The Impact of Non-uniform Label Noise on the Classification of Hyperspectral Images

Meizhu Li*, Shaoguang Huang and Aleksandra Pižurica
Department of Telecommunications and Information Processing, TELIN-GAIM, Ghent University, Belgium
Ghent, Belgium
{meizhu.li, shaoguang.huang, aleksandra.pizurica}@ugent.be

***Abstract*—Erroneous labels affect the learning models in supervised classification, deteriorate the classification performance and hinder thereby subsequent tasks. These erroneous labels are referred to as label noise. The influence of label noise on the classification performance has been so far mainly studied using simulations with a uniform distribution of noisy labels accross the classes. In this paper, we propose a new label noise simulation approach for hyperspectral images (HSIs) where similar classes have a higher chance to be mixed up. Under such a realistic label noise simulation model, we compare the behaviour of different types of classifiers for HSI in remote sensing, including traditional machine learning and deep learning approaches. Our analysis reveals which levels of label noise are acceptable for a given tolerance in the classification accuracy and how robust are different learning models in this respect.**

***Index Terms*—Robust classification, hyperspectral images, label noise**

I. Introduction

Hyperspectral images (HSIs) have been extensively used in numerous applications in various domains, including geosciences [1], agriculture [2], defense and security [3] and environment monitoring [4]. Image classification, which assigns labels to each pixel, plays an essential role in the automatic analysis and interpretation of HSIs [5].

The performance of supervised classification models depends greatly on the amount of training data. Most supervised classification methods are designed under the assumption that the training data does not contain erroneous labels. However, imprecise labels are inevitable in practice as labeling is often labor intensive and involves a lot of manual work [6], [7]. Resent work [8] identified label errors in the test sets of 10 of the most commonly-used computer vision, natural language and audio data sets. They estimated an average of 3.4% errors across the 10 data sets, which could cause problems in artificial intelligence systems that use these data sets. The erroneous labels affect adversely model training, resulting in a degraded classification performance. We refer to the erroneous labels as label noise.

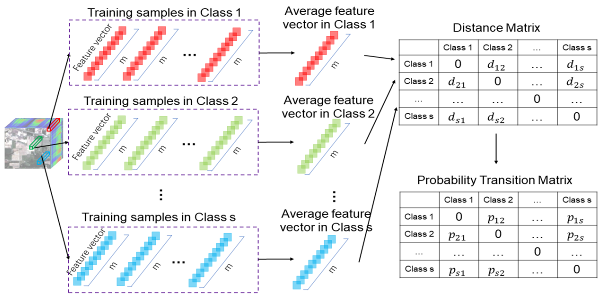
Most of the works that studied in the influence of label noise on the image classification performance consider the uniform label noise, with which the erroneous labels are chosen uniformly at random [9]–[11]. This results in an equal probability of labelling a data point to other incorrect classes. However, mutually similar classes are in practice more easily mixed up with each other in the labelling process than dissimilar classes. Thus, the probability of wrongly assigning a data point to other classes is non-uniform. In order to study the effect of label noise on classification performance of classifiers in a more practical way, we here propose a new label noise simulation method, which is based on the spectral feature distances. The proposed simulation method quantifies the distances between spectral features of classes, and then constructs the probability transition matrix according to these distances. This probability transition matrix is used to simulate the label noise for each class with a corresponding probability distribution, which admits that similar classes have a higher probability to be mixed up with each other.

Under the proposed label noise simulation approach, we study thoroughly the behaviour of several representative supervised classification approaches in the scenario where different levels of label noise are present in the training data. We characterize statistically the effect of errors in labels on the spectral signatures and compare them to those spectral signatures under uniform label noise. The empirical results show how erroneous labels affect the spectral signatures and deteriorate the classification performance. We also analyze the classifiers' tolerance to label noise given an acceptable overall accuracy degradation with the proposed simulation approach.

II. Proposed label noise simulation approach

Most work relating to label noise generate the erroneous labels uniformly at random. Although it is the easiest way to simulate label noise, it does not reflect well the situation in practice. Mutually similar classes have a higher chance to be mixed during the labelling of pixels. Therefore, we here propose a new label noise simulation approach, which takes these considerations into account.

