, F \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij} f_{ij}. \tag{6}$$

The goal is to find an optimal coupling matrix F^* that minimizes the overall cost function, and the least cost function over all coupling functions is the solution to this optimal transport problem, called EMD.

$$F^* = \arg\min_F t(P,Q), \tag{7}$$

$$EMD = \frac{\min\limits_F t(P,Q)}{\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{m} f_{ij}}, \tag{8}$$

s.t.

$$f_{ij} \geq 0,$$

$$\sum_{j=1}^{m} f_{ij} \leq w_{p_i},$$

$$\sum_{i=1}^{m} f_{ij} \leq w_{q_j},$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} = \min(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{m} w_{q_j}).$$

EMD has a complicated calculation method for finding the optimal solution and is not suitable as a loss function. However, when solving the distance between distributions, it can increase the influence of the inter-class distance on the cost function by reasonably presetting the distance matrix. Thus, we introduce the Sinkhorn distance as loss function which is the approximate value of EMD. It smooths the classic optimal transport problem with an entropic regularization term. The solution to the problem can be rewritten as:

$$For \lambda > 0, SD := \left\langle D, F^{\lambda} \right\rangle, \tag{9}$$

where

$$F^{\lambda} = \arg\min_F t(P,Q) - \frac{1}{\lambda} h(F),$$

$$h(F) = -\sum_{ij} F_{ij} \log F_{ij}.$$

λ is the regularization coefficient. When λ grows, a slower convergence can be observed as F^{λ} gets closer to the optimal vertex F^{*}, but the computational complexity will also rise at the same time. Thus we take $\lambda = 10$ where the computational complexity and the accuracy of the approximate solution reach the compromise. By introducing entropy regularization term, the transport problem is turned into a strictly convex problem that can be solved with Sinkhorn's matrix scaling algorithm at a speed which is several orders of magnitude faster than that of transport solvers. For $\lambda > 0$, the solution F^{λ} of the problem is unique and has the form $F^{\lambda} = \mathrm{diag}(u)K\mathrm{diag}(v)$, where u and v are two non-negative vectors of $\mathbb{R}^{m}$ and K is the element-wise exponential of $-\lambda D$.

3.2. Integrating Deep Learning with Binary Coding for Texture and Remote Sensing Image Classification

Nowadays, the networks used for image classification are generally trained and tested through an end-to-end network, and the classification accuracy is improved by optimizing the parameters of the feature extractor and classifier. However, the features extracted by the deep network have limitations. In order to improve the performance of the classification algorithm, the local texture information obtained by the ULBP of the image is used as the supplementary features. This paper combines it with the deep features as the input of fully connected layer, and the optimization of network parameters is guided by Sinkhorn loss. The framework of the two stream model is shown in Figure 7.

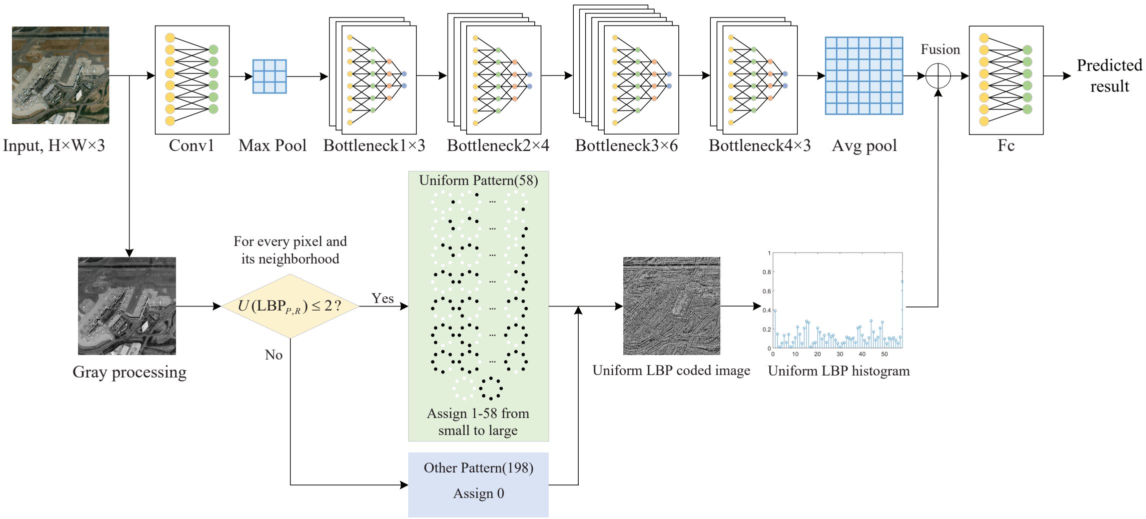
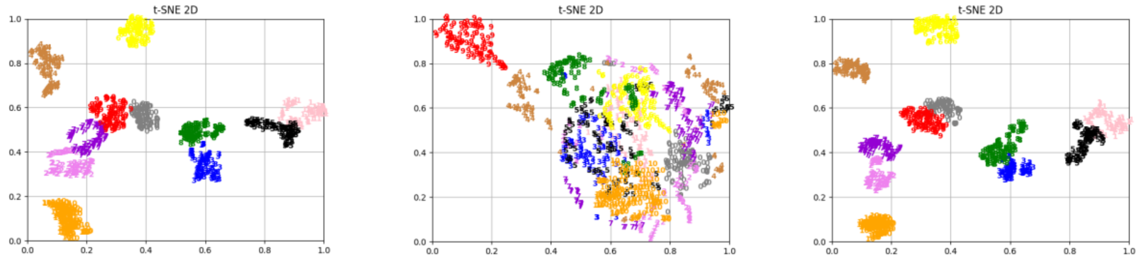


