machine learning & knowledge extraction

MDPI

Article

Voting in Transfer Learning System for Ground-Based Cloud Classification

Mario Manzo [1,*], Simone Pellino [2]

[1] Information Technology Services, University of Naples "L'Orientale", 80121 Naples, Italy
[2] Department of Applied Science, I.S. Mattei Aversa M.I.U.R., 81031 Rome, Italy; simonepellino@gmail.com
* Correspondence: mmanzo@unior.it; Tel.: +39-08-16909229 (ext. 80121)

Abstract: Cloud classification is a great challenge in meteorological research. The different types of clouds, currently known and present in our skies, can produce radioactive effects that impact the variation of atmospheric conditions, with consequent strong dominance over the earth's climate and weather. Therefore, identifying their main visual features becomes a crucial aspect. In this paper, the goal is to adopt pretrained deep neural networks-based architecture for clouds image description, and subsequently, classification. The approach is pyramidal. Proceeding from the bottom up, it partially extracts previous knowledge of deep neural networks related to original task and transfers it to the new task. The updated knowledge is integrated in a voting context to provide a classification prediction. The framework trains the neural models on unbalanced sets, a condition that makes the task even more complex, and combines the provided predictions through statistical measures. An experimental phase on different cloud image datasets is performed, and the results achieved show the effectiveness of the proposed approach with respect to state-of-the-art competitors.

Keywords: cloud classification; deep learning; transfer learning; voting-based classification; climate change

Citation: Manzo, M.; Pellino, S. Voting in Transfer Learning System for Ground-Based Cloud Classification. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 542–553. https://doi.org/10.3390/make3030028

Academic Editor: Andreas Holzinger

Received: 16 June 2021
Accepted: 8 July 2021
Published: 12 July 2021

1. Introduction

Clouds are a constant presence in our skies; combined with the role assumed by ecosystems, they are important in determining the atmospheric conditions, hours of sunshine and temperature. Nowadays, dynamic atmospheric conditions, attributed to climate change, have led to an increase in attention to the behavior of clouds [1]. Climate models allow to predict climate changes, but their precision degree is currently insufficient and attributable to alteration in the conditions determined by different phenomena. Consequently, cloud behavior prediction is important for estimating climate change. Furthermore, cloud changes are also a reason for the influence on the earth radiation budget and energy balance [2–4]. A great deal of effort has been made by the scientific community to acquire many datasets, but they have not fully managed, due to a lack of devices with adequate processing resources. In recent years, however, the state of the art has seen the birth of numerous studies and analyses that can automate clouds attributes and classes detection that are scientifically relevant. Due to the amount of ground cloud images, the recognition phase has been extensively studied in the literature as of late. Most of the standard algorithms adopt hand-made features, such as brightness, texture, shape, and color, to represent image contents but without obtaining model generalization, due to the complex distribution of data. Indeed, the visual information contained in the image is unable to accurately describe the clouds, due to the large variations in appearance. Finally, the non-visual features, also known as multimodal information, obtainable from the cloud process formation, such as temperature, humidity, pressure and wind speed, can be of help. According to recent studies, deep learning has proven effective for image management, analysis, representation and classification in the cloud recognition field. In particular, the success of deep neural networks, applied to the image classification task, concern several

interesting aspects mainly connected to software development and the large amount of data available. Specifically, for cloud images analysis, deep neural networks are adopted both in the segmentation and detection phases. However, image content and poor data balance among classes have a decisive impact on performance, causing uncertainty in the model generalization. With the purpose of addressing the above problems, we present a framework based on deep transfer and voting learning for cloud image classification. It is built based on three steps: (1) performing image preprocessing operations such as resizing, essential for neural networks training; (2) modifying and retraining multiple deep neural networks, exploiting previous knowledge; and (3) looking at the different predictions provided by the deep neural networks and combining them in order to provide the best decision in the classification phase. The main contributions of the proposed framework can be summarized in some key points. First, the framework based on deep and voting learning is designed to address the imbalance between classes in the cloud recognition task. Second, the framework is built on multiple classification models based on deep transfer learning. Third, we demonstrate that several models, suitably combined, can strengthen the decision in classification with respect to a single one. Finally, the experimental demonstrations are compared to the established existing methods on the datasets recognized by field experts. The paper is structured as follows. Section 2 provides an overview of the state of the art about cloud classification approaches. Section 3 describes in detail the proposed framework. Section 4 provides a wide experimental phase, while Section 5 concludes the paper.