We define the level of label noise ρ as the proportion of training samples that have wrong labels. We denote by $\mathbf{x} = (x_1, \cdots, x_m)$ a training sample and $\mathbf{y} = (y_1, \cdots, y_m)$ a test sample, where x_i and y_i are the corresponding i-th features. Both of these are vectors of pixel values in an HSI at a given spatial position in m spectral bands. Let C denote the class variable that is assigned to these samples and that takes values c in a finite set $\mathcal{C}$, with cardinality s. Further on, let $\{\mathbf{x}_i^c\}_{i=1}^l$ be a set of feature vectors with label c in an HSI for training, where $c \in \mathcal{C}$, l is the number of feature vectors that are labelled as class c. The average feature vector in class c, denoted by $\overline{\mathbf{x}}_c$, is obtained by computing the average


Training samples in Class 1
Feature vector
m
…
Average feature vector in Class 1
Training samples in Class 2
Average feature vector in Class 2
⋮
Training samples in Class s
Average feature vector in Class s
Distance Matrix
Probability Transition Matrix

	Class 1	Class 2	…	Class s
Class 1	0	d_{12}	…	d_{1s}
Class 2	d_{21}	0	…	d_{2s}
…	…	…	0	…
Class s	d_{s1}	d_{s2}	…	0

	Class 1	Class 2	…	Class s
Class 1	0	p_{12}	…	p_{1s}
Class 2	p_{21}	0	…	p_{2s}
…	…	…	0	…
Class s	p_{s1}	p_{s2}	…	0

Fig. 1. The framework of the proposed label noise simulation approach.

of all samples belonging to class c, i.e., $\mathbf{x}_i^c$. The generation of erroneous labels is governed by the probability transition matrix.

To this end, we define the distance matrix D_t as:

$$D_t = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1s} \\ d_{21} & d_{22} & \dots & d_{2s} \\ \dots & \dots & \dots & \dots \\ d_{s1} & d_{s2} & \dots & d_{ss} \end{bmatrix} = \begin{bmatrix} 0 & d_{12} & \dots & d_{1s} \\ d_{21} & 0 & \dots & d_{2s} \\ \dots & \dots & 0 & \dots \\ d_{s1} & d_{s2} & \dots & 0 \end{bmatrix}, \quad (1)$$

where d_{ij} is defined as the Euclidean Distance between the average feature vectors $\overline{\mathbf{x}}_{c_i}$ and $\overline{\mathbf{x}}_{c_j}$:

$$d_{ij} = \sqrt{\sum\nolimits_m (\overline{x}_{c_i m} - \overline{x}_{c_j m})^2}. \quad (2)$$

The probability transition matrix P_t has the general form

$$P_t = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1s} \\ p_{21} & p_{22} & \dots & p_{2s} \\ \dots & \dots & \dots & \dots \\ p_{s1} & p_{s2} & \dots & p_{ss} \end{bmatrix} = \begin{bmatrix} 0 & p_{12} & \dots & p_{1s} \\ p_{21} & 0 & \dots & p_{2s} \\ \dots & \dots & 0 & \dots \\ p_{s1} & p_{s2} & \dots & 0 \end{bmatrix}, \quad (3)$$

and we define its entries in terms of the corresponding entries in the distance matrix:

$$p_{ij} = e^{-d_{ij}} / \sum_{k=1, k \neq i}^{m} e^{-d_{ik}}. \quad (4)$$

The sum of the probabilities in each row of P_t is 1. The smaller distance between two classes, the larger probability for the transition from one class to the other. Note that P_t is not symmetric, the transition is only available through the probabilities in the same row of P_t. Fig. 1 shows an illustration of the generation of probability transition matrix. The proposed label noise simulation implements a practical scenario that similar classes have higher chance to be mixed up.

III. Spectral Signature Analysis with Different Types of Label Noise

We now move on to analyze the effect of label noise on the estimation of spectral signatures of landcover classes. In particular, the erroneous labels are chosen in two ways: uniformly at random and using our proposed label noise simulation approach. These average spectral signatures are obtained by taking the mean value of spectral intensities for each class. We randomly select 50% samples as the training samples and then select at random a given portion ρ of the total training samples (from various classes) and flip each of them to one of the remaining classes by the two ways.

