



Block-based cloud classification with statistical features and distribution of local texture features

H.-Y. Cheng¹ and C.-C. Yu²

¹Department of Computer Science and Information Engineering, National Central University, 300 Jongda Rd., Zhongli Dist., Taoyuan City 32001, Taiwan

²Department of Computer Science and Information Engineering, Vanung University, 1 Wanneng Rd., Zhongli Dist., Taoyuan City 32061, Taiwan

Correspondence to: H.-Y. Cheng (breeze.cheng@gmail.com)

Received: 5 September 2014 – Published in Atmos. Meas. Tech. Discuss.: 25 November 2014

Revised: 19 February 2015 – Accepted: 19 February 2015 – Published: 10 March 2015

Abstract. This work performs cloud classification on all-sky images. To deal with mixed cloud types in one image, we propose performing block division and block-based classification. In addition to classical statistical texture features, the proposed method incorporates local binary pattern, which extracts local texture features in the feature vector. The combined feature can effectively preserve global information as well as more discriminating local texture features of different cloud types. The experimental results have shown that applying the combined feature results in higher classification accuracy compared to using classical statistical texture features. In our experiments, it is also validated that using block-based classification outperforms classification on the entire images. Moreover, we report the classification accuracy using different classifiers including the k-nearest neighbor classifier, Bayesian classifier, and support vector machine.

reserved power is not used, it is a waste. If the reserved power is not enough, a blackout will happen. To utilize solar energy more effectively, integrated and large-scale PV systems need to overcome the unstable nature of solar resources. PV grid operators desire mechanisms of scheduling, dispatching, and allocating energy resources adaptively. Obtaining an accurate estimation of the resources that can be exploited is helpful for reducing costs and achieving better efficiency. Therefore, the ability to perform accurate short-term forecast on surface solar irradiance is desired.

The unstable and intermittent nature of solar resources is due to the influences of cloud cover and cloud types. The height and the thickness of the clouds vary for different types of clouds. Therefore, the impact on the irradiance caused by different types of clouds also varies a lot (Martínez-Chico et al., 2011; Fu and Cheng, 2013). Large-scale cloud information is available from satellite images. However, the spatial and temporal resolutions provided by satellite images are not high enough for short-term prediction. As a consequence, devices that capture all-sky images are designed to monitor the sun and the clouds. Devices developed more recently include a whole-sky imager developed by Scripps Institute of Oceanography at the University of California (Li et al., 2004; Kassianov et al., 2005), a whole-sky camera designed by Spain's University of Girona (Long et al., 2006), an all-sky imager developed by Japan's Communications Research Laboratory (Kubota et al., 2003), and a total-sky imager by Yankee Environmental Systems (Pfister et al., 2003; Calbo and Sabburg, 2008). With the all-sky images captured by these devices, analyzing the cloud activities on a basis of

1 Introduction

The demand for sustainable and green energy is growing as fossil fuel bases decline and gas emissions increase. Solar energy is one emerging green energy that has been improved significantly in recent years. Recently, a large number of photovoltaics (PV) were installed worldwide. However, the main challenge of PVs is that the produced electricity is often variable and intermittent. The fluctuation of the supply makes the energy expensive and prevents it from prevalence. Due to the unpredictable nature, the grid operators usually need to adopt a more conservative strategy and reserve enough power. If the

more refined scales is feasible. Such analysis on cloud activities include cloud-cover detection, cloud tracking, and cloud classification. The purpose of cloud classification is to distinguish the cloud types and hopefully figure out their impacts on the change of irradiance.

In the work of Martínez-Chico et al. (2011), the clouds were classified into different attenuation groups according to different levels of attenuation of the direct solar radiation reaching the surface. The authors also analyzed the annual and seasonal frequencies of each cloud group. However, this work did not propose any method for extracting features from images and performing classification based on image features. For works of cloud classification using sky image features, we review the following existing methods. The research by Calbo and Sabburg (2008) used features based on Fourier transform along with simple statistics such as standard deviation, smoothness, moments, uniformity, and entropy. The features are extracted from intensity images and red-to-blue components ratio (R/B) images. The classifier they used was based on the supervised classification parallel technique.

In the work of Heinle et al. (2010) statistical features such as mean, standard deviation, skewness, and difference are utilized. Also, textural features including energy, entropy, contrast, and homogeneity are computed from the grey-level co-occurrence matrices (GLCM). Instead of extracting features from intensity images, the authors reported the color component for which each individual feature should be calculated. This work used a k -nearest neighbor (k-NN) classifier to classify the clouds into seven different types. Kazantzidis et al. (2012) improved the method of Heinle et al. by dividing the data set into subclasses according to solar zenith angle, cloud coverage, and visible fraction of the solar disk. Other features such as autocorrelation, edge frequency, Law's features, and primitive length are also tested for cloud classification (Singh and Glennen, 2005).

The statistical features utilized in these works are basic and simplified descriptors. The abilities of these descriptors are more restricted since a certain amount of information is lost in the simplification process. In addition to the simple statistical features, we extract the local texture features using local binary patterns (LBPs) (Suruliandi et al., 2012). The texture information encoded by LBP forms higher dimensional feature vectors compared to traditional statistical features. Therefore, we perform dimension reduction on the extracted feature vector before performing classification.