Figure 7. The detailed framework of the proposed algorithm: DBSNet.

The ResNet-50 is pre-trained on the ImageNet 2012 dataset used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [35] and then the original softmax with cross-entropy loss is replaced with the Sinkhorn loss to get a new network (RSNet). Finally, we fine-tuned the RSNet on different datasets and removed the fully connected layer and the classifier to get the deep feature extraction network. The binary coded feature extractor is the ULBP algorithm. The input of the model is an RGB image. Firstly, 2048 dimensional features are extracted through the deep feature extractor. At the same time, the image is grayscale processed and encoded by ULBP to get the 59 dimensional local texture features. After the two sets of features are fused, the class of image is predicted by the output of the fully connected layer.

In order to clearly observe the difference before and after the feature fusion, t-distributed stochastic neighbor embedding (t-SNE) [36] is used to visualize the pre-fusion deep features, ULBP features and the merged DBSNet features extracted on KTH-TIPS2-b texture dataset in the 2D space. The results are shown in Figure 8. As shown in the figure, the deep features have good image characterization capabilities, but the samples of the same class are more scattered. The LBP features have certain image characterization capabilities but the discriminability is not good. The DBSNet features combine the deep features and the LBP features. It can be seen from the reduced-dimensional image features that the image feature representation capability of the DBSNet features is better than the deep features and the ULBP features and the samples of the same class are more compact, indicating that the ULBP features complement deep features.



(a) RSNet features (b) ULBP features (c) DBSNet features

Figure 8. Comparison of feature maps of RSNet, ULBP, and DBSNet algorithms on KTH-TIPS2-b dataset.

4. Experiment

The performance of the proposed algorithm will be verified on two texture datasets and five remote sensing scene datasets. Firstly, the texture recognition performance of the algorithm is verified on two texture datasets and compared with the ResNet-50, RSNet, and several typical LBP-derived algorithms. Then, the remote sensing scene classification performance of this algorithm is evaluated on five remote sensing scene datasets and compared with the ResNet-50, RSNet, and the representative remote sensing scene classification algorithm.

4.1. Experimental Data

4.1.1. Texture Dataset

The performance of the proposed algorithm is firstly validated on two classic texture datasets: KTH-TIPS2-a dataset and KTH-TIPS2-b dataset.

The KTH-TIPS2-a dataset includes 11 classes of texture images. Most classes of the textures are shot in nine different scales, three poses, and four different lighting conditions, for a total of 4608 images, each with a pixel size of 200 × 200. We use three sets of samples as the train set and one set of samples as the test set and perform four experiments, with the average of four results as the final result.

The KTH-TIPS2-b dataset includes 11 classes of texture images, each of which is shot in nine different scales, three poses, and four different lighting conditions, for a total of 4752 images, each with a pixel size of 200 × 200. We use one set of samples as the train set and three sets of samples as the test set and perform four experiments, with the average of four results as the final result.