2. Related Work

In this section, we briefly describe the most important studies about cloud images classification in the literature. In this field, numerous works address the task, according to different aspects, such as image characterization, segmentation algorithm application to obtain new descriptors, complex mechanisms of learning and classification, and much more.

In [5], the authors present a layer named joint fusion (JFCNN) to jointly learn two kinds of cloud features under one framework. After training the proposed JFCNN, they extract the visual and multimodal features from two subnetworks, which are based on the well-known Resnet50 [6] and integrate them using a weighted strategy. The architecture consists of five parts: two subnetworks, one joint fusion layer, one FC layer and the loss function. The subnetworks are used for learning cloud visual features. The authors work with two kind of extracted features, combined in a multimodal way, which contain some complementary information and different characteristics of the ground-based cloud.

An approach called the multi-evidence and multi-modal fusion network (MMFN) is proposed in [7]. The idea is to learn extended cloud information by fusing heterogeneous features (global and local) in a unified framework. MMFN takes advantage of multiple pieces of evidence using a main network and an attentive network. In the attentive network, local visual features are extracted from attentive maps, which are obtained by refining salient patterns from convolutional activation maps. Meanwhile, the main network learns multi-modal features for ground-based cloud. In order to combine the multi-modal and multi-evidence visual features, the authors design two fusion layers in MMFN to incorporate multi-modal features with global and local visual features, respectively.

In [8], the authors propose to use deep convolutional activations-based features (DCAFs). Cloud images are directly fed into a CNN model. Then, the features from different convolutional and FC layers are extracted through different pooling strategies. Finally, a multilabel linear support vector machine (SVM) model is used for the classification step.

A convolutional neural network model, called CloudNet, for accurate ground-based meteorological cloud classification is proposed in [9]. The model consists of five convolutional layers and two FC layers. In addition, to optimize the network training, the image input is processed through a robust strategy that subtracts the mean red-green-blue value of each pixel over the training set to improve training speed and accuracy. Furthermore, the authors create a clouds dataset, called Cirrus Cumulus Stratus Nimbus (CCSN), which consists of 11 categories under meteorological standards.

In [10], the authors propose an approach called deep multimodal fusion (DMF). In order to learn the visual features, CNN models are applied to capture texture information. The extracted features, from deeper layers, have several eligible properties, such as invariance and discrimination. Subsequently, the authors employ a weighted strategy to integrate visual and multimodal features. Finally, the SVM algorithm to train the classification model is adopted.

In [11], a deep tensor fusion network is presented in order to hold spatial information of ground-based cloud images. It fuses cloud visual and multimodal features at the tensor level.

In [12], the author proposes an approach called hierarchical multimodal fusion (HMF), which fuses deep multimodal and deep visual features in different levels. The architecture is composed of two subnetworks: the visual subnetwork and multimodal subnetwork. The visual subnetwork is defined in order to extract deep visual features from ground-based cloud images, employing Resnet50 [6]. The multimodal subnetwork is used to learn features from a vector composed of six FC layers. The classification step through SVM is managed.

In [13], the author proposes a classification method of sky-condition based on whole sky infrared cloud images, where the local binary patterns operator (LBP) and the contrast of local cloud image texture (VAR signal) are combined to classify the sky conditions. The correspondence relationship among traditional cloud classes and instrument-measured cloud classes is suggested. The approach analyzes the LBP spectra and VAR characteristics for five classes of clouds.

An automatic cloud classification algorithm is developed in [14]; the approach uses an image-mask created by visually identifying image regions containing discriminative information. Furthermore, the approach extracts a set of mainly statistical features describing the color as well as the texture of an image. The classification step adopts the k-nearest neighbor algorithm.

A modified texton-based classification approach that integrates both color and texture information to improve classification results is proposed in [15]. The color channel is adopted to generate image descriptors and filter responses of images across all the categories, aggregating them together. K-means clustering is applied on the concatenated filter responses, producing the different cluster centers. These clusters centers are the modified-textons and constitute the texton dictionary. The discriminative histogram model for each image category is generated by comparing the filter responses of the pixels with the generated textons in the dictionary.

An ensemble learning method and resource allocation scheme for cloud observation and classification is proposed in [16]. Ensemble methods, such as Bagging, AdaBoost and Snapshot, are used as a base classifier to take the cross-semantic and structure features of cloud images.