Figure 1 illustrates the proposed system framework. An all-sky image is divided into blocks before the features are extracted. The existing works classified the clouds based on the entire scene. However, very often there are mixed cloud types in the scene of an all-sky image as can be observed in Fig. 2. Therefore, we divide the scene into blocks and perform classification based on the feature of each block. After block division, the system extracts statistical and texture features based on local patterns from each block. Then,

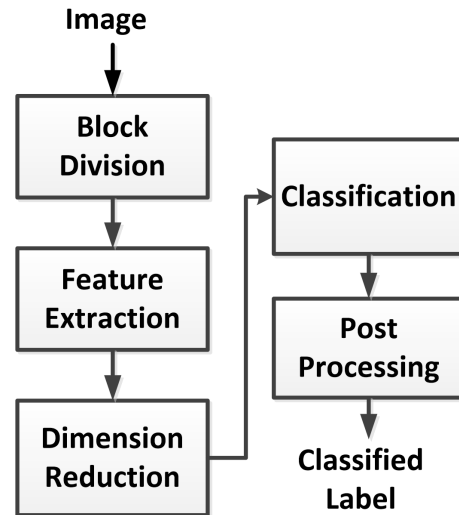


Figure 1. System framework.

principal component analysis (PCA) (Duda et al., 2001) is performed to reduce the dimensionality of the extracted feature vectors. For classification, we compare several classifiers, including k-NN, Bayesian classifier with regularized discriminant analysis (Cheng et al., 2010), and support vector machine (SVM) (Cristianini and Shawe-Taylor, 2000). In this work, the blocks are classified into cirrus, cirrostratus, scattered cumulus or altocumulus, cumulus or cumulonimbus, stratus, and clear sky. In the post-processing step, the classification results from the classifier are examined using the cloud-cover information. Furthermore, a voting scheme is proposed to summarize the classified label of the entire image from the class labels of all the blocks.

2 Data and methodology

This section outlines the data sources and samples as well as the methodology used for classification.

2.1 All-sky images

The all-sky images used in this research are captured by the all-sky camera manufactured by the Santa Barbara Instrument Group. The charge-coupled device is Kodak KAI-0340. The lens of the camera is Fujinon FE185C046HA-1. The focal length is 1.4 mm and focal ratio range is $f/1.4$ to $f/16$. The device covers a field of view of 185° . The RGB images are stored in bitmap format with resolution 640×480 . The data set is provided by the Industrial Technology Research Institute of Taiwan.

Figure 3 displays the six types of clouds on which the system will perform classification. Cirrus clouds and cirrostratus clouds are high and thin clouds. The main difference between cirrus clouds and cirrostratus clouds is that the

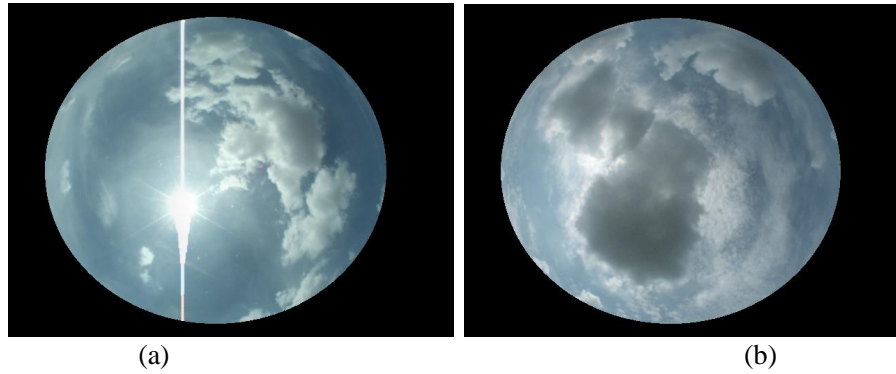


Figure 2. Conditions of mixed cloud types.

area of cirrostratus is larger. Altocumulus or scattered cumulus clouds are mid- to low-altitude clouds which look like blobs of cotton. Cumulus or cumulonimbus clouds are lower-altitude clouds which have noticeable vertical development and are often darker and larger. Stratus clouds are flat, wide-area clouds at lower altitude.

2.2 Block division

In practice, there might be more than one cloud type in one sky image, as shown in Fig. 2. In Fig. 2a, some cumulus clouds present in the scene, and there are some cirrostratus clouds around the sun area. In Fig. 2b, a cumulus cloud blocks the sun, and some altocumulus and cirrus clouds also exist in other regions of the image. Mixing up the features of cumulus, altocumulus, and cirrostratus clouds tends to confuse the classifier. Therefore, under such conditions of mixed cloud types, it is not appropriate to use the features of the entire image and classify the whole image as a certain cloud type. To solve this problem, we divide the entire scene into blocks and perform classification based on blocks. An example of the divided block is shown in Fig. 4 with block size 60×80 pixels. The feature vector of a block represents the characteristics of the cloud type in the block only. Such design will reduce the confusing conditions of mixing up features of different cloud types. Additionally, we can obtain more detailed information about the location of each cloud type. This information is very helpful since the clouds in the regions closer to the sun have higher impact on the irradiance changes.

2.3 Feature extraction

This work combines the statistical features proposed in the work by Heinle et al. (2010) and the distribution of local texture features using LBP codes (Suruliandi et al., 2012). The statistical features represent the spectral and texture information in a global view. On the contrary, the LBP codes encode the local characteristics of the gradient and texture features.

2.3.1 Statistical features

The statistical feature vector used in the work by Heinle et al. (2010) includes statistical spectral features and statistical textual features. The statistical spectral features include the following dimensions: mean of R components, mean of B components, standard deviation of B component, skewness of B component, and differences of $R-G$, $R-B$, and $G-B$ components. The statistical textual features are statistical measures computed from GLCM (Haralick et al., 1973), including energy, entropy, contrast, and homogeneity of the GLCM. Also, the cloud-cover ratio is considered as a feature. The details of these statistical features can be found in the work by Heinle et al. (2010).

2.3.2 Distribution of local texture features

In addition to the above-mentioned statistical features, we enhance the texture features by applying LBPs (Suruliandi et al., 2012). The $LBP_{P,R}$ code for a pixel (x_c, y_c) is defined in Eq. (1). In this equation, g_c denotes the grey-level value of the center pixel (x_c, y_c) , and g_p denotes the grey-level value of its neighboring pixel. The parameter P sets the number of neighboring pixels that are considered when computing the binary codes. The parameter R sets the distance between the center pixel and its neighbors. For $LBP_{8,1}$ codes, we consider the eight neighboring pixels whose distance with the center pixel is 1. The code represents the local texture characteristics around (x_c, y_c) .

