Half-FFT Features and Hierarchical Algorithms for Cloud Type Classification Using Ground-Based Images

Thitinan Kliangsuwan

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering

Prince of Songkla University
2016

Half-FFT Features and Hierarchical Algorithms for Cloud Type Classification Using Ground-Based Images

Thitinan Kliangsuwan

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering

Prince of Songkla University

2016

Thesis Title Half-FFT Features and Hierarchical Algorithms for Cloud Type Classification Using Ground-Based Images

Author Mr. Thitinan Kliangsuwan

Major Program Computer Engineering

Major Advisor :

……………………………………..
(Asst.Prof.Dr. Apichat Heednacram)

Examining Committee :

………………………...Chairperson
(Dr. Noppon Lertchuwongsa)

……………………….....Committee
(Asst.Prof.Dr. Apichat Heednacram)

…………………………Committee
(Asst.Prof.Dr. Chatklaw Jareanpon)

The Graduate School, Prince of Songkla University, has approved this thesis as partial fulfillment of the requirements for the Doctor of Philosophy Degree in Computer Engineering.

…………………………….…….....
(Assoc.Prof.Dr. Teerapol Srichana)
Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate’s own investigations. Due acknowledgement has been made of any assistance received.

....Signature
(Asst.Prof.Dr. Apichat Heednacram)
Major Advisor

…..Signature
(Mr. Thitinan Kliangsuwan)
Candidate

I hereby certify that this work has not been accepted in substance for any degree, and is not being currently submitted in candidature for any degree.

…..Signature
(Mr. Thitinan Kliangsuwan)
Candidate

Thesis Title	Half-FFT Features and Hierarchical Algorithms for Cloud Type Classification Using Ground-Based Images
Author	Mr. Thitinan Kliangsuwan
Major Program	Computer Engineering
Academic Year	2015

ABSTRACT

The study of cloud type classification has played significant role in weather forecast, air traffic control, and global warming analysis. Therefore, being able to classify clouds accurately is important. This thesis proposes new features and algorithms for automatic cloud type classification system using ground-based images. The input to the system is the images of seven sky conditions, namely, cirriform clouds, high cumuliform clouds, stratocumulus clouds, cumulus clouds, cumulonimbus clouds, stratiform clouds, and clear sky. The output is a class name for an image which belongs to one of the seven sky conditions. There are four steps involved in developing the system. First, the suitable classifier is chosen from commonly used classifiers; and the well-performed classifier will be used in our system design and implementation. Second, different feature extraction techniques are compared with the *k*-FFTPX which is our newly proposed technique based on Fourier transform; and the most effective feature extraction technique will be used in the classification system. Third, more features and hierarchical classification technique are introduced to improve the system performance to achieve accuracy higher than 95%. Fourth, a complete cloud classification system is implemented in practice for local cloud monitoring at Prince of Songkla University, Phuket Campus. The best accuracy for classifying seven cloud types using hierarchical classification tree technique integrating with the meteorological data is as high as 99.82%. In addition, a mobile application is developed for online classification using the approach presented in the thesis. Live cloud image and live meteorological data from our station can be viewed through mobile application from anywhere in the world.

Keywords: pattern recognition, cloud classification, ground-based images

ACKNOWLEDMENT

First of all, I would like to thank my advisor, Asst.Prof.Dr. Apichat Heednacram, for his attention, guidance, patience, taking time, and great support all along during this research.

I would like to also express my gratitude to all lecturers who teach image processing and pattern recognition for knowledge and great suggestions.

I am grateful to Mr. Anupong Rongmak and Mr. Aekawat Wachiaphan for his kind assistance with the installation of weather station and cloud station.

I would like to thank all lecturers and staffs of Department of Computer Engineering for guidance, consulting, assistance, and supporting various devices in this research.

I especially acknowledge the scholarship of Faculty of Engineering for financially supporting my PhD research.

Finally, I am greatly thankful to my family and parents who are encouraging and driving force for me to skip through difficult times until the research is completed.

Thitinan Kliangsuwan

CONTENT

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

LIST OF TABLES

Table **Page**

Table **Page**

Table **Page**

LIST OF FIGURES

Figure **Page**

Figure **Page**

LIST OF ABBREVIATIONS

TSI	Total sky imager
WSC	Whole sky camera
FOV	Field of view
PNN	Probabilistic neural network
MSVM	Multi-category support vector machine
SVD	Singular value decomposition
WSI	Whole-sky imager
k-NN	*k*-nearest neighbor
SLBP	Salient local binary pattern
MLP	Multilayer perceptron
BoMs	Bag of micro-structures
SVM	Support vector machine
IHS	Illuminant-Hue-Saturate
CCT	Color census transform
ELM	Extreme learning machine
RGB	Red-Green-Blue
HSV	Hue-Saturation-Value
RGB2HSV	Transformation of RGB to HSV color model

FFT	Fast Fourier transform
WMO	World Meteorological Organization
GLCM	Grey Level Co-occurrence Matrices
ANN	Artificial neural network
LOOCV	Leave-one-out cross-validation
2D-FFT	Two dimensional fast Fourier transform
k-FFTPX	Fast Fourier Transform Projection on the *x*-axis
CC	Correlation with clear
SI	Spectral intensity
DWT	Discrete wavelet transform
DTCWT	Dual-tree complex wavelet
LBP	Local Binary Patterns
CA	Coefficient approximated
CH	Coefficient horizontal edge
CV	Coefficient vertical edge
CD	Coefficient diagonal edge
CCTA	Cloud Classification Tree Algorithm
Half-FFT	Features from the half of FFT image

CHAPTER 1

INTRODUCTION

1.1 Introduction and Motivation

Weather conditions have received increasing attention in recent years because severe weather has caused damages in many countries around the world. In 2012, the destruction of hurricane Sandy in the United States caused serious damages; at least 159 people have died, more than 650,000 homes and hundreds of thousands of businesses were severely damaged [1]. Therefore, the study of weather conditions is vital in help preparing for incoming natural disasters and extreme weather changes such as storm, lightning, heavy rain, landslides and flash flood. This is because it helps to reduce the losses that may occur. Analyzing weather conditions involves essential elements of celestial phenomena and clouds. The appearance of each cloud type changes upon the weather conditions at the time and in the near future. Therefore, we can recognize weather conditions using cloud classification. Nowadays, with the help of image processing, weather conditions can be monitored without any use of sophisticated instruments, even in the remote or rural areas. Hence, cloud classification is worth exploring.

Automatic cloud classification has been highly appreciated over the last decades [2]–[16] because the traditional cloud type classification requires specialists to do it manually. Nevertheless, the number of specialists in this area is limited. Moreover, this approach lead to high cost, speed of manual classification is also slowly and human errors are occurring. Unique shape and disorder of clouds can lead to different results for different observers. Moreover, the observation of cloud requires experts and there is limited number of experts in the area. Therefore, automatic cloud classification has been developed since 1977 [17] to serve the purpose of auto-recognition.

Satellite images have been used as input in the analysis of cloud classification. It can provide information and overview but is unable to provide local details on a specific area. Furthermore, satellite images are sometimes expensive, and they are limited for public use [10], [13]. Thus, ground-based images which obtained by imager devices have been utilized more extensively. Two types of imagers are developed [18], namely, the total sky imager (TSI) and the whole sky camera (WSC). Both of the imagers take an image with wide angle of more than 160° field of view (FOV) that can cover large area of sky. However, these devices are used for commercial and they are expensive which may not be suitable for small research groups. Therefore, a normal digital camera is introduced. Although, this type of camera provides a fraction of sky image, it has more specific information needed, low cost, and easier acquisition.

In this thesis, we will develop an automatic ground-based cloud type classification using digital camera. Image processing and pattern recognition technique are used. Seven different sky conditions are considered in the classification, namely, cirriform clouds, high cumuliform clouds, stratocumulus clouds, cumulus clouds, cumulonimbus clouds, stratiform clouds, and clear sky. The characteristics and the sample images of these cloud types will be explained in details in Chapter 2.

1.2 Literature Review

1.2.1 Satellite Images

Several researchers use satellite images as an input for automatic cloud type classification. Lee *et al*. [2] presented a classification of sub-regions into one of three cloud types using a neural network with texture features. The overall accuracy of classification is 93%. Heinzmann [3] used fuzzy logic approach to classify four cloud classes in form of cloud cover percentage. Bankert [4] suggested a probabilistic neural network (PNN) with spectral, textural, and physical measures for classifying each region into one of ten cloud classes which achieve an accuracy of 79.80%. Aha and Bankert [5] provided feature selection algorithms for classifying ten cloud classes using IB1. The best accuracy here is 88%. Fan *et al*. [7] used a bispectral cloud classification method based on man-computer interactive way to classify six types of clouds, land, and water through looking up table. This method provides an accuracy of 87.10%. Ambroise *et al*. [8] used distribution of pixels with hierarchical clustering to classify nine cloud types. The method achieves an accuracy of 63%. Lee *et al*. [9] applied multi-category support vector machine (MSVM) to classify radiance profiles as one of three cloud classes. The result of classification is 90.11%. Shangguan *et al*. [12] studied texture feature analysis combined with Variational theory to extract texture features. Kaur and Ganju [14] used singular value decomposition (SVD) to extract the salient spectral and textural features to classify clouds based on their heights. This method has an accuracy of 70-90%. All the papers mentioned above used satellite images.

1.2.2 All Sky Images

Recently, ground-based images have been used increasingly in cloud classification. Buch *et al*. [6] used images from two whole-sky imagers (WSIs) to create three-dimensional volume. Binary decision trees with three groups of features (texture measures, position information, and pixel brightness) are used to classify each pixel into one of five sky conditions. The accuracy of classification is 61%. Calbó and Sabburg [13] used images from two ground-based imagers, namely, TSI and WSC. They applied parallelepiped technique with three types of features (texture, Fourier transform, and cloudy pixels). When classifying eight sky conditions, it yields accuracy of 62% and is increased to 76% when classifying five sky conditions. Heinle *et al*. [15] used whole sky images as an input for cloud type classification. Their *k*-nearest neighbor (*k*-NN) classifier applied 12 features from spectral features and textural features to classify seven sky conditions. The accuracy classification is as high as 97% but in general case with unseen data, the accuracy is between 75% and 88%. Martínez-Chico *et al*. [16] used radiation data and images from TSI to classify clouds according to their heights. The result was shown as the frequency of occurrence for each class. Afterwards, there were two papers based on the previous work of Heinle *et al*. [15]. First, Kazantzidis *et al*. [19] proposed method to detect raindrops for feature extraction. The average of classification is 87.9%. Second, Liu *et al*. [20] proposed the new feature method called the salient local binary pattern (SLBP). Their accuracy classified by the nearest neighborhood using chi-square metric is 93.65%. Taravat *et al*. [21] used pixel values of the whole-sky images to classify pixels into either cloud coverage or others (cloud-free and sun). They used Multilayer perceptron (MLP) neural networks in the classification and the overall accuracy is 95.07%. Cheng and Yu [22] used block-based classification on all-sky images to classify six sky conditions. In each block, statistical texture features and local binary pattern are extracted. Then, these features are classified using Bayesian classifier which gives the accuracy of 90%. Li *et al*. [23] proposed new method named bag of micro-structures (BoMs) for classifying five sky conditions using s support vector machine (SVM) classifier. Their result yields an accuracy of 90.9%.

1.2.3 Digital Images without Fisheye Lens

The input images captured from digital cameras have received high attention from several researchers since around 2005. Singh and Glennen [10] classified five sky conditions using *k*-NN and neural network classifiers with five different feature extraction methods. The best result has an accuracy of 64%. Souza-Echer *et al*. [11] presented the new algorithm based on a criteria decision process on Illuminant-Hue-Saturate (IHS) space to classify each pixel as clear sky, cloudy or undefined using parallelepiped method. This method estimates the percentage of sky cloud coverage, the output yields better than 94% for classifying clear sky and better than 99% for classifying cloudy sky. Most recently, Zhuo *et al*. [24] introduced color census transform (CCT) and automatic block assignment method. Then, texture and structure information are extracted in every block and are concatenated as a feature vector. SVM classifier is used to classify six sky conditions and the result is 81.17%. Xia *et al*. [25] presented texture features, color features, and shape features which performed with a hybrid method based on *k*-NN and extreme learning machine (ELM). The overall accuracy of classifying four sky conditions is 84.82%. Wu *et al.* [26] used ELM classifier to classify four sky conditions. By combining three features, namely, texture features, color features, and SIFT features, it provides accuracy is 86.64%.

Table 1.1. Summary of literature survey on existing methods using satellite images.

Year	**Proposed method**	**Classifier**	**No. class**	**Accuracy (%)**	**Author**
1990	Texture features	Neural network	3	93	Lee *et al.* [2]
1994	Spectral, textural, and physical measures	PNN	10	79.80	Bankert [4]
1994	Feature selection algorithms	IB1	10	88	Aha and Bankert [5]
1997	Bispectral cloud classificatio n method	Look up table	8	87.10	Fan *et al.* [7]

Year	**Proposed method**	**Classifier**	**No. class**	**Accuracy (%)**	**Author**
2000	Distribution of pixels	Hierarchical clustering	9	63	Ambroise *et al.* [8]
2004	Radiance profiles	MSVM	3	90.11	Lee *et al.* [9]
2008	SVD	Unclear	3	70-90	Kaur and Ganju [14]

Table 1.2. Summary of literature survey on existing methods using all sky images.

Year	**Proposed method**	**Classifier**	**No. class**	**Accuracy (%)**	**Author**
1995	Texture, position info, and pixel brightness	Binary decision trees	5	61	Buch *et al.* [6]

Year	**Proposed method**	**Classifier**	**No. class**	**Accuracy (%)**	**Author**
2008	Fourier transform, texture, and cloudy pixels	Parallelepiped technique	5	76	Calbó and Sabburg [13]
2010	Spectral and textural features	*k*-NN	7	75-88	Heinle *et al*. [15]
2012	Existence of raindrops	*k*-NN	7	87.90	Kazantzidis *et al*. [19]
2013	SLBP feature	Nearest neighbor	7	93.65	Liu *et al*. [20]
2015	Pixel values	MLP neural network	2	95.07	Taravat *et al*. [21]
2015	Block-based classification	Bayesian	6	90	Cheng and Yu [22]

Year	**Proposed method**	**Classifier**	**No. class**	**Accuracy (%)**	**Author**
2016	BoMs	SVM	5	90.90	Li *et al.* [23]

Table 1.3. Summary of literature survey on existing methods using digital images.

Year	**Proposed method**	**Classifier**	**No. class**	**Accuracy (%)**	**Author**
2005	Five feature extraction methods	Neural network and *k*-NN	5	64	Singh and Glennen [10]
2006	Criteria decision process on IHS	Parallelepiped method	2	94-99	Souza-Echer *et al.* [11]
2014	CCT and automatic block assignment	SVM	6	81.17	Zhuo *et al.* [24]

Year	**Proposed method**	**Classifier**	**No. class**	**Accuracy (%)**	**Author**
2015	Texture, color, and shape features	ELM and *k*-NN	4	84.82	Xia *et al.* [25]
2015	Texture, color, and SIFT features	ELM	4	86.64	Wu *et al.* [26]

From the literature surveys in Table 1.1 – 1.3, the texture features are commonly used for feature extraction. However, the classification accuracy using only texture features is not very high; therefore, they must be combined with other features to achieve higher accuracy. Although some result using a digital camera shows the high accuracy of 94%, the number of cloud classes being considered is only two.

In this research, we will use a digital camera and the number of cloud classes being considered is increased to seven. Our cloud classification system for the seven cloud classes is aimed to have accuracy higher than 95%. We will begin our experiment by finding a set of suitable texture features. Then, using these features we will compare the performance of two classifiers, namely, *k*-nearest neighbor and artificial neural network. We will show that artificial neural network performs better than *k*-nearest neighbor. In the second experiment, we will find a suitable feature extraction technique that is effective for cloud type classification. We will also develop a novel feature based on Fourier transform to enhance the accuracy of classification. In the third experiment, we will introduce three new features based on Fourier transform and new classification tree algorithm to enhance the accuracy even further. Finally, a complete cloud classification system will be constructed.

1.3 Objective

1.3.1 To select a suitable classifier from the two commonly used classifiers, namely, *k*-nearest neighbor and artificial neural network.

1.3.2 To select a suitable feature extraction technique that is effective for cloud type classification.

1.3.3 To propose novel features and algorithms that are different from the previous works.

1.3.4 To propose a complete cloud classification system for classifying seven cloud types using ground-based images which has accuracy higher than 95%.

1.3.5 To implement hardware system for capturing cloud images and to build a low-cost local cloud monitoring station.

1.4 Scope

To study and to design automatic cloud-type classification system using digital camera for classifying seven sky conditions. The color images have a resolution at least 640 × 480 pixels in Red-Green-Blue (RGB) and JPEG format and they are taken in daylight. There are four steps of system development. In the first three steps we will use simple digital images without fisheye lens.

The first step is to test a less-sophisticated system using texture features in order to find a suitable classifier to be used in the final complete system. An experiment will be conducted to compare two classifiers between *k*-nearest neighbor and artificial neural network.

The second step is to conduct more experiments in order to test different feature extraction techniques with our proposed feature based on Fourier transform. The most effective feature extraction technique will be selected for later cloud type classification.

The third step is to use the suitable classifier and suitable features to construct a hierarchy classification tree for high–accuracy classification.

The final step is to integrate and implement hardware system (see Figure 1.1) to capture cloud images with 170° FOV fisheye lens every 5 minutes. The fisheye lens is used in the final step to enhance the system accuracy as it can cover larger area in the sky. A low-cost local cloud monitoring station is built and installed at Prince of Songkla University, Phuket Campus. Moreover, a mobile application will be developed for online classification and for mobile users to view live images from our station.

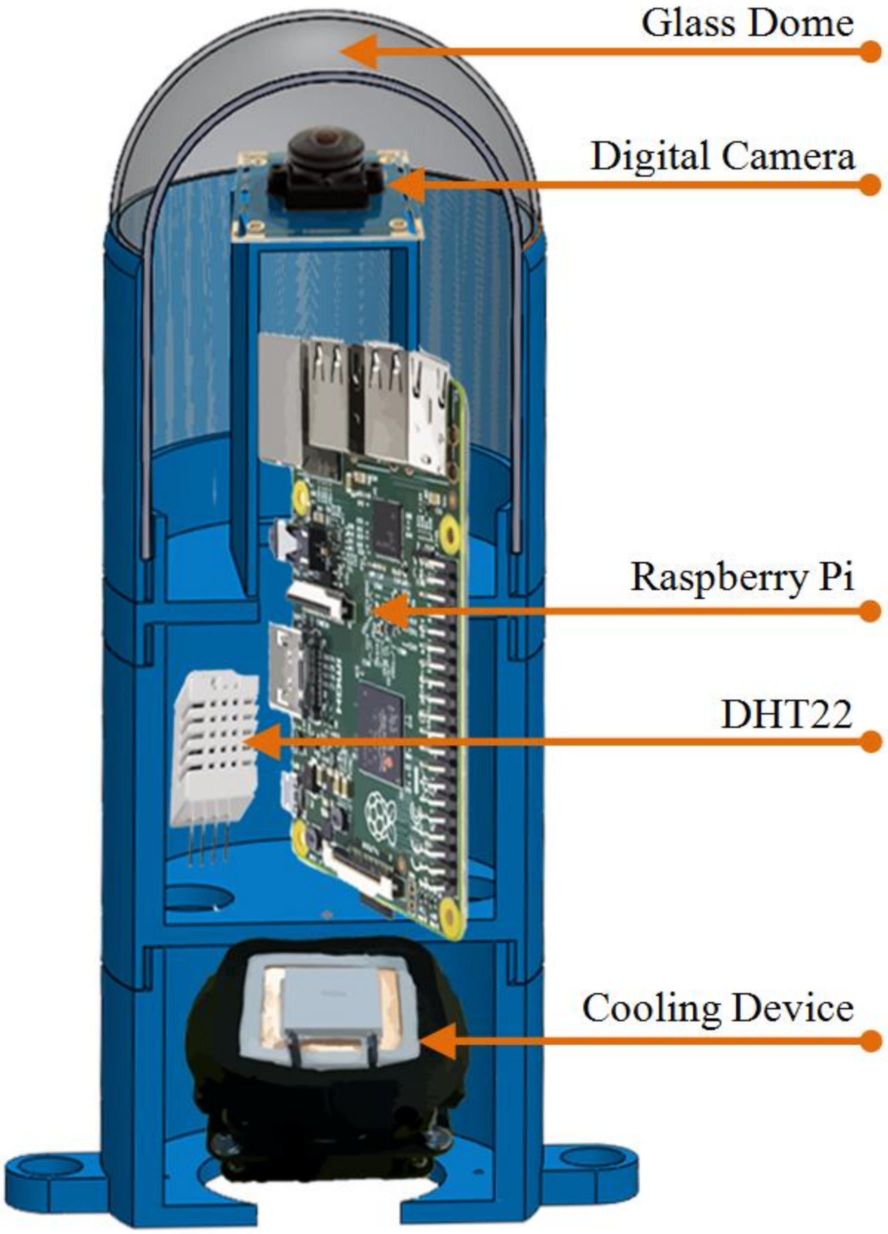


Figure 1.1. Hardware architecture.