There are some examples of these texture datasets shown in Figure 9.

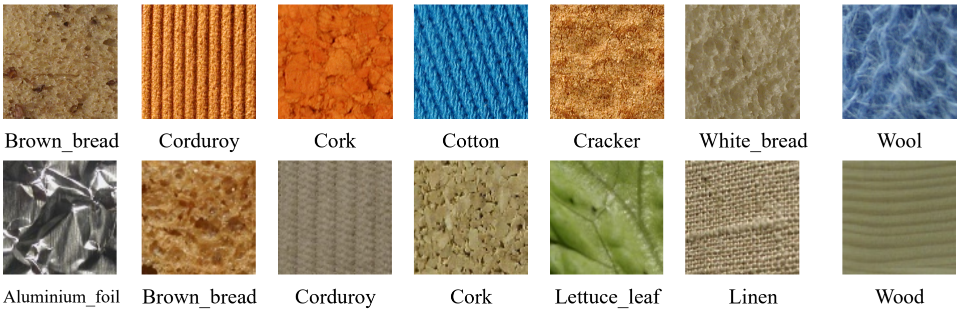


Figure 9. Example images of two texture datasets from top to bottom: KTH-TIPS2-a and KTH-TIPS2-b.

4.1.2. Remote Sensing Scene Dataset

Besides the texture image classification, the performance of the algorithm is also validated on five remote sensing scene datasets: AID dataset, RSSCN7 dataset, UC Merced Land-Use dataset, WHU-RS19 dataset, and OPTIMAL-31 dataset.

AID dataset [37] contains 30 classes of scene images, each class has about 200 to 400 samples, a total of 10,000, and each image has a pixel size of 600 × 600. Each class of images is randomly selected with ratio of 20:80 to obtain the train and test set.

RSSCN7 dataset [38] contains seven classes of scene images, each with 400 samples, a total of 2800, and each image has a pixel size of 400 × 400. Each class of images is randomly selected with ratio of 50:50 to obtain the train and test set.

UC Merced Land-Use dataset [39] contains 21 classes of scene images, each with 100 samples, a total of 2100, and each image has a pixel size of 256 × 256. Each class of images is randomly selected with ratio of 50:50 to obtain the train and test set.

WHU-RS19 dataset [40] contains 19 classes of scene images, each with about 50 samples, a total of 1005, and each image has a pixel size of 600 × 600. Each class of images is randomly selected with ratio of 60:40 to obtain the train and test set.

OPTIMAL-31 dataset [41] contains 31 classes of scene images, each with 60 samples, a total of 1860, and each image has a pixel size of 256 × 256. Each class of images is randomly selected with ratio of 80:20 to obtain the train and test set.

There are some examples of these remote sensing scene datasets shown in Figure 10.