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

$$s(g_p - g_c) = \begin{cases} 1 & g_p - g_c \geq 0 \\ 0 & g_p - g_c < 0 \end{cases} \quad (2)$$

For each pixel in the image, a P bit binary number is computed. When representing the LBP texture feature of a region using a feature vector, the convention is to construct an LBP

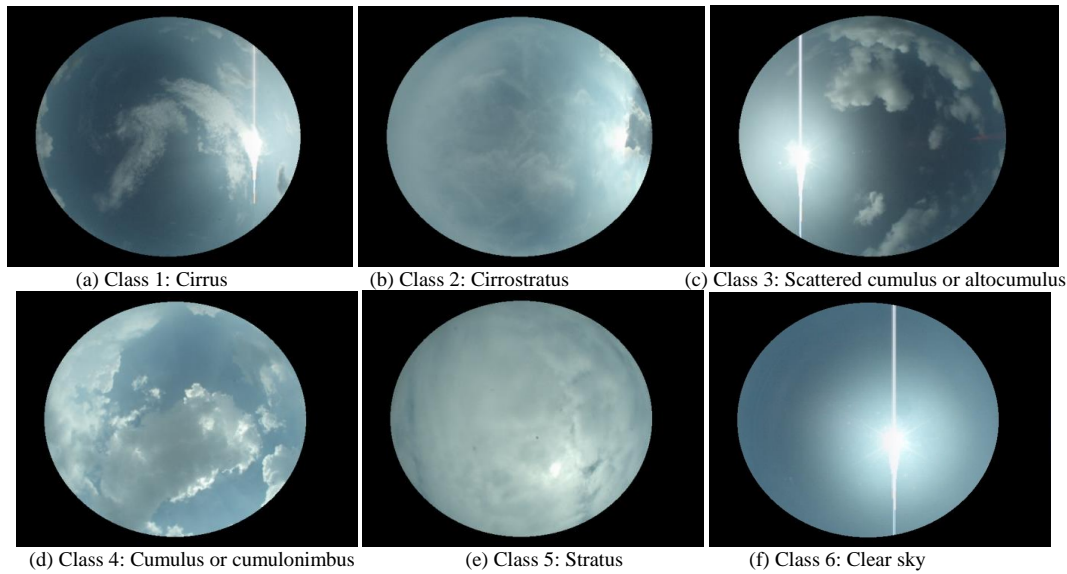


Figure 3. Six different types.

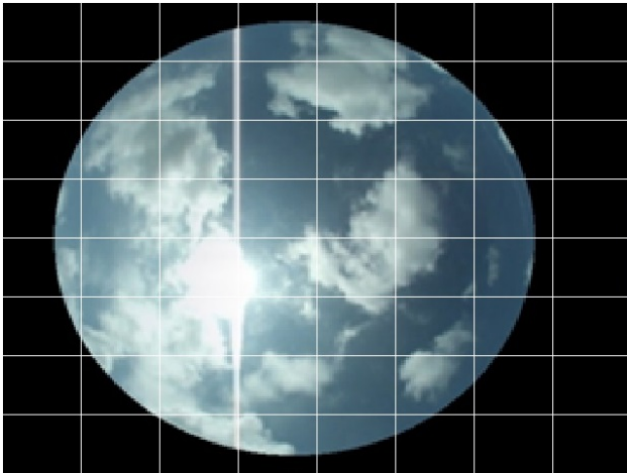


Figure 4. Block division example.

histogram by voting with the codes of all the pixels in the region. The LBP histogram characterizes the distribution of local texture features of the region.

We apply the $LBP_{P,R}$ codes with $P = 8$ and $R = 1$ to extract local texture features in this work. For $LBP_{8,1}$ codes, there are 256 distinct values since the code is an 8 bit binary number. Therefore, 256 histogram bins are required for all the distinct codes. However, it has been shown that some codes appear more frequently than others, concentrating the votes in the histogram in a few bins. The codes that appear with higher frequencies are the uniform LBP codes. Researches have shown that uniform LBP codes account for over 90 % of all LBP codes. The uniform LBP codes are the codes that have at most two zero-to-one or one-to-zero transi-

tions. Among the 256 distinct LBP codes, 58 LBP codes are uniform. As a consequence, we can use 58 bins for the uniform LBP codes and one extra bin for all the non-uniform LBP codes in the histogram. In total, the number of histogram bins is reduced to 59 instead of 256.

Because clouds of a certain type might be rotated, we further consider rotation invariant LBP code. To make the LBP code invariant to rotation, the code is circularly shifted to a minimum code number. In Eq. (3), $ROR(LBP_{P,R}, i)$ performs a circular bit-wise right shift on $LBP_{P,R}$ for i times. For rotation invariant LBP, there are nine uniform patterns. Therefore, only 10 bins are required for the histogram of uniform rotation invariant LBPs.

$$LBP_{P,R}^{RI} = \min\{ROR(LBP_{P,R}, i) | i = 0, 1, \dots, P-1\} \quad (3)$$

To obtain the distribution of the local texture patterns and to retain the localized information as well, we divide each block into N_{cell} cells when constructing the feature vector. One LBP histogram is generated for each cell. And then the N_{cell} histograms are concatenated to form the feature vector. In other words, for each image block, we generate a $59 \times N_{\text{cell}}$ dimensional feature vector for uniform LBPs and a $10 \times N_{\text{cell}}$ dimensional feature vector for uniform rotation invariant LBPs.

2.3.3 Combining statistical features and distribution of local texture features

The feature vectors described in Sect. 2.3.1 and 2.3.2 can be concatenated to obtain the combined feature vector. We denote combined feature A as the vector obtained by concatenating statistical features and uniform LBP histogram. We denote combined feature B as the vector obtained by concatenating statistical features and uniform rotation invariant

LBP histogram. Since the statistical feature vector has 12 dimensions, combined feature A and combined feature B have $12 + 59 \times N_{\text{cell}}$ and $12 + 10 \times N_{\text{cell}}$ dimensions, respectively.

