A Novel Robust Classification Method for Ground-Based Clouds

Aihua Yu, Ming Tang, Gang Li, Beiping Hou*, Zhongwei Xuan, Bihong Zhu and Tianliang Chen

School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China; yuaihua@zust.edu.cn (A.Y.); 222007855035@zust.edu.cn (M.T.); ieligang@zjut.edu.cn (G.L.); 1180350017@zust.edu.cn (Z.X.); 212007802004@zust.edu.cn (B.Z.); 222007855005@zust.edu.cn (T.C.)
* Correspondence: bphou@zust.edu.cn

Abstract: Though the traditional convolutional neural network has a high recognition rate in cloud classification, it has poor robustness in cloud classification with occlusion. In this paper, we propose a novel scheme for cloud classification, in which the convolutional neural networks are used for feature extraction and a weighted sparse representation coding is adopted for classification. Three such algorithms are proposed. Experiments are carried out using the multimodal ground-based cloud dataset and the results show that in the case of occlusion, the accuracy of the proposed methods can be much improved over the traditional convolutional neural network-based algorithms.

Keywords: cloud classification; fusion convolutional neural network; sparse representation; robustness

1. Introduction

Recently, with the expansion of big data combined with the improvement in algorithms and the exponential growth in computing power, machine learning has become a research highlight. Machine learning, especially deep learning, is widely used in computer vision, speech recognition, natural language processing, data mining and meteorological information processing.

In the early days, Buch et al. extracted features from cloud images, including texture measurements, location information and pixel brightness, then used binary decision trees to divide them into four kinds of clouds: altocumulus, cirrus, cumulus and stratiform [1]. Singh et al. proposed a procedure to test the texture feature used for the automatic training of a cloud classifier, in which five feature extraction methods were examined, namely autocorrelation, co-occurrence matrix, edge frequency, Law’s feature and primitive length. These tests can help us better understand the advantages and disadvantages of different feature extraction methods and classification techniques [2]. Heinle et al. made use of k-nearest neighbor classifier to distinguish 7 kinds of cloud images by extracting color and texture features from cloud images [3]. Neto et al. used the multivariate color space features to classify clouds and sky pixels using the statistical features of Euclidean geometric distance [4]. Liu et al. proposed a multiple random projection algorithm to obtain textons and discriminative features [5]. However, the accuracy of these cloud classification methods based on manual feature extraction is far from expectation, and we must manually design features carefully for each classification.

Cloud classification plays a key role in a wide range of applications such as solar power generation, weather forecasting [6,7], deep space climate observatory mission [8], rainfall estimation [9] and optical remote sensing [10]. However, cloud classification is a challenging task. Different clouds show different meteorological information features such as temperature, humidity, pressure, wind speed and color distribution. To achieve cloud classification, it is necessary to accurately estimate its characteristics from the shape, thickness, degrees of sparseness and other cloud features. However, the traditional cloud
classification [11–13] is in the charge of professional observers. In addition to consuming a lot of labor, there will also be human errors, and it is more difficult to classify the clouds with a satisfactory accuracy when there exist interferences such as sunlight, fog and many others.

Recently convolution neural networks (CNN) have been widely used in image classification and recognition applications. Such networks have many advantages. First, they do not need to extract features manually. In fact, the CNNs automatically extract features through many training pictures and yield a very good performance. Secondly, they can process many data well. Moreover, they can also reflect a strong image discrimination under a large amount of data, so they are a good choice for dealing with the ever-changing clouds. Yang et al. proposed CNN model, known as LeNet-5, yielding an accuracy of handwritten digit recognition as high as 99% [14]. Krizhevsky proposed a deep neural network model, denoted as Alexnet, which is relatively smaller than the LeNet in size, and changed the sigmoid activation function to a simpler ReLU activation function. Such a model makes the training easier under different parameter initialization methods and can reduce overfitting by dropout layers [15]. In [16], an efficient network, called Inception-v3, was proposed, but the deeper network is more difficult to train. He et al. proposed a residual learning framework to simplify the training of deeper networks and the network, denoted as ResNet, takes the residual blocks as the basic blocks and can prevent gradient from disappearing [17]. Liu et al. used multimodal information and visual features to achieve cloud classification through support vector machine [18], where a fusion of convolutional neural networks, including visual sub-network and multimodal sub-network, is proposed. After extracting visual and multimodal features, a weighted integration was used for cloud classification [19]. Although the recognition accuracy of these CNN-based cloud classification methods is very high, they have a common weak point—lack of robustness to perturbation.