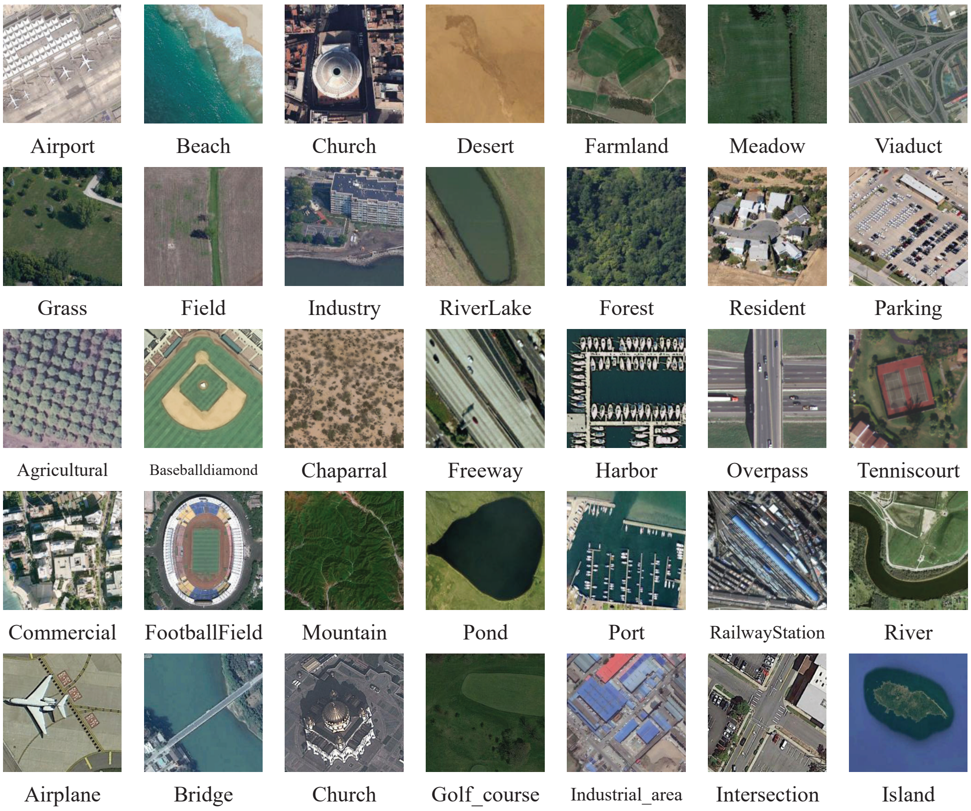


Figure 10. Example images of five remote sensing scene classification datasets from top to bottom: AID, RSSCN7, UC-Merced, WHU-RS19, and OPTIMAL-31.

4.2. Experimental Setup

Performance of the algorithms in the experiments is measured by the overall accuracy (OA) and the confusion matrix (CM) on the test set. The classification accuracy over all scene categories in a dataset is calculated according to $\frac{S_P}{S_T}$, where S_P is the number of correct predictions in the test set and S_T is the total number of images in the test set. The CM allows us to clearly see the classification accuracy of the algorithm for each type of image in the dataset.

In order to verify the performance of the proposed algorithm, we compare the proposed algorithm DBSNet with several representative algorithms on texture datasets and remote sensing datasets. For texture datasets, we compare the DBSNet with the hand-crafted texture feature descriptors ULBP and some efficient and recently proposed LBP derived algorithms such as COV-LBPD, MRELBP, and fast LBP-TOP. We experiment using the source code on the texture datasets. After the feature extraction by the texture feature descriptors, classification using nearest neighbors is conducted. In the proposed algorithm DBSNet, ResNet-50 is one solution for deep feature extractors. Because the ResNet-50 model is complex and the dimension of extracted features is large, we replace ResNet-50 with a shallow CNN model shown in Figure 11 and do the classification experiments on texture datasets to further verify the complementary effect of the hand-crafted texture features on the deep features. The network is trained and tested on four different train-test sets respectively, and then four feature extractors are obtained after removing the fully connected layer. For each feature extractor, we extract the deep features and classify them by the fully connected layer. The deep features fused with ULBP features are also classified by the fully connected layer to obtain the performance of the fused features.

For remote sensing datasets, we compare the proposed method DBSNet with the classic image classification algorithms IFK-SIFT [10], CaffeNet [42], VGG-VD-16 [18], GoogLeNet, ARCNet-VGGNet16 [41], and GBNet + global feature [43]. In addition to comparing the original results in the references [37,41,43], we do experiments on OPTIMAL-31 dataset with IFK-SIFT, CaffeNet, VGG-VD-16, and GoogLeNet referring to the parameter settings in reference [37]. We extract the deep features using the pretrained models without the fully connected layers on ImageNet and the IFK-SIFT features and then classify them respectively by the liblinear [44] for 10 times and take the mean accuracy as the result. Considering that the ResNet-50 used in the proposed methods are fine-tuned for better performance, we fine-tune the deep models CaffeNet, VGG-VD-16, and GoogLeNet for further comparison. We change the output channels of the last fully connected layer and optimize the parameters of deep models with the stochastic gradient descent (SGD). The detailed parameter settings are listed in Table 1.

Table 1. Parameter settings of the deep models.

Parameter	**Batch Size**	**Learning Rate**	**Momentum**	**Weight Decay**
Value	60	0.0001	0.9	0.0001

Besides the comparison methods mentioned above, three different algorithms are to be compared based on the difference of feature extraction method and the loss function on both texture datasets and remote sensing datasets, which are the fine-tuned ResNet-50, RSNet, and DBSNet algorithms. These three comparison algorithms are respectively tested to verify whether the deep features extracted by the RSNet and the statistical texture features obtained by the ULBP are complementary and whether the proposed Sinkhorn loss has robust performance.