2.4 Dimension reduction

PCA (Duda et al., 2001) is a commonly used way to reduce the dimensions of the feature vectors. To reduce the dependency among different feature dimensions, PCA seeks to find a set of new orthogonal bases to re-express the data more effectively. The new orthogonal bases, which are called principal components, are linear combinations of the original bases. Considering the variability in the data as an important and desired characteristic, PCA will preserve most of the data variability to the first (often few) principal components. Suppose that the original data set X has D_1 dimensions and there are N samples in the data set. The matrix X is a $D_1 \times N$ matrix whose columns are the original feature vectors. The PCA will select the first D_2 eigenvectors corresponding to the first largest D_2 eigenvalues of the matrix $X^T X$, which is proportional to the empirical sample covariance matrix of the original data set X . These D_2 eigenvectors define the principal component directions. Then the original data are projected on to the principal components to obtain the data with reduced dimensions in the new coordinate system. The criterion to select D_2 is usually based on the following equation. In Eq. (4), λ_k denotes the k th eigenvalue of the matrix $X^T X$. In other words, we preserve the first D_2 eigenvectors so that ratio between the sum of the absolute values of the first D_2 eigenvalues and the sum of the absolute values of all the eigenvalues is larger than a threshold Thr_{PCA} .

$$\frac{\sum_{k=1}^{D_2} |\lambda_k|}{\sum_{k=1}^{D_1} |\lambda_k|} > \text{Thr}_{\text{PCA}} \quad (4)$$

2.5 Classifiers

In addition to the basic k-NN classifier, this work also utilizes a Bayesian classifier with regularized discriminant analysis and a support vector machine in the experiments.

2.5.1 Bayesian classifier with regularized discriminant analysis

Given an unknown sample x the Bayesian classifier will classify it as the most probable class, ω_k , with the highest posterior probability, $P(\omega_k|x)$. According to Bayes' theorem, the posterior probability can be decomposed into several terms as shown in Eq. (5). In Eq. (5), the denominator is the probability of the sample $P(x)$, which does not depend on the class label and thus does not affect the decision process. The numerator includes the prior probability $P(\omega_k)$ and class-conditional probability $P(x|\omega_k)$. The prior probability

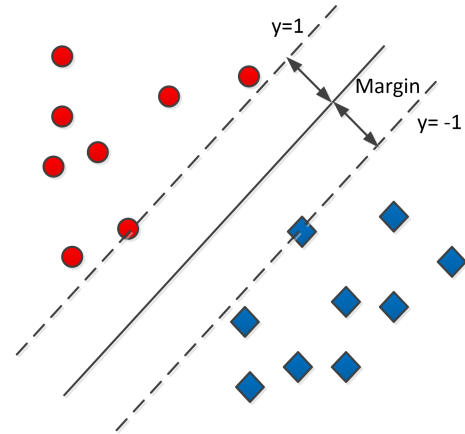


Figure 5. Decision boundary of support vector machine.

is the probability of observing a certain class before the feature of unknown sample x is taken into account. The class-conditional probability is learned from the training samples. It is usually modeled using Gaussian functions, as defined in Eq. (6). For simplicity, we can assume that all the classes have the same prior probabilities. It is also possible to set the prior probabilities according to the frequency of appearance of each class in the training data set.

$$P(\omega_k|x) = \frac{P(\omega_k)P(x|\omega_k)}{P(x)} \quad (5)$$

$$P(x|\omega_k) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)\Sigma_k^{-1}(x-\mu_k)^T} \quad (6)$$

To model class-conditional probabilities as Gaussians, we need to estimate the parameters of the Gaussians from the training data. Regularization techniques help reduce variance without adding too much model bias when estimating the parameters for high-dimensional data (Cheng et al., 2010). In eigenvalue decomposition regularized discriminant analysis (EDRDA) (Bensmail and Celeux, 1996), the covariance matrix Σ_k for the k th class is re-parameterized in terms of its eigenvalue decomposition $\Sigma_k = \alpha_k D_k \mathbf{A}_k D_k^T$, where $\alpha_k = |\Sigma_k|^{1/p}$ and D_k is the matrix of eigenvectors of Σ_k . The matrix \mathbf{A}_k is a diagonal matrix such that $|\mathbf{A}_k| = 1$ with the normalized eigenvalues of Σ_k on the diagonal in a decreasing order. By allowing each of the parameters α_k , \mathbf{A}_k , D_k to be either the same or different among different classes, eight discriminant models can be obtained. Furthermore, six more models are obtained by modeling the covariance matrix as a diagonal matrix or a scalar multiple of the identity matrix. More specifically, $\Sigma_k = \alpha_k \mathbf{B}_k$ leads to four more less complex models, where \mathbf{B}_k is a diagonal matrix with $|\mathbf{B}_k| = 1$. The models requiring the smallest numbers of parameters are to assume spherical shapes, i.e., \mathbf{A}_k is an identity matrix, which leads to model $\alpha_k I$ and model αI . Among the 14

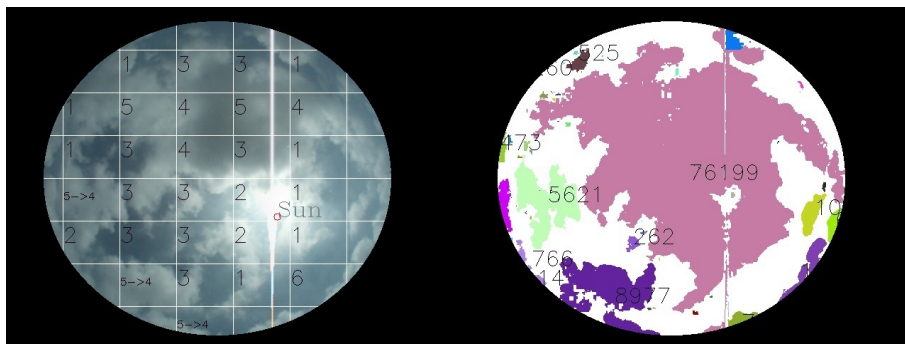


Figure 6. Example of correcting a stratus block as cumulus.