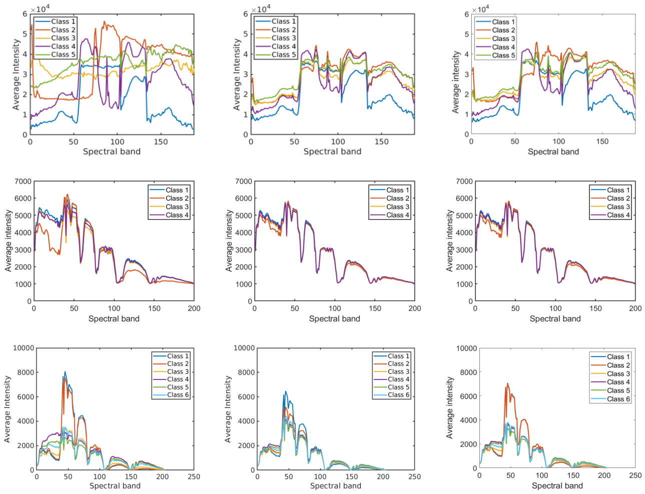


Fig. 2. Average spectral signatures for the *HYDICE Urban* data set (top three), *Indian Pines* data set (middle three) and the *Salinas Scene* data set (bottom three) with $\rho = 0$ (left column) and with $\rho = 0.5$ obtaining noisy labels by uniform transition (middle column) and by probability distribution transition (right column).

A. Data sets

We conduct our experiments on three real HSI data sets: *HYDICE Urban*, *Indian Pines* and *Salinas Scene*. The *HYDICE Urban* data set was captured by the HYDICE sensor with size $200 \times 200 \times 188$. There are five classes in this data set: trees, concrete, soil, grass and asphalt. The *Indian Pines* data set has a spatial size of 85×70 and contains 200 spectral bands, including four classes: corn-notill, grass-trees, soybean-notill and soybean-minitill. The size of the *Salinas Scene* data is $100 \times 80 \times 224$, including six classes: brocoli-green-weeds-1, brocoli-gree-weeds-2, grapes-untrained, lettuce-romaine-4wk, lettuce-romaine-5wk and lettuce-romaine-6wk.

B. Spectral signatures analysis with noisy labels

Fig. 2 illustrates the effect of label noise on the average spectral signatures in the three data sets. In *HYDICE Urban* data set, the spectral signatures of different classes are rather different from each other without label noise. In the presence of label noise, they wrongly appear to be more similar to each other with both methods of label noise simulation. The spectral signatures with our non-uniform label noise show slightly different from those with uniform label noise. This is because the five classes in this data set are rather different from each other, causing the distances between features in different classes not very distinguishable. In *Indian Pines* data set, the spectral signatures of all the four classes are quite similar without label noise. In the presence of label noise under two simulation methods, the spectral signatures of the four class are quite similar, and falsely appear to be more similar to each other.

In *Salinas Scene* data set, without label noise, the spectral signatures of Class 1 and Class 2 are quite similar and the other four classes are similar. With the uniform label noise, the spectral signatures of all classes wrongly appear to be more

TABLE I
PROBABILITY TRANSITION MATRIX FOR *Salinas Scene* DATA SET.

	Class 1	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Class 6**
Class 1	0	0.6043	0.1197	0.0843	0.0779	0.1140
Class 2	0.5611	0	0.1319	0.0938	0.0865	0.1267
Class 3	0.0816	0.0969	0	0.2115	0.2651	0.3449
Class 4	0.0592	0.0710	0.2180	0	0.3435	0.3065
Class 5	0.0504	0.0603	0.2516	0.3181	0	0.3196
Class 6	0.0676	0.0809	0.2999	0.2587	0.2928	0

similar to each other. However, with our proposed simulation approach, the first two classes tend to mix up with each other, and the other four classes appear to be more similar to each other. The probability transition matrix of this case is shown in Table I. In this table, the probabilities for the classes with similar spectral signatures are much higher than the others. This example shows clearly that the proposed label noise simulation method achieves the goal successfully: similar classes have higher chance to be mixed with each other. The analysis also indicates that erroneous labels lead to model uncertainties, which will in turn affect the classification performance. In the following section, we will study the performance of the representative classifiers with the proposed label noise simulation approach and explore which level of label noise can be tolerated depending on the acceptable drop in the classification accuracy.