In Figure 1.1, the hardware architecture for capturing whole sky images has 5 main components. The first component is a glass dome for protecting camera from the rain and other particles. The second component consists of digital camera and fisheye lens for capturing color image 170° FOV with a resolution of 2048 × 1536 pixels in JPEG format. The third component is a Raspberry Pi 2 Model B for controlling system such as capture images and control cooling device. The fourth component is a humidity and temperature sensor named DHT22. Finally, we place cooling device for controlling heat inside the hardware, it will run when a temperature inside is more than 30°C.

1.5 Benefit

1.5.1. The research proposes several novel features for cloud type classification with our own hardware implementation that is low-cost and operational as a complete system.

1.5.2. The research proposes several algorithms for cloud type classification and can be applied to various applications by changing the user-expected accuracy or changing the number of cloud classes being considered.

1.5.3. Being able to classify cloud types more accurately can lead to the more accurate weather prediction and other applications that are related to metrological events.

1.5.4. The research can be further extended to predict cloud type which is related to earthquakes. A recognition of earthquake clouds is interesting and it is currently under study by Nanyang Normal University [27], [28], Institute of Geography and Natural Resource [28], and others [29]–[34].

1.6 Overall System

In Figure 1.2, the input image to our system will be in RGB color. Then, the image will pass to cloud classification algorithm and return the output class which belongs to one of the seven sky conditions. Each stage is explained in more details as follows.

In the preprocessing step, the image resolution is scaled for a suitable size. Channel splitting is used to convert an RGB image to a grayscale image. Moreover, the transformation of RGB to Hue-Saturation-Value (HSV) color model (RGB2HSV) is used before channel splitting of HSV.

Then, we use a binary mask image for segmentation. A segmentation of each pixel is made by multiplying a pixel value of image with a mask pixel in the same position; the result is set to zero if the pixel value is multiplied by zero, otherwise the result is equal to the pixel value itself if it is multiplied by one. By removing the unwanted parts such as buildings and trees, the output of segmentation will give only clouds and sky.

There are several feature extraction methods. We extract the features from image using relation of pixels in image. Some feature is not suitable for identifying cloud types. Therefore, after the feature extraction we also need feature selection algorithm for selecting suitable features.

The classification is performed based on the extracted features. There are two parts involved in this step which are the training and the testing. The training process is used to construct a classifier model by machine learning algorithm using feature vectors together with their pre-defined classes (label vectors).

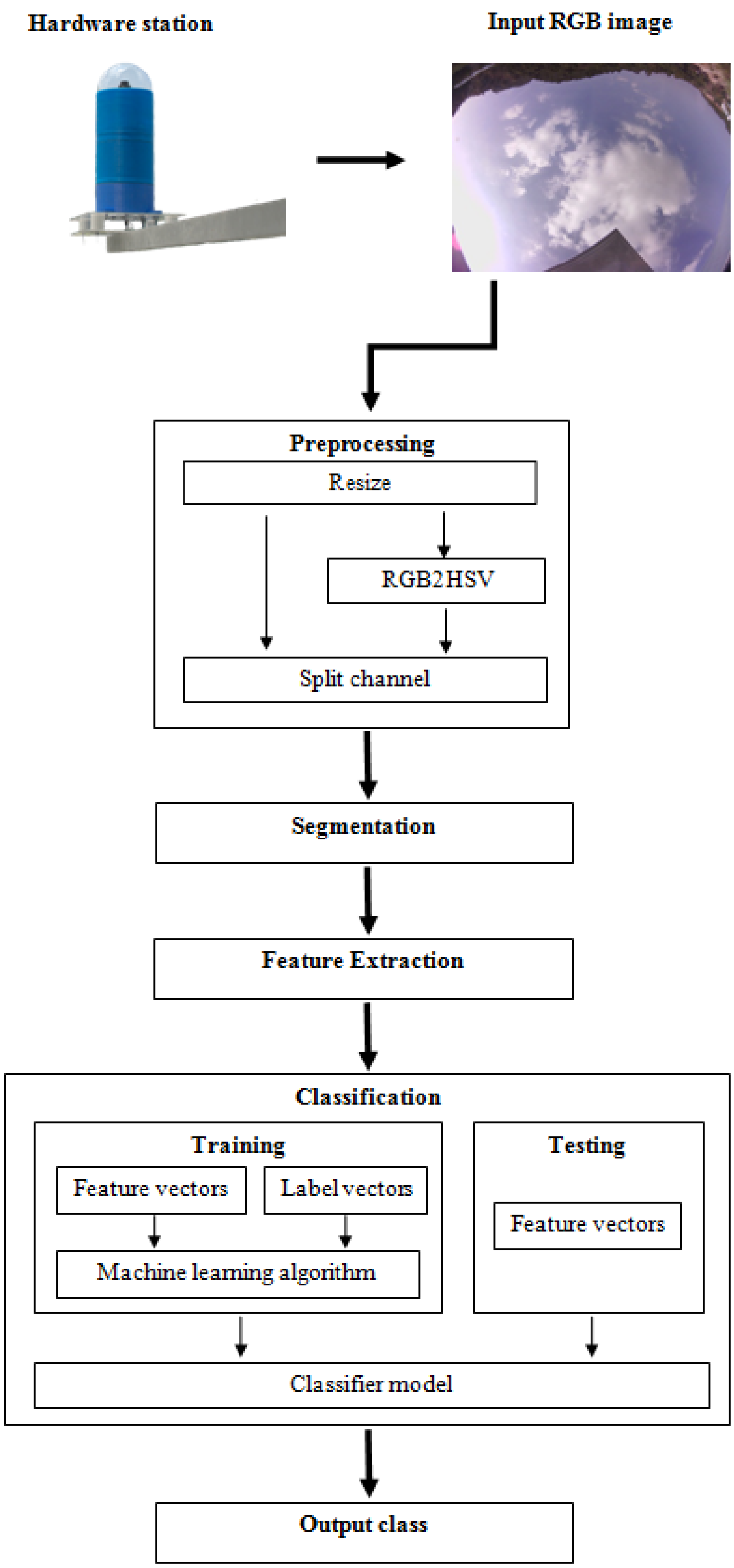


Figure 1.2. System overview of proposed cloud classification.

Then, the classifier model is tested by utilizing feature vectors in the testing process. We use artificial neural network and *k*-nearest neighbor as our classifiers; we then compare the results from both classifiers and choose the best one. Our classification system is expected to perform better than the existing one in the literatures.

Finally, the system will be tested and integrated. The automatic cloud classification system will be implemented and installed at Prince of Songkla University, Phuket campus.

1.7 Organization of the Thesis

The organization is as follows:

- This chapter described introduction and motivation, objective, scope and literature review.
- Chapter 2 introduces the theoretical background and related algorithms.
- Chapter 3 describes an experiment for finding a suitable classifier.
- Chapter 4 discusses a series of experiments to select a suitable feature extraction technique.
- Chapter 5 presents three novel fast Fourier transform (FFT) features and a complete cloud classification tree algorithm.
- Chapter 6 explains the system integration test and the hardware installation.
- Finally, Chapter 7 concludes our works and contributions of the thesis and also suggests possible future work.

CHAPTER 2

THEORETICAL AND PRINCIPLE

This chapter presents theoretical background that is important for readers to understand each element in the cloud classification system. The cloud names, their characteristics, the features used in the later experiments, and the classifiers used in our system will be described.

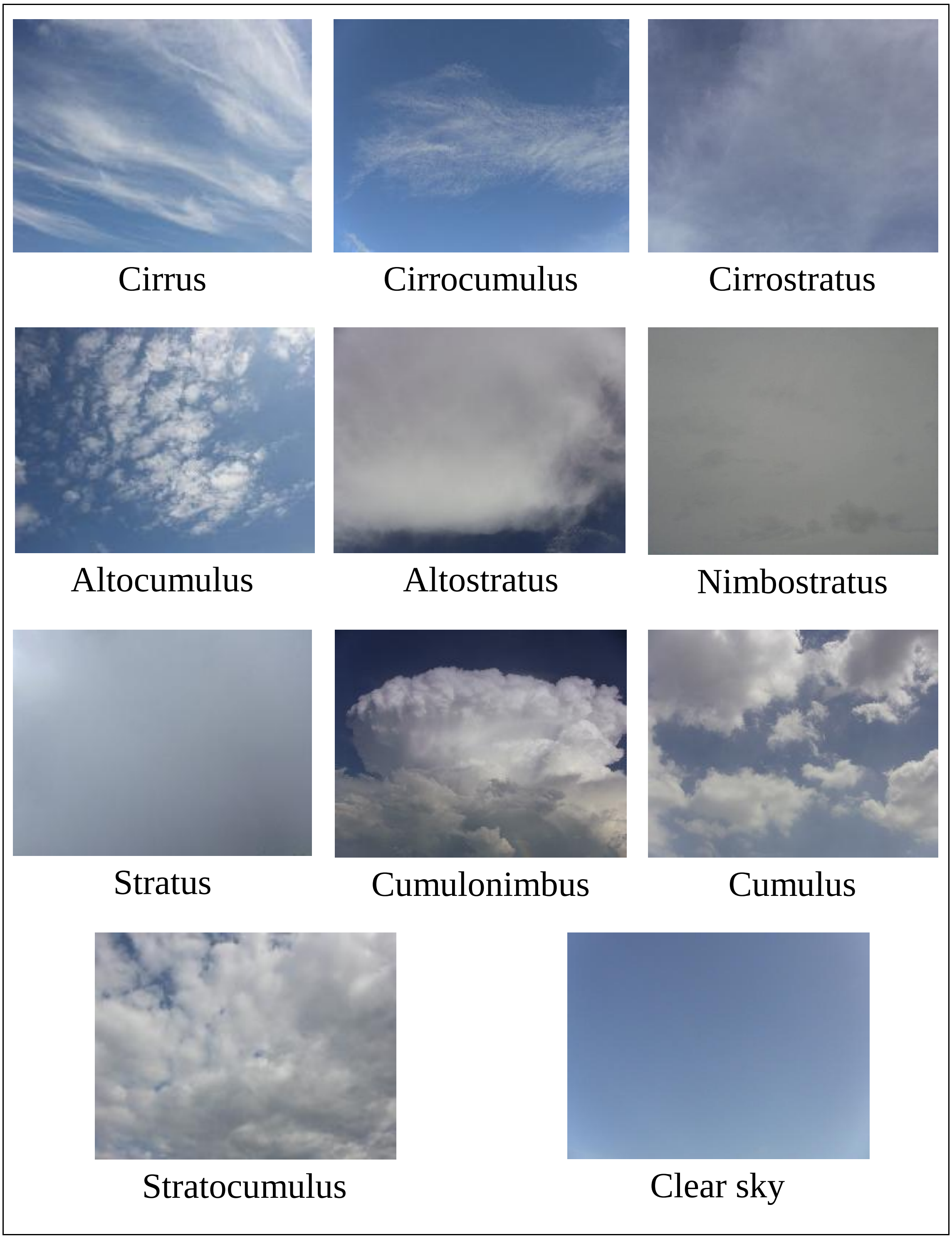
2.1 Cloud Names and Classifications

In the study of Meteorological Office (Met Office), clouds are difficult to recognize since they are always changing and are in many shapes and forms. A meteorologist, Luke Howard has written a book on cloud classification in 1803 which he described structure of various clouds and names [35].

Afterward, Luke Howard's classifications were extended by the World Meteorological Organization (WMO) as 10 cloud types or genera, namely [36], cirrus, cirrocumulus, cirrostratus, altocumulus, altostratus, nimbostratus, stratus, cumulonimbus, cumulus, and stratocumulus. Figure 2.1 show eleven different cloud types (clear sky is included).

1) Cirrus Clouds

Cirrus clouds occur above about 18,000 - 40,000 feet. They are short, detached, and look like hair. They are whiter than any other clouds in daytime. Furthermore, they are colors of the sunset while the sun is rising [35].



Cirrus

Cirrocumulus

Cirrostratus

Altocumulus

Altostratus

Nimbostratus

Stratus

Cumulonimbus

Cumulus

Stratocumulus

Clear sky

Figure 2.1. Eleven different sky conditions.

2) Cirrocumulus Clouds

Cirrocumulus clouds occur above about 20,000 - 40,000 feet. Often cirrocumulus clouds are small white ripples and compose mostly ice crystals. They are often rare than altocumulus clouds and no shading [35].

3) Cirrostratus Clouds

Cirrostratus clouds occur above about 18,000 - 40,000 feet. They are transparent and covering big areas of the sky, and may be smooth or fibrous, and usually have cirrus clouds around them. When the sun shines through cirrostratus clouds, they have shadows which are difference from nimbostratus clouds [35].

4) Altocumulus Clouds

Altocumulus clouds occur above about 2,000 - 18,000 feet. They are white or grey color and the sides are shaded when they are away from the sun. Usually, altocumulus clouds are rounded clump shape and composed of droplets and may contain ice crystals which are found in settled weather. The appearance like shading can help distinguish between altocumulus clouds and cirrocumulus clouds. Cirrocumulus clouds are only white but altocumulus clouds can be white or grey and the sides may be shaded [35].

5) Altostratus Clouds

Altostratus clouds occur above about 7,000 - 18,000 feet. They consist of the mixture of water droplets and ice crystals. In addition, they are in grey or bluish layers and appear thin where we can see the sun shining weakly through the cloud.

The difference between altostratus and nimbostratus clouds is that there is no shadow when the sun shining through altostratus [35].

6) Nimbostratus Clouds

Nimbostratus clouds occur above about 2,000 - 10,000 feet. They are dark grey or bluish grey covering large area of sky with no layers. Moreover, they can block out the sun and usually comes with continuous heavy rain or snow [35].

7) Stratus Clouds

Stratus clouds occur above about 0 - 6,500 feet. They are in grey layers or white or even tract of clouds with blurry edges. Sometimes, they appear like fog at ground level. The sun or the moon may shine through stratus clouds, if there is no other cloud above. Stratus clouds often come with drizzle or snow [35].

8) Cumulonimbus Clouds

Cumulonimbus clouds occur above about 1,100 - 6,500 feet. They are heavy and dense, often flat and very dark. This type of clouds extends high into the sky like towers or mushroom shape. Moreover, they are related with extreme weather conditions such as heavy torrential downpours, hail storms, lightning and tornados [35].

9) Cumulus Clouds

Cumulus clouds occur above about 1,200 - 6,500 feet. The top of clouds is white but their base is rather dark. Usually, they appear in fair weather which is similar to the cauliflower shape. Sometimes, they produce showers when they get larger [35].

Table 2.1. Seven sky conditions and their descriptions.

Class No.	Cloud types	Description	Subtypes
1	Cirriform	Hair-like, milky, thin, whitening of the blue, white.	Cirrus and cirrostratus
2	High cumuliform	Rounded, patched clouds of small cloudlets, white or gray.	Cirrocumulus and altocumulus
3	Stratocumulus	Patches of clouds, white or gray.	Stratocumulus
4	Cumulus	Puffy with flat bases, white or light-gray.	Cumulus
5	Cumulonimbus	Mushroom-like, dark base, gray.	Cumulonimbus
6	Stratiform	Layer of cloud, uniform, usually overcast, gray or dark.	Altostratus, nimbostratus, and stratus
7	Clear sky	No cloud or a very few of clouds, blue.	Clear sky

10) Stratocumulus Clouds

Stratocumulus clouds occur above about 1,200 - 6,500 feet. Their colors are changing from bright white to dark grey which are patches of cloud. They tend to stick together or separate and often appear in every weather condition [35].

11) Clear Sky

Clear sky is when there is no cloud or a very few of clouds in the sky. Usually, it appears as blue.

The above cloud types will be grouped into seven classes for the experiments in Chapter 3 to 5. The seven classes are concluded in Table 2.1. In this table, some cloud types are combined together because they have similar characteristics. For example, cirrus and cirrostratus are grouped into one class because cirrostratus rarely occurs in nature and it usually appears together with cirrus. Similarly, cirrocumulus is hardly found in nature. Hence, it is grouped with altocumulus and being classified under the name of high cumuliform clouds. Our grouping of cloud types here follows Heinle *et al*. [15]. However, nimbostratus is grouped together with stratiform in accordance with Calbó and Sabburg [13] and Li *et al*. [23].

2.2 Features

2.2.1 Texture Features

Texture description is used to describe the texture of region. There are two main approaches for texture description which are grouped by data sources used in calculation. The first type is when texture features are extracted straight from images. These features are mean (*ME*), standard deviation (*SD*), difference of mean (D_{ij}) between each channel, and uniformity (U) defined by the equations below [13], [15].

- Mean

$$ME = \sum_{i=0}^{N-1} x_i p(x_i) \qquad (2.1)$$

- Standard deviation

$$SD = \left(\sum_{i=0}^{N-1} (x_i - ME)^2 p(x_i)\right)^{1/2} \qquad (2.2)$$

- Difference of mean R-G, R-B, and G-B

$$D_{ij} = ME_i - ME_j \; for \; i, j \in \{R, G, B\} \; where \; i \neq j \qquad (2.3)$$

- Uniformity

$$U = \sum_{i=0}^{L-1} p(z_i) \qquad (2.4)$$

The notation N in the equation is the number of pixels in image, x_i is a value of pixel i-th, and $p(x_i)$ is a probability of x_i. This type of texture features is used to distinguish between dark clouds and white clouds as well as to separate thin clouds like cirrus from others. However, some clouds have the same color tone such as cumulus and stratocumulus; hence we cannot separate them by these features alone [15]. Therefore, other texture features will be combined to solve this problem.

The second type is when texture features are calculated from Grey Level Co-occurrence Matrices (GLCM) which is a square matrix where the number of columns equals the number of grey levels. Each element in the matrix refers to the frequency that two pixels occurred ($P^{\Delta}(a,b)$) [15]. We will use the following Haralick texture features [15], [37] later on in the experiments.

- Homogeneity

$$HOM = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} \frac{P^{\Delta}(a,b)}{1+|a-b|} \qquad (2.5)$$

- Contrast

$$CON = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} (a-b)^2\, P^{\Delta}(a,b) \qquad (2.6)$$

- Energy

$$EN = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} P^{\Delta}(a,b)^2 \qquad (2.7)$$

- Variance

$$EN = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} (a-\mu)^2\, P^{\Delta}(a,b) \qquad (2.8)$$

- Inverse Difference Moment

$$IDM = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} \frac{P^{\Delta}(a,b)}{1+|a-b|^2} \qquad (2.9)$$

- Sum Average

$$SA = \sum_{a=1}^{2(G-1)} aP_{x+y}(a) \qquad (2.10)$$

- Sum Variance

$$SV = \sum_{a=1}^{2(G-1)} (a-SE)^2 P_{x+y}(a) \qquad (2.11)$$

- Sum Entropy

$$SE = -\sum_{a=1}^{2(G-1)} P_{x+y}(a) log P_{x+y}(a) \qquad (2.12)$$

- Entropy

$$ENT = -\sum_{a=0}^{G-1}\sum_{b=0}^{G-1} P^{\Delta}(a,b) log P^{\Delta}(a,b) \qquad (2.13)$$

- Difference Variance

$$D = variance\ of\ P_{x+y} \qquad (2.14)$$

- Difference Entropy

$$DEN = -\sum_{a=0}^{G-1} P_{x+y}(a) log P_{x+y}(a) \qquad (2.15)$$

- Information Measures of Correlation

$$C = \frac{ENT-HXY}{\max\{HX,HY\}} \qquad (2.16)$$

The GLCM matrix size is denoted by *G* where HX and HY are entropies of summing row and column of GLCM matrix, respectively, and HXY is the entropy of multiplying row and column together.

$$HXY = -\sum_{a=0}^{G-1}\sum_{b=0}^{G-1} P^{\Delta}(a,b)\log\{P_x(a)P_y(b)\} \qquad (2.17)$$

$P_{x+y}(a)$ is a sum of GLCM element where row plus column equals a. In contrast, $P_{x-y}(a)$ is a sum of GLCM element where row minus column equals a.

2.2.2 Moments of Two-Dimensional Functions

Moments of two-dimensional functions are used because of their resistance to any transformation. That is, the value of moments after transformation is not varied too much. The Zernike moments is shown in equation below [38].

$$A_{nl} \frac{n+1}{\Pi} \sum_x \sum_y P_{xy} \, [V_{nl}(x, y)]^* \qquad (2.18)$$

Note that *n* is an order with repetition *l* for a digital image, P_{xy} is the current pixel, $V_{nl}(x,y)$ is the Zernike polynomial and * is the complex conjugate.