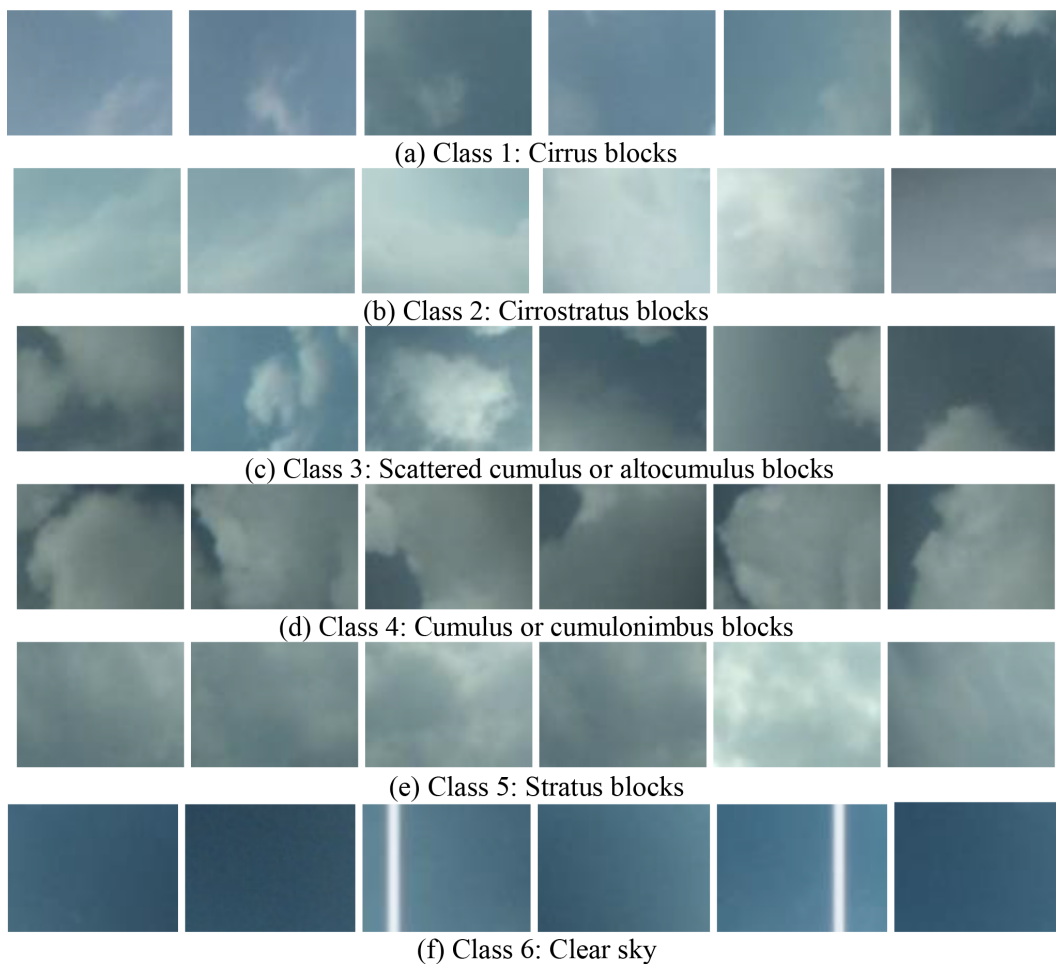


Figure 7. Example of training blocks.

models, there are nine models whose maximum likelihood (ML) estimation of the covariance matrix can be computed in closed form. For other models, the ML estimation needs to be computed through an iterative procedure. To accelerate the model selection process, this work only considers the nine EDRDA models that have closed-form solutions for ML parameter estimation.

2.5.2 Support vector machine

Given a set of training samples, the SVM will learn linear decision boundaries that maximize the margins between the decision boundaries and the training samples. Using a two-class case as an example, the margins are illustrated in Fig. 5. The intension is to lower the generalization error of

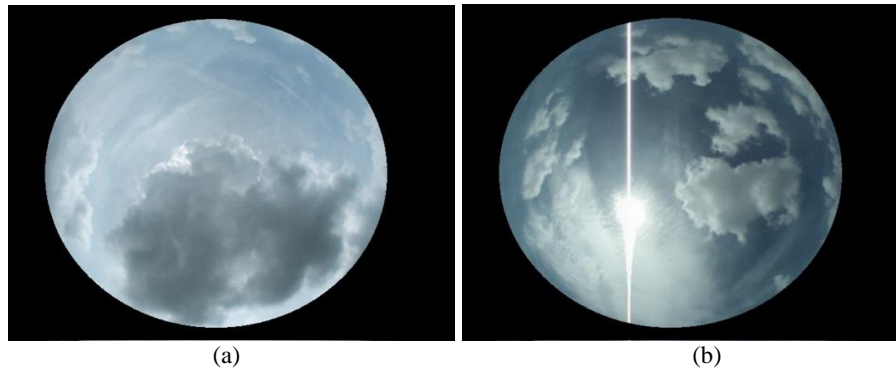


Figure 8. Examples of images with two ground truth labels.

the trained classifier with the large margins. The reason is that unseen testing samples may fall within the large margin and hopefully will still be correctly classified. To determine the hyper-plane that results in maximized margin, the support vector machine solves the quadratic programming optimization problem. Furthermore, to effectively handle non-linear separable data in the real world, the concept of soft margin and the usage of kernel functions are applied in the SVM. The details can be found in the work of Cristianini and Shawe-Taylor (2000). In this work, we apply SVM with radial basis functions as kernel functions.