3. Materials and Methods

In this section, we describe the proposed framework, which includes two methodologies: deep neural networks [17] and voting learning [18]. The goal is to combine several deep neural networks with the purpose of classifying images of clouds. Specifically, a set of competitive models are aligned and provide a range of confidential decisions useful for making choices during classification. The framework is composed of three blocks. The first performs preprocessing in terms of image resize. The second learns different deep neural networks, previously redesigned for the specific task. The third combines different potential indications through voting rules, provided by deep neural networks for the classification purpose. Finally, the framework runs a predetermined number of iterations in a supervised learning context. An overview of the proposed framework in Figure 1 is shown.

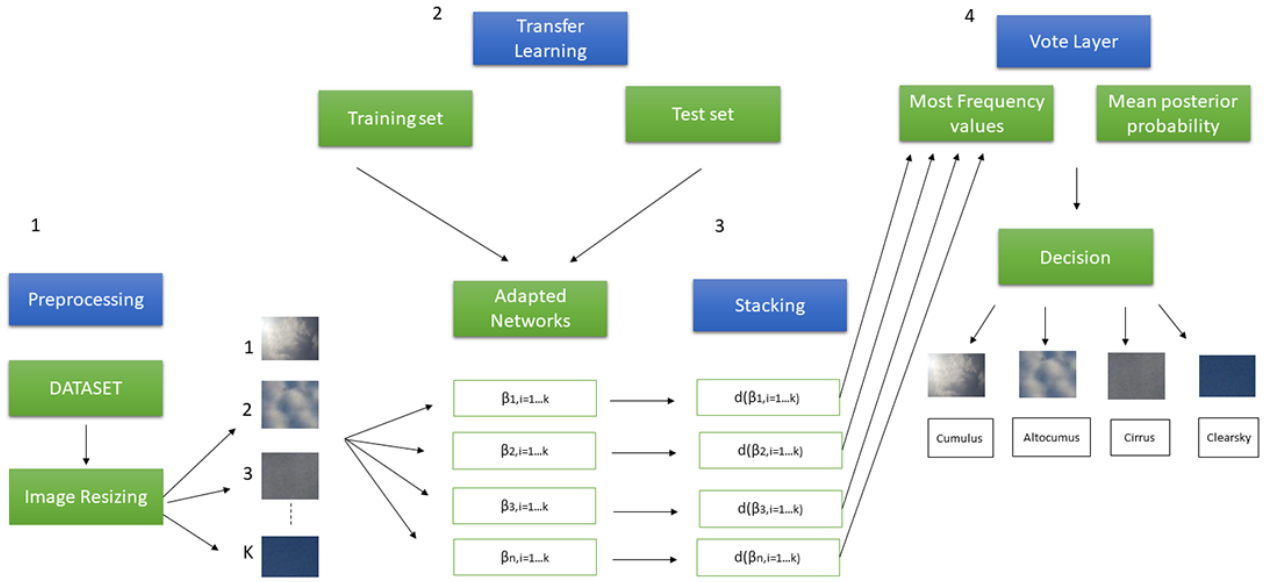

Figure 1. Overview of the proposed framework. The steps of the pipeline are numbered consecutively and the direction of the arrows indicates the data flow. Step 1 indicates image preprocessing described in Section 3.1. Step 2 describes how the different deep neural networks are redesigned and retrained for the classification task as described in the Section 3.2. Steps 3 and 4 illustrate, respectively, the stacking approach adopted to combine the deep neural networks and the vote process for final decision as described in Section 3.3.

3.1. Image Resize

One of the drawbacks of neural networks concerns the fixed dimension about the input layer with reference to the images to be processed (details about adopted neural networks can be found in Table 1 at column 5). Size normalization, according to the input layer dimension, is essential because it is not possible to process different or large sized images for the network training and classification stages. This step does not alter the content of the image information in any way.

Table 1. Description of adopted pretrained network. Only about Nasnetlarge no depth information is available.