2.2.3 Features based on Fourier Transform

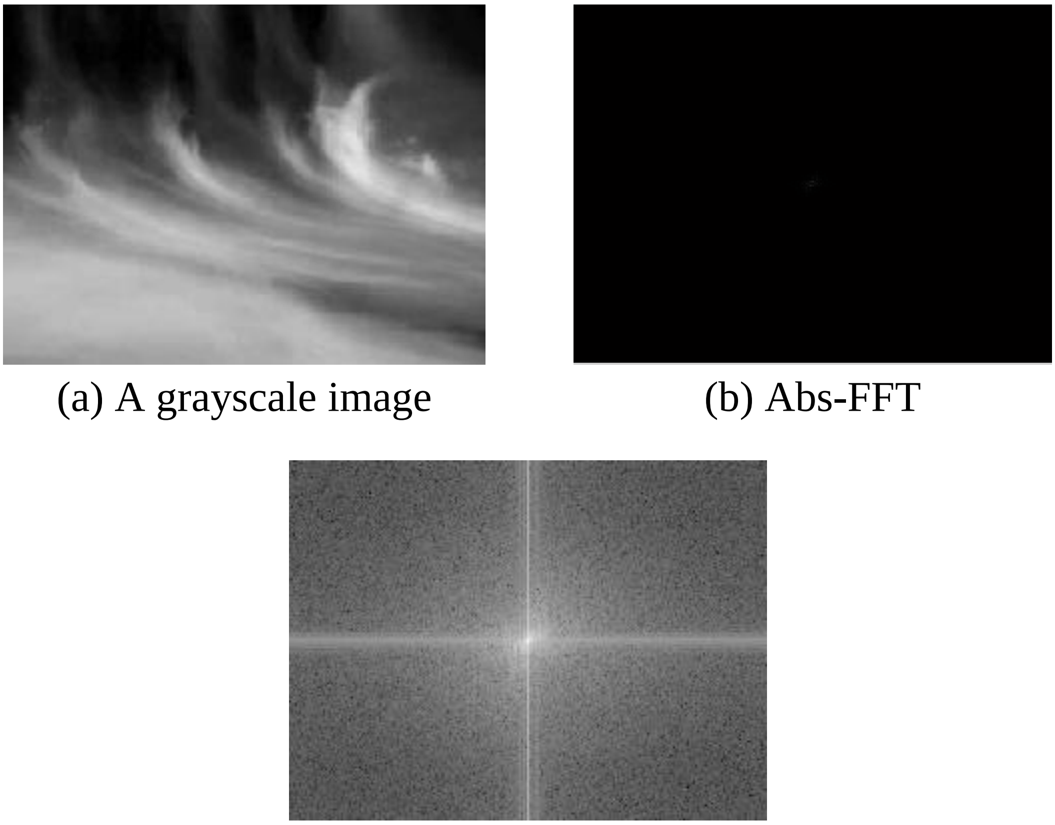
We use a grayscale image to transform pixels into frequency domain by two dimensional fast Fourier transform (2D-FFT) and use FFT shift to move the low frequency pixels into the center of the image (see Figure 2.2). There are three types of features based-on Fourier transform that we will use in the experiments of Chapter 4, namely, abs-FFT, log-FFT, and *k*-FFTPX.

1) Abs-FFT

Abs-FFT is an absolute-magnitude of Fourier transform image. Figure 2.2(a) is a grayscale image, we use 2D-FFT and FFT shift to derive abs-FFT (see Figure 2.2(b)) by calculating an absolute value of each pixel. The transformed image in Figure 2.2(b) is later on passed to the texture feature extraction process.

2) Log-FFT

We use a grayscale image in Figure 2.2(a) followed by 2D-FFT and FFT shift. Then, we calculate the logarithmic of each pixel in frequency domain which is a logarithmic magnitude of Fourier transform image (log-FFT) as shown in Figure 2.2(c). The log-FFT image is then used in a calculation of texture feature extraction.



(a) A grayscale image

(b) Abs-FFT

(c) Log-FFT

Figure 2.2. Fast Fourier transform of a grayscale image.

3) *k*-FFTPX

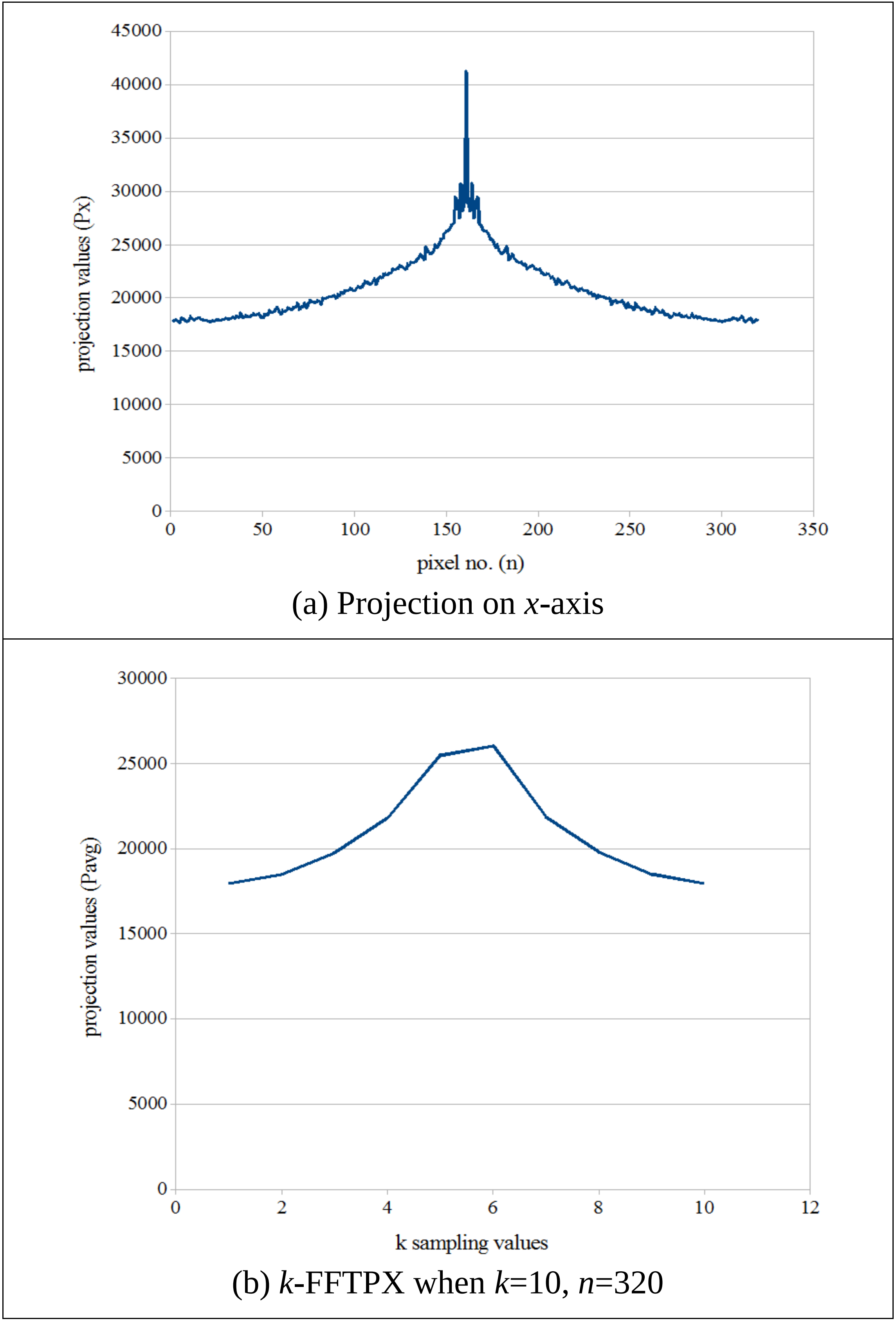
This is a new FFT feature that we introduce in Chapter 4. The proposed feature extraction method called Fast Fourier Transform Projection on the *x*-axis (*k*-FFTPX) is named after the process of projecting the log-FFT values of an image onto the *x*-axis (see Figure 2.3(a)) before selecting *k* sampling values of the data as *k* dimensions of a feature vector as depicted in Figure 2.3(b). The *k*-FFTPX is based on FFT technique which has more sub methods inside. First, DWT is used to extract the key characteristics by frequency separation of an image. Second, the image is transformed to the frequency domain by FFT which is useful in distinguishing shapes of clouds and reduces the effects of unequal brightness of the image. Third, the characteristics and key features are extracted from the logarithmic magnitude of FFT image and the values are projected on the *x*-axis. Then, these projection values are split into *k* blocks in order to reduce the dimension of the feature vector. Each block is then represented by an average value. After that, values in the feature vectors are sorted in descending order to increase the performance because we found that the sorting technique works well in practice. The algorithm is explained in full in Chapter 4.

2.3 Classifier

A classifier is a function that assigns the input images to a desired output class. Two commonly used classifiers, *k*-nearest neighbor and artificial neural network are described below.

2.3.1 *k*-Nearest Neighbor

The *k*-nearest neighbor (*k*-NN) technique is used in many research papers [10], [15], [19] for cloud type classification because it is simple to implement and understand. In addition *k*-NN has low computation and also can be used to solve a complex problem [10], [15].



(a) Projection on *x*-axis

(b) *k*-FFTPX when *k*=10, *n*=320

Figure 2.3. *k*-FFTPX projection and sampling values.

The *k*-NN algorithm is a non-parametric method. It uses a training data which consists of a set of vectors with class label of each vector. In classification, we define a *k* constant which is the number of neighbors, and we use a test data (a vector with unknown class) which will be classified by considering *k* neighbors.

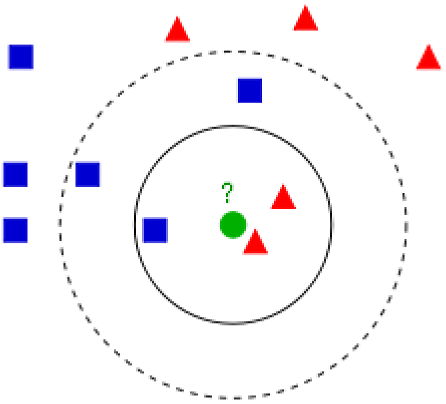


Figure 2.4. The *k*-nearest neighbor algorithm [39].

If *k* neighbors of the test data come from many classes, we will classify the test data as it belongs to a class with the majority of neighbors [40] as shown in Figure 2.4. In the Figure above, we use *k*=3, the green circle is then classified as in the same class as the red triangle because among three neighbors there are two red triangles and one blue rectangle. If *k*=5, the green circle is classified as in the blue square class because among five neighbors there are two red triangles and three blue rectangles.

2.3.2 Artificial Neural Networks

Artificial neural network (ANN) technique is used for cloud type classification [2], [10] because it is a nonlinearity classifier which is suitable for solving complex issues. Back propagation algorithm which applies gradient descent often used to train the network for the best-fit value. There are three layers of neural network which are composed of input layer, hidden layer, and output layer [40] as depicted in Figure 2.5.

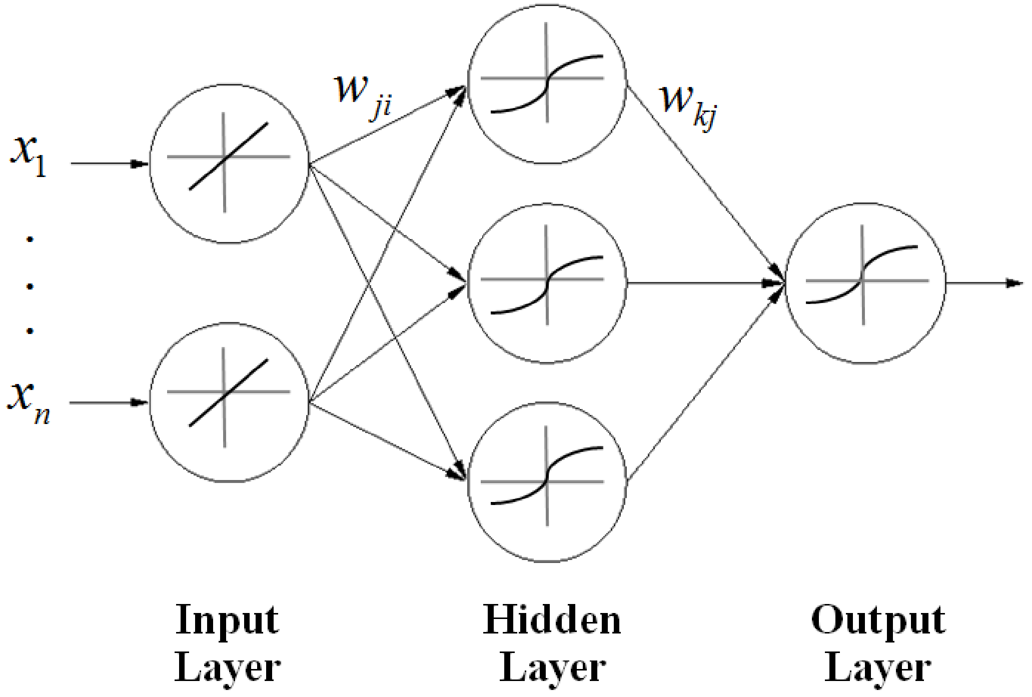


Figure 2.5. Multilayer Neural Networks [41].

Each node uses a net activation equation which is a sum of weights and inputs as shown in the equation below.

$$net_j = \sum_{i=1}^{d} x_i w_{ji} + w_{j0} = \sum_{i=0}^{d} x_i w_{ji} \qquad (2.19)$$

In the equation above, the subscript *i* is the indexes of input layer, where *j* is the indexes of hidden layer; w_{ji} is the weights from layer *i* to *j*. The output of each node uses a nonlinear function known as the activation function. There are many activation functions with difference properties. For example, the equation below shows a hyperbolic tangent function [40].

$$f(net) = sgn(net) \equiv \frac{1-e^{-net}}{1+e^{-net}} \qquad (2.20)$$

There are several parameters of ANN that can be set in order to achieve high performance. In the next chapter, we will conduct the experiment to find a proper setting for these parameters. The number of hidden layers is usually set between one and three. However, only one hidden layer is enough to estimate a result of any problems, if the number of hidden nodes is sufficient [42]. The number of hidden nodes is normally set between the number of input nodes and the number of output nodes. Learning rate should be set to a reasonable small value to construct an accurate model. Momentum is used for avoiding local minimum in the training model and stopping criteria is a criterion to stop learning process. The setting of these parameters will be mentioned more in the next chapter. Note that, all experiments in this thesis are executed on a desktop computer using an Intel Core i5-3550 Quad-Core processor with clock speed 3.30 GHz and 4 GB of RAM.

CHAPTER 3

EXPERIMENT A – FINDING A SUITABLE CLASSIFIER

In this chapter, we begin the experiment by using 18 texture features to distinguish seven sky conditions. The important parameters of two classifiers are fine-tuned in the experiment, namely, *k*-nearest neighbor (*k*-NN) and artificial neural network (ANN). The performances of the two classifications are compared. Advantages and limitations of both classifiers are discussed. Our result reveals that the *k*-NN model performs at 72.99% accuracy while the ANN model has higher performance at 86.93% accuracy. We show that our result is better than previous studies. Finally, seven most effective texture features are recommended to be used for more compact cloud type classification system.

3.1 Introduction

As mentioned before, texture features are commonly used in cloud classification to describe characteristics of cloud. There are several feature extraction methods to extract texture feature and the accuracy depends on how to choose features to use as a feature vector. In this chapter, we will experiment with automatic cloud classification system using ground-based sky images from a digital camera. Several texture features will be combined and we will select the 18 suitable features for classifying images into one of seven sky conditions which again are cirrus, high cumuliform, stratocumulus, cumulus, cumulonimbus, stratus, and clear sky. Two classifiers will be tested in the system, namely, *k*-nearest neighbor and artificial neural network. Then, we will compare the performance of both classifiers in terms of percentage of accuracy.

3.2 Technical Background

We perform the feature extraction, before we apply a classifier to compute a class type based on the given features. Features are extracted from grayscale image by splitting channels of the image into Red (R), Green (G), or Blue (B) channel. Although, there are several feature extraction methods, we will begin our experiment using texture features because they are commonly used in cloud classification [10], [13], [15] Thus, it is worth testing these features.

3.2.1 Texture Features

Texture features describe texture of image region. This chapter uses two types of texture features in the feature extraction process. The first type of texture features, *ME* of R channel, *SD* of B channel, and D_{ij} between each channel are used. All equations are as defined in Chapter 2 (Eq. (2.1) to (2.3)). The second type of texture features is described by 13 features. We use Eq. (2.6) to (2.16) to compute 11 texture features on the R channel and also compute 2 texture features of *HOM* (Eq. (2.5)) and *CON* (Eq. (2.6)) on the B channel. Note that, there are three features from Eq. (2.3). In total there are 5 texture features of the first type and 13 texture features of the second type. Therefore, there are 18 texture features for feature extraction. Most of these features are calculated from the R channel. We explain why this is the case.

Cloud texture generally appears white or gray because it is equally composed of red and blue pixel values. In contrast, clear sky appears blue because it is composed of blue pixel values more than red [15]. Therefore, the distinction between clear sky and cloud is better described by R channel which gives higher contrast of image than other channels. Features from R channel can be used to distinguish cloud types better than the B channel which will be shown later by experiment in Section 4.3.1.

Two commonly used classifiers [10], [15], pattern recognition tools for distinguish classes, for cloud classification in this experiment are *k*-nearest neighbor and artificial neural network.

3.2.2 *k*- Nearest Neighbor

The reason we choose *k*-nearest neighbor (*k*-NN) for a comparison is, because it is quite a popular technique in cloud classification because of low computation and its ability to solve complex problem. Most of all, it is simple to implement [10], [15]. A brief detail of *k*-NN algorithm is that it works by considering *k* neighbors of a sample before classifying that sample as a class by a majority vote of its neighbors; that is the sample is assigned to the class most frequent among its *k* nearest neighbors. In theory, the larger *k* is more suitable for infinite number of samples. However, this is impossible in practice [43]. Hence, we are required to choose the optimal *k* within the available samples.

3.2.3 Artificial Neural Network

The reason we choose artificial neural network (ANN) is, because when a problem is complex, this classifier is often used [10]. ANN is composed of three main layers, namely, input layer, hidden layer, and output layer. It can be used to solve nonlinear problems without considering the relationship between input and output. However, the classifier requires the tuning of multiple parameters in the model [44] for the best performance. Therefore, to achieve high accuracy we will examine key parameters of ANN in the experimental section. In this experiment, multilayer feed forward neural network is used.

3.3 Experimental Results

We used 696 ground-based cloud images taken from a digital camera for training and testing process. These images are scaled to 320 × 240 pixels for all experiments. The 18 texture features explained in Section 3.2 using Eq. (2.1) to (2.3) and Eq. (2.5) to (2.16) are implemented and the two classifiers used are ANN and *k*-NN. The main objective of the experiment is to find the suitable classifier for cloud classification. First, the parameters of two classifiers are tuned for providing high accuracy. Then, we compared the performance of the two classifiers and discussed the result. Some parameters are fixed for all experiments such as, a learning rate of ANN to 0.01 and a momentum to 0.9. Leave-one-out cross-validation (LOOCV) [44] is used to evaluate the percentage of classification.

3.3.1 Selection of *k* Value for *k*-NN

The number of neighbors is assigned for classification. The results are shown in Table 3.1. The accuracy (%) values are the average percentages of all correctly classified instances.

The *k* values from 1 to 14 were tested with a distance measure set to the Euclidean distance. In practice, *k* should not be too large as it can lead to over-smoothed boundaries [43].

Table 3.1. The *k* value of *k*-NN model between 1 to 14.

***k*-value**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Accuracy (%)	60.20	50.00	49.86	50.14	51.72	50.72	52.01
***k*-value**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Accuracy (%)	51.29	51.72	49.86	48.42	49.71	49.86	50.00

According to Table 3.1, when $k = 1$ the accuracy is highest among others. This means that $k = 1$ can separate noises from a correct sample better than other *k* values. Note that, the suitable *k* value depends on the distribution of features. For this problem, other *k* values ($k > 1$) may include points from other classes which can easily lead to the misclassification. Therefore, the remaining of this chapter will apply $k = 1$ as the parameter of *k*-NN model.

3.3.2 Selection of Distance Measures for *k*-NN

Distance function is an important parameter of *k*-NN classifier for determining nearest neighbor of the test data.

There are many distance measures for *k*-NN and choosing a suitable measure is an important problem in supervised learning techniques [45]. We used seven distance measures as shown in Table 3.2. From Table 3.2, each distance measure provides similar results except that Standardized

Euclidean distance has around 10% higher accuracy than others. We, therefore, resolve that Standardized Euclidean distance measure is compatible with a form of our training data distribution.

Table 3.2. Accuracy when using seven different distance measures.

Distance measures	**Accuracy (%)**
City Block	61.06
Chebychev	61.49
Correlation	59.34
Cosine	59.91
Euclidean	60.20
Min-kowski	60.20
Standardized Euclidean	72.99

3.3.3 Selection of Activation Function for ANN

ANN model has many parameters that we can vary. The activation function of the hidden layer is examined here. Note that the activation function of the input layer is fixed as linear to keep the same input values throughout the model. For the output layer, we fixed as hyperbolic tangent because this function can generate binary outputs [46].

Table 3.3. Different activation functions in hidden layer.

Activation function	**Radial basis**	**Tangent**	**Linear**	**Log-sigmoid**
Accuracy (%)	80.45	82.04	77.87	80.32

From Table 3.3, there are 5 hidden nodes in the hidden layer. The result shows that the hyperbolic tangent function outperforms the other 3 functions because the hyperbolic tangent can provide binary outputs which are suitable for classification problems. However, it provides a good result if this function is used in both hidden and output layers [46].

3.3.4 Selection of Number of Hidden Nodes for ANN

The number of hidden nodes in a hidden layer is varied as 3, 5, 7, and 9. The activation function is set to hyperbolic tangent.

Table 3.4. The number of hidden nodes is chosen as 3, 5, 7, and 9.

Number of hidden nodes	**3**	**5**	**7**	**9**
Accuracy (%)	77.44	82.04	86.93	82.90

From Table 3.4, the number of hidden nodes equal to 7 gives the best result. The number of hidden nodes affects the generalization error. When there are fewer hidden nodes such as 3 hidden nodes, this might cause underfitting and high statistical bias. In contrast, if we design our hidden layer with many hidden nodes such as 9 hidden nodes, it might cause overfitting and high variance [47].

3.3.5 Comparison of *k*-NN and ANN

In the past four sub-experiments, it may seem that ANN classifier is cumbersome to use than *k*-NN due to the fine-tuning of various parameters. However, if we can tune the parameters appropriately, it can lead to a more accurate classification over *k*-NN.