2.6 Post-processing

In the process of block division, the important information of global cloud-cover percentage is inevitably lost. Therefore, we examine the classification result of each block in the post-processing step. Connected component analysis is performed on the cloud detection results. If a block is classified as stratus but the size of the cloud component is lower than the threshold, the label of the block is revised to cumulus. An example is shown in Fig. 6. The cloud detection result with connected component labeling is shown in Fig. 6b. Different connected components are illustrated in different colors. The numbers on each component denote the number of pixels in the component. We can observe that three blocks are re-labeled as cumulus. In our experiments, the threshold for revising the classification result is 12 000 pixels.

The subsequent application modules can utilize the classification result of each individual block with the knowledge of the location of the block. The classification results of all the blocks in an image can also be gathered to obtain a summarized label for the entire image. A simple way to summarize the labels in an image is to perform voting. From the classification results in Fig. 6, we have the knowledge that there are more votes for class 3 than other classes in this all-sky image.

3 Experiments and discussions

In this section, we report experimental results and discuss the performance of the proposed block-based cloud classification framework. For training purposes, we select 1800 blocks from the images and manually label the ground truth of these blocks. Selected training blocks for the six classes are shown in Fig. 7. Note that the block size used in our experiments is 60×80 pixels. We manually classified the ground truth of 3000 images in the data set in order to calculate the summarized classification accuracy for whole images. Since there are mixed cloud conditions in many images, each image can be associated with at most two ground truth labels. For a mixed cloud type image, the voting result is considered correct if the classified label matches any of the two ground truth labels. Figure 8 displays some examples of images that are associated with two ground truth labels. Figure 8a is labeled as both class 2 and class 4. Figure 8b is labeled as both class 1 and class 3. Due to the privacy issue of the data provider, we use a mask on the image to eliminate the surrounding buildings. The experiment data set includes all-sky images from 08:30 to 15:30 (UTC + 8 h). Therefore, the data set does not include the cases when the sun is close to the mask limits.

To select the proper threshold Thr_{PCA} for dimension reduction, we plot the accuracy using different Thr_{PCA} in Fig. 9. We use the 1800 blocks with ground truth labels and perform 10-fold cross validation (CV) when conducting this experiment. Note that the CV accuracy in Fig. 9 is based on the classification result of the Bayesian classifier. Both PCA and non-centered PCA (Cadima and Jolliffe, 2009) are considered in our experiment. The classification accuracy of applying PCA is higher than applying non-centered PCA. We observe that when Thr_{PCA} ranges from 93 to $\sim 94\%$, the CV accuracy is higher for both uniform LBPs and combined feature A. Therefore, we select $\text{Thr}_{\text{PCA}} = 93\%$ for uniform LBPs and combined feature A in the rest of the experiments. According to Fig. 9, we select $\text{Thr}_{\text{PCA}} = 95\%$ for uniform rotational invariant LBPs and combined feature B. In Fig. 9, when Thr_{PCA} equals 100%, it is equivalent to not applying

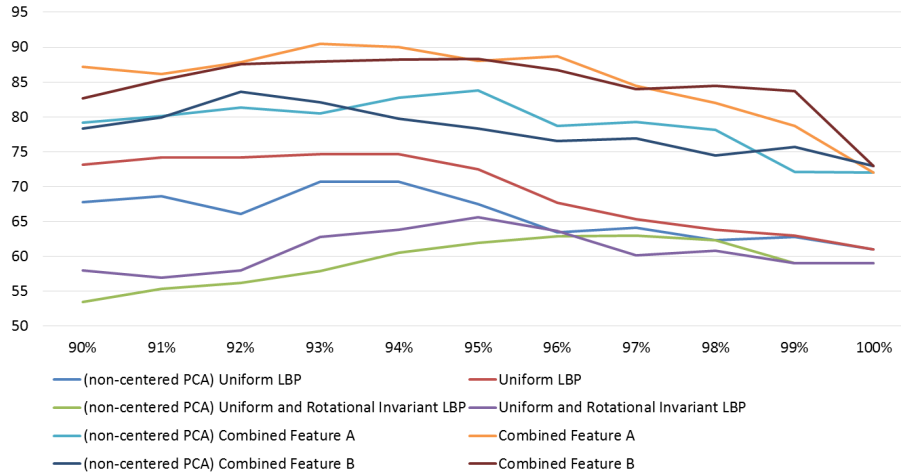


Figure 9. Selection of the threshold Thr_{PCA} for dimension reduction.

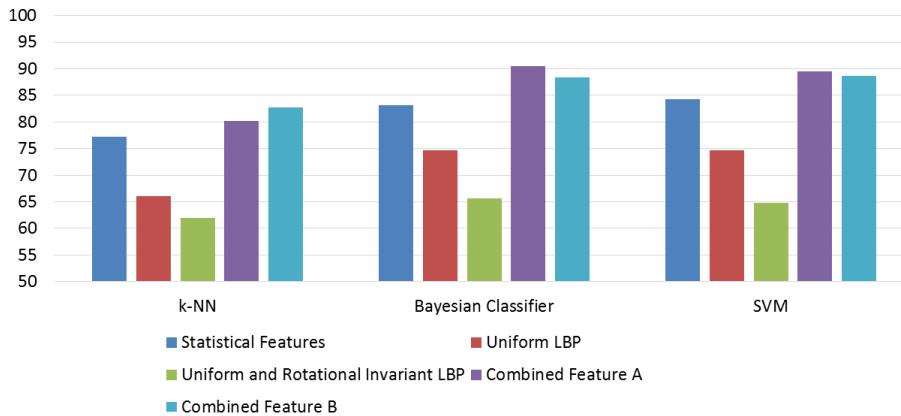


Figure 10. Classification accuracy on blocks using different feature and classifier combinations.

dimensionality reduction. For combined feature A and combined feature B, the advantage of applying PCA is more obvious since the dimensionality is higher. The statistical feature vector has only 12 dimensions. Therefore, there is no need to apply PCA on the statistical feature vector.