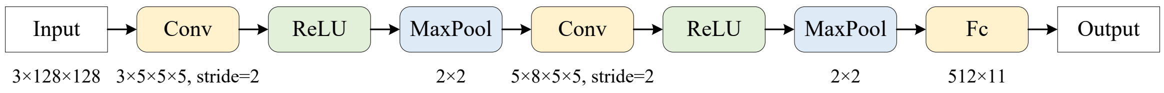


Figure 11. The framework of the shallow convolutional neural network (CNN).

4.3. Experimental Results and Analysis

In this section, we report the classification performance of the proposed DBSNet and other methods for comparison on challenging texture datasets and remote sensing scene classification datasets respectively.

4.3.1. Experiments on Texture Dataset

For the texture recognition, the classification results given in Table 2 show the performance comparison of the different algorithms on KTH-TIPS2-a and KTH-TIPS2-b texture datasets. The accuracy of the best performing algorithm is bolded for different databases. It can be seen that on the KTH-TIPS2-a and KTH-TIPS2-b datasets, the traditional hand-crafted methods are not competitive, and the ResNet-50, the RSNet, and the DBSNet provide incremental performance, which proves that the performance of the Sinkhorn loss is excellent and the features obtained by the ULBP are complementary to the deep features.

Table 2. Classification accuracy of different algorithms on KTH-TIPS2-a and KTH-TIPS2-b texture datasets.

	ULBP	**COV-LBPD**	**MRELBP**	**Fast LBP-TOP**	**ResNet-50 (Fine-Tuning)**	**RSNet**	**DBSNet**
KTH-TIPS2-a	0.6014	0.6291	0.6342	0.6058	0.8247	0.8321	**0.8359**
KTH-TIPS2-b	0.2628	0.5588	0.5475	0.2499	0.7379	0.7458	**0.7511**

Tables 3 and 4 are the confusion matrices of the RSNet algorithm and the DBSNet algorithm on KTH-TIPS2-b texture dataset which clearly reflect the classification performance on each category in the dataset. We compare these two confusion matrices and find that among the 11 classes, DBSNet algorithm outperforms RSNet algorithm in seven classes, which are aluminium foil, brown bread, cork, cracker, lettuce leaf, linen, and wood, and is inferior to the RSNet algorithm in three classes, which are corduroy, cotton, and wool. The overall classification performance of DBSNet is better than the RSNet algorithm, which proves the superiority of the proposed feature extraction method over the normal deep feature based method.

To further verify the complementary effect of the hand-crafted texture features on the deep features, we replace the RSNet feature extractor with a shallow CNN feature extractor. In Table 5, the accuracy of the best performing algorithm is bolded for different databases. It can be seen that the classification performance of the fused features is better than the deep features on four train-test sets of both KTHTIPS2-a and KTHTIPS2-b datasets. Consequently, the ULBP features complement the low-dimensional deep features of shallow CNN in classification task and even though the dimensions of deep features increase, the complement of ULBP features still exists, which has been proved in Table 2.

Table 3. Confusion matrix (CM) of RSNet algorithm on KTH-TIPS-2b dataset.

	Aluminium Foil	**Brown Bread**	**Corduroy**	**Cork**	**Cotton**	**Cracker**	**Lettuce Leaf**	**Linen**	**White Bread**	**Wood**	**Wool**
aluminium foil	0.9846	0	0	0	0	0	0	0.0154	0	0	0
brown bread	0	0.8549	0	0	0	0.0494	0	0	0.0957	0	0
corduroy	0.0123	0.0031	0.8117	0.0802	0.0062	0.0123	0	0.0463	0.0031	0.0031	0.0216
cork	0	0	0	0.8549	0	0.1204	0	0	0.0247	0	0
cotton	0	0	0.1358	0	0.2531	0	0	0.3827	0.0031	0.0463	0.1790
cracker	0	0.4846	0	0.0710	0	0.4414	0	0.0031	0	0	0
lettuce leaf	0	0	0	0	0.0062	0	0.9938	0	0	0	0
linen	0	0	0.0093	0	0.1790	0	0	0.8117	0	0	0
white bread	0	0	0	0	0	0	0	0	0.9877	0.0123	0
wood	0	0	0	0	0.0247	0	0	0	0.0278	0.9475	0
wool	0.0062	0.0031	0.0309	0.1636	0.0123	0	0	0.5216	0	0	0.2623

Table 4. CM of DBSNet algorithm on KTH-TIPS-2b dataset.