Compressed sensing (CS) [20,21] is a recently developed technology. It is based on the concept of sparse representation of signals [22]. Let \( x \in \mathbb{R}^{n \times 1} \) be a signal vector. It is called \( \|a\|_0 \)-sparse in a matrix \( D \in \mathbb{R}^{n \times m} \) if \( x = Da \), where \( D \) is called the dictionary and \( \|a\|_0 \) denotes the number of non-zero entries in the coefficient vector \( a \in \mathbb{R}^{m \times 1} \). Sparse representation has been used in the fields of image processing [23] and face recognition [24,25]. A weighted sparse representation was proposed in [26] and simulations show that it is more robust to occlusion, destruction and other interferences. Gan et al. [27] proposed a dual norm bounded sparse coding, which classifies cloud images by extracting nine-dimensional features, including five color features and four statistical features, but they did not execute the robustness analysis.

In this paper, we investigate classification of ground-based clouds that are observed by ground-based cameras. The main contributions of this paper are as follows:

- A novel structure for robust cloud classification is proposed, in which the features are extracted by convolution neural networks and the classification is executed using the weighted sparse representation;
- A two-channel-neural network is proposed for extracting features of the ground-based clouds.

Experimental results show that our proposed method yields a satisfactory performance for classifying ground-based clouds. In particular, the robustness of the proposed two-channel-neural network-based classifier is better than the one with a single convolutional neural network in the case of less occlusion.

The rest of this paper is organized as follows: in Section 2, we briefly introduce some related work, we describe a novel robust cloud classification structure and specific algorithm flow in Section 3, Section 4 presents experimental analysis. To end this paper, some remarking points are given in Section 5.
2. Related Work

Now, we introduce some existing results that will be used for our proposed robust classification algorithm for ground-based clouds.

2.1. ResNet Model

It has been observed that after adding too many layers in a CNN, the training error tends to rise. Though the numerical stability brought by the batch normalization makes it easier to train deeper networks, this problem still exists. The ResNet is proposed for addressing this problem. The residual block is the basic block of the ResNet. In a residual block, the input can propagate forward faster through cross layer data lines, and the occurrence of gradient disappearance can be reduced. Two additional major benefits of rectified linear units (ReLU) are sparsity and a reduced likelihood of vanishing gradient [17], see Figure 1.

![Figure 1. Flowchart of a residual block in the ResNet, where x is the input data vector and y is the output vector, while ReLU is rectified linear units: ReLU(z) = Max (0, z).](image)

Table 1 shows the parameters involved in the ResNet-50 model.

<table>
<thead>
<tr>
<th>Type</th>
<th>Patch Size $(k \times k \times c)/$Stride(s)</th>
<th>Input Size $(m \times n \times c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv</td>
<td>7 × 7 × 64/2</td>
<td>224 × 224 × 3</td>
</tr>
<tr>
<td>max-pooling</td>
<td>3 × 3 × 64/2</td>
<td>112 × 112 × 64</td>
</tr>
<tr>
<td>3 × residual block</td>
<td></td>
<td>56 × 56 × 64</td>
</tr>
<tr>
<td>4 × residual block</td>
<td></td>
<td>28 × 28 × 256</td>
</tr>
<tr>
<td>6 × residual block</td>
<td></td>
<td>14 × 14 × 512</td>
</tr>
<tr>
<td>3 × residual block</td>
<td></td>
<td>7 × 7 × 1024</td>
</tr>
<tr>
<td>average pooling</td>
<td>7 × 7 × 2048</td>
<td>7 × 7 × 2048</td>
</tr>
<tr>
<td>softmax</td>
<td>1 × 1 × 2048</td>
<td>1 × 1 × 2048</td>
</tr>
</tbody>
</table>

2.2. Inception Model

The efficient convolutional neural network based on Inception model, proposed in [16], consists of a set of basic convolution blocks, i.e., the Inception blocks. The naive version of such a network is showed in Figure 2.

![Figure 2. Block-diagram of the Inception CNN—a naive version.](image)
As shown in Figure 2, the $k \times k$ convolutions, for $k = 1, 3$ and $5$ ($k$ is 2D of convolution kernel sizes), are employed to extract information in different spatial sizes. However, this structure increases in the number of outputs from stage to stage [28].