To compare the effect of various features and classifiers, Fig. 10 shows the 10-fold cross-validated classification accuracy on the 1800 blocks with ground truth labels using different features and classifiers. Compared with the statistical features and k-NN classifier used in the work by Heinle et al. (2010), the proposed combined features with Bayesian classifier or SVM demonstrate higher classification rates. It is clear that local texture feature alone does not perform better than statistical features. However, when combined with statistical features, additional information provided by distribution of local texture features can significantly improve the classification accuracy. We can observe that combined feature A slightly outperforms combined feature B when using the Bayesian classifier and SVM. Although intuitively we think that features with rotation invariant characteristics

should be preferable for cloud classification, combined feature A performs slightly better in practice. It might be due to the small dimensionality of combined feature B. Overall, the method using combined feature A and the Bayesian classifier with regularized discriminant analysis has the highest cross-validated classification accuracy in our experiments. Figure 11 displays selected classification results using combined feature A and the Bayesian classifier with regularized discriminant analysis. Although there are inevitably some misclassified blocks, most blocks are correctly classified in Fig. 11. Note that classification labels are not displayed on incomplete blocks and the block where the sun resides in Fig. 11.

To observe the advantage of block-based classification, Fig. 12 shows the classification accuracy on the 3000 images in the data set with and without the block voting scheme. In this experiment, the classifier is Bayesian classifier. Since features from mixed cloud conditions will not be mixed up in a single feature vector, the classification rates using block voting schemes are higher. Moreover, another advantage of

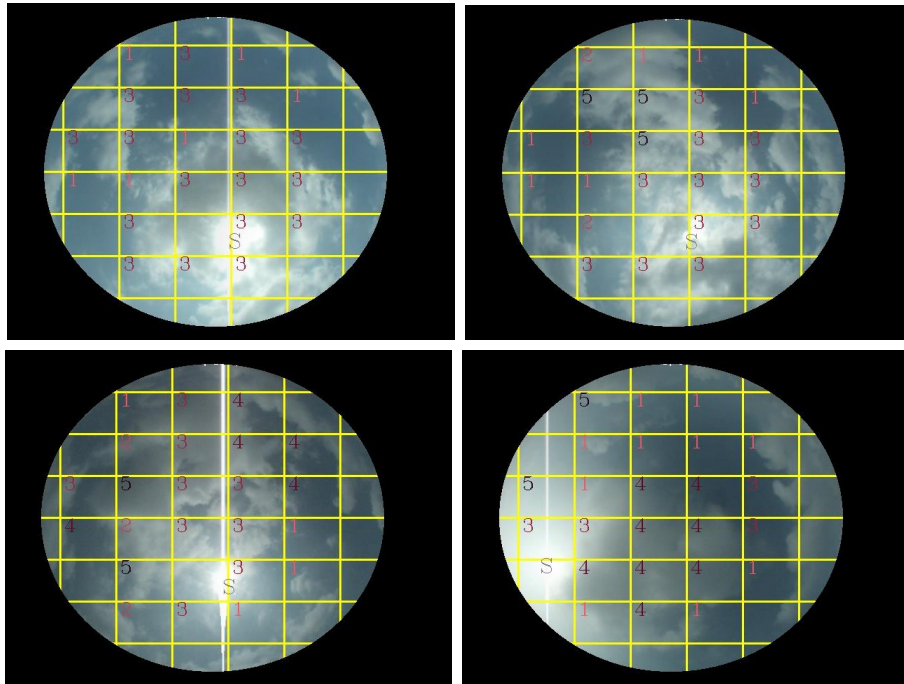


Figure 11. Selected classification results.

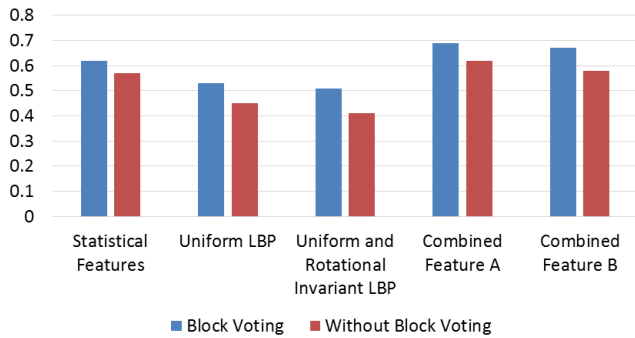


Figure 12. Comparison of whole-image classification and block-based classification with voting scheme.

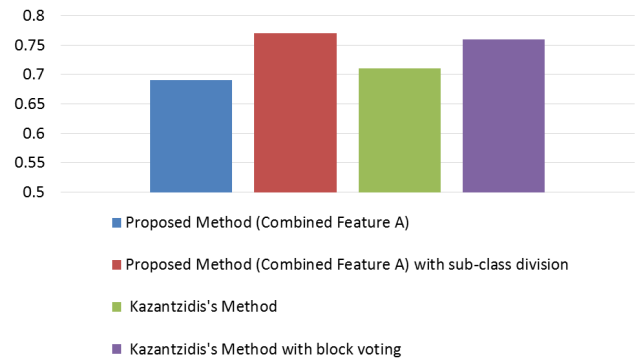


Figure 13. Comparison of classification results of different methods on the effect of subclass division and block voting.

block-based classification is that the classification result of each individual block with the knowledge of the block location can be utilized by subsequent application modules.