Network	Depth	Size (MB)	Parameters (Millions)	Input Size
Densenet201	201	77	20	224 × 224
Alexnet	8	227	61	227 × 227
Googlenet	8	27	7	224 × 224
Resnet18	18	44	11.7	224 × 224
Resnet50	50	96	25.6	224 × 224
Nasnetlarge	*	332	88.9	331 × 331

3.2. Network Design and Transfer Learning

Transfer learning is selected as the training strategy. The basic idea is to transfer the knowledge extracted from a source domain to a destination one, in our case, cloud classification. Generally, a pretrained network is chosen as a starting point in order to learn a new task. It turns out to be the most convenient and forthcoming solution to adopt the representational power of pretrained deep neural networks. Clearly, it is easier and faster to tune a network with transfer learning than training a new network from scratch with randomly initialized weights. For clouds recognition, deep learning architectures are selected based on their task compliance. The goal is to train networks on images by

redesigning their structures in the final layer according to different outgoing classes. Table 1 supports the description below about adopted neural models.

Alexnet [19] consists of five convolutional layers and three fully connected layers. It includes the non-saturating ReLU activation function, which is better then Tanh and Sigmoid during the training phase.

Googlenet [20] is composed of 22 deep layers. The network is inspired by LeNet [21] but implements a novel element, which is dubbed an inception module. This module is based on several very small convolutions in order to drastically reduce the number of parameters. The architecture reduces the number of parameters from 60 million (AlexNet) to 4 million. Furthermore, it includes batch normalization, image distortions and the root mean square propagation algorithm.

Densenet201 [22] is a convolutional neural network with 201 deep layers. Unlike standard convolutional networks composed of L layers with L one-to-one connections between the current layers and the nexts, it contains $\frac{L(L+1)}{2}$ direct connections. Specifically, each layer adopts the feature maps of all the preceding layers and its own feature maps into all subsequent layers as inputs.

Resnet18 and Resnet50 [6] are inspired by pyramidal cells contained in the cerebral cortex. They use particular skip connections or shortcuts to jump over some layers. They are composed of 18 and 50 deep layers, which, with the help of a technique known as skip connection, have paved the way for residual networks.

Nasnetlarge [23] is designed on a search space, called NASNet search space, which enables transferability. The model works by looking for the best convolutional layer, or cell, and subsequently replicating this layer in a stack, each with its own parameters to design a convolutional architecture. Additionally, a regularization technique, called Scheduled-DropPath, which significantly improves generalization in the model, is introduced.

Deep neural networks are adapted to the clouds classification problem. Originally, the Imagenet dataset [24], which includes one million images divided into 1000 classes, is adopted to perform the main training phase. Generally, a network elaborates an image and provides a prediction about a class to which it might belong with an attached probability. Indeed, a network is structured to work on different layers. The first concerns the input image and requires three color channels. Next, convolutional layers, which work with the purpose to extract image features, are placed. The last learnable and the final classification layers are adopted to classify the input image. To make the pretrained network compliant to the classification of new images, the last two layers are replaced with new layers. Often, the last layer, with its learnable weights, is completely connected. It is removed and replaced by a new one that is completely connected with the outputs related to classes of new data (clouds types). Furthermore, the learning of the new layer, connected with the transferred layers, can be sped up by increasing the learning rate factors. Optionally, the weights of the previous levels can be left unchanged by resetting the learning rate to zero. This modification avoids weight updates during training and the consequent flattening of the execution time, as it is not necessary to calculate the gradients of the relative layers. This improvement has a strong impact in the case of small datasets to avoid overfitting.

3.3. Voting Based Learning

A voting based learning approach is adopted to manage the classification phase. In particular, among all possible strategies, we select stacking. It works by training a single classifier and, subsequently, combines it with further classifiers. Unlike a standard approach, where weak or strong learners are adopted, we basically combine several equally powerful models that predict an outcome with a certain probability. Finally, we join all the predictions for a classification result. The general model can be summarized by the following matrix:

$$CN = \begin{bmatrix} \beta_1 i_1 & \dots & \beta_1 i_k \\ \vdots & \ddots & \\ \beta_n i_1 & & \beta_n i_k \end{bmatrix} \quad (1)$$

where each i_k represents an image to be classified, taken from the set $Imgs = \{i_1, i_2, \ldots, i_k\}$ with cardinality k, belonging to one of x classes. Furthermore, each β_n represents a deep neural network, taken from the set $C = \{\beta_1, \beta_2, \ldots, \beta_n\}$ with cardinality n, which provides a decision $d \in I\{1, \ldots, x\}$, with reference to $i_k \in Imgs$ and x membership classes. The set of decisions can be rearranged through the following matrix D:

$$D = \begin{bmatrix} d_{\beta_1 i_1} & \ldots & d_{\beta_1 i_k} \\ \vdots & \ddots & \\ d_{\beta_n i_1} & & d_{\beta_n i_k} \end{bmatrix} \tag{2}$$