Table 3.5. The accuracy comparison of *k*-NN and ANN.

Classifiers	***k*-NN**	**ANN**
Accuracy (%)	72.99	86.93

Table 3.5 shows that the accuracy of ANN is about 14% higher than *k*-NN. While the result of Singh and Glennen [10] is 64% accurate for a classification of five sky conditions using ANN, we obtained 86.93% accuracy for a classification of seven sky conditions using also ANN. Nevertheless, our *k*-NN did not perform badly in the comparison with Singh and Glennen. For the *k*-NN, we received 72.99% accuracy while Singh and Glennen have 59.5% accuracy for a classification of less sky conditions. Although Souza-Echeret *et al*. [11] have 94% accuracy, their classification is only for clear sky condition. If we classify only this class, we would get 96.34%.

$$\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & \mathbf{58.77} & 7.89 & 1.75 & 17.54 & 11.40 & 0.88 & 1.75 \\
2 & 13.98 & \mathbf{64.52} & 13.98 & 5.38 & 2.15 & 0.00 & 0.00 \\
3 & 1.20 & 7.23 & \mathbf{75.90} & 6.02 & 3.61 & 6.02 & 0.00 \\
4 & 17.12 & 0.90 & 1.80 & \mathbf{69.37} & 10.81 & 0.00 & 0.00 \\
5 & 8.94 & 4.07 & 6.50 & 13.82 & \mathbf{65.04} & 1.63 & 0.00 \\
6 & 1.11 & 0.00 & 5.56 & 0.00 & 0.00 & \mathbf{92.22} & 1.11 \\
7 & 2.44 & 1.22 & 0.00 & 0.00 & 0.00 & 1.22 & \mathbf{95.12}
\end{array}$$

(a) Confusion matrix of *k*-NN classifier

$$\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & \mathbf{79.82} & 4.39 & 2.63 & 10.53 & 1.75 & 0.00 & 0.88 \\
2 & 4.30 & \mathbf{86.02} & 5.38 & 4.30 & 0.00 & 0.00 & 0.00 \\
3 & 3.61 & 14.46 & \mathbf{72.29} & 4.82 & 2.41 & 2.41 & 0.00 \\
4 & 8.11 & 0.00 & 0.00 & \mathbf{82.88} & 8.11 & 0.00 & 0.90 \\
5 & 1.63 & 0.00 & 0.00 & 3.25 & \mathbf{95.12} & 0.00 & 0.00 \\
6 & 2.22 & 0.00 & 1.11 & 0.00 & 1.11 & \mathbf{95.56} & 0.00 \\
7 & 0.00 & 0.00 & 0.00 & 1.22 & 1.22 & 1.22 & \mathbf{96.34}
\end{array}$$

(b) Confusion matrix of ANN classifier

Figure 3.1. Confusion matrix for the ground-based cloud image using two different classifiers.

The percentage of the correct classification for each class is described in Figure 3.1, using confusion matrix. According to Figure 3.1, each row represents the percentage of the instances in a true class and each column represents the percentage of the instances in a predicted class. Note that Class 1 refers to cirrus and Class 7 refers to clear sky which is in the same order as explained in the beginning of this chapter. In Figure 3.1(a), four classes have less than 75% accuracy leading to a poorer result when using *k*-NN. On the other hand, in Figure 3.1(b), the majority of correctly classified instances for ANN are over 80%, some are even as high as 96.34%. To further improve this result, we could try to find more effective features for distinguishing cirrus and stratocumulus from the rest of the other clouds.

3.3.6 Recommended Texture Features

The 18 texture features used in the experiment did not produce the same impact. Therefore, we rank the most effective texture features using a knock-out method. The results of ranking are as follows: 1) *SD* of B channel, 2) D_{ij} of R and B channel, 3) *HOM* of B channel, 4) *EN* of R channel, 5) *V* of R channel, 6) *ENT* of R channel, and 7) *C* of R channel. With merely 7 texture features above, we get the accuracy as high as 77.44% using ANN with 7 hidden nodes.

3.3.7 Standard Cloud Images

In this section, we conduct an additional experiment to show the generalization of our features and classifier. Since there is no official standard library for the seven cloud-type images, we will use 636 pre-classified cloud images from the two trusted locations [48], [49] as our standard cloud images for the testing purposes. The ANN classifier with the same parameter settings as Section 3.3.5 is tested using the standard images. The 18 texture features are applied in the same way as the previous experiments

Table 3.6. Confusion matrix for classifying standard cloud images.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**85.83**	3.94	1.57	0.79	6.30	0	1.57
2	15.07	**65.75**	4.11	2.74	6.85	4.11	1.38
3	12.12	33.33	**36.36**	3.03	3.03	12.12	0
4	5.45	7.27	0.91	**54.55**	30	1.82	0
5	1.67	0.56	0.56	5	**92.22**	0	0
6	0	2.90	5.80	1.45	4.35	**82.61**	2.90
7	2.28	0	0	2.28	0	2.27	**93.18**

Table 3.6 shows the classification results using a set of standard cloud images. Four classes are correctly classified with high accuracy of more than 82%. Class 4 is highly misclassified as Class 5 because most of Class 4 images are towering cumulus which is quite similar to cumulonimbus in Class 5. The texture features alone cannot distinguish these shapes of clouds. The shape information is also necessary. This problem will be addressed in the next chapter. Also, Class 3 is highly misclassified as Class 2. This is a problem of the zoomed

images in Class 3. With the images from different angles and scales, it becomes more difficult to separate Class 2 and Class 3. Nevertheless, the overall classification accuracy using standard cloud images is 72.93%. This is rather the same with the result of *k*-NN (72.99%). Hence, the generalization and effectiveness of our method are rather acceptable for general cloud images.

Note that the standard cloud images from different places of the world are merely for the testing purposes, they will no longer be used in the future experiments. Instead, a database of real cloud images captured locally at Prince of Songkla University, Phuket Campus will be experimented upon as the final cloud monitoring station will be built at this location.

3.4 Chapter Summary

The seven texture features in recognizing each sky condition were recommended. If the 18 texture features were combined, this would give even better results. Although, ANN was difficult to fine-tune, ANN provided a better accuracy than *k*-NN with 86.93% of instances are correctly classified. This result is about 23% higher than the previous studies which were using also a digital camera and ANN method but the number of classes in those studies was only for five sky conditions, in our case we have seven sky conditions. In particular, our classification for clear sky condition returned 96.34% accuracy which is higher than the previous studies.

According to the results of *k*-NN and ANN, they show that when a problem is complex such as a problem of classify seven sky conditions, ANN can handle the complex problem better than *k*-NN. In the next experiment, we, therefore, will use

ANN as our main classifier. In addition, we will explore not only texture features but also other kinds of features such as a two dimensional fast Fourier transform (2D-FFT) and moments of two-dimensional functions. Moreover, we will conduct the experiments to find a suitable feature extraction technique to achieve a higher percentage of accuracy.

CHAPTER 4

EXPERIMENT B – FINDING A SUITABLE FEATURE EXTRACTION TECHNIQUE

This chapter studies several feature extraction techniques for the classification of cloud types using ground-based images. Seven sky conditions are again considered. We present an algorithm that computes a matrix of feature vectors for cloud classification with five alternative ways of extracting cloud features. The five feature extraction techniques include textures, moments of two-dimensional functions, abs-FFT, log-FFT, and the new technique called Fast Fourier Transform Projection on the *x*-axis (*k*-FFTPX). We propose the *k*-FFTPX algorithm that extracts features by projecting the values of logarithmic magnitude of FFT images on the *x*-axis of the frequency domain before selecting *k* sampling values of the data as *k* dimensions of a feature vector. To the best of our knowledge, there is no research on ground-based cloud type classification using such technique before. Then, a comparison of the techniques is made through a series of five experiments and the accuracies are ranged between 80.76% and 90.40%. Our new method provides the highest accuracy. The advantages are that we can now classify more cloud types than the existing methods with further improved in accuracy, and our method requires no expensive tools, only a digital camera is used to obtain ground-based images. This suggests a variety of practical solutions in combination with other meteorological sensors to report weather conditions inexpensively.

4.1 Introduction

In this chapter, we will develop an automatic cloud type classification system for ground-based digital camera using image processing and pattern recognition. Seven different cloud types for our recognition are cirrus, high cumuliform (cirrocumulus and altocumulus), stratocumulus, cumulus, cumulonimbus, stratus, and clear sky. We will extract texture features from cloud images and will use this information in the training process of the classification. ANN is then used for classifying instances. Moreover, we will add three types of features based on Fourier transform. The first two types use logarithmic and absolute magnitudes for extracting texture features of FFT images. The last type uses logarithmic magnitude but we project these values on the *x*-axis. Our main contribution is a novel feature that uses a projection of logarithmic magnitude of the FFT onto the *x*-axis. We call this feature *k*-FFTPX.

4.2 Technical Background

Our new approach will incorporate the strength of texture analysis found in the previous chapter into the new technique of FFT feature extraction that focuses more on the shape of cloud. In addition to Table 1.1 – 1.3, Table 4.1 shows specifically various uses of FFT techniques, some are incorporated with other methods. Calbó and Sabburg [13] used features based on Fourier transform to discriminate cloud shapes. They extracted the characteristics of the spectral power image using correlation with clear (CC) and spectral intensity (SI). Daowieng *et al.* [50] used FFT and discrete wavelet transform (DWT) for word recognition. More recently, Chen [51] extracted dual-tree complex wavelet (DTCWT) features from EEG signals and perform the FFT to the DTCWT features subbands. Soltana *et al*. [52] applied FFT with Local Binary Patterns (LBP) histogram to calculate features from lace images. Stępniowski *et al*. [53] calculated the radial average of FFT for arrangement analysis of the aluminum nanopores.

Table 4.1. Related works on fast Fourier transform techniques.

Year	**Proposed method**	**Application**	**Author**
2008	Extracting CC and SI from FFT	Cloud classification	Calbó and Sabburg [13]
2010	DWT with FFT	Word recognition	Daowieng *et al*. [50]
2014	DTCWT with FFT	EEG analysis	Chen [51]
2014	LBP and FFT with *k*-NN	Analysis of lace images	Soltana *et al*. [52]
2014	Radial average of FFT	Aluminum nanopores	Stępniowski *et al*. [53]

To the best of our knowledge, there is no research on ground-based cloud type classification that performs feature extraction by projecting the values of logarithmic magnitude of FFT images on the *x*-axis of the frequency domain. Furthermore, the new idea of introducing the *k*-sampling and sorting techniques in the settings of feature vector will be incorporated into our proposed algorithms. These techniques will be explained later.

4.2.1 Features

We use a grayscale image which is computed by splitting channels of image as R, G, and B channels for extracting features. There are three groups of features which are used in the experiments, namely, the texture feature, the moments of two-dimensional, and the features based on Fourier transform.

1) Texture Features

There are two types of texture features used in this chapter. The first type of texture features are extracted from images directly. These are *ME* of R channel, *SD* of B channel, and D_{ij} between each channel defined by the Eq. (2.1) to (2.3) in Chapter 2. The second type of texture features are computed from GLCM. We will use the Haralick texture features Eq. (2.6) to (2.16) are computed on R channel and Eq. (2.5) to (2.6) are computed on B channel. There are 18 texture features to be used in the algorithm which is the same as Chapter 3.

2) Moments of Two-Dimensional Functions

Moments of two-dimensional functions or 2D-moments are used because of their resistance to any transformation. The Zernike moments is shown in Eq. (2.18) for calculating 0-th to 7-th moments on R channel.

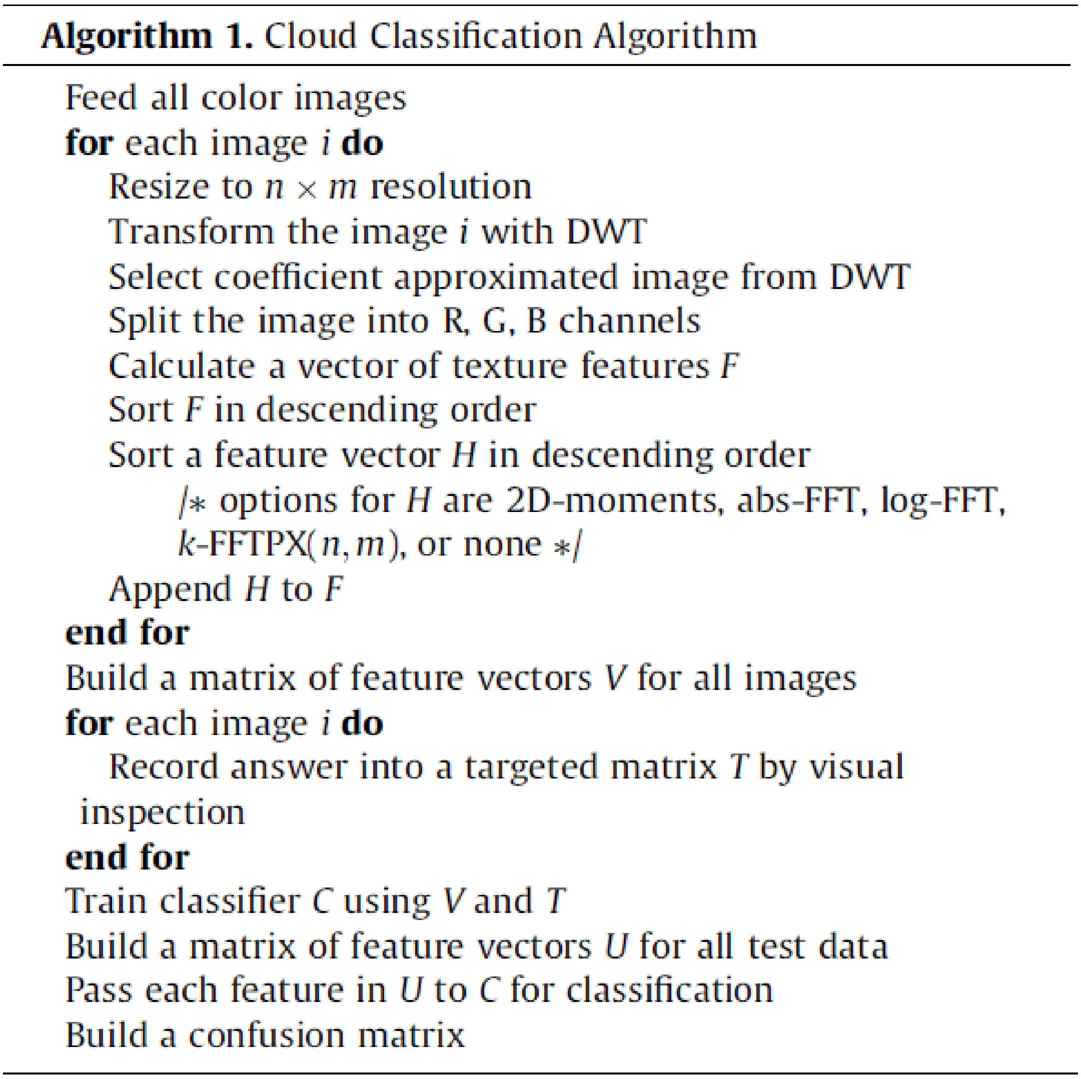
3) Features Based on Fourier Transform

We use a grayscale image (R channel) to transform pixels into frequency domain by two dimensional Fast Fourier Transform (2D-FFT) and use FFT shift to move the low frequency pixels into the center of the image. There are three types of features based-on Fourier transform that we exploit in the experiment, namely, abs-FFT, log-FFT, and *k*-FFTPX. These features were, as explained earlier in Chapter 2.

4.2.2 Algorithms

Two algorithms used in our experiments are explained. We use a set of digital camera images with no more than 36 degree FOV as the input. Algorithm 1 explains our methodology for cloud classification starting from preprocessing the input, extracting dominant features, training the classifier, classifying the instances and returning the confusion matrix as the answer.

In the preprocessing stage of Algorithm 1, we scale down each image to the resolution of n × m size. By this process we achieve a much smaller computational time. For each image we perform the feature extraction method using discrete wavelet transform (DWT). This transformation gives four images which are coefficient approximated (CA), coefficient horizontal edge (CH), coefficient vertical edge (CV), and coefficient diagonal edge (CD). The size of four images is reduced by half. We select only CA image for a calculation of our feature vector.



Algorithm 1. Cloud Classification Algorithm

```
Feed all color images
for each image $i$ do
   Resize to $n \times m$ resolution
   Transform the image $i$ with DWT
   Select coefficient approximated image from DWT
   Split the image into R, G, B channels
   Calculate a vector of texture features $F$
   Sort $F$ in descending order
   Sort a feature vector $H$ in descending order
       /* options for $H$ are 2D-moments, abs-FFT, log-FFT,
       $k$-FFTPX($n, m$), or none */
   Append $H$ to $F$
end for
Build a matrix of feature vectors $V$ for all images
for each image $i$ do
   Record answer into a targeted matrix $T$ by visual
 inspection
end for
Train classifier $C$ using $V$ and $T$
Build a matrix of feature vectors $U$ for all test data
Pass each feature in $U$ to $C$ for classification
Build a confusion matrix
```

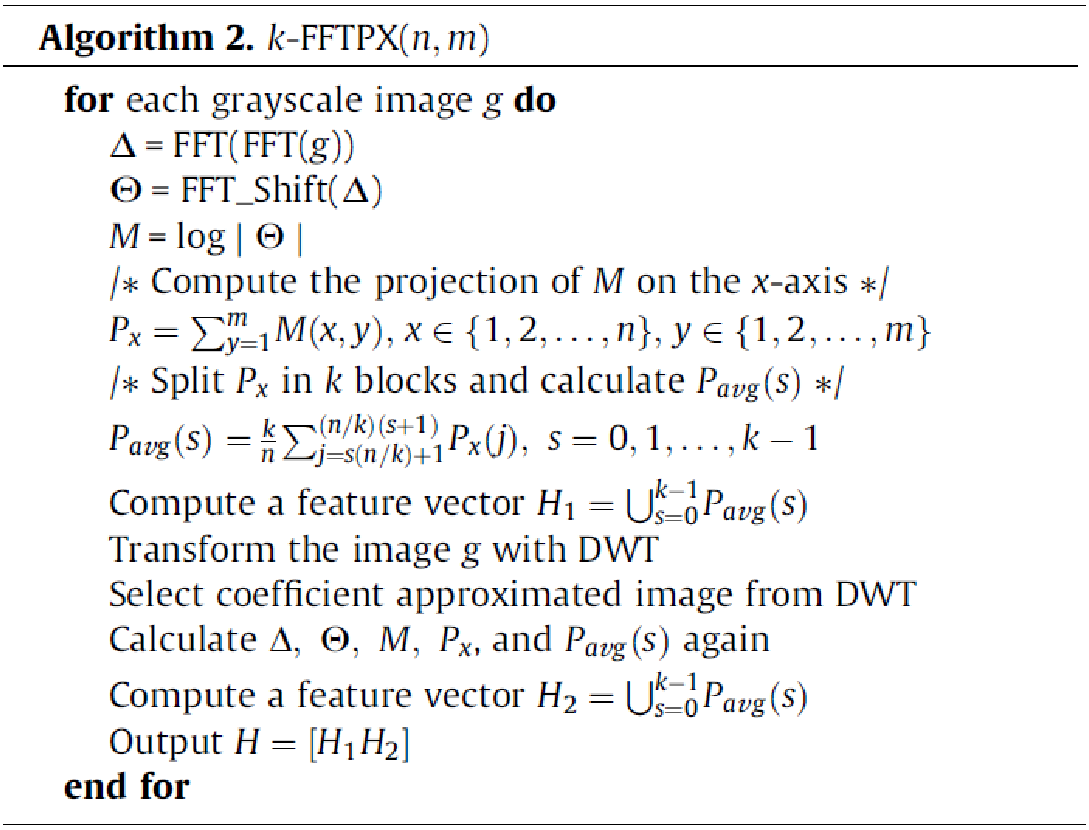
In feature extraction stage, we split channels into R, G, and B channels as grayscale images before calculating a vector of texture features *F*. The vector *F* is a based feature in our algorithm which is calculated by Eq. (2.1) to (2.3) and Eq. (2.5) to (2.16) explained in Section 3.2. There are a total of 18 texture features implemented in the algorithm. However, to improve the accuracy of the classification we propose to add one of the four following feature extractions, 2D-moments, abs-FFT, log-FFT, or *k*-FFTPX. We call this additional feature vector, *H*. The moments of two-dimensional functions extract eight features using Zernike moments in Eq. (2.18) of order $n = 0$ to $n = 7$ and repetition $m = 2$. The features based on Fourier transform are referred to abs-FFT, log-FFT, and our proposed feature (also operated on frequency domain) called *k*-FFTPX. We test and evaluate each feature separately and discuss the performance in the experimental section. Note that the method of computing *k*-FFTPX is presented in Algorithm 2 which we will explain later.

In the training stage of classifier, we build a matrix of feature vectors *V* for holding the trained features. We label the answer for each feature vector and call it a targeted matrix *T*. Based on these information, we train classifier *C* using *V* and *T*.

In the classification stage, we build a matrix of feature vectors *U* for holding the tested features. We pass each feature in *U* to *C* for the classification process before building confusion matrix for final results.