	Aluminium Foil	**Brown Bread**	**Corduroy**	**Cork**	**Cotton**	**Cracker**	**Lettuce Leaf**	**Linen**	**White Bread**	**Wood**	**Wool**
aluminium foil	1	0	0	0	0	0	0	0	0	0	0
brown bread	0	0.9352	0	0	0	0.0062	0	0	0.0586	0	0
corduroy	0.0741	0.0062	0.7130	0.0401	0.0031	0.0525	0.0031	0.0617	0.0062	0.0401	0
cork	0	0.0062	0	0.9475	0	0.0432	0	0	0.0031	0	0
cotton	0	0	0.2222	0	0.1265	0	0	0.4043	0.0031	0.0617	0.1821
cracker	0.0062	0.3457	0.0031	0.0432	0	0.5988	0	0	0	0.0031	0
lettuce leaf	0	0	0	0	0.0031	0	0.9969	0	0	0	0
linen	0	0	0.0031	0	0.1204	0.0031	0.0031	0.8673	0	0.0031	0
white bread	0	0.0062	0	0	0	0	0	0	0.9877	0.0062	0
wood	0	0	0	0	0.0062	0.0123	0	0	0.0216	0.9599	0
wool	0.1821	0.0062	0.0648	0.0772	0.0123	0	0	0.4630	0	0.0648	0.1296

Table 5. Classification accuracy of different feature sets on KTHTIPS2-a and KTHTIPS2-b texture datasets.

		Train-Test 1	**Train-Test 2**	**Train-Test 3**	**Train-Test 4**	**OA**
KTHTIPS2-a	Deep features	0.5097	0.4722	0.4318	0.5892	0.5007
	Deep features+ULBP features	0.5182	0.4882	0.4840	0.6002	**0.5227**
KTHTIPS2-b	Deep features	0.3131	0.3151	0.3361	0.4458	0.3525
	Deep features+ULBP features	0.3527	0.3561	0.4234	0.5208	**0.4133**

4.3.2. Experiments on Remote Sensing Scene Dataset

For the remote sensing scene classification, the results given in Table 6 show the performance comparison of the different algorithms on the five challenging remote sensing datasets. The accuracy of the top three best performing algorithms for different databases is bolded. It can be seen that the DBSNet algorithm provides better performance than the RSNet algorithm and the RSNet algorithm performs better than ResNet-50 algorithm on these five datasets, which demonstrates that the features obtained by ULBP still have the performance complementary to the deep features on remote sensing datasets and the proposed Sinkhorn loss can better guide the learning process of the network than the commonly used softmax loss. Compared with the mid-level method IFK-SIFT, the advanced deep feature based methods CaffeNet, VGG-VD-16, GoogLeNet, ARCNet-VGGNet16, and GBNet + global feature achieve improvement performance but the advanced deep feature based methods still have limitations in feature extraction. Based on the deep features, the algorithm DBSNet adds texture features that are instructive for image classification and uses a more suitable loss function. Compared with these representative methods, DBSNet always ranks in the top three on all five datasets.

Table 6. Classification accuracy of different algorithms on AID, RSSCN7, UC-Merced, WHU-RS19, and OPTIMAL-31 remote sensing scene classification datasets.

	AID	**RSSCN7**	**UC-Merced**	**WHU-RS19**	**OPTIMAL-31**
IFK-SIFT [37]	0.7192	0.8509	0.7874	0.8742	0.6022
CaffeNet [37]	0.8686	0.8825	0.9398	0.9624	0.8586
CaffeNet (fine-tuning)	0.8953	0.9043	0.9525	0.9550	0.8623
VGG-VD-16 [37]	0.8659	0.8718	0.9414	0.9605	0.8610
VGG-VD-16 (fine-tuning)	0.9036	0.9293	0.9552	0.9651	0.8737
GoogLeNet [37]	0.8344	0.8584	0.9270	0.9471	0.8454
GoogLeNet (fine-tuning)	0.9015	**0.9368**	0.9580	0.9650	0.8900
ARCNet-VGGNet16 [41]	0.8875	-	0.9681	**0.9975**	0.9270
GBNet + global feature [43]	0.9220	-	**0.9705**	**0.9925**	**0.9328**
ResNet-50 (fine-tuning)	**0.9233**	0.9312	0.9622	0.9751	0.9301
RSNet	**0.9281**	**0.9400**	**0.9762**	0.9800	**0.9328**
DBSNet	**0.9293**	**0.9521**	**0.9790**	**0.9875**	**0.9344**