which describes the result of the deep neural networks combination and images of the matrix CN in terms of position, such as $\beta_n i_k \rightarrow d_{\beta_n i_k}$. In addition, a score value $s \in S\{0, \ldots, 1\}$ is associated to each decision d and provides the posterior probability $P(i|x)$ that an image i could belong to class x. Finally, all the score values relating to the results of the possible combinations of matrix CN are collected in the matrix S:

$$S = \begin{bmatrix} P(i_1|x)_{d_{\beta_1 i_1}} & \ldots & P(i_k|x)_{d_{\beta_1 i_k}} \\ \vdots & \ddots & \\ P(i_1|x)_{d_{\beta_n i_1}} & & P(i_k|x)_{d_{\beta_n i_k}} \end{bmatrix} \tag{3}$$

where each element of posterior probability in the matrix S refers to element of the matrix CN, such as $\beta_n i_k \rightarrow d_{\beta_n i_k} \rightarrow P(i_k|x)_{d_{\beta_n i_k}}$. Moving on, each column of the matrix D is analyzed with the statistical mode and stored in the vector DM:

$$DM = \{dm_{d_{\beta_{1,\ldots,n} i_1}}, \ldots, dm_{d_{\beta_{1,\ldots,n} i_k}}\}, \tag{4}$$

where the generic value dm contains the modal value of the class to which image i could belong with the average probability score ds. In essence, this is the class to which an image could belong based on the votes given by different deep neural networks. In this regard, the concept of the statistical mode is introduced. It can be defined as the value that repeatedly occurs in a given set:

$$mode = l + \left(\frac{f_1 - f_0}{2f_1 - f_0 - f_2} \right) \times h \tag{5}$$

where l is the lower limit of the modal class, h is the size of the class interval, f_1 is the frequency of the modal class, f_0 is the frequency of the class which precedes the modal class and f_2 is the frequency of the class which successes the modal class. The columns of matrix D are analyzed in order to obtain the values of the most frequent decisions. This step is performed in order to verify the highest voted classes from different deep neural networks, contained in the CN set. Moreover, the aim of the mode application is twofold: first, to extract the most frequent value, and second, to extract its occurrences in terms of indices. For each most frequent occurrence, modal value, the corresponding score from the matrix S is extracted. To this end, DS vector is built as follows:

$$DS = \{ds_{P(i_1|x)_{d_{\beta_{1,\ldots,n} i_1}}}, \ldots, ds_{P(i_k|x)_{d_{\beta_{1,\ldots,n} i_k}}}\}, \tag{6}$$

where each element ds contains the average decision scores with higher frequency, extracted through the mode with reference to the corresponding column of matrix D.

4. Experimental Results

This section describes the experimental phase. In order to train the neural models, with the purpose of performing the classification task in a supervised context, labeled data are need. Consequently, the issue to be addressed concerns the quantity of data sufficient to produce experimental results. The contents of a large dataset, useful for training and

testing, strongly affect the classification performance. Therefore, the discriminating factor for the effectiveness of neural models is the amount of data. Contextually, with the purpose of producing compliant performance, the settings reported in recent cloud classification methods are adopted.

4.1. Datasets

The proposed framework on a state-of-art datasets, containing ground-based clouds images, is tested. The following datasets are adopted:

1. The multimodal ground-based cloud database (MGCD) [7,25] is collected in China and consists of cloud images captured by a sky camera with a fisheye lens under a variety of conditions and multimodal cloud information. It includes a total amount of 1720 cloud data. The images are divided into seven classes: cumulus, cirrus, altocumulus, clear sky, stratus, stratocumulus, and cumulonimbus. The number of item of each class varies from 140 to 350, and the detailed numbers are listed in Table 2.
2. The Singapore whole sky imaging categories database (SWIMCAT) dataset [15] is composed of 784 sky/cloud patch images with 125 × 125 pixels captured using a wide-angle high-resolution sky imaging system, a calibrated ground-based WSI designed by [26]. The dataset is split into five distinct categories: clear sky, patterned clouds, thick dark clouds, thick white clouds, and veil clouds. The details are presented in Table 3.
3. The cirrus cumulus stratus nimbus (CCSN) dataset [9] contains only 2543 unique cloud images with 256 × 256 pixels in the JPEG format and contains 10 different forms in cloud observation. It is characterized by a large set of images, making it the largest of the available public cloud datasets. Details are shown in Table 4.