Our proposed *k*-FFTPX is shown in Algorithm 2. We extract the *k*-FFTPX feature from 2 levels which are DWT level 1 (DWT1) and DWT level 2 (DWT2) using R channel of the CA image *g*. The two levels of DWT are used because they are useful for multiresolution analysis of cloud shapes and this technique is less affected by the zooming of image. Higher levels of DWT (more than 2) will take longer to calculate and they are not recommended because the image will be too small and the important details will be lost. After the transformation of DWT1 (in Algorithm 1), we first transform the image *g* into a frequency domain using 2D-FFT to get Δ as presented in Algorithm 2. After that, Θ is obtained by shifting 2D-FFT of Δ.

Then, we calculate the logarithmic magnitude of Θ to find *M*. We project the magnitude *M* on the *x*-axis of the transformed image and called it a vector P_x. We split P_x into *k* blocks and take the average of the magnitude in each block to get *k* projection values. Note that the suitable *k* value is determined experimentally. The *k* value should be able to represent a dominant characteristic of each cloud class. The small *k* value may not be able to describe characteristics of each class while the large *k* value may represent too specific characteristics and cannot generalize to each class. A feature vector H_1 is constructed by joining the *k* projection values of $P_{avg}(s)$ in order from $s = 0$ until $s = k - 1$. The process is repeated with DWT2 to obtain the second feature vector H_2. The output of *k*-FFTPX algorithm is H_1 concatenating with H_2.



Algorithm 2. *k*-FFTPX(n, m)

```
for each grayscale image g do
    $\Delta$ = FFT(FFT(g))
    $\Theta$ = FFT_Shift($\Delta$)
    $M = \log |\Theta|$
    /* Compute the projection of $M$ on the $x$-axis */
    $P_x = \sum_{y=1}^{m} M(x, y),\ x \in \{1, 2, \ldots, n\},\ y \in \{1, 2, \ldots, m\}$
    /* Split $P_x$ in $k$ blocks and calculate $P_{avg}(s)$ */
    $P_{avg}(s) = \frac{k}{n} \sum_{j=s(n/k)+1}^{(n/k)(s+1)} P_x(j),\ s = 0, 1, \ldots, k-1$
    Compute a feature vector $H_1 = \bigcup_{s=0}^{k-1} P_{avg}(s)$
    Transform the image g with DWT
    Select coefficient approximated image from DWT
    Calculate $\Delta$, $\Theta$, $M$, $P_x$, and $P_{avg}(s)$ again
    Compute a feature vector $H_2 = \bigcup_{s=0}^{k-1} P_{avg}(s)$
    Output $H = [H_1 H_2]$
end for
```

The overall feature extraction process is summarized in Figure 4.1. According to Figure 4.1, two features vectors, F_1 and H_1, are extracted from DWT1. The feature vector F_1 contains 18 texture features derived from Algorithm 1. The feature vector H_1 is a k dimensional FFTPX feature. The feature vector H_2 is extracted from DWT2. This feature vector is also a k dimensional FFTPX feature. Finally, we concatenate all vectors to get a feature vector F which will be the input of our classifier in Algorithm 1. Each F represents one image. In the training stage of the classifier, we then build a matrix of feature vectors called V which comes from all vectors F combined.

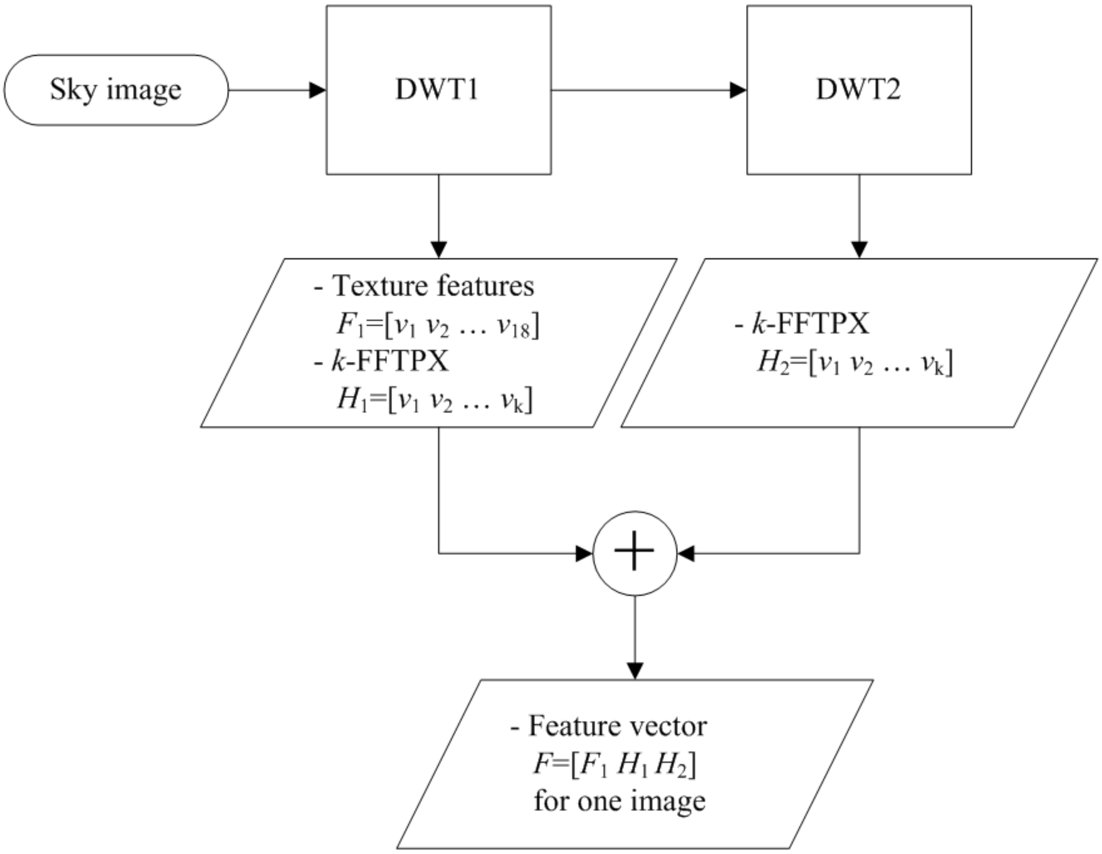


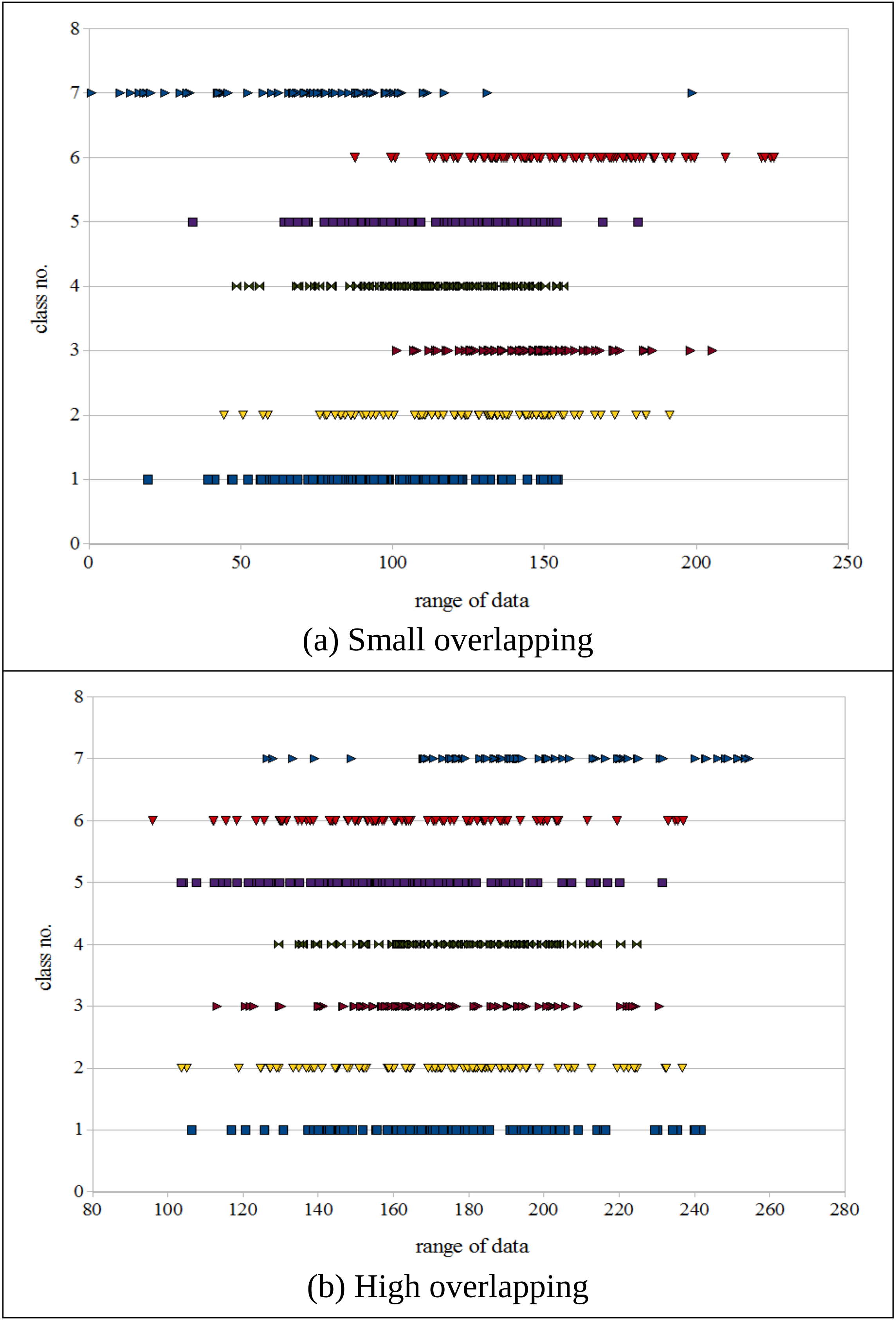
Figure 4.1. Feature extraction process for k-FFTPX.

4.3 Experimental Results

In the following experiments, we begin to exploit the real cloud images captured from the interested area of study which is Prince of Songkla University, Phuket Campus. We used over 353 ground-based cloud images. Each image has a resolution of 640 × 480 pixels. The methodology used for all experiments is as per depicted in Algorithm 1 except in the last experiment Algorithm 1 and 2 are both used. LOOCV is used to evaluate the accuracy of the classification. Our classifier is a multilayer feed forward neural network with a single hidden layer for classifying seven sky conditions. The hyperbolic tangent function is used as the activation function in hidden layer and in output layer as shown in Figure 2.5. The number of input nodes is equal to the size of each feature vector in each experiment. The number of output nodes is seven. The suitable number of hidden nodes will be determined experimentally. Other parameters of ANN are fixed with a learning rate of 0.01 and a momentum of 0.9. There are five experiments based on different features being tested in this chapter. In the first experiment, we test the performance of correctly classification with our 18 chosen texture features. Experiment 2–5, the 18 texture features are used in conjunction with 2D-moments, abs-FFT, log-FFT, and *k*-FFTPX, respectively. Furthermore, we will analyze the strengths and weaknesses of each feature used in each experiment.

4.3.1 Experiment 1 - Texture Features

There are many features that we used in the classification of cloud types. Therefore, we must find a way to choose the suitable features for classification.



(a) Small overlapping

(b) High overlapping

Figure 4.2. Distribution of chosen features.

We select the features by observing a distribution of each feature where the overlapping is minimized between each class. Figure 4.2 shows a plot of the mean distribution of the two considered features. The horizontal axis displays a range of distribution of a chosen feature whereas the vertical axis is a class number identifying the cloud types. In Figure 4.2(a) we give an example of the suitable feature (mean-R) where the distribution has small overlapping among classes. Figure 4.2(b) is an example of the unsuitable feature (mean-B) where the distribution has high overlapping among classes. This latter feature cannot distinguish a class from each other. Although, the mean-R feature from Figure 4.2(a) is suitable we still cannot use one feature solely for cloud classification because the distribution cannot distinguish all classes at once. Hence, based on the above technique we checked the distributions of 39 texture features appeared in [13], [15], [37] and selected the best 18 texture features such that the distributions can separate classes most effectively.

The chosen 18 texture features are implemented in the experiment using Eq. (2.1) to Eq. (2.3) and Eq. (2.5) to Eq. (2.16). The 18 texture features are used in the training of ANN classifier. The ANN model is constructed and fine-tuned by varying the number of hidden nodes. According to Heaton [54], this value is usually lied between the number of input nodes and output nodes. We conduct separate experiments to heuristically find which number of hidden nodes optimizes the performance of our ANN model. For this experiment (Experiment 1 – Texture features) the suitable number of hidden nodes is set to 15. Note that the number of hidden nodes for the remaining four experiments will be optimized in the same manner. This value will be varied experimentally based on the size of feature vector.

Table 4.2. Confusion matrix classified using 18 texture features.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**79.14**	6.30	2.47	7.41	0.86	1.23	2.59
2	6.50	**84.25**	4.25	2.13	1.25	1.63	0
3	1.92	13.46	**79.42**	1.92	1.35	1.92	0
4	15.59	9.15	1.19	**70.00**	1.19	0	2.88
5	8.70	0	0	4.35	**73.91**	13.04	0
6	5.26	0	5.26	0	0	**89.47**	0
7	1.79	0	0	0	2.56	0.77	**94.87**

From Table 4.2, the result of classification using 18 texture features is presented in the form of confusion matrix. Each row of the confusion matrix is a true class while each column represents the output class given by our classifier. For example, the element in the first row and third column is a percentage of the accuracy classified as Class 3 while in fact, it is Class 1. The diagonal of the matrix is, therefore, a correct classification for each class. From the table, the classification accuracies of Class 2, 6 and 7 are more than 80%, however, class 4 and 5 have quite low accuracy in comparison with the other classes since the misclassification of Class 4 as Class 1 and Class 5 as Class 6 are still high. On the average, the accuracy of a classification using 18 texture features is 80.76%.

4.3.2 Experiment 2 - Moments of Two-Dimensional Functions

Moments of two-dimensional functions or 2D-moments are used in the experiment because their ability to tolerate any transformation changes of images. They are calculated from CA images of DWT level 1 using Eq. (2.18). We split this experiment into 3 sub-experiments. First, we add in the feature vector each *n*-th order for *n* = 0, 1, . . . , 7 one by one to the existing 18 texture features. Second, only 0-th order to 7-th order are in the feature vector. And third, the 18 texture features combined with all *n*-th orders are in the feature vector.

Table 4.3. Texture features with one additional Zernike moments of order *n*.

***n*-th order**	**0**	**1**	**2**	**3**
Accuracy (%)	81.66	82.15	81.73	82.04
***n*-th order**	**4**	**5**	**6**	**7**
Accuracy (%)	82.58	80.52	78.56	81.78

Table 4.3 shows the result of the first sub-experiment where the 18 texture features are used in combination with each *n*-th order. The accuracy of classification is peaked at 82.58% when using texture features with the 4-th order of Zernike moments. However, in the second sub-experiment when the texture features were removed, the accuracy is reduced to 63.03%. Therefore, we have learnt that 2D-moments work better in the presence of 18 texture features.

Table 4.4. Confusion matrix classified using textures features and 2D-moments.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**81.73**	8.27	2.59	4.07	0	0.99	2.35
2	6.25	**85.13**	2.00	4.25	0	1.25	1.25
3	8.08	4.04	**84.62**	0.19	0	3.08	0
4	16.95	4.24	1.19	**73.39**	2.71	0.85	0.68
5	4.35	11.30	1.30	5.22	**66.96**	10.87	0
6	2.63	0	1.58	3.68	0	**92.11**	0
7	1.79	2.56	0	0	0	0	**95.64**

In the third sub-experiment when the 18 texture features and all 2D-moments are combined, we obtained more satisfiable result with the accuracy of 82.66%. This is better than using 18 texture features alone (see Experiment 1). The confusion matrix for this sub-experiment is presented in Table 4.4. In particular, the accuracy of Class 7 is increased to 95.64%. However, the accuracy of Class 5 is reduced to 66.96%. Therefore, we will explore another technique to improve these results in the next experiments.

4.3.3 Experiment 3 - Absolute FFT

Absolute FFT or abs-FFT computes texture features using the absolute magnitude of FFT. There are two sub-experiments. First, we select four best texture features which operate on this magnitude of FFT and add each of these features one by one to the existing 18 texture features. Second, the 18 texture features combined with all four abs-FFT features are in the feature vector.

Table 4.5. Texture features with one additional abs-FFT feature.

Additional feature	**Max**	**Average**	**Energy**	**Variance**
Accuracy (%)	81.30	85.44	80.74	82.55

Table 4.5 yields the result of the first sub-experiment. The best result is at 85.44% when using average-features in conjunction with the 18 texture features and this is superior than the result in Experiment 2. Therefore, it is still worth to use 18 texture features with our test features as the performance has been improved by 2.78%. The second sub-experiment gives

85.06% of the accuracy. This result is close to using the average-feature. However, the average-feature yields slightly higher percentage. The confusion matrix of the average-feature is, therefore, shown in Table 4.6. The accuracies in most classes are improved from Experiment 2. Especially, Class 4 and 5 we had problems before in Experiment 2, now Class 4 is improved by 1.02% while Class 5 is improved by 9.13%. The accuracy of Class 7 is now climbed up to 97.18%, and the average accuracy of Table 4.6 is 85.44%.

Table 4.6. Confusion matrix classified using texture features and the abs-FFT average-feature.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**85.41**	3.70	0.37	5.43	0.62	0.86	1.61
2	7.5	**85.75**	3.63	1.63	0.25	1.25	0
3	2.69	5.00	**90.00**	1.15	0.58	0.58	0
4	14.25	6.78	2.20	**74.41**	1.53	0.17	0.68
5	4.35	0.43	3.04	9.13	**76.09**	6.96	0
6	8.42	0	1.58	0.53	4.21	**84.74**	0.53
7	1.54	0	0	0.51	0.51	0.26	**97.18**

4.3.4 Experiment 4 - Logarithmic FFT

Logarithmic FFT or log-FFT closely resembles the abs-FFT but log-FFT uses the logarithmic magnitude of FFT. There are two sub-experiments. First, we use one additional log-FFT feature with 18 texture features. Second, we use all log-FFT features together with 18 textures features in order to compare which sub-experiments give better results.

Table 4.7. Texture features with one additional log-FFT feature.

Additional feature	**Max**	**Average**	**Energy**	**Variance**
Accuracy (%)	62.46	86.11	79.96	84.99

The result of the first sub-experiment is given in Table 4.7. The average of log-FFT feature outperforms the other three features. The best accuracy is at 86.11% which is higher than the abs-FFT average-feature in Experiment 3. In the second sub-experiment when we use all log-FFT features with 18 texture features, the overall accuracy is down to 78.47%. We believe that the max-feature and energy-feature are the plausible causes since their accuracies are not too high as shown before in Table 4.7. Therefore, only average-feature is recommended with the 18 texture features. The confusion matrix when using texture features and the log-FFT average-feature is presented in Table 4.8. The result reveals that most of the classes have the accuracies higher than Experiment 3 with the slight drop of Class 1 and Class 4 performances. However, Class 5 to 7 give rather excellent results, all are above 90%. The performances of

Class 5 and Class 6 are enhanced by 15.21% and 10%, respectively. Moreover, Class 7 accuracy is now 100%.

Table 4.8. Confusion matrix classified using texture features and the log-FFT average-feature.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**82.72**	4.94	1.23	4.94	2.47	1.23	2.47
2	6.25	**87.50**	3.75	1.25	0	1.25	0
3	3.85	5.77	**90.38**	0	0	0	0
4	10.17	8.47	3.39	**71.18**	3.39	0	3.39
5	0	0	0	0	**91.30**	8.70	0
6	0	0	5.26	0	0	**94.74**	0
7	0	0	0	0	0	0	**100**

Until now, the accuracy of Class 4 is still less than 75%. In the next experiment, we will show how our new technique of feature extraction can lead to a significant improvement of Class 4 and the rest of the remaining classes. We expect to have no less than 87% accuracy for every class.

4.3.5 Experiment 5 – FFT Projection on *x*-axis

The experiment uses the logarithmic magnitude of FFT presented in Algorithm 2 (*k*-FFTPX) along with the 18 texture features that are obtained from Algorithm 1. Algorithm 2 is called by Algorithm 1 when the step of appending optional features *H* to *F* is reached.

There are two sub-experiments. First, we investigate the suitable value of *k* in the *k*-FFTPX algorithm. Table 4.9 shows the accuracy of the *k*-FFTPX algorithm when *k* = 1, 5, 10, 15, and 20, respectively. The accuracy increases as the *k* value increases until it reaches a peak at *k* = 10 and the accuracy begins to decrease when *k* when *k* is greater than 10.

Table 4.9. Performance of *k*-FFTPX when *k* is varied.

***k* values**	***k* = 1**	***k* = 5**	***k* = 10**	***k* = 15**	***k* = 20**
Accuracy (%)	71.84	89.94	90.40	30.03	32.44

In the second sub-experiment, we apply *k* = 10 and derive the confusion matrix as shown in Table 4.10. Most of the correctly classified instances are now over 87% which are better than the previous four experiments. Moreover, the correctly classified instances of all the classes are higher than 80%. Furthermore, the accuracy of Class 4 is improved by 8.99%. The overall (average) accuracy is at 90.40% which is better than the results of the previous four experiments.