The confusion matrices of RSNet algorithm and DBSNet algorithm are compared on RSSCN7 dataset to analyze the classification performance more carefully. It can be seen from Tables 7 and 8 that among the seven classes, DBSNet algorithm outperforms RSNet algorithm in five classes, which are

Grass, Industry, Forest, Resident, and Parking, and is second to the RSNet algorithm in two classes, which are Field and RiverLake. Generally speaking, the overall classification performance of DBSNet algorithm is better than the RSNet algorithm. As a complement, texture features play a role in the classification task.

Table 7. CM of RSNet algorithm on RSSCN7 dataset.

	Grass	Field	Industry	RiverLake	Forest	Resident	Parking
Grass	0.9150	0.0500	0.0100	0.0100	0	0.0100	0.0050
Field	0.0400	0.9600	0	0	0	0	0
Industry	0	0	0.8800	0.0150	0	0.0350	0.0700
RiverLake	0.0050	0.0150	0	0.9650	0.0150	0	0
Forest	0.0100	0.0150	0	0.0050	0.9700	0	0
Resident	0	0	0.0250	0.0050	0	0.9600	0.0100
Parking	0	0.0050	0.0600	0	0	0.0050	0.9300

Table 8. CM of DBSNet algorithm on RSSCN7 dataset.

	Grass	Field	Industry	RiverLake	Forest	Resident	Parking
Grass	0.9550	0.0300	0.0050	0.0100	0	0	0
Field	0.0450	0.9550	0	0	0	0	0
Industry	0	0	0.9000	0.0100	0	0.0350	0.0550
RiverLake	0.0150	0.0100	0.0100	0.9600	0.0050	0	0
Forest	0.0100	0	0	0.0050	0.9850	0	0
Resident	0	0	0.0250	0.0050	0	0.9700	0
Parking	0	0.0050	0.0450	0	0.0050	0.0050	0.9400

5. Conclusions

In this paper we have proposed a robust image classification algorithm based on deep learning integrated with binary coding and Sinkhorn distance. Taking into account the characteristics of hand-crafted features and deep features, we combine their advantages and supplement the deep features with the statistical texture features to fully describe the image. In order to remove redundant information from the fused features and train the model quickly and efficiently, we introduced the Sinkhorn loss where an entropy regularization term plays a key role. In this paper, experiments are carried out on two classic texture datasets and five remote sensing classification datasets. The experimental results show that compared with the ResNet-50, the proposed two stream model DBSNet can improve the overall performance when achieving image classification tasks. In addition, compared with the classic classification algorithms for remote sensing scene classification, the algorithm DBSNet can still provide better results. In the future, we will study how to combine the traditional feature extraction framework with the deep learning framework so that they guide and improve each other.

Author Contributions: Conceptualization, C.H.; Funding acquisition, C.H.; Investigation, Q.Z.; Methodology, C.H. and Q.Z.; Writing—original draft, Q.Z.; Writing—review & editing, T.Q., D.W. and M.L.

Funding: This research was funded by the National Key Research and Development Program of China (No. 2016YFC0803000), the National Natural Science Foundation of China (No. 41371342, No. 61331016), and the Hubei Innovation Group (2018CFA006).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, D.; Weng, Q. A survey of image classification methods and techniques for improving classification performance. *Int. J. Remote Sens.* **2007**, *28*, 823–870. [CrossRef]