IV. REPRESENTATIVE CLASSIFICATION METHODS

Here, we review briefly the classifiers that we use for the analysis in this paper.

A. Naive Bayes classifiers (NBCs)

NBCs are simple Bayesian classifiers. For any given feature vector $\mathbf{x}$, an NBC returns the Maximum a Posteriori (MAP) estimate of the class variable C, assuming the conditional independence $P(\mathbf{x}|c) = \prod_{i=1}^{m} P(x_i|c)$. The estimated class is thus:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} P(c|\mathbf{x}) = \arg\max_{c \in \mathcal{C}} P(c) \prod_{i=1}^{m} P(x_i|c). \quad (5)$$

B. K-nearest-neighbor classifier (k-NN)

In k-NN algorithm, the test sample $\mathbf{y}$ is classified by the majority voting of its k nearest neighbors, which are often measured by the Euclidean distance as follows:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{m} (x_m - y_m)^2}. \quad (6)$$

Let $\mathcal{N}_y$ be the set of k nearest neighbors of $\mathbf{y}$ according to Equation (6). The test sample $\mathbf{y}$ is assigned to the class that is most common among $\mathcal{N}_y$.

C. Support vector machine (SVM)

SVM learns a separating hyperplane from a given set of training data with an optimal decision boundary to each class [12], and categorizes new data points by the learned hyperplane. Let $K(\mathbf{x}_i, \mathbf{x}_j)$ be a kernel function which defines an inner product in the feature space. The decision function implemented by SVM can be written as:

$$f(\mathbf{y}) = sgn(\sum_{i=1}^{N} c_i \alpha_i K(\mathbf{y}, \mathbf{x}_i) + b), \quad (7)$$

where c_i is the corresponding label of sample $\mathbf{x}_i$, b is a real number and the coefficients α_i are obtained by solving the convex Quadratic Programming (QP) problem [13].

D. Sparse representation classification (SRC)

SRC identifies the label of test data in two steps: sparse representation and classification. Sparse representation represents a test data $\mathbf{y}$ by a linear combination of a few atoms from a dictionary $\mathbf{D} \in \mathbb{R}^{m \times d}$, which in SRC is constructed specially by the training samples $\{\mathbf{x}_i\}_{i=1}^{d}$. We denote by $\mathbf{D}_i \in \mathbb{R}^{m \times d_i}$ the i-th subdictionary in $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_c]$ where each column of $\mathbf{D}_i$ is a training sample of i-th class. The resulting sparse coefficients vector $\boldsymbol{\alpha} \in \mathbb{R}^{d}$ of $\mathbf{y}$ can be obtained by solving the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \qquad s.t. \qquad \|\boldsymbol{\alpha}\|_0 \le K, \quad (8)$$

where $\|\boldsymbol{\alpha}\|_0$ denotes the number of non-zero elements in $\boldsymbol{\alpha}$ and K is the sparsity level, i.e., the largest number of atoms in dictionary $\mathbf{D}$ needed to represent any input sample $\mathbf{y}$. The optimization problem in Eq. (8) is typically solved with a greedy algorithm, such as Orthogonal Matching Pursuit (OMP) [14]. Then, the class of the test sample is identified by calculating the class-specific residuals r_i [15]:

$$\begin{aligned} class(\mathbf{y}) &= \arg\min_{i=1,2,\dots,C} r_i(\mathbf{y}) \\ &= \arg\min_{i=1,2,\dots,C} \|\mathbf{y} - \mathbf{D}_i \boldsymbol{\alpha}_i\|_2, \end{aligned} \quad (9)$$

where $\boldsymbol{\alpha}_i$ are the sparse coefficients associated with class i.