The Inception module with dimension reduction is shown in Figure 3, where the $1 \times 1$ convolutions are used to compute reductions before the expensive $3 \times 3$ and $5 \times 5$ convolutions. Besides being used for dimension reduction, they are also used for the rectified linear activation, which makes them dual-purpose.

![Figure 3. Block-diagram of the Inception module with dimension reduction.](image)

In general, an Inception network is a network consisting of modules of the above type, stacked upon each other with occasional max-pooling layers with stride 2 to halve the resolution of the grid and therefore, reducing the computational complexity and improving performance [28].

Table 2 shows the parameters involved in the Inception-v3 model.

<table>
<thead>
<tr>
<th>Type</th>
<th>Patch Size $(k \times k \times c)/$Stride(s)</th>
<th>Input Size $(m \times n \times c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv</td>
<td>$3 \times 3 \times 32$/2</td>
<td>$299 \times 299 \times 3$</td>
</tr>
<tr>
<td>conv</td>
<td>$3 \times 3 \times 32$/1</td>
<td>$149 \times 149 \times 32$</td>
</tr>
<tr>
<td>conv padded</td>
<td>$3 \times 3 \times 64$/1</td>
<td>$147 \times 147 \times 32$</td>
</tr>
<tr>
<td>pool</td>
<td>$3 \times 3 \times 64$/2</td>
<td>$147 \times 147 \times 64$</td>
</tr>
<tr>
<td>conv</td>
<td>$3 \times 3 \times 80$/1</td>
<td>$73 \times 73 \times 64$</td>
</tr>
<tr>
<td>conv</td>
<td>$3 \times 3 \times 192$/2</td>
<td>$71 \times 71 \times 80$</td>
</tr>
<tr>
<td>conv</td>
<td>$3 \times 3 \times 288$/1</td>
<td>$35 \times 35 \times 192$</td>
</tr>
<tr>
<td>3 × Inception</td>
<td></td>
<td>$35 \times 35 \times 288$</td>
</tr>
<tr>
<td>5 × Inception</td>
<td></td>
<td>$17 \times 17 \times 768$</td>
</tr>
<tr>
<td>2 × Inception</td>
<td></td>
<td>$8 \times 8 \times 1280$</td>
</tr>
<tr>
<td>average pooling</td>
<td>$8 \times 8 \times 2048$</td>
<td>$8 \times 8 \times 2048$</td>
</tr>
<tr>
<td>softmax</td>
<td></td>
<td>$1 \times 1 \times 2048$</td>
</tr>
</tbody>
</table>

2.3. Robust Sparse Coding-Based Classification

The traditional sparse coding problem is formulated as

$$\tilde{\alpha} = \arg \min_{\alpha} \| y - Da \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_1 \leq \kappa,$$

where $\kappa$ is the prescribed sparsity level. This coding model actually assumes that the coding residuals $e = y - Da$ follow Gaussian or Laplacian distribution.

When applied for image recognition, each image is represented by a (feature) vector $x \in \mathbb{R}^{n \times 1}$ and the columns of the dictionary $D \in \mathbb{R}^{n \times m}$ are formed with the $m$ training images and it is assumed that the test images $y$ can be well represented as a linear combination of training samples with a few terms involved, i.e., not more than $\kappa$ as (1) suggested.
In practice, the residual $e = y - Da$ may be far away from Gaussian distribution or Laplacian distribution, especially when the images contain occlusion or damage, so the traditional sparse coding model has a poor robustness. To improve the robustness, we must make a good use of those pixels which are not perturbed and neglect those greatly distorted in the sparse coding-based classification (1). This can be done by assigning each pixel as weighting factor, leading to the following robust sparse coding problem:

$$\hat{\alpha} = \arg \min_{\alpha} \| W(y - Da) \|_2^2 \quad \text{s.t.} \quad \| \alpha \|_1 \leq \kappa,$$

(2)

where the diagonal matrix $W$ is selected based on the residual vector $\hat{e}$ in such a way that if $|\hat{e}(i)| < |\hat{e}(j)|$, then $W(i, i) > W(j, j)$. Here, we choose $W$ with

$$W(i, i) = \sqrt{\frac{e^{\beta(\varphi^2(i))}}{1 + e^{\beta(\varphi^2(i))}}} \forall i,$$

(3)

where the parameters $\beta, \varphi$ are determined by experiments [26]. Usually, $W$ is normalized $\text{trace}(W^2) = 1$. See the next section.