Table 4.10. Confusion matrix classified using texture features and 10-FFTPX.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**89.26**	1.85	0.12	7.28	0.12	0	13.58
2	2.38	**93.88**	1.23	1.25	0.12	1.25	0
3	2.88	4.42	**92.69**	0	0	0	0
4	11.53	6.78	0.68	**80.17**	0.85	0	0
5	1.74	0	4.35	1.74	**87.83**	4.35	0
6	6.32	0	0	0	0	**93.68**	0
7	0	0	0	2.05	0	0	**97.95**

4.3.6 Comparison of Each Feature Extraction Method

Table 4.11 shows the best results from Experiment 2 to 5 where four different feature extraction techniques are used in conjunction with the 18 texture features. First, when we append 2D-moments to the 18 texture features, we obtain the accuracy of 82.66% which comes from using textures features combined with eight features of Zernike moments.

Table 4.11. Each feature extraction method and its accuracy.

Features	**Accuracy (%)**
Texture + 2D moments	82.66
Texture + Abs-FFT	85.44
Texture + Log-FFT	86.11
Texture + *k*-FFTPX	90.40

Later, the abs-FFT average-feature improves the accuracy to 85.44%. Then, the log-FFT average-feature increases the result even further to 86.11%. Finally, the highest accuracy arises from using the 18 texture features with *k*-FFTPX where the confusion matrix shows the overall accuracy of 90.40%. This suggests that the 18 texture features with *k*-FFTPX is the most effective among those feature extraction techniques we presented. In addition, when the magnitude of FFT is plotted in the logarithmic scale, the magnitude differences are far more prominent than the scale of the absolute magnitude (see Figure 2.2(b) and (c), for example). Hence, the projection of log-FFT image on the *x*-axis works very well. Note that the suitable *k* value may require tuning and it can be varied from problems to problems. That is why we stated in the algorithm as *k*-FFTPX.

4.4 Running Time Analysis

In this section, we estimate the running time of our two algorithms which are the cloud classification algorithm and the *k*-FFTPX. The results are shown in Table 4.12.

Table 4.12. The approximate running times of the two proposed algorithms.

Algorithms	**Settings**	**Number of images**	**Approx. running time**
Cloud classification	LOOCV	353	14.1 min
	Training	353	1.7 min
	Testing	1	813.9 ms
The *k*-FFTPX	*k*=5	1	52.7 ms
	k=10		52.9 ms
	k=15		54.0 ms
	k=20		54.5 ms

The 18 texture features and the *k*-FFTPX (with the same setting as Section 4.3.5) are applied to the cloud classification algorithm. The running time of the *k*-FFTPX algorithm is around 52 ms which is slightly increased for the larger *k* value. The algorithm for the *k*-FFTPX is executed once for each image. Therefore, a multiplication with the number of images is requited to calculate the total running time.

4.5 Chapter Summary

Two algorithms were presented. The first algorithm computes a matrix of feature vectors for all images before using this information in the process of cloud classification. The second algorithm is provoked by the first one if the feature extraction technique is set to use *k*-FFTPX. None of the literatures in Table 4.1 has provided such algorithms before. Besides, our *k*-FFTPX is different from other FFT techniques in many aspects. We project the values of logarithmic magnitude of FFT images on the *x*-axis of the frequency domain and split the projection values into *k* blocks and take the average of the magnitude in each block to get *k* projection values whereas the past research has calculated the characteristics of FFT using CC, SI, or the radial average. We also add a sorting technique to sort values in the feature vector for a better performance. Although, there were research using DWT with FFT for word recognition and using DTCWT with FFT for EEG analysis; the merit of this research is that we used DWT with texture features in conjunction with the new *k*-FFTPX features for cloud type classification. By using merely a digital camera available anywhere on the market today, our method is inexpensive. Our next contribution is the increase of cloud types to seven different cloud types and yet a good result, 90.40% accuracy, was obtained. Among digital camera images used in the literatures (see Table 1.3), there was a result showing more than 90% accuracy; however, their output classes were limited to two cloud types. In this chapter, we also delivered more than one extraction technique; in fact, five different combinations of feature extraction techniques were presented and the accuracies are 80.76%, 82.66%, 85.44%, 86.11%, and 90.40%, respectively. These results, therefore, suggest a variety of practical solutions from the simple to the sophisticated functionality that requires no satellite images or expensive tools. Note that our approach will be developed until it can combine with other inexpensive meteorological sensors to report weather conditions and display them on a smart phone.

In addition to the practical advantages above, our algorithm is rather simple to implement while the accuracy of our proposed method was improved from the conventional methods with the capability of distinguishing cloud types of up to seven classes. By using simply a digital camera with a resolution of at least 640 × 480 pixels, our method is less cumbersome and less expensive than those using TSI/WSI imagers. However, our method cannot classify more than one type of clouds that appear on the same image. Also, our algorithm has not yet been tested with the night time images. Our algorithm must be used in conjunction with the texture features to achieve the best performance because in Chapter 3 we proved that texture features are necessary as they convey key characteristics of clouds. Since the *k*-FFTPX provides the results better than the other three feature extraction techniques; therefore, we will use the *k*-FFTPX as the main feature extraction technique in the next chapter. However, we will further improve any drawbacks of the *k*-FFTPX especially in the area of a computation time while the good characteristics will be maintained.

CHAPTER 5

EXPERIMENT C – VALIDATING PROPOSED FEATURES AND NOVEL ALGORITHMS

In the last chapter, the *k*-FFTPX has proven to perform better than other feature extraction techniques. However, the computation time of the *k*-FFTPX can still be improved. In this chapter, we modify and bring in three more features based on the previous *k*-FFTPX. The classification accuracy will be increased from 90.40% because we use the new algorithms based on hierarchical classification. We still perform cloud classification on seven cloud types, namely, cirriform, high cumuliform, stratocumulus, cumulus, cumulonimbus, stratiform, and clear sky. In the chapter, we present eight algorithms that are used in automatic cloud classification with ground-based images as input. The main Cloud Classification Tree Algorithm (CCTA) uses the technique called hierarchical classification which is composed of three levels of tree. The design of our tree helps reduce the number of competitions among the cloud classes. We show that this method provides the highest accuracy at 98.08% through a series of four experiments. The result confirms that the hierarchical classification performs better than a single classification.

In addition, the tree can be adapted to classify lesser number of cloud types. Our experiment reveals that the accuracy for classifying two classes, cloud and no-cloud, is high as 100%. Moreover, users have freedom to specify their expected accuracy to gain higher speed in calculation.

5.1 Introduction

In this chapter, both texture and Fourier transform features are extracted from images. Then, ANN is used for training and classifying instances based on Cloud Classification Tree Algorithm (CCTA). Our classification tree splits sky conditions into clouded sky and clear sky before further separating clouded sky into three forms of clouds which are cirriform, cumuliform, and stratiform. Then, cumuliform clouds with four different shapes are classified last. Along with the classification tree, we also proposed three new features which are the modified *k*-FFTPX, the half *k*-FFTPX, and the *h* × *k*-FFT. Eight algorithms are introduced and their validated results are presented in our four experiments.

5.2 Technical Background

In addition to Table 1.1 – 1.3, Table 5.1 shows specifically various uses of hierarchy classification. Hansen *et al*. [55] used hierarchical tree structure, a decision tree that applies training data to generate tree structure using pruning method to separate data into two sets. One data set is used to grow the tree and the other is used to prune errors. This method classifies satellite images (AVHRR) into one of twelve classes for land cover classification. Polat and Güneş [56] used a decision tree classifier and FFT based on Welch method to classify EEG signals as either patient or normal. Their method is used to detect epileptic seizure. Pang *et al*. [57] introduced a binary classification tree algorithm for face membership authentication. Their classification tree is constructed by clustering one data set from the root node into two subsets.

Then, clustering procedure is repeated on both child nodes. The procedure terminates when all nodes remain only either membership or non- membership data. The result from each node of tree is trained by SVM and all SVM classifiers are combined into SVM classification tree. From what mentioned above, the tree structure is generated automatically from training data.

Table 5.1. Related works on classification tree and decision tree.

Year	**Proposed method**	**Application**	**Author**
2000	Hierarchical tree structure	Land cover classification	Hansen *et al.* [55]
2000	Hierarchical SVM structure	Cloud classification	Azimi-Sadjadi and Zekavat [58]
2005	SVM classification tree	Face authentication	Pang *et al.* [57]
2007	Decision tree and FFT	Detection epileptic seizure	Polat and Güneş [56]

However, some researchers prefer to construct tree based on characteristics of each class. Azimi-Sadjadi and Zekavat [58] used satellite images (GOES 8) IR channel to classify areas into ten classes which composed of six cloud classes and four no-cloud classes. They used SVM classifier together with mean and standard deviation features to classify each block of image. Hierarchical SVM structure was then formed to classify at most two classes on each level of hierarchy. Moreover, Parikh [17] suggested that hierarchical classification leads to better results than using solely a single classification. To the best of our knowledge, apart from SVM classifiers there is no research on hierarchy classification using ANN classifiers. In this paper, we will design tree structure and use ANN to classify at most four classes on each level of hierarchy. Other research on SVM used only binary classes (two classes) per hierarchical level.

5.2.1 Features

Two color models, namely, RGB and HSV (Hue, Saturation, and Value) are used for feature extraction. In this paper, digital images are in RGB color while HSV color codes are computed by equations below [59]. The notation R′, G′ and B′ are R, G, and B which are scaled to [0, 1]. The H, S, and V values are all varied in a range of [0, 1]. The H values corresponds to colors varying from red through yellow, green, cyan, blue, magenta, and back to red, so at 0 and 1 they are both red. The S values are varied so that the corresponding H colors change from unsaturated to fully saturated (no white component) whereas the V values are brightness values in scale of [0, 1].

$$V = \max(R', G', B') \tag{5.1}$$

$$X = \max(R', G', B') \tag{5.2}$$

$$S = \begin{cases} 0 & if\ V = X \\ \frac{V-X}{V} & if\ V \neq X \end{cases} \tag{5.3}$$

$$H = \begin{cases} 0\ if\ V = X \\ 5 + \frac{V-B'}{V-X}\ if\ V = R'\ and\ X = G' \\ 1 - \frac{V-G'}{V-X}\ if\ V = R'\ and\ X \neq G' \\ 1 + \frac{V-R'}{V-X}\ if\ V = G'\ and\ X = B' \\ 3 - \frac{V-B'}{V-X}\ if\ V = G'\ and\ X \neq B' \\ 3 + \frac{V-G'}{V-X}\ if\ V = B'\ and\ X = R' \\ 5 - \frac{V-R'}{V-X}\ if\ V = B'\ and\ X \neq R' \end{cases} \tag{5.4}$$

We will use a grayscale image which is computed by splitting channels of image as R, G, B, H, S or V channels in feature extraction. There are two groups of features used in the experimental section which are the texture features and the new features based on Fourier transform. These features are used together because no single feature extraction method is best suited for recognizing all classes [10]. Each method has its own merits.

1) Texture Features

In this chapter, there are five types of texture features depending on data sources used in the calculation. The first type is when texture features are extracted straight from images. These features are mean (*ME*) of R, G and B, difference of mean (*Di j*) between each channel, and uniformity (*U*) of R defined by Eq. (2.1) to Eq. (2.2) and Eq. (2.4) [13], [15] in Chapter 2.

The second type is when texture features are calculated from GLCM which is a square matrix [15]. We will use four of Haralick texture features [37]. These type of features are homogeneity (HOM) of B, contrast (*CON*) of B, energy (*EN*) of B and entropy (*ENT*) of R, G, B, and S as depicted by Eq. (2.5) to Eq. (2.7) and Eq. (2.13).

The third type of texture feature is computed from edge of image which is calculated by canny edge detection [60] on R channel. The number of edge pixels is different for each cloud class; hence it can be used to distinguish clear sky and stratiform clouds from other cloud classes. The *ENT* (Eq. (2.13)) is calculated again, but on edge of image, this texture feature is called entropy of edge image (*EE*). The sum of edge pixels (*SE*) is calculated by equation below. The annotation *e*(i, j) is a pixel value from the i-th row and the j-th column while A and B are the size of edge image A × B pixels, width and height, respectively.

$$SE = \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} e(i,j) \tag{5.5}$$

The fourth type of texture feature is computed from binary image or threshold image on S channel. Threshold algorithm is used, if pixel value is less than threshold value then the pixel is set to one otherwise it is set to zero. The remaining pixels in the image are cloud pixels and we can use them to separate cloud types by considering the number of cloud pixels for each type. The sum of cloud pixels (*SC*) using threshold image is calculated by equation below. The annotation $c(i, j)$ is a pixel value from the i-th row and j-th column.

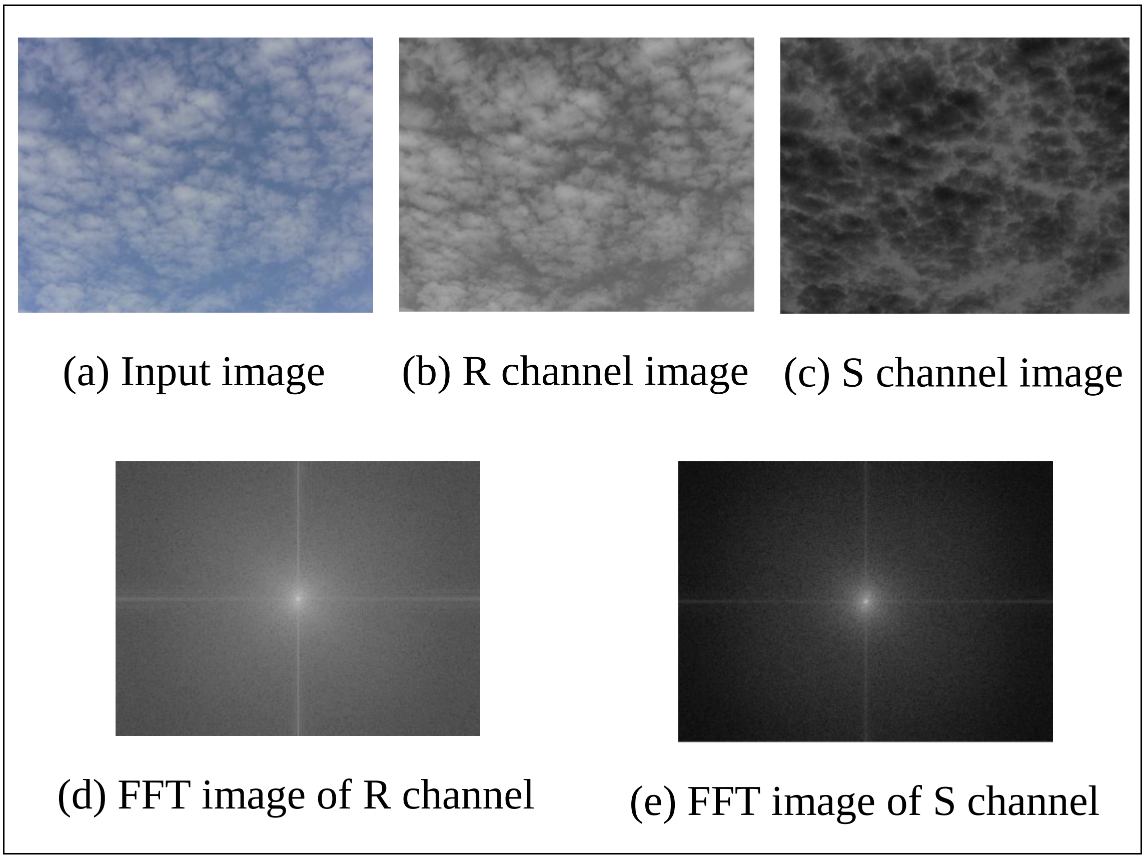
$$SC = \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} c(i,j) \tag{5.6}$$

The fifth type of texture feature is computed from two gradient images on S channel using Eq. (5.7) called energy of image gradient (*EG*). The notation I_x and I_y describe image gradients of row and column directions, respectively [61]. This feature is used to measure sharpness of grayscale image.

$$EG = \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} \left(I_x^2(i,j) + I_y^2(i,j)\right) \tag{5.7}$$

2) Feature Based on Fourier Transform

From Figure 5.1, we take input RGB image (Figure 5.1(a)) and transform it to grayscale image by splitting channel into R, G, or B channel. The sample of grayscale image from R channel is shown in Figure 5.1(b). The channel splitting transforms HSV image to H, S or V image in grayscale. The S channel image is shown in Figure 5.1(c). We transform pixels of the grayscale image into frequency domain by 2D-FFT and we use FFT shift to move the low frequency pixels into the center of the image (see Figure 5.1(d) and Figure 5.1(e)).



(a) Input image

(b) R channel image

(c) S channel image

(d) FFT image of R channel

(e) FFT image of S channel

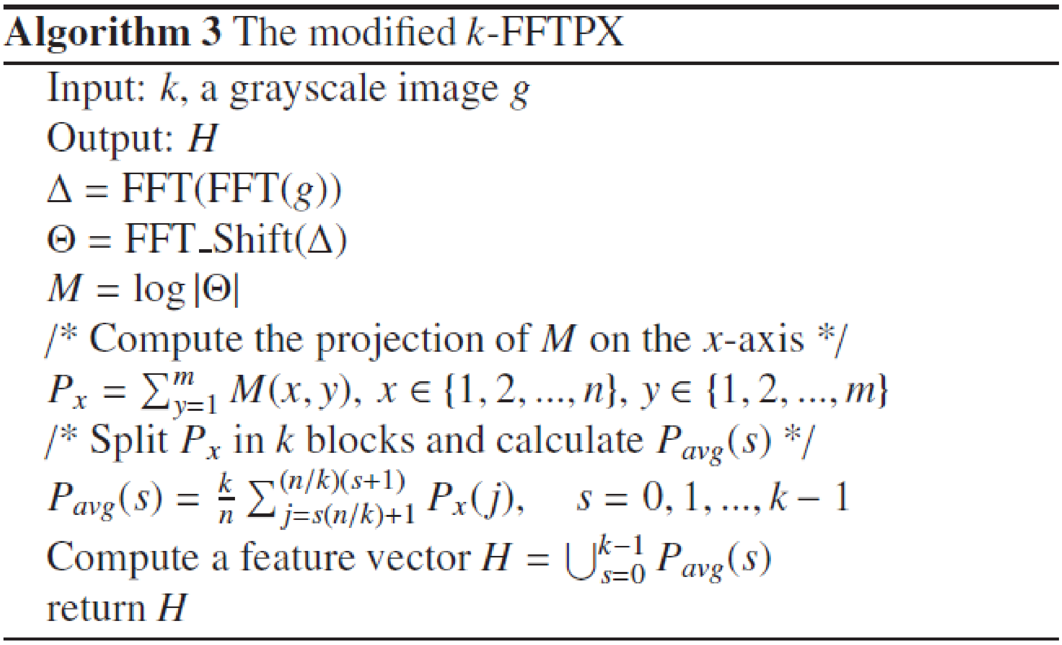
Figure 5.1. A comparison of Fourier transform image of R and S channels.

In Figure 5.1(b) the white color in cloud areas means that the areas have high red value while Figure 5.1(c) shows the complement of the image which presents a purity of pixels. The pixels with the highest purity have the highest values and are represented as white. The non-white color in cloud areas corresponds to a mixture of colors. Furthermore, the S image provides more details or better contrast than the R image. When both images are transformed to FFT images, the FFT image of S channel (Figure 5.1(e)) also has better contrast than the FFT image of R channel (Figure 5.1(d)). Therefore, we will use the FFT image of S channel and the verification of results will be presented later in the experimental section.

The shape of clouds cannot be explained by texture features alone. Therefore, we introduced features based-on Fourier transform for differentiating the shape of clouds. Moreover, it helps reduce the effect of unequal brightness in cloud images. There are three types of features based-on Fourier transform that we exploit in the experiment, namely, the modified *k*-FFTPX, the half *k*-FFTPX, and the *h* × *k*-FFT.

2.1) The modified *k*-FFTPX

The original *k*-FFTPX (in Chapter 4) used coefficient approximated image from DWT to transform the image to frequency domain using 2D-FFT. The projection of logarithmic magnitude of Fourier transform image on the *x*-axis was used. Then, we chose *k* uniform sampling values of the projection data as *k* dimensions of a feature vector. All steps are repeated twice and the second feature vector found was concatenated to the first one. To reduce computation time, the modified *k*-FFTPX is proposed (see Algorithm 3). The DWT is no longer used in the algorithm; a simple grayscale image is used instead. Moreover, it does not repeat the step of finding the second *k* dimensions of a feature vector. Only the first feature vector is required.