2. Caicedo, J.C.; Cruz, A.; Gonzalez, F.A. Histopathology image classification using bag of features and kernel functions. In Proceedings of the Conference on Artificial Intelligence in Medicine in Europe, Verona, Italy, 18–22 July 2019; Springer: Berlin/Heidelberg, Germany, 2009; pp. 126–135.
3. Szummer, M.; Picard, R.W. Indoor-outdoor image classification. In Proceedings of the 1998 IEEE International Workshop on Content-Based Access of Image and Video Database, Bombay, India, 3 January 1998; pp. 42–51.
4. Marr, D.; Hildreth, E. Theory of edge detection. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **1980**, *207*, 187–217.
5. Davidson, M.W.; Abramowitz, M. *Molecular Expressions Microscopy Primer: Digital Image Processing-Difference of Gaussians Edge Enhancement Algorithm*; Olympus America Inc. and Florida State University: Tallahassee, FL, USA, 2006.
6. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision (ICCV 2019), Seoul, Korea, 27 October –2 November 1999; Volume 99, pp. 1150–1157.
7. Ojala, T.; Pietikäinen, M.; Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
8. Förstner, W.; Gülch, E. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In Proceedings of the ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, 2–4 June 1987; pp. 281–305.
9. Yang, J.; Jiang, Y.G.; Hauptmann, A.G.; Ngo, C.W. Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval, Augsburg, Germany, 24–29 September 2007; ACM: New York, NY, USA, 2007; pp. 197–206.
10. Perronnin, F.; Sánchez, J.; Mensink, T. Improving the fisher kernel for large-scale image classification. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 143–156.
11. Ahonen, T.; Hadid, A.; Pietikainen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041. [CrossRef]
12. Hong, X.; Zhao, G.; Pietikäinen, M.; Chen, X. Combining LBP difference and feature correlation for texture description. *IEEE Trans. Image Process.* **2014**, *23*, 2557–2568. [CrossRef]
13. Liu, L.; Lao, S.; Fieguth, P.W.; Guo, Y.; Wang, X.; Pietikäinen, M. Median robust extended local binary pattern for texture classification. *IEEE Trans. Image Process.* **2016**, *25*, 1368–1381. [CrossRef] [PubMed]
14. Hong, X.; Xu, Y.; Zhao, G. Lbp-top: A tensor unfolding revisit. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Springer: Cham, Switzerland, 2016; pp. 513–527.
15. Druzhkov, P.; Kustikova, V. A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognit. Image Anal.* **2016**, *26*, 9–15. [CrossRef]
16. Ball, J.E.; Anderson, D.T.; Chan, C.S. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *J. Appl. Remote Sens.* **2017**, *11*, 042609. [CrossRef]
17. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
18. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256.
21. Liu, L.; Fieguth, P.; Wang, X.; Pietikäinen, M.; Hu, D. Evaluation of LBP and deep texture descriptors with a new robustness benchmark. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 69–86.
22. Courbariaux, M.; Bengio, Y.; David, J.P. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*; Curran Associates: New York, NY, USA, 2015; pp. 3123–3131.
23. Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Binarized neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates: New York, NY, USA, 2016; pp. 4107–4115.
24. Aly, M. Survey on multiclass classification methods. *Neural Netw.* **2005**, *19*, 1–9.

25. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 499–515.
26. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. In Proceedings of the 2016 International Conference on Machine Learning (ICML 2016), New York, NY, USA, 19–24 June 2016; Volume 2, p. 7.
27. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*; Curran Associates: New York, NY, USA, 2013; pp. 2292–2300.
28. Rubner, Y.; Tomasi, C.; Guibas, L.J. A metric for distributions with applications to image databases. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), Bombay, India, 7 January 1998; pp. 59–66.
29. Nanni, L.; Lumini, A.; Brahnam, S. Survey on LBP based texture descriptors for image classification. *Expert Syst. Appl.* **2012**, *39*, 3634–3641. [CrossRef]
30. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
31. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5265–5274.
32. He, C.; He, B.; Liu, X.; Kang, C.; Liao, M. Statistics Learning Network Based on the Quadratic Form for SAR Image Classification. *Remote Sens.* **2019**, *11*, 282. [CrossRef]
33. Heikkila, M.; Pietikainen, M. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 657–662. [CrossRef]
34. Zhao, G.; Pietikainen, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 915–928. [CrossRef]
35. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
36. Maaten, L.V.D.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
37. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
38. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
39. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; ACM: New York, NY, USA, 2010; pp. 270–279.
40. Sheng, G.; Yang, W.; Xu, T.; Sun, H. High-resolution satellite scene classification using a sparse coding based multiple feature combination. *Int.l J. Remote Sens.* **2012**, *33*, 2395–2412. [CrossRef]
41. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1155–1167. [CrossRef]
42. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 675–678.
43. Sun, H.; Li, S.; Zheng, X.; Lu, X. Remote Sensing Scene Classification by Gated Bidirectional Network. *IEEE Trans. Geosci. Remote Sens.* **2019**. [CrossRef]
44. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.