The class that image $y$ belongs to, denoted as $j(y)$, is then determined by

$$j(y) = \arg \min_j \{ |\hat{e}_j|^2 \} \quad \text{s.t.} \quad \hat{e}_j = W(y - Da(j)).$$

(4)

3. Our Proposed Methods

We propose a novel robust classification scheme for ground-based clouds. The basic idea is to use CNNs for feature extraction and sparse coding for classification.

3.1. A Two-Channel-Neural Network-Based Feature Extraction

A more sophisticated approach is depicted in Figure 4, which contains two parts. The first one is for feature extraction, which converts an image into a (feature) vector $z$. The second part is for classifying the feature vector $z$.

![Figure 4. The structure of our proposed method. In Resnet-50, Residual block uses the structure similar to Figure 1. In Inception-v3 net, Inception uses the structure of Figure 3.](image)

As each CNN performs differently for different set of conditions, an intuitive idea is to fuse the features obtained with more than one CNNs. Here, we propose a scheme, shown in Figure 4, where two CNNs are used, yielding feature vectors $y_i \in R^{n_i \times 1}$ for $i = 1, 2$ that are simply stacked as

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \in R^{(n_1 + n_2) \times 1}.$$

(5)
Let \( n_1 + n_2 = 2n \). The system of feature selection aims to convert the feature vector \( y \) of dimension \( 2n \) into a vector \( z \) of dimension \( n \). Here, we propose such a system consisting of two projections \( P_I \) and \( P_E \):

\[
y \in \mathbb{R}^{2n \times 1} \mapsto z = P_I(y) \in \mathbb{R}^{1.5n \times 1}
\]

\[
z \in \mathbb{R}^{1.5n \times 1} \mapsto \tilde{z} = P_E(z) \in \mathbb{R}^{n \times 1},
\]

where the two projections \( P_I \) and \( P_E \) are determined with the training samples \( Y = [ Y_1 \cdots Y_k \cdots Y_K ] \) and the total dictionary \( V = [ v_1 \cdots v_k \cdots v_K ] \) with \( V_k \in \mathbb{R}^{2n \times K} \). Instead of using one sample for each class, we make use of all the dictionary for robust sparse coding is formed based on the feature vectors of the training samples. Instead of using one sample for each class, we make use of all the dictionary for robust sparse coding is formed based on the feature vectors of the training samples.

- \( P_I \) — For a given \( Y_k \in \mathbb{R}^{2n \times m_k} \), denote the mean and variance vectors \( v_k, \sigma_k \in \mathbb{R}^{2n \times 1} \) with the following problem (see [20,21]).

\[
\left\{ \begin{aligned}
\bar{v}_k &\triangleq \frac{1}{m_k} \sum_{i=1}^{m_k} Y_k(:,i) \\
\bar{\sigma}_k(i) &\triangleq \sqrt{\frac{1}{m_k} \sum_{j=1}^{m_k} (Y_k(i,j) - \bar{v}_k(i))^2}
\end{aligned} \right. \quad (7)
\]

Define \( V_\ell \triangleq \frac{1}{K} \sum_{k=1}^{K} \sigma_k \), which is a vector determined by the matrix \( Y \) with (7). Let \( \{ V_\ell(j_p) \} \) with \( j_p < j_{p+1} \) for \( p = 1, 2, \cdots, 1.5n - 1 \) be the set of the \( 1.5n \) smallest entries of \( V_\ell \), then the projection \( P_I \) such that \( \tilde{z} = P_I(y) \) is given by

\[
z(p) = y(j_p), \quad \forall \quad 1 \leq p \leq 1.5n. \quad (8)
\]

Thus, \( Y_k \mapsto Z_k = P_I(Y_k) \) for \( k = 1, 2, \cdots, K \), i.e., \( Y \mapsto Z = P_I(Y) \). As understood, the projection \( P_I \) intends to keep those entries of the feature vector \( y \), which are clustered within the each of the classes.