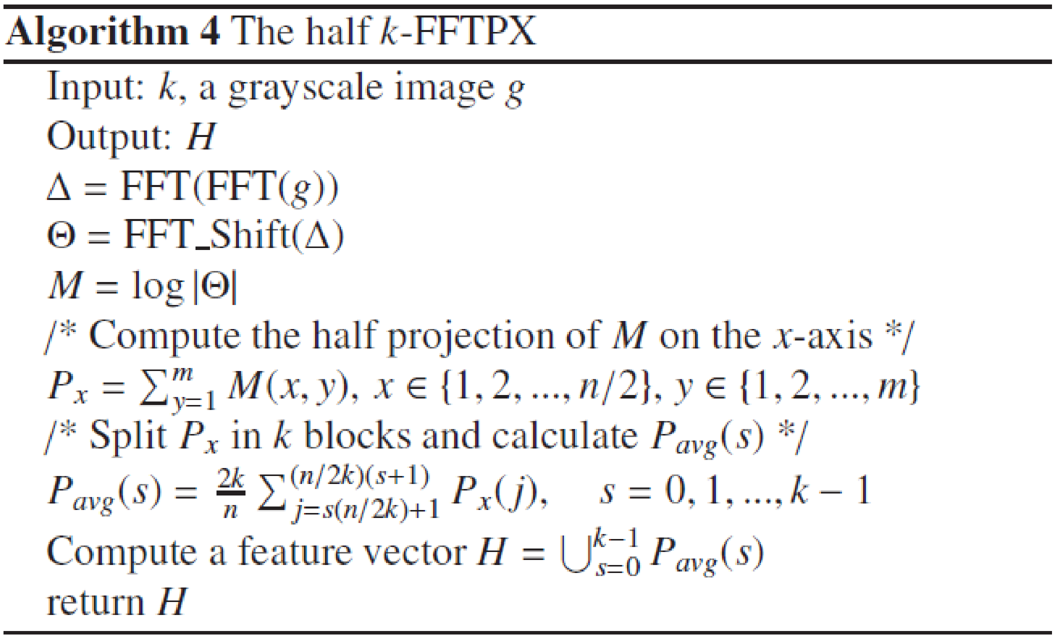


Algorithm 3 The modified k-FFTPX

```
Input: $k$, a grayscale image $g$
Output: $H$
$\Delta = \text{FFT}(\text{FFT}(g))$
$\Theta = \text{FFT\_Shift}(\Delta)$
$M = \log|\Theta|$
/* Compute the projection of $M$ on the $x$-axis */
$P_x = \sum_{y=1}^{m} M(x,y),\ x \in \{1,2,...,n\},\ y \in \{1,2,...,m\}$
/* Split $P_x$ in $k$ blocks and calculate $P_{avg}(s)$ */
$P_{avg}(s) = \frac{k}{n} \sum_{j=s(n/k)+1}^{(n/k)(s+1)} P_x(j), \quad s = 0,1,...,k-1$
Compute a feature vector $H = \bigcup_{s=0}^{k-1} P_{avg}(s)$
return $H$
```

2.2) The half *k*-FFTPX

This feature is similar to the modified *k*-FFTPX, but half *k*-FFTPX operates with a half projection of logarithmic magnitude of Fourier transform image on the *x*-axis to reduce calculation and processing time even further (See Algorithm 4).

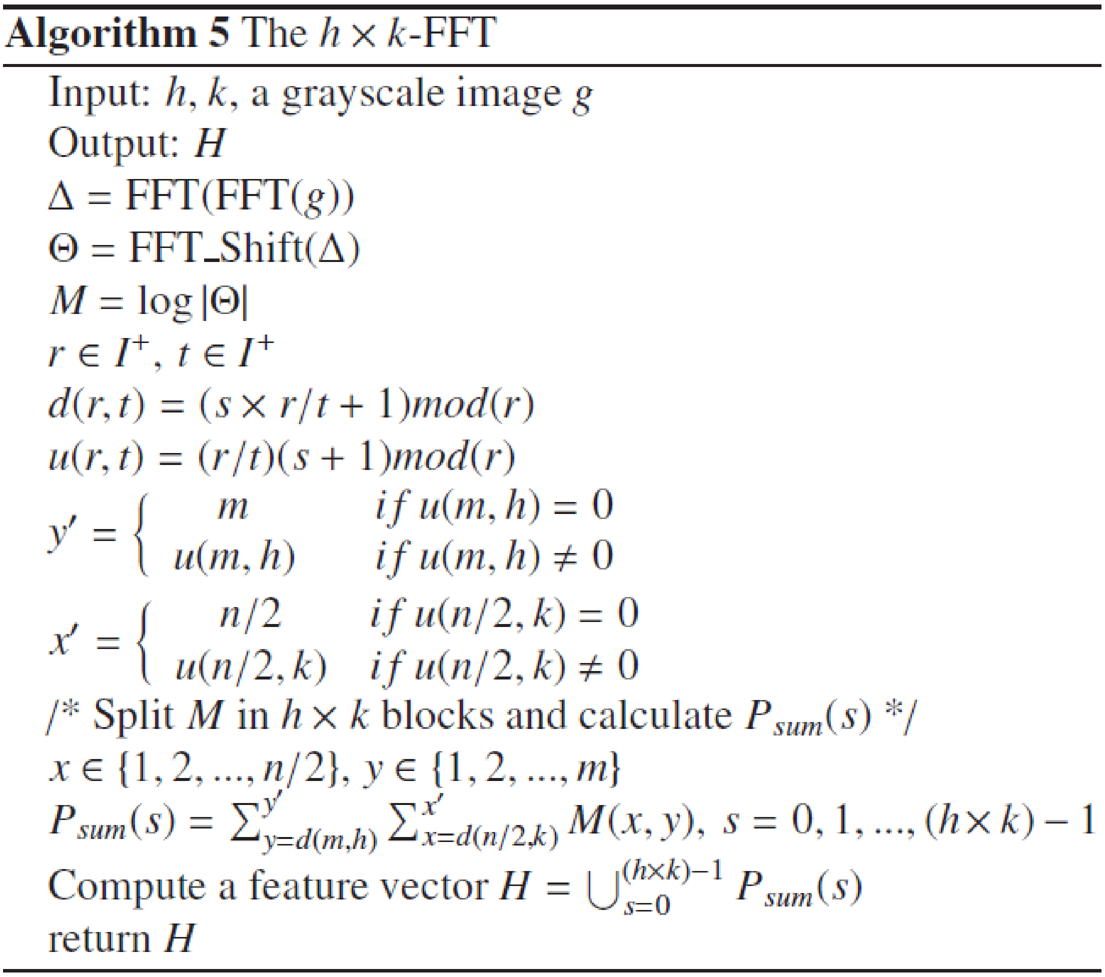


Algorithm 4 The half k-FFTPX

```
Input: $k$, a grayscale image $g$
Output: $H$
$\Delta = \text{FFT}(\text{FFT}(g))$
$\Theta = \text{FFT\_Shift}(\Delta)$
$M = \log|\Theta|$
/* Compute the half projection of $M$ on the $x$-axis */
$P_x = \sum_{y=1}^{m} M(x,y),\ x \in \{1,2,...,n/2\},\ y \in \{1,2,...,m\}$
/* Split $P_x$ in $k$ blocks and calculate $P_{avg}(s)$ */
$P_{avg}(s) = \frac{2k}{n} \sum_{j=s(n/2k)+1}^{(n/2k)(s+1)} P_x(j), \quad s = 0,1,...,k-1$
Compute a feature vector $H = \bigcup_{s=0}^{k-1} P_{avg}(s)$
return $H$
```

2.3) The *h* × *k*-FFT

The *h* × *k*-FFT is used to describe the shape of clouds in greater detail than the above two FFT features because it uses a sampling block technique on the FFT image. After logarithmic magnitude of Fourier transform image is calculated, we take half of the image and split it into *h* rows and *k* columns. This becomes the sub-images as depicted in Figure 5.2. The sum of pixel values in each sub-image is then calculated. Then, each sum value is concatenated as a feature vector $V = [v_1, v_2, v_3, ..., V_{h\times k}]$ (See Algorithm 5)



Algorithm 5 The $h \times k$-FFT

```
Input: $h$, $k$, a grayscale image $g$
Output: $H$
$\Delta = \mathrm{FFT}(\mathrm{FFT}(g))$
$\Theta = \mathrm{FFT\_Shift}(\Delta)$
$M = \log|\Theta|$
$r \in I^+, t \in I^+$
$d(r,t) = (s \times r/t + 1) mod(r)$
$u(r,t) = (r/t)(s + 1) mod(r)$
$y' = \begin{cases} m & if\ u(m,h) = 0 \\ u(m,h) & if\ u(m,h) \neq 0 \end{cases}$
$x' = \begin{cases} n/2 & if\ u(n/2,k) = 0 \\ u(n/2,k) & if\ u(n/2,k) \neq 0 \end{cases}$
/* Split $M$ in $h \times k$ blocks and calculate $P_{sum}(s)$ */
$x \in \{1, 2, ..., n/2\}, y \in \{1, 2, ..., m\}$
$P_{sum}(s) = \sum_{y=d(m,h)}^{y'} \sum_{x=d(n/2,k)}^{x'} M(x,y),\ s = 0, 1, ..., (h \times k) - 1$
Compute a feature vector $H = \bigcup_{s=0}^{(h\times k)-1} P_{sum}(s)$
return $H$
```

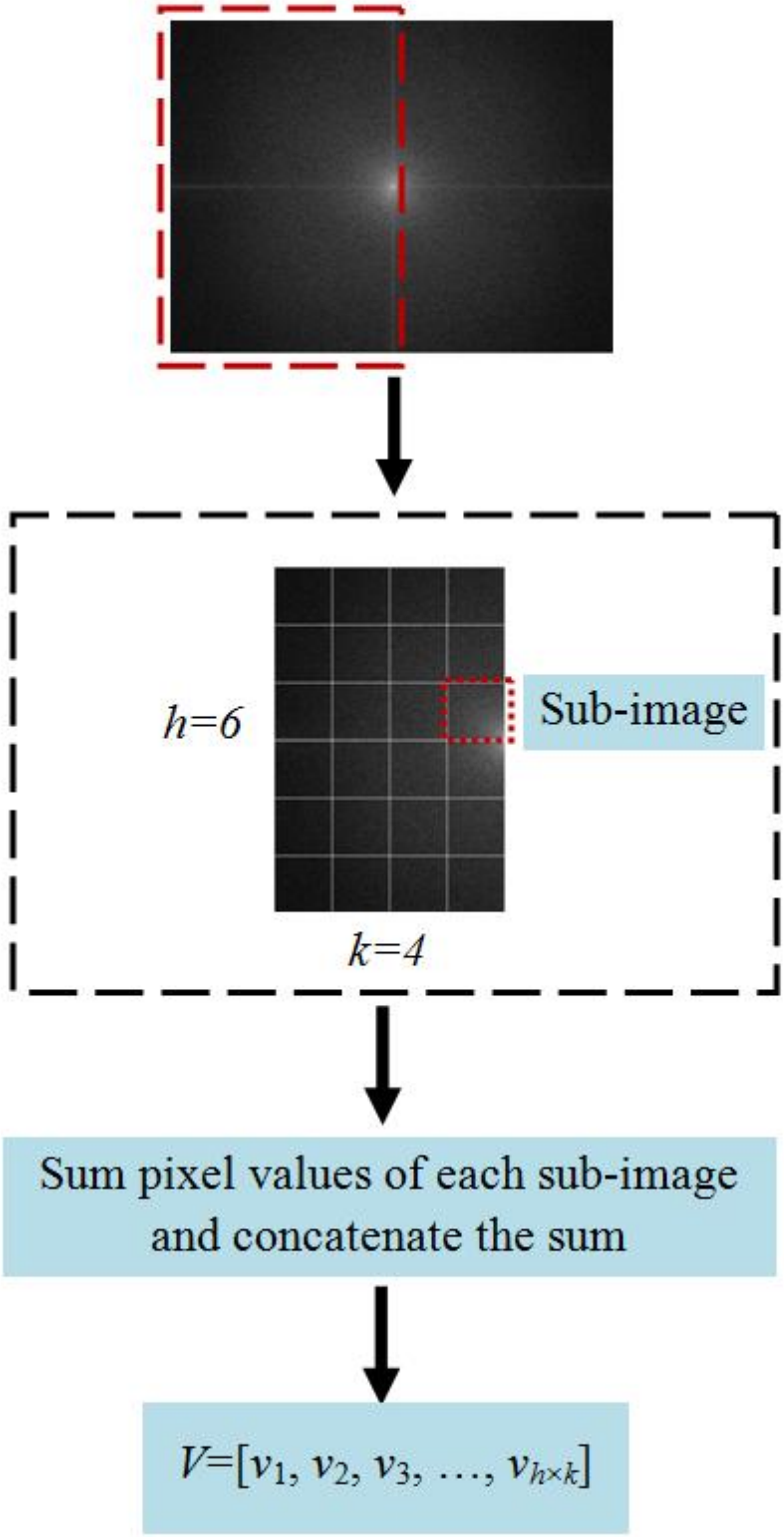


Figure 5.2. The *h* × *k*-FFT diagram.

Both the half *k*-FFTPX and the *h* × *k*-FFT used the left half of logarithmic magnitude of Fourier transform image for feature extraction. Since Fourier transform is the process of signal transformation into series of sine and cosine which are odd and even function respectively; therefore, we can use only half image for calculated features. These novel features are referred to as features from the half of FFT image, or half-FFT for short.

5.2.2 Algorithms

We introduce the algorithm for cloud type classification based on classification tree. Figure 5.3 shows a hierarchy classification of clouds in a tree-like fashion. There are three levels of classification. In each level of classification, instances are classified at the internal node (black circle in Figure 5.3) using ANN classifier. Different features are used depending on the types of clouds being classified. For seven types of cloud, their subtypes and their descriptions we recommend readers to revisit Table 2.1. Level 1, clouded sky and clear sky are separated by the texture features. At this level, we select the most suitable five features introduced in Section 5.2.1. Level 2 deals with three different forms of clouds which are cirriform, cumuliform, and stratiform. At this level, the modified *k*-FFTPX, the half *k*- FFTPX, and the *h* × *k*-FFT are tested and the best performed feature is used to distinguish the three forms of clouds. Level 3, cumuliform clouds with different shapes are classified using two texture features and one of the best performed feature from the modified *k*-FFTPX, the half *k*-FFTPX or the *h* × *k*-FFT. Note that leaves of the tree are the result of classification.

If we consider the computation time, it is not necessary the case that the single classification (one level) for seven cloud types will be faster than the hierarchical classification. This is because if the result is clear sky, only the first level of classification tree is executed and this will take less time than classifying of seven cloud types all at once. Similarly, using two levels of tree can classify cirriform and stratiform faster than the single classification in Chapter 4 (see Table 4.12 and 5.12). Although using all three levels of tree may take longer to classify seven cloud types than the single classification, the results from the cloud classification tree is more accurate. This is a simple trade-off between accuracy and computation time.

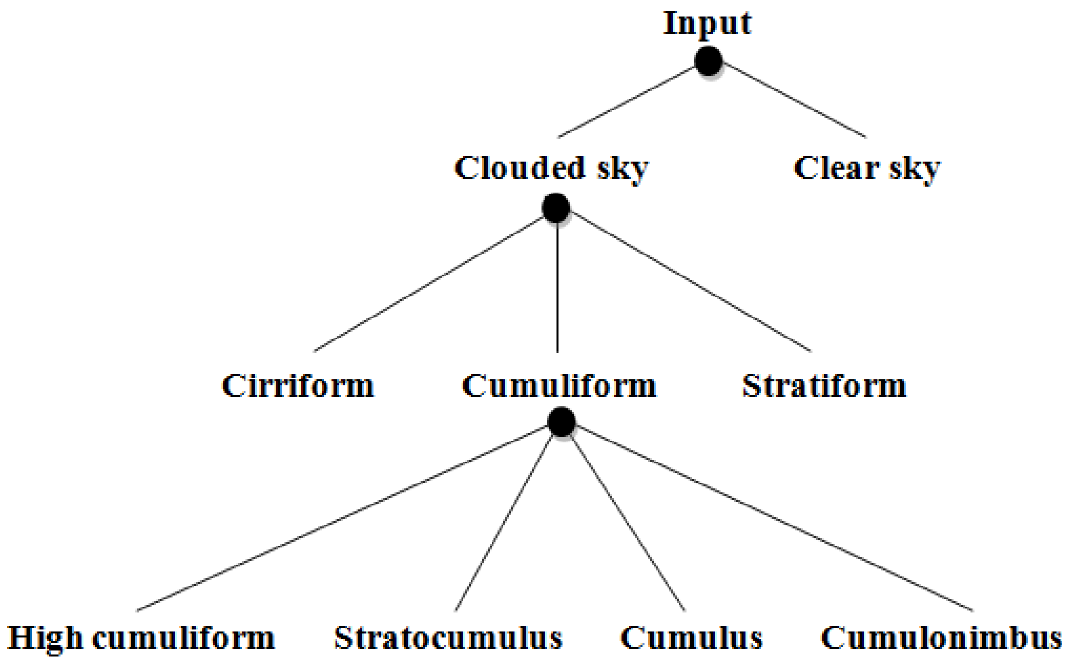
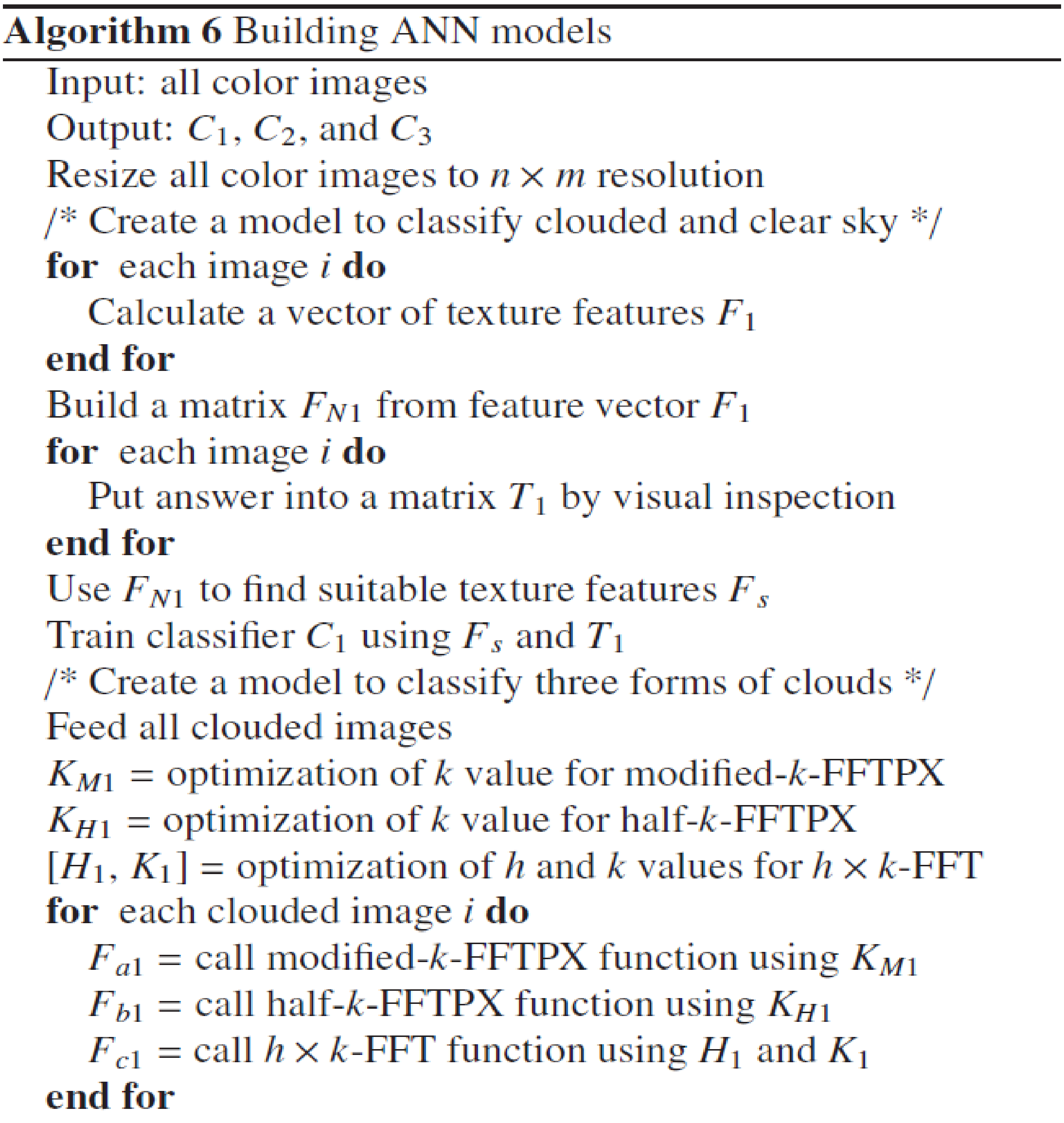


Figure 5.3. Cloud classification tree.

There are two main algorithms which are used to construct ANN models and to classify instances based on the classification tree. To construct ANN models, Algorithm 6 is presented. The color ground-based images are scaled to 240 × 320 pixels before transforming to grayscale images for feature extraction. The texture features described in Section 5.2.1 are selected by Algorithm 7 and ANN model for Level 1 is then constructed. After testing the three FFT features, ANN model for Level 2 is trained by the best performed FFT feature. Note that the optimization of k value for the first two FFT features uses Algorithm 8 while the optimization of h and k values for the h × k-FFT uses Algorithm 9. The three FFT feature vectors are computed using Algorithm 3, 4, and 5, respectively.