E. SRC-based classifier with spectral-spatial features

We also consider a representative of SRC-based method where spatial information is included, and in particular we will use in our analysis the method of [16], called SJSRC, which employs super-pixel segmentation and encodes jointly all the pixels within one super-pixel. It assumes that similar pixels in local regions, which are defined by super-pixel segmentation, can be represented by a few common atoms in $\mathbf{D}$. This results in a row sparsity pattern on the coefficients matrix of the pixels within the same super-pixel. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ represent a super-pixel composed of n pixels in m spectral bands and $\mathbf{A} \in \mathbb{R}^{d \times n}$ the corresponding coefficients matrix. SJSRC solves the following problem

$$\arg\min_{\mathbf{A}} \ \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 \ \ s.t. \ \|\mathbf{A}\|_{row,0} \le K_0, \quad (10)$$

where $\|\mathbf{A}\|_{row,0}$ denotes the number of non-zero rows of $\mathbf{A}$ and K_0 is the row-sparsity level. After finding $\mathbf{A}$, the class for the whole super-pixel $\mathbf{X}$ is decided as:

$$class(\mathbf{X}) = \arg\min_{i=1,2,\dots,C} \|\mathbf{X} - \mathbf{D}_i \mathbf{A}_i\|_F, \quad (11)$$

where $\mathbf{A}_i$ is the sub-matrix of $\mathbf{A}$ corresponding to class i.

F. Deep learning based spectral-spatial classifier

Deep learning methods have been increasingly used in HSI classification [17]–[19]. As a representative of these methods, we select the SSUN [20], SSRN [21] and CBSP algorithm [22]. All the three algorithms combine spectral and spatial feature extraction.

1) Spectral-spatial unified network (SSUN): The SSUN algorithm [20] integrates the spectral feature extraction, spatial feature extraction and classifier training into a unified neural network. It incorporates long short-term memory (LSTM) [23] network for band grouping and spectral feature extraction and the multiscale CNN (NSCNN) for spatial feature extraction.

2) Spectral-spatial residual network (SSRN): The SSRN algorithm [21] is an end-to-end spectral-spatial residual network that takes raw 3-D cubes as input data for hyperspectral image classification. In SSRN, the spectral and spatial residual blocks consecutively learn discriminative features from abundant spectral signatures and spatial contexts in hyperspectral imagery. More details can be found in [21].

3) Convolution based spectral partitioning architecture (CBSP): The CBSP algorithm [22] aims to develop a deep learning architecture using 3-D convolutional neural networks with spectral partitioning to extract features. It first performs a spatial transformation via 2-D convolution. The transformed image is partitioned on the spectral level and split into segments for efficient processing. 3-D convolution is then applied to each segment. Finally, convoluted segments are concatenated and fed to two fully-connected layers with dropout as regularization. The detailed description of CBSP can be found in [22].

V. Comparisons of Classification Performance with Noisy Labels

We conduct experiments on two real HSI data sets: *HYDICE Urban* and *Indian Pines*.

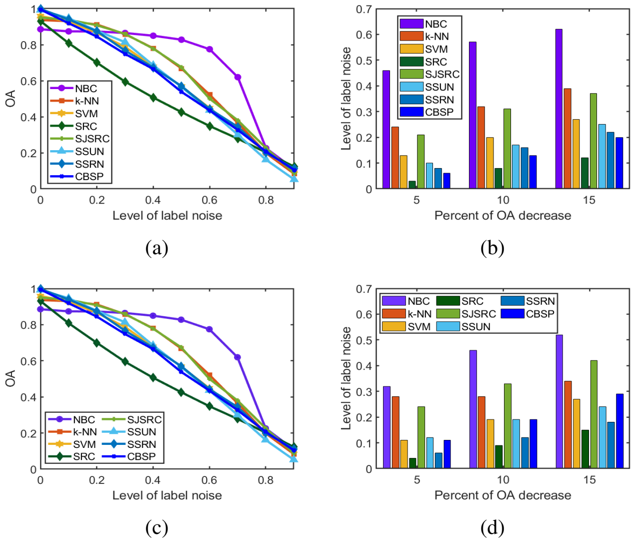
A. Experimental setting

The effect of erroneous labels is studied by evaluating the performance of the eight representative classification algorithms described in Section IV. Four of these (NBC, k-NN, SVM and SRC) are based on spectral features alone, and the remaining four (SJSRC, SSUN, SSRN and CBSP) make use of both spectral and spatial features.

In the following experiments, 10 percent of samples are randomly selected for training and the rest are for testing. The reported results are averaged values over 10 runs with different training samples. We evaluate the classification performance by overall accuracy (OA), which is the ratio between correctly classified testing samples and the total number of testing samples.