- \( P_E \) — With \( V_\ell \in \mathbb{R}^{1.5n \times K} \) obtained, we can compute the mean and variance vectors, liked as \( \bar{v}^*, \sigma^* \in \mathbb{R}^{1.5n \times 1} \), using (7). Let \( \{ v^*(j_p) \} \) with \( j_p < j_{p+1} \) for \( p = 1, 2, \cdots, n - 1 \) be the set of the \( n \) largest entries of \( \sigma^* \), then the projection \( P_E \) such that \( z = P_E(\tilde{z}) \) is given by

\[
z(p) = \tilde{z}(j_p), \quad \forall \quad 1 \leq p \leq n. \quad (9)
\]

Unlike \( P_I \), \( P_E \) aims at enhancing the discrimination between the classes by keeping those entries of vector \( \tilde{z} \) that are of a big variance.

3.2. Robust Sparse Coding with Extended Dictionary

As assumed before, there are \( K \) different classes of clouds to be considered. The dictionary for robust sparse coding is formed based on the feature vectors of the training samples. Instead of using one sample for each class, we make use of all the \( m_k \) training pictures for the \( k \)th class. Precisely speaking, the \( k \)-sub-dictionary, denoted as \( D_k \), is given by

\[
D_k = \mathcal{P}_E(\mathcal{P}_I(Y_k))
\]

\[
k = 1, 2, \cdots, m_k
\]

and the total dictionary \( D \in \mathbb{R}^{n \times m} \) with \( m = \sum_{k=1}^{K} m_k \), named extended dictionary, is formed with \( \{ D_k \} \):

\[
D = [ D_1 \cdots D_k \cdots D_K ].
\]

The optimal sparse representation-based robust classification (2) is usually attacked with the following problem (see [20,21]).

\[
\hat{\alpha} = \arg \min_{\alpha} \| \mathcal{W}(z - D\alpha) \|^2_2 + \lambda \| \alpha \|_1,
\]

where \( \mathcal{W} \) is the weighting matrix, which is diagonal and a function of \( e = z - D\alpha \).
The problem defined by (11) is an alternative version of (2) and is very difficult to solve because \(\mathbb{W}\) is a function of \(e = z - D\alpha\). Practically, the weighting matrix \(\mathbb{W}\) is obtained with an iterative procedure. Actually, \(\mathbb{W}\) updated based on the previous estimate of \(\bar{\pi}\): let \(\bar{\pi}^{(l)}\) be the estimate of \(\bar{\pi}\) at the \(l\)th iteration of Algorithm 1 below. Denote

\[
\hat{e}^{(l)} = z - D\bar{\pi}^{(l)}
\]  

(12)

The weighting matrix \(W\) is then updated with the above \(\hat{e}^{(l)}\) via (3), i.e.,

\[
W^{(l)}(i,i) = \sqrt{\frac{e^{\beta(\varphi - |\hat{e}^{(l)}(i)|^2)}}{1 + e^{\beta(\varphi - |\hat{e}^{(l)}(i)|^2)}}, \quad \forall i
\]

(13)

and hence \(\mathbb{W}^{(l)}\) is then updated with

\[
\mathbb{W}^{(l)}(i,i) = W^{(l)}(i,i)/||W^{(l)}||_F \quad \forall i
\]

(14)

with \(||.||_F\) denoting the Frobenius norm.

With the obtained \(\bar{\pi} = [\bar{\pi}_1 \cdots \bar{\pi}_k \cdots \bar{\pi}_K]\) and \(\mathbb{W}\), compute

\[
\delta_k = \mathbb{W}(z - D_k\bar{\pi}_k), \quad \forall k \leq K.
\]

(15)

and hence

\[
g_k = ||\delta_k||_2, \quad \forall k \leq K.
\]

(16)

The class of the test image represented by \(z\) is determined with

\[
k(z) = \arg\min_k \{g_k\}.
\]

(17)

The entire procedure of the proposed WSR algorithm is outlined in Algorithm 1.

**Outline of Robust Sparse Representation-based Classification.**

---

**Algorithm 1:** Weighted Sparse Representation.  

**Require:** Dictionary \(D \in \mathbb{R}^{n \times m}\), test sample \(z \in \mathbb{R}^{n \times 1}\) and initial error \(\hat{e}^{(0)} = z - \frac{1}{m} \sum_{j=1}^{m} D(:,j)\); set \(\lambda = 0.4\) and \(N_{ite}\)—the number of iterations.  