We perform an experiment to compare the proposed method with the method of Kazantzidis et al. (2012). The method of Kazantzidis et al. outperforms the proposed work with the concept of subclass division. In addition to comparing the method proposed by Kazantzidis et al., we also apply the concept of subclass division in our framework in the experiment. Since we have the information of the source image from which a training or testing block is selected, we could obtain the information needed to separate a block into subclasses. The subclasses are divided according to the solar zenith angle, global cloud coverage of the all-sky image,

and the visible fraction of the solar disk. In Fig. 13 we can observe that applying subclass division is indeed helpful for improving the classification accuracy. However, we perform an experiment on applying block-based classification using the features in the work of Kazantzidis et al. In the work of Kazantzidis et al., the existence of raindrops is used as a feature in the feature vector. The existence of raindrops is based on the indicator of the image in which the block resides in the case of applying block-based classification. As shown in Fig. 13, applying block-based classification with the voting mechanism is also helpful for improving the classification accuracy in this case. It can be observed that using combined statistical features and distribution of local texture features,

block voting, and subclass division would yield the best result.

4 Conclusions

Cloud classification is an important task for improving short-term solar irradiance prediction since different types of clouds have different effects on the change of solar irradiance. In this work, an automatic cloud-classification method for all-sky images is proposed. The classification is performed based on fixed-size blocks in the all-sky images. In addition to the statistical features in the literature, we combine the histogram of local texture patterns in the feature vector. With more discriminate features provided by local texture patterns, the proposed combined feature can improve the classification accuracy. Replacing k-NN classifier with more sophisticated supervised learning methods can further enhance the recognition results. Bayesian classifier with regularized discriminant analysis outperforms other classifiers on this data set in our experiments. This work also compares the classification accuracy with and without the voting scheme. With block-based classification and the voting scheme, the classification results on images with mixed cloud type conditions were shown to be better. Although the global cloud coverage feature is lost in the block-based feature extraction process, the global cloud coverage information can still be used to divide the data set into subclasses, as suggested in the work of Kazantzidis et al., to improve the classification accuracy of the proposed framework. For future work in component-based cloud classification, each detected connected component of cloud can be classified separately. In this way, the situation of mixed cloud types could be analyzed with even better precision and the information of the cloud coverage can be preserved. However, the performance of component-based classification would be highly dependent on the cloud detection accuracy. Therefore, current cloud detection methods need to be improved in order to lead to satisfactory component-based classification results. In addition to component-based cloud classification, another potential future work is to integrate the proposed cloud classification method in a short-term irradiance prediction system to obtain more accurate prediction results.

Acknowledgements. This work was supported in part by the Ministry of Science and Technology of Taiwan.

Edited by: V. Amiridis

References

- Bensmail, H. and Celeux, G.: Regularized Gaussian discriminant analysis through eigenvalue decomposition, *J. Am. Stat. Assoc.*, 91, 1743–1748, 1996.
- Cadima, J. and Jolliffe, I.: On relationships between uncentered and column-centered principal component analysis, *Pak. J. Statist.*, 25, 473–503, 2009.
- Calbo, J. and Sabburg, J.: Feature extraction from whole-sky ground-based images for cloud-type recognition, *J. Atmos. Ocean. Tech.*, 25, 3–14, 2008.
- Cheng, H. Y., Yu, C. C., Tseng, C. C., Fan, K. C., Hwang, J. N., and Jeng, B. S.: Environment classification and hierarchical lane detection for structured and unstructured roads, *IET Computer Vision*, 4, 37–49, 2010.
- Cristianini, N. and Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, Cambridge, England, 2000.
- Duda, R. O., Hart, P. E., and Stork, D. G.: Pattern classification, John Wiley & Sons, New York City, NY, USA, 2nd edn, 2001.
- Fu, C. L. and Cheng, H. Y.: Predicting solar irradiance with all-sky image features via regression, *Solar Energy*, 97, 537–550, 2013.
- Haralick, R. M., Shanmugam, K., and Dinstein, I.: Textural features for image classification, *IEEE Transactions on Systems, Man and Cybernetics*, 3, 610–621, 1973.
- Heinle, A., Macke, A., and Srivastav, A.: Automatic cloud classification of whole sky images, *Atmos. Meas. Tech.*, 3, 557–567, doi:10.5194/amt-3-557-2010, 2010.
- Kassianov, E., Long, C. N., and Ovtchinnikov, M.: Cloud sky cover versus cloud fraction: Whole-sky simulations and observations, *J. Appl. Meteor.*, 44, 86–98, 2005.
- Kazantzidis, A., Tzoumanikas, P., and Bais, A. F.: Fotopoulos S, Economou G, Cloud detection and classification with the use of whole-sky ground-based images, *Atmos. Res.*, 113, 80–88, 2012.
- Kubota, M., Nagatsuma, T., and Murayama, Y.: Evening corotating patches: A new type of aurora observed by high sensitivity all-sky cameras in Alaska, *Geophys. Res. Lett.*, 30, 1612, doi:10.1029/2002GL016652, 2003.
- Li, Z., Cribb, M. C., Chang, F. L., and Trishchenko, A. P.: Validation of MODIS-retrieved cloud fractions using whole sky imager measurements at the three ARM sites, Proc. 14th ARMScience Team Meeting, Albuquerque, NM, Atmospheric Radiation Measurement Program, 6, 2–6, 2004.
- Long, C. N., Sabburg, J., Calbó, J., and Pagès, D.: Retrieving cloud characteristics from ground-based daytime color all-sky images, *J. Atmos. Ocean. Tech.*, 23, 633–652, 2006.
- Martínez-Chico, M., Batlles, F. J., and Bosch, J. L.: Cloud classification in a mediterranean location using radiation data and sky images, *Energy*, 36, 4055–4062, 2011.
- Pfister, G., McKenzie, R. L., Liley, J. B., Thomas, A., Forgan, B. W., and Long, C. N.: Cloud coverage based on all-sky imaging and its impact on surface solar irradiance, *J. Appl. Meteorol.*, 42, 1421–1434, 2003.
- Singh, M. and Glennen, M.: Automated ground-based cloud recognition, *Pattern Anal. Appl.*, 8, 258–271, 2005.
- Suruliandi, A., Meena, K., and Reena Rose, R.: Local binary pattern and its derivatives for face recognition, *IET Computer Vision*, 6, 480–488, 2012.