Table 2. Details of MGCD dataset.

Label	Cloud Type	Number of Samples
1	Cumulus	160
2	Cirrus	300
3	Altocumulus	340
4	Clear sky	350
5	Stratocumulus	250
6	Stratus	140
7	Cumulonimbus	180

Table 3. Details of SWIMCAT dataset.

Label	Cloud Type	Number of Samples
A	Clear Sky	224
B	Patterned clouds	89
C	Thick dark clouds	251
D	Thick white clouds	135
E	Veil clouds	85

Table 4. Details of CCSN dataset.

Label	Cloud Type	Number of Samples
Ci	Cirrus	139
Cs	Cirrostratus	287
Cc	Cirrocumulus	268
Ac	Altocumulus	221
As	Altostratus	188
Cu	Cumulus	182
Cb	Cumulonimbus	242
Ns	Nimbostratus	274
Sc	Stratocumulus	340
St	Stratus	202
Ct	Contrails	200

4.2. Results

Table 5 provides indications about the adopted neural models with respect to datasets. As can be seen, different combinations are provided due to the variable composition (total images, images per class, etc.) of each dataset. In fact, two additional neural models, Resnet18 and Nasnetlarge, are added for CCSN processing, as it is the most complex to manage. Moreover, the framework consists of different modules written in Matlab language. The neural models are trained based on different parameters. Stochastic gradient descent (SGDM) with momentum is adopted as the solver of the training process. Its main peculiarity concerns the oscillation along the steepest descent path toward the optimum. Adding a momentum term to the parameter update is one way to reduce this oscillation. Carrying on, the MiniBatchSize value, the subset size of the training set adopted to evaluate the gradient of the loss function and update the weights, is set to 10, which is optimal for the obtained results. Regarding MaxEpochs, the maximum number of epochs to use for training, the right compromise is reached with the value of 6, optimizing execution time and performance. An iteration is a step performed by SGDM to minimize the loss function using MiniBatchSize. An epoch concerns the complete cycle of the training process on the training set. InitialLearnRate is set to 3e-4. If it is too low, this results in a high training time. Otherwise, if it is too high, the result may be suboptimal or the training may diverge. The right compromise is found for the latter. To avoid discarding the same data every epoch, the shuffle parameter is to every epoch. Finally, ValidationFrequency, the number of iterations between evaluations of validation metrics, is set as the ratio between the training set size and the MiniBatchSize.

Table 5. Deep neural networks adopted with respect to datasets.

Networks \ Datasets	MGCD	SWIMCAT	CCSN
Densenet201	✔	✔	✔
Alexnet	✔	✔	✔
Googlenet	✔	✔	✔
Resnet18	✗	✗	✔
Resnet50	✔	✔	✔
Nasnetlarge	✗	✗	✔

The classification accuracy of the MGCD dataset is presented in Table 6. In order to produce a comparison with further methods that work on the same ground-based cloud classification task, the settings described in [7] are adopted. Looking at the results, several conclusions can be drawn. The proposed model, composed of different pretrained networks, produces promising recognition accuracy, giving high representation of the

cloud images. The implemented voting-based architecture, compared with the competitors, has better performance.

Table 6. Experimental results on MGCD dataset.

Method	Acc
Our	99.98
MMFN [7]	88.63
DCAFs + MI [7]	82.97
BOVW + MI [7]	67.20
PBOVW + MI [7]	67.15
LPB + MI [7]	50.53
CLPB + MI [7]	69.68
CloudNet + MI [7]	80.37
BoVW [27]	66.15
PBoVW [27]	66.13
LBP [28]	55.20
CLBP [29]	69.18
VGG-16 [30]	77.95
DCAFs [8]	82.67
CloudNet [9]	79.92
DMF [10]	79.05
DTFN [11]	86.48
HMF [12]	87.90

The classification performance of the SWIMCAT dataset is summarized in Table 7. In this phase, the settings present in [8] are adopted. In particular, a cross validation on 2,3,4,5 folds is performed first. Subsequently, 40 images per class for training and 45 ones for testing are selected randomly. The average accuracy of 50 random runs is reported. In the training phase, the neural models that do not contribute to improve both the performance and the execution time are discarded, as can be seen in Table 5. The results highlight that combining different multiple classification predictions is useful to capture more spatial and local layout information of clouds, with the purpose of outperforming the compared methods. Furthermore, it is important to underline that the proposed approach is even better than neural models, such as VGG-16 [30] and CloudNet [9], for which a single classification confidence value is provided, compared to a multiple voting based mechanism.