B. Experiments on HYDICE Urban

Fig. 3 (a) shows the overall accuracy of the eight algorithms on *HYDICE Urban* with ρ ranging from 0 to 0.9. When there is low-to-moderate amounts of label noise, the four spectral-spatial methods show much better performance than the four spectral-based methods. When there is no label noise ($\rho = 0$) the deep learning method SSRN yields the best OA, while the naive Bayesian classifier (NBC) is inferior to all other methods. This can partly be attributed to the fact that this particular NBC makes use of only spectral features while the other better performing methods (SJSRC, SSUN, SSRN and CBSP) incorporate spatial and spectral features. With the increasing levels of label noise, spectral-based algorithms k-NN, SVM and SRC show similar behaviour, but SRC performs worse than the other two and shows approximately linear decrease. The performance of NBC is the most stable, and drops sharply when ρ exceeds 0.6. The overall accuracy of spectral-spatial methods SJSRC, SSUN, SSRN and CBSP deteriorate significantly with the increasing label noise and the three deep learning methods (SSUN, SSRN and CBSP) are especially vulnerable in this respect.



(a) (b)

(c) (d)

Fig. 3. (a) Influence of label noise on OA and (b) the classifiers' tolerance to label noise in the data set *HYDICE Urban* with respect to OA drop. (c) Influence of label noise on OA and (d) the classifiers' tolerance to label noise in the data set *Indian Pines* with respect to OA drop.

Fig. 3 (b) shows the maximum level of label noise that a classifier can tolerate given a decreasing rate in the OA compared to the case without label noise ($\rho = 0$). We analyze the tolerance of the eight classification models in the cases with OA decreasing by 5%, 10% and 15% compared to the OA of $\rho = 0$. We assume that the OAs between any two successive ρ (in steps of 0.1) decrease linearly as in Fig. 3 (a). NBC shows the highest tolerance to label noise, which means that if 5% decrease in OA can be tolerated, NBC allows 40% of erroneous labels. The three deep learning approaches (SSUN, SSRN and CBSP) exhibit very low tolerance to label noise, although they make use of both spectral and spatial features. The sparse coding approach based on spectral features alone (SRC) also shows low tolerance to label noise, but its version with spatial information (SJSRC) is much more robust, both compared to the basic SRC and to the deep learning methods.

C. Experiments on Indian Pines

Fig. 3 (c) shows the performance of the eight algorithms on *Indian Pines*. Spectral-based algorithms k-NN, SVM and SRC show similar behaviour as in *HYDICE Urban* data set. NBC is again the most stable method. Its performance drops significantly when ρ exceeds 0.6. The spectral-spatial methods (SJSRC, SSUN, SSRN and CBSP) also behave similarly as in *HYDICE Urban* and their overall accuracy deteriorates significantly with the increasing label noise.

Results in Fig. 3 (d) show similar trends as in *HYDICE Urban* data set in terms of the tolerance to label noise. NBC shows again the highest tolerance to label noise in the three cases. The sparse coding approach based on spectral alone (SRC) shows very low tolerance to label noise, but the version with spatial information (SJSRC) is much more robust to label noise, both compared to basic SRC and to the deep learning methods SSUN, SSRN and CBSP.

VI. Conclusion

We proposed a new label noise simulation approach, and employed it in the evaluation of classifiers for HSI images in remote sensing. The proposed approach admits the fact that similar classes have a higher chance to be mixed up with each other. With the proposed label noise model, we analysed the effect of erroneous data labeling on the estimated spectral signatures of different classes. We also compared the performance of representative supervised classification approaches when erroneous labels simulated by our proposed method are present. The results reveal that Bayesian classifiers, even under the simplest naive Bayesian model (NBC) are more robust to the label noise simulated by the proposed method than methods based on support vector machines (SVM), sparse coding and deep learning. Deep learning approaches exhibited in all our experiments the highest vulnerability to label noise. This agrees with recent studies that show susceptibility of deep learning to various other perturbations, such as noise in the data and adversarial attacks. The k-NN method also demonstrated very robust performance, which can be attributed to its majority voting strategy. Our analysis shows also clearly the importance of using spatial context not only to improve the classification accuracy in ideal settings but also to improve the robustness to label noise. Sparse coding methods that make use of both spectral and spatial information showed excellent performance and can be considered as a good choice of a classifier, which enables a high classification accuracy and a robust performance to label noise.