**Ensure:**  
While \(0 \leq l \leq N_{ite}\) do  
(1) update \(\mathbb{W}\) via (13) and (14), yielding \(\mathbb{W}^{(l)}\);  
(2) update \(\bar{\pi}\) by solving the following using any of the basis pursuit-based algorithms:

\[
\hat{\alpha}^{(l)} = \arg\min_\alpha \|\mathbb{W}^{(l)}(z - D\alpha)\|_2^2 + \lambda \|\alpha\|_1
\]

(3) update the sparse representation error \(\hat{e}^{(l)}\) via (12);  
(4) \(l = l + 1\);  
End while if \(l > N_{ite}\) and output \(\bar{\pi}\) as well as \(\mathbb{W}\), and execute the classification with (15)–(17).  

Return

---

4. Experiments

In this section, we present some experimental results to examine our proposed approach. We first introduce the dataset and setup used in the experiments, then present the experimental results and discussions.
4.1. Dataset

The multimodal ground-based cloud dataset (MGCD) [19] collected in China mainly contains two kinds of ground-based cloud information, i.e., the cloud images and the multimodal cloud information. The cloud images with the size of $1024 \times 1024$ pixels are captured at different times by a sky camera with fisheye lens. The fisheye lens could provide a wide range observation of the sky conditions with the horizontal and vertical angles of 180 degrees. The multimodal cloud information is collected by a weather station, including temperature, humidity, pressure, wind speed, maximum wind speed and average wind speed. Each cloud image corresponds to a set of multimodal data. The 8000 pictures used are classified into 7 groups of clouds: cumulus, altocumulus, cirrus, clear sky, stratocumulus and mixed. In addition, it should be noted that the cloud images with less than 10% cloud cover belong to clear sky. The detailed distribution of samples for each class is illustrated in Table 3.

Table 3. Distribution of the 8000 samples from the MGCD.

<table>
<thead>
<tr>
<th>Label</th>
<th>Cloud Class</th>
<th>Number of Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cumulus</td>
<td>1438</td>
</tr>
<tr>
<td>2</td>
<td>Altocumulus</td>
<td>731</td>
</tr>
<tr>
<td>3</td>
<td>Cirrus</td>
<td>1323</td>
</tr>
<tr>
<td>4</td>
<td>Clear sky</td>
<td>1338</td>
</tr>
<tr>
<td>5</td>
<td>Stratocumulus</td>
<td>963</td>
</tr>
<tr>
<td>6</td>
<td>Cumulonimbus</td>
<td>1187</td>
</tr>
<tr>
<td>7</td>
<td>Mixed</td>
<td>1020</td>
</tr>
</tbody>
</table>

Figure 5 shows three samples for each of the 7 classes of clouds.

Figure 5. Samples with each for one of the seven classes of cloud pictures.

The occluded testing samples are demonstrated in Figure 6.
4.2. Parameter Setting

Random partition ensures that the learning features will be uniformly distributed, while non-random allocation leads to uneven samples, which will seriously affect the convergence of network training phase and the generalization of network testing phase. Therefore, the data set is randomly divided into training set and testing set. The first one contains 2/3 of the cloud samples of each class, and the second one contains 1/3 of the cloud samples of each class.

All experiments are carried out with the same experimental setup. We use transfer learning to train the convolutional neural network models of Inception-v3 network and ResNet-50 network. We change the size of the fully connected output layer to fit the cloud categories, and the input cloud samples are automatically cropped to input layer size.

A small batch random gradient descent method is used to adjust the parameters continuously, we set minimum batch size to 10, maximum epochs to 6, initial learn rate to 0.00001, validation frequency to 250.

4.3. Results

We will examine five methods for classifying ground-based Clouds, among which there are two existing ones:

- ICNN—using the Inception-v3 convolutional neural network (ICNN) for feature extraction and classification;
• RCNN—it is similar to ICNN, but the CNN used is the ResNet-50 convolutional neural network.

Three methods that we proposed:
• IWSRC—using the Inception-v3 convolutional neural network for feature extraction and the weighted sparse representation coding for classification;
• RWSRC—exactly the same as IWSRC but with the Inception-v3 CNN replaced by the ResNet-50;
• RIWSRC—this is the proposed two-channel CNN-based sparse representation coding method, depicted in Figure 4.

Train loss of convolutional neural network is displayed in Figure 7.

![Figure 7. Train loss of the two convolutional neural networks. An epoch means one complete pass of the training dataset through the algorithm.](image)

Train accuracy of convolutional neural network is shown in Figure 8.