Table 7. Experimental results on SWIMCAT dataset.

Methods \ Folds	2	3	4	5	40/45
Our	99.36	99.49	99.49	99.75	99.91
LPB [13]	85.26	81.60	83.51	85.03	93.47
Heinle Feature [14]	90.26	91.89	92.91	93.43	93.09
Text-based method [15]	-	-	-	-	95.00
DCAF [8]	98.72	98.46	98.97	98.84	99.56

Table 8 shows the performance of the CCSN dataset. In order to compare the results with further methods that worked on same task, the settings described in [16] are adopted. Additionally, in this case, the combination of the neural models leads to better performance. As shown in Table 5, for this experimental phase, we stack all analyzed deep neural networks. Once again, the results demonstrate that the multiple base learners may lead to better performance, according to the combination with the different number of base learners in the stacking.

Table 8. Experimental results on CCSN dataset.

Method	Acc
Our	95.08
Cloudnet [9]	90.00
[16]	80.00
MMI [5]	75.42
M_DF [5]	78.21
M_JFCNN [5]	84.55
V_DF [5]	85.10
V_JFCNN [5]	86.79
V_DF + MMI [5]	86.33
V_JFCNN + MMI [5]	89.40
V_DF + M_DF [5]	90.21
J_JFCNN [5]	78.82
JFCNN [5]	93.37

4.3. Discussion

The presented satisfactory results are attributable to many relevant aspects. The first regards the features extracted through convolutional layers of the deep neural network. They provide good image representation, although they are completely abstract and devoid of real meaning. The second regards the framework's capability to provide multiple representation models, which lead to a significant improvement in performance. Another issue concerns image size normalization, tackled by many methods in the field. It is performed before features extraction to avoid performance degradation. Again, we can look at the robustness with respect to the underrepresented classes in the datasets. In fact, the framework does not fail, even though the samples are not sufficient for class representation in specific cases. The latter appears to be an open problem in the literature, as ad hoc classifiers are often designed for unbalanced classification different from the standard ones that produce untrue results. Contrarily, a weak point concerns the computational aspect. First, the time required for training the pretrained model is high but less than that of a model created from scratch. Second, the classification step, which provides multiple choices in decision making at each iteration, requires a lot of effort. The latter works for the purpose of choosing which classifiers are suitable for specific clouds images included in the test set. Finally, we have shown that although the framework is more expensive from a computational point of view, it produces better results, compared to a single classifier.

5. Conclusions and Future Works

The challenge in ground-based cloud recognition is specifically interesting, and not only for its multiple aspects and variety of data. The complexity of the task is linked to several factors, such as the type of clouds and the visual patterns contained in them. In support, convolutional neural networks lend a big hand in understanding the meaning of images with the consequent goal of their classification. In this regard, we proposed a framework that combines convolutional neural networks, adapted to the cloud recognition task through a transfer learning approach, using voting rules. The results produced certainly strengthen the theoretical thesis. A multiple model, based on several deep neural networks, compared to a single one is a powerful factor. Through a large experimental phase, it is shown that the proposed approach is competitive and, in some cases, better, compared to the more advanced methods. Although pretrained models were adopted, the main weakness concerns the computational complexity of the learning phase, which requires a long time and is sensitive to the growth of the data. Future work will certainly concern the study and analysis of still-unexplored convolutional neural networks for this type of problem and the application of the proposed framework to further datasets with the aim of taking a step forward in cloud recognition.

Author Contributions: M.M. and S.P. conceived the study and contributed to the writing of the manuscript and approved the final version. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Acknowledgments: We would like to thank Alfredo Petrosino. He followed us in our first steps toward computer science through a whirlwind of goals, ideas, and especially love and passion for the work. We will be forever grateful to him, our great teacher.