Acknowledgment

This work was supported by the China Scholarship Council (CSC), by the Fonds voor Wetenschappelijk Onderzoek (FWO) project under Grant G.OA26.17N and the Flanders AI Research Programme under grant 174B09119.

References

[1] S. J. Buckley, T. H. Kurz, J. A. Howell, and D. Schneider, “Terrestrial lidar and hyperspectral data fusion products for geological outcrop analysis,” *Comput. Geosci.*, vol. 54, pp. 249–258, 2013.

[2] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. J. Sousa, “Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry,” *Remote Sens.*, vol. 9, no. 11, p. 1110, 2017.

[3] M. T. Eismann, A. D. Stocker, and N. M. Nasrabadi, “Automated hyperspectral cueing for civilian search and rescue,” *Proc. IEEE*, vol. 97, no. 6, pp. 1031–1055, 2009.

[4] C. Wu, L. Zhang, and B. Du, “Kernel slow feature analysis for scene change detection,” *IEEE Trans. Geosci. Remote Sensing*, vol. 55, no. 4, pp. 2367–2384, 2017.

[5] S. Huang, H. Zhang, and A. Pižurica, “Semisupervised sparse subspace clustering method with a joint sparsity constraint for hyperspectral remote sensing images,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 3, pp. 989–999, 2019.

[6] J. Jiang, J. Ma, Z. Wang, C. Chen, and X. Liu, “Hyperspectral image classification in the presence of noisy labels,” *IEEE Trans. Geosci. Remote Sensing*, vol. 57, no. 2, pp. 851–865, 2018.

[7] B. Tu, C. Zhou, W. Kuang, L. Guo, and X. Ou, “Hyperspectral imagery noisy label detection by spectral angle local outlier factor,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 9, pp. 1417–1421, 2018.

[8] C. G. Northcutt, A. Athalye, and J. Mueller, “Pervasive label errors in test sets destabilize machine learning benchmarks,” *arXiv preprint arXiv:2103.14749*, 2021.

[9] X. Kang, P. Duan, X. Xiang, S. Li, and J. A. Benediktsson, “Detection and correction of mislabeled training samples for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 5673–5686, 2018.

[10] M. Li, S. Huang, and A. Pižurica, “Robust dynamic classifier selection for remote sensing image classification,” in *Proc. ICSIP*. IEEE, 2019, pp. 101–105.

[11] M. Li, S. Huang, J. De Bock, G. De Cooman, and A. Pižurica, “A robust dynamic classifier selection approach for hyperspectral images with imprecise label information,” *Sensors*, vol. 20, no. 18, p. 5262, 2020.

[12] V. N. Vapnik, “An overview of statistical learning theory,” *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, 1999.

[13] S. Hua and Z. Sun, “Support vector machine approach for protein subcellular localization prediction,” *Bioinformatics*, vol. 17, no. 8, pp. 721–728, 2001.

[14] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[15] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2008.

[16] S. Huang, H. Zhang, and A. Pižurica, “A robust sparse representation model for hyperspectral image classification,” *Sensors*, vol. 17, no. 9, p. 2087, 2017.

[17] X. Li, M. Ding, and A. Pižurica, “Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sensing*, vol. 58, no. 4, pp. 2615–2629, 2019.

[18] Y. Chen, K. Zhu, L. Zhu, X. He, P. Ghamisi, and J. A. Benediktsson, “Automatic design of convolutional neural network for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 7048–7066, 2019.

[19] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, “Deep learning ensemble for hyperspectral image classification,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 6, pp. 1882–1897, 2019.

[20] Y. Xu, L. Zhang, B. Du, and F. Zhang, “Spectral–spatial unified networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sensing*, vol. 56, no. 10, pp. 5893–5909, 2018.

[21] Z. Zhong, J. Li, Z. Luo, and M. Chapman, “Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2017.

[22] R. S. Chu, H.-C. Ng, X. Wang, and W. Luk, “Convolution based spectral partitioning architecture for hyperspectral image classification,” in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 3962–3965.

[23] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.