![Figure 8. Train accuracy of the two convolutional neural networks.](image)

Table 4 shows the accuracy of all the methods used in this paper without occlusion.
Table 4. The accuracy(%) of all the methods used in this paper without occlusion.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICNN</td>
<td>96.97</td>
</tr>
<tr>
<td>RCNN</td>
<td>97.09</td>
</tr>
<tr>
<td>IWSRC</td>
<td>99.56</td>
</tr>
<tr>
<td>RWSRC</td>
<td>99.28</td>
</tr>
<tr>
<td>RIWSRC</td>
<td>99.81</td>
</tr>
</tbody>
</table>

Table 5 shows the accuracy of each method when the occluded cloud samples are used.

Table 5. The accuracy(%) of all the methods used in this paper with occlusion.

<table>
<thead>
<tr>
<th>Method</th>
<th>Occ. 5%</th>
<th>Occ. 10%</th>
<th>Occ. 15%</th>
<th>Occ. 20%</th>
<th>Occ. 25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICNN</td>
<td>84.03</td>
<td>83.18</td>
<td>82.24</td>
<td>79.43</td>
<td>77.52</td>
</tr>
<tr>
<td>RCNN</td>
<td>90.47</td>
<td>89.68</td>
<td>83.99</td>
<td>78.77</td>
<td>75.15</td>
</tr>
<tr>
<td>IWSRC</td>
<td>98.62</td>
<td>97.2</td>
<td>93.94</td>
<td>87.21</td>
<td>79.4</td>
</tr>
<tr>
<td>RWSRC</td>
<td>99.03</td>
<td>97.97</td>
<td>97.62</td>
<td>96.56</td>
<td>93.28</td>
</tr>
<tr>
<td>RIWSRC</td>
<td>99.37</td>
<td>98.87</td>
<td>98.06</td>
<td>95.53</td>
<td>90.28</td>
</tr>
</tbody>
</table>

4.4. Discussion

In Figure 7, the overall trend is downward, and generally the loss of RCNN is lower than that of ICNN. In Figure 8, the recognition rate of RCNN is faster than that of ICNN.

As see Table 4, the RCNN is better than the ICNN since the ResNet-50 increases gradient propagation and hence is more effective in clouds classification. As to the proposed methods IWSRC, RWSRC and RIWSRC, they are all better than ICNN and RCNN because the noise interference can be suppressed with weighted representation learning during classification. However, RIWSRC achieves the best performance with the optimized feature selection process. The testing cloud images are with 5%, 10%, 15%, 20% and 25% occlusion, respectively.

In Table 5, experimental results with various occlusion show that the proposed IWSRC and RWSRC are better than ICNN and RCNN. The neural network is sensitive to perturbation with occlusion, and weighted sparse representation (WSR) will adjust the weights according to the size of the error. The larger error leads to smaller weights, which helps to reduce the perturbation on classification. In general, the proposed RIWSRC is more robust and effective than both IWSRC and RWSRC. The multiple features fusion can further improve the robustness of the algorithm. The code is now publicly available on https://github.com/tangming666/NRC (accessed on 18 July 2021).

5. Conclusions

A novel robust classification scheme is proposed for ground-based clouds in this paper. The basic idea behind this is to use the CNNs for feature extraction and a weighted sparse representation coding for classification. Two classification algorithms are proposed directly along this line and the third one is based on the fusion of two CNNs. Experimental results show that the three proposed algorithms can enhance the robustness greatly. The proposed methods can be applied in most of deep learning neural networks.

As future research, we will consider adding some multimodal information to the selected neural network features to improve the robustness of the system. More applications of the proposed methods will be explored.

Author Contributions: All authors made significant contributions to the manuscript. A.Y. and M.T. conceived, designed and performed the experiments; G.L. wrote the paper; B.H. performed
the experiments and analyzed the data; Z.X., B.Z. and T.C. revised the paper and provided the background knowledge of cloud classification. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** No applicable.

**Informed Consent Statement:** No applicable.

**Data Availability Statement:** The models and code used during the study are available in a repository (https://github.com/tangming666/NRC, accessed on 1 August 2021). The data is available from the corresponding author by request (shuangliu.tjnu@gmail.com).

**Acknowledgments:** This work was supported by Key Research & Development Project of Zhejiang Province 2021C04030 & the Public Project of Zhejiang Province LGG20F020007 and LGG21F030004. We would like to express my gratitude to Liu Shuang for providing the MGCD datasets. Our deepest gratitude goes to the reviewers for their careful work and thoughtful suggestions that have helped improve this paper substantially.

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

**References**