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

References

1. Duda, D.P.; Minnis, P.; Khlopenkov, K.; Chee, T.L.; Boeke, R. Estimation of 2006 Northern Hemisphere contrail coverage using MODIS data. *Geophys. Res. Lett.* **2013**, *40*, 612–617. [CrossRef]
2. Rossow, W.B.; Schiffer, R.A. ISCCP cloud data products. *Bull. Am. Meteorol. Soc.* **1991**, *72*, 2–20. [CrossRef]
3. Chen, T.; Rossow, W.B.; Zhang, Y. Radiative effects of cloud-type variations. *J. Clim.* **2000**, *13*, 264–286. [CrossRef]
4. Stephens, G.L. Cloud feedbacks in the climate system: A critical review. *J. Clim.* **2005**, *18*, 237–273. [CrossRef]
5. Liu, S.; Li, M.; Zhang, Z.; Xiao, B.; Cao, X. Multimodal Ground-Based Cloud Classification Using Joint Fusion Convolutional Neural Network. *Remote Sens.* **2018**, *10*, 822. [CrossRef]
6. 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.
7. Liu, S.; Li, M.; Zhang, Z.; Xiao, B.; Durrani, T.S. Multi-evidence and multi-modal fusion network for ground-based cloud recognition. *Remote Sens.* **2020**, *12*, 464. [CrossRef]
8. Shi, C.; Wang, C.; Wang, Y.; Xiao, B. Deep Convolutional Activations-Based Features for Ground-Based Cloud Classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 816–820. [CrossRef]
9. Zhang, J.; Liu, P.; Zhang, F.; Song, Q. CloudNet: Ground-based cloud classification with deep convolutional neural network. *Geophys. Res. Lett.* **2018**, *45*, 8665–8672. [CrossRef]
10. Liu, S.; Li, M. Deep multimodal fusion for ground-based cloud classification in weather station networks. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 48. [CrossRef]
11. Li, M.; Liu, S.; Zhang, Z. Deep tensor fusion network for multimodal ground-based cloud classification in weather station networks. *Ad Hoc Netw.* **2020**, *96*, 101991. [CrossRef]
12. Liu, S.; Duan, L.; Zhang, Z.; Cao, X. Hierarchical multimodal fusion for ground-based cloud classification in weather station networks. *IEEE Access* **2019**, *7*, 85688–85695. [CrossRef]
13. Sun, X.; Liu, L.; Gao, T.; Zhao, S. Classification of whole sky infrared cloud image based on the LBP operator. *Trans. Atmos. Sci.* **2009**, *32*, 490–497.
14. Heinle, A.; Macke, A.; Srivastav, A. Automatic cloud classification of whole sky images. *Atmos. Meas. Tech.* **2010**, *3*, 557–567. [CrossRef]
15. Dev, S.; Lee, Y.H.; Winkler, S. Categorization of cloud image patches using an improved texton-based approach. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; IEEE: New York, NY, USA, 2015; pp. 422–426.
16. Zhang, J.; Liu, P.; Zhang, F.; Iwabuchi, H.; de Moura, A.A.; de Albuquerque, V.H.C. Ensemble Meteorological Cloud Classification Meets Internet of Dependable and Controllable Things. *IEEE Internet Things J.* **2020**, *8*, 3323–3330. [CrossRef]
17. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
18. Peteiro-Barral, D.; Guijarro-Berdiñas, B. A survey of methods for distributed machine learning. *Prog. Artif. Intell.* **2013**, *2*, 1–11. [CrossRef]
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
20. 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.
21. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
22. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
23. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2018; pp. 8697–8710.

24. 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; IEEE: Miami, FL, USA, 2009; pp. 248–255.
25. Liu, S.; Li, M.; Zhang, Z.; Cao, X.; Durrani, T.S. Ground-Based Cloud Classification Using Task-Based Graph Convolutional Network. *Geophys. Res. Lett.* **2020**, *47*, e2020GL087338. [CrossRef]
26. Dev, S.; Savoy, F.M.; Lee, Y.H.; Winkler, S. WAHRSIS: A low-cost high-resolution whole sky imager with near-infrared capabilities. In Proceedings of the Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXV, Baltimore, MD, USA, 6–8 May 2014; International Society for Optics and Photonics: Bellingham, DC, USA, 2014; Volume 9071, p. 90711L.
27. Csurka, G.; Dance, C.; Fan, L.; Willamowski, J.; Bray, C. Visual categorization with bags of keypoints. In Proceedings of the Workshop on Statistical Learning in Computer Vision, ECCV, Prague, Czech Republic, 11–14 May 2004; Volume 1, pp. 1–2.
28. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
29. Guo, Z.; Zhang, L.; Zhang, D. A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Trans. Image Process.* **2010**, *19*, 1657–1663. [PubMed]
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.