Algorithm 6 Building ANN models

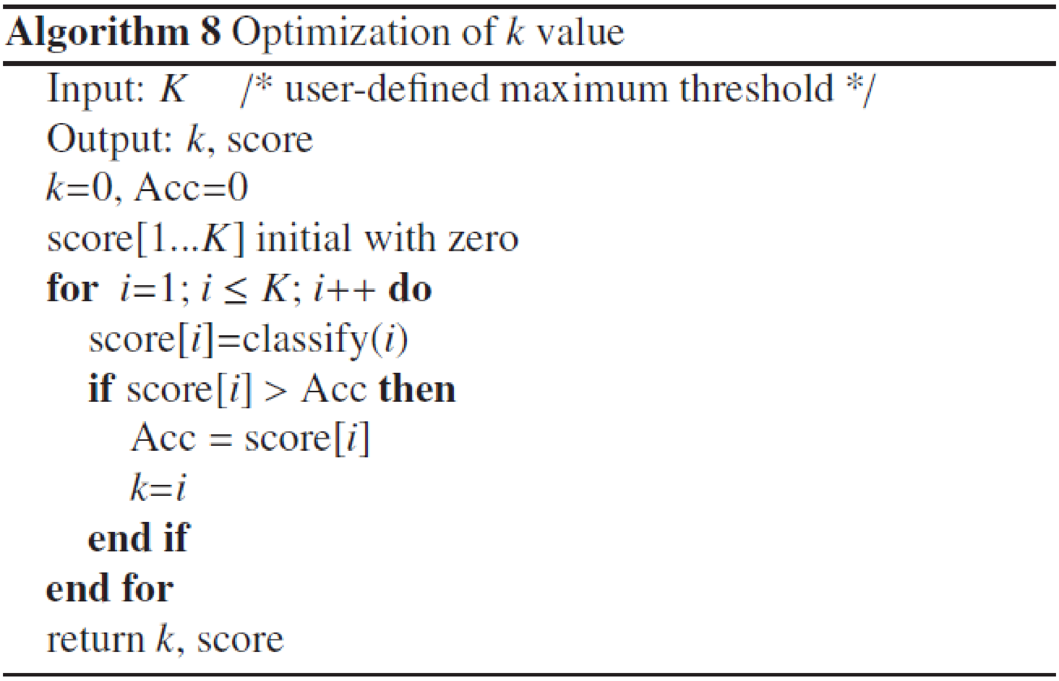
```
Input: all color images
Output: $C_1$, $C_2$, and $C_3$
Resize all color images to $n \times m$ resolution
/* Create a model to classify clouded and clear sky */
for each image $i$ do
    Calculate a vector of texture features $F_1$
end for
Build a matrix $F_{N1}$ from feature vector $F_1$
for each image $i$ do
    Put answer into a matrix $T_1$ by visual inspection
end for
Use $F_{N1}$ to find suitable texture features $F_s$
Train classifier $C_1$ using $F_s$ and $T_1$
/* Create a model to classify three forms of clouds */
Feed all clouded images
$K_{M1}$ = optimization of $k$ value for modified-$k$-FFTPX
$K_{H1}$ = optimization of $k$ value for half-$k$-FFTPX
$[H_1, K_1]$ = optimization of $h$ and $k$ values for $h \times k$-FFT
for each clouded image $i$ do
    $F_{a1}$ = call modified-$k$-FFTPX function using $K_{M1}$
    $F_{b1}$ = call half-$k$-FFTPX function using $K_{H1}$
    $F_{c1}$ = call $h \times k$-FFT function using $H_1$ and $K_1$
end for
```

```
Build a matrix $F_{K1}$ from feature vector $F_{a1}$
Build a matrix $F_{H1}$ from feature vector $F_{b1}$
Build a matrix $F_{HK1}$ from feature vector $F_{c1}$
**for** each clouded image $i$ **do**
    Put answer into a matrix $T_2$ by visual inspection
**end for**
$F_{M1}$ = testFeatures($F_K$, $F_H$, $F_{HK}$)
/* testFeatures return the best FFT feature */
Train classifier $C_2$ using $F_{M1}$ and $T_2$
/* Create a model to classify four cumuliform clouds */
Feed all cumuliform images
$K_{M2}$ = optimization of $k$ value for modified-$k$-FFTPX
$K_{H2}$ = optimization of $k$ value for half-$k$-FFTPX
$[H_2, K_2]$ = optimization of $h$ and $k$ values for $h \times k$-FFT
**for** each cumuliform image $i$ **do**
    $F_{a2}$ = call modified-$k$-FFTPX function using $K_{M2}$
    $F_{b2}$ = call half-$k$-FFTPX function using $K_{H2}$
    $F_{c2}$ = call $h \times k$-FFT function using $H_2$ and $K_2$
    Calculate a vector of texture features $F_2$
**end for**
Build a matrix $F_{K2}$ from feature vector $F_{a2}$
Build a matrix $F_{H3}$ from feature vector $F_{b2}$
Build a matrix $F_{HK2}$ from feature vector $F_{c2}$
Build a matrix $F_T$ from feature vector $F_2$
**for** each cumuliform image $i$ **do**
    Put answer into a matrix $T_3$ by visual inspection
**end for**
$F_{M2}$ = testFeatures($F_{K2}$, $F_{H2}$, $F_{HK2}$)
Use $F_T$ and $F_{M2}$ to find suitable texture features $F_{N2}$
$F_P$ = appendFeatures($F_{N2}$, $F_{M2}$)
Train classifier $C_3$ using $F_P$ and $T_3$
return $C_1$, $C_2$, and $C_3$
```

Algorithm 7 Selection of a suitable minimal set of features

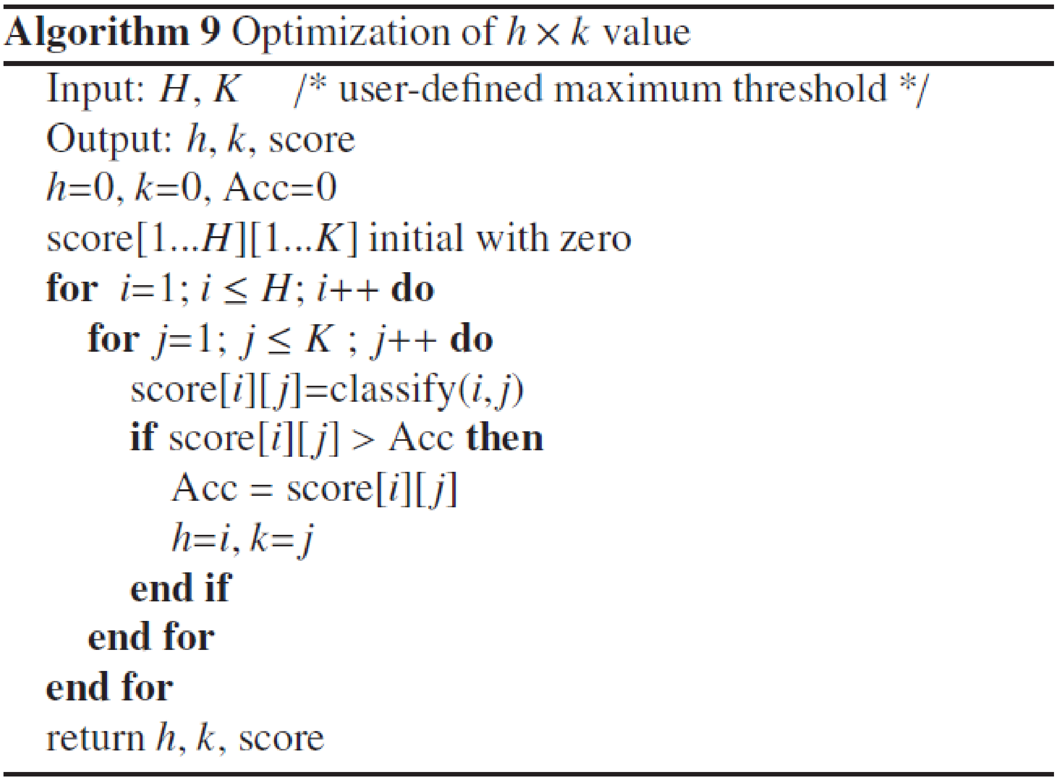
```
Input: $F[1...n]$, $H[1...(h \times k)][1...N]$, Acc
Output: result, resultIdx, score
/* $F$ is a vector of $n$ texture features, $H$ is a matrix of feature
vectors, Acc is a user-defined accuracy */
result = an empty sequence
resultIdx = an empty sequence
score$[1...n][1...n]$ initial with zero     /* record % accuracy */
temp = $F$, seq$[1...n]$, $c$=$n$
for $i$=1; $i \leq n$; $i$++ do
    seq[$i$]=$i$
end for
for $i$=1; $i \leq n$; $i$++ do
    for $j$=1; $j \leq c$; $j$++ do
        if $H$ is empty then
            score[$i$][$j$]=classify(result $\frown$ temp[$j$])
        else
            score[$i$][$j$]=classify(result $\frown$ temp[$j$],$H$)
        end if
    end for
    [maxVal, maxIdx]=max(score[$i$])
    result=result $\frown$ temp[maxIdx]
    resultIdx=resultIdx $\frown$ seq[maxIdx]
    Remove temp[maxIdx]
    Remove seq[maxIdx]
    $c$=$c$-1

    if maxVal $\geq$ Acc then
        return result, resultIdx, score
    end if
end for
return result, resultIdx, score
```



Algorithm 8 Optimization of k value

```
Input: $K$     /* user-defined maximum threshold */
Output: $k$, score
$k$=0, Acc=0
score[1...$K$] initial with zero
for  $i=1$; $i \le K$; $i$++ do
    score[$i$]=classify($i$)
    if score[$i$] > Acc then
        Acc = score[$i$]
        $k=i$
    end if
end for
return $k$, score
```



Algorithm 9 Optimization of $h \times k$ value

```
Input: $H$, $K$     /* user-defined maximum threshold */
Output: $h$, $k$, score
$h$=0, $k$=0, Acc=0
score[1...$H$][1...$K$] initial with zero
for  $i=1$; $i \le H$; $i$++ do
    for $j=1$; $j \le K$ ; $j$++ do
        score[$i$][$j$]=classify($i,j$)
        if score[$i$][$j$] > Acc then
            Acc = score[$i$][$j$]
            $h=i$, $k=j$
        end if
    end for
end for
return $h$, $k$, score
```

Algorithm 10 Cloud Classification Tree Algorithm (CCTA)

```
/* Test ANN models using the test data */
Input: all test images, $C_1$, $C_2$, $C_3$
Output: $C_M$
for each test image $i$ do
    Extract a feature vector $F_1$ for model $C_1$
    $R_1$ = Classify($F_1$, $C_1$)
    if $R_1$ is clear sky then
        $CS$ = Class No. 7
    else
        /* Three forms of clouds */
        Extract a feature vector $F_2$ for model $C_2$
        $R_2$ = Classify($F_2$, $C_2$)
        if $R_2$ is cirriform then
            $CS$ = Class No. 1
        else if $R_2$ is stratiform then
            $CS$ = Class No. 6
        else
            /* Cumuliform clouds */
            Extract a feature vector $F_3$ for model $C_3$
            $R_3$ = Classify($F_3$, $C_3$)
            if $R_3$ is high cumuliform then
                $CS$ = Class No. 2
            else if $R_3$ is stratocumulus then
                $CS$ = Class No. 3
            else if $R_3$ is cumulus then
                $CS$ = Class No. 4
            else
                $CS$ = Class No. 5
            end if
        end if
    end if
end for
Build a confusion matrix $C_M$ from $CS$
return $C_M$
```

To construct ANN model for Level 3, the best performed FFT feature is appended to the suitable set of texture features. The set of texture features are derived from Algorithm 7 using the 18 texture features and the best FFT feature vectors as the inputs. Finally, to classify instances based on the classification tree, CCTA is elaborated in Algorithm 10. The images are scaled and transformed in the same manner as described before in Algorithm 6. The same feature extraction methods which used to train ANN models are used to form feature vectors for classifying instances according to the classification tree in Figure 5.3.

5.3 Experimental Results

In the following experiments, we used 1,660 ground-based images from digital camera. Some cloud types are rare naturally-occurring types; hence the number of images per class is collected based on the frequency of cloud occurrence in nature. The classification of cloud images is very challenging because these images are taken from different views and come in different sizes but they all have at least 640 × 480 pixels in JPEG format. Therefore, we can scale all the images down to 320 × 240 pixels. LOOCV is used for result evaluation. For the setting parameter of ANN in these experiments, a hyperbolic tangent function is set as the activation function for both hidden and output layers. The activation function of input layer is set to linear so the input remains unchanged. The number of hidden layers is one. The number of hidden nodes is set to 9 for the first level of the tree and 11 for the second and third level. Learning rate and momentum are set to 0.01 and 0.9, respectively. We set stopping criteria when error in the training process reaches

0.001. All weights are initially fixed for our first three experiments. In the final experiment, random weights are used to achieve better classification performance. There are four separate experiments and the first three experiments match to the three levels of classification tree in Figure 5.3. In the first experiment, we classify instances into two classes either cloud or no-cloud (clear sky). In the second experiment, we classify three groups of clouds by considering their forms. In the third experiment, we classify four cloud classes by considering lumpy appearances of cumuliform clouds. Finally, all levels in the classification tree are combined together for the final classification.

5.3.1 Level 1 - Cloud or No-cloud

The set of *n* texture features are sent to Algorithm 7. In this experiment, *n* is set to 18 which refers to the 18 texture features are described in Section 5.2.1. Algorithm 7 is used to select a suitable minimal set of features.

Table 5.2 shows the five-iteration results from Algorithm 7 which keep in the score matrix when *Acc* or expected accuracy value is set to 100%. Each iteration, one suitable feature is selected from the feature set (*F*) that has a maximum outcome of accuracy. On the next iteration, the previously selected feature will be concatenated with the remaining features for another classification round. The iteration is repeated until the accuracy reaches the expected value.

Table 5.2. Accuracy (%) in selection of suitable features for classifying between cloud and no-cloud.

Features	**Iteration**				
	1	**2**	**3**	**4**	**5**
ME (R)	50	50	97.71	96.29	99.97
ME (G)	50	50	97.71	98.39	**100**
ME (B)	54.94	50	96.93	94.73	100
D (R-G)	52.03	50	99.03	99.48	99.77
D (R-B)	82.75	50	98.84	50	99.74
D (G-B)	81.81	50	98.65	**99.97**	-
U	69.83	49.97	98.28	92.91	99.77
CON	60.50	49.37	97.57	98.37	99.77
HOM	50.23	50	96.93	89.72	99.77
EN	50	86.75	97.38	95.86	99.77
ENT (R)	73.47	50	**99.19**	-	-
ENT (G)	50	50	97.76	50	99.77

Features	**Iteration**				
	1	**2**	**3**	**4**	**5**
ENT (B)	53.76	50	96.55	50	99.77
ENT (S)	49.97	50	97	96.41	99.77
EE	85.65	50	95.99	50	99.77
SE	85.44	**92.25**	-	-	-
SC	50	92.17	99.06	50	99.77
EG	**86.04**	-	-	-	-

As a result, we selected five texture features from Table 5.2 and referred to them as a suitable minimal set of features. These features in order of being added to the set (on each iteration) are *EG*, *SE*, *ENT* (*R*), *D* (*G* − *B*), and *ME* (*G*), respectively. Note that on the final iteration, *ME* (*B*) can also be chosen in place of *ME* (*G*) as they both yielded the same result. The five features are used in the process of building ANN model for Level 1 to classify clouded sky and clear sky in Algorithm 6. After executing Algorithm 10, we obtain the result for classifying cloud and no-cloud. The accuracy is as high as 100%. The advantage of Algorithm 7 is the flexibility of user-defined accuracy (*Acc*). For example, if *Acc* is expected at 92.25%, the algorithm will run only two iterations and only two features are used to classify cloud and no-cloud. Our algorithm is designed to have freedom to trade-off accuracy with computation time.

5.3.2 Level 2 - Three Forms of Clouds

The aim of this level is to classify three forms of clouds which are cirriform, cumuliform, and stratiform. In this experiment, features based on Fourier transform are used, namely, the modified *k*-FFTPX, the half *k*-FFTPX, and the *h* × *k*-FFT. However, only one of these features will be selected by Algorithm 6.

Table 5.3. Accuracy of the modified 20-FFTPX when channel is varied.

Channel	**R**	**G**	**B**	**H**	**S**	**V**
Accuracy (%)	74.30	63.23	61.22	79.39	81.25	61.64

Table 5.3 shows the accuracy of classification using the modified *k*-FFTPX for *k* equal to 20. Each color channel is tested in the feature extraction process. The accuracy of S channel is higher than the five other channels. This result confirms that the use of FFT image of S channel has paid off. Hence, we will use the information obtained from this channel for the remaining experiments.

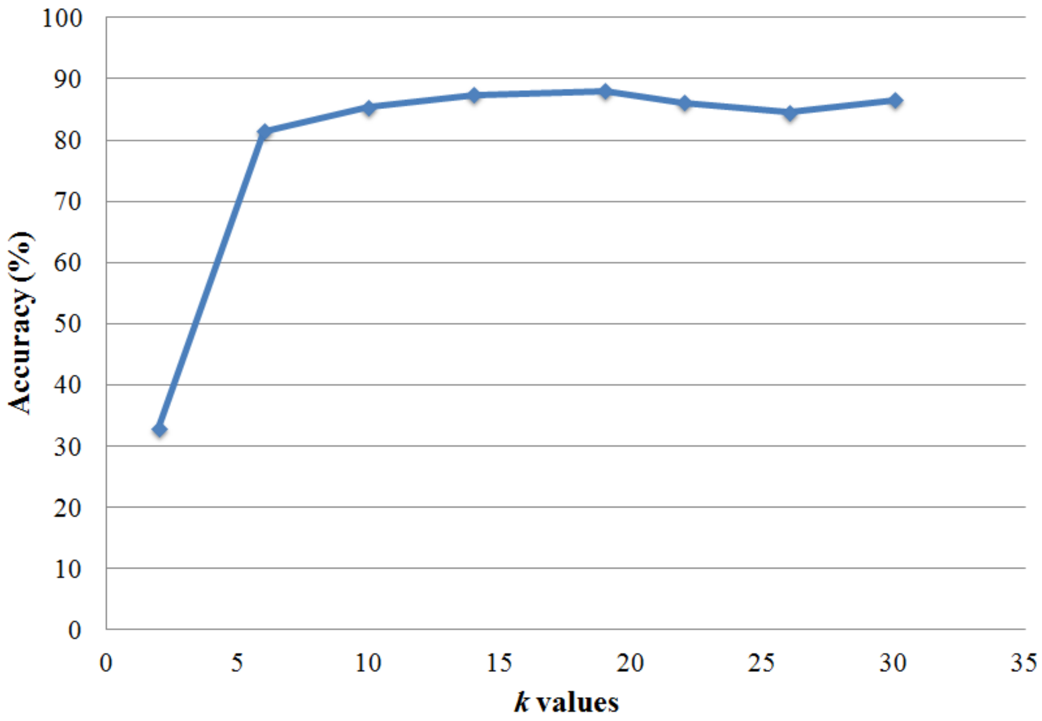


Figure 5.4. Accuracy of the modified *k*-FFTPX on S channel when *k* is varied.

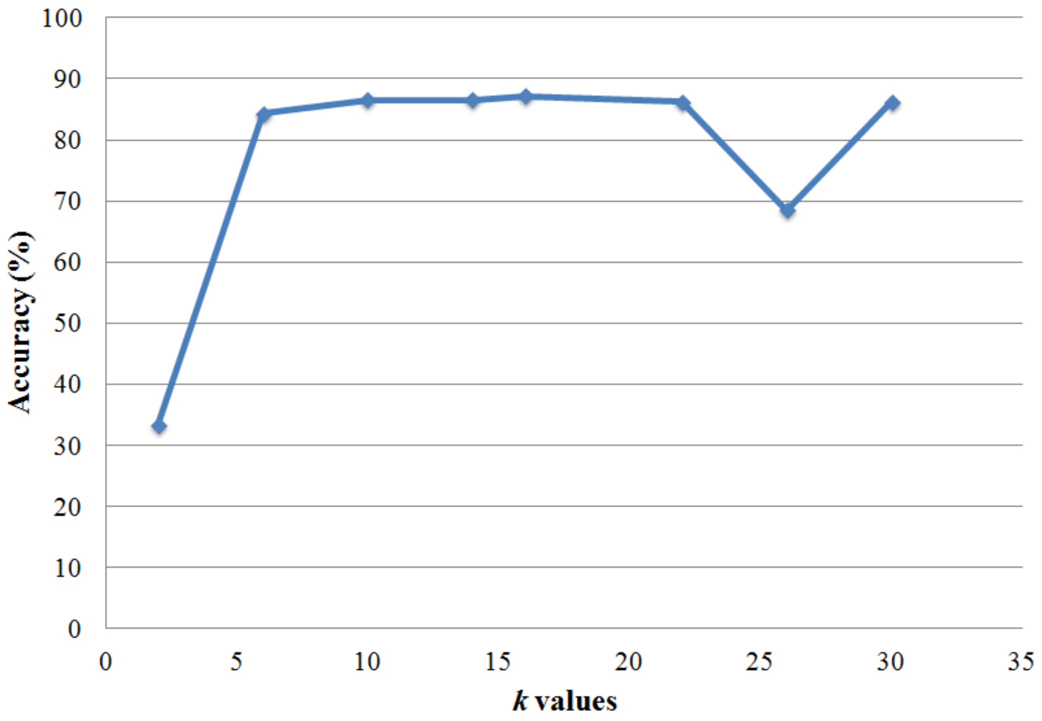


Figure 5.5. Accuracy of the half *k*-FFTPX on S channel when *k* is varied.

Table 5.4. Accuracy (%) of the *h* × *k*-FFT when *h* and *k* are varied (*k* = 1 – 6).

***h* value**	***k* value**					
	1	**2**	**3**	**4**	**5**	**6**
1	48.27	45.71	76.59	78.83	33.33	84.35
2	34.02	80.72	82.59	80.18	86.27	64.96
3	81.6	88.19	86.69	88.05	78.72	89.51
4	83.02	63.53	65.31	71.95	71.25	77.79
5	84.14	88.81	89.2	55.3	90.38	91.24
6	85.6	80.94	78.06	61.58	91.16	92.56
7	84.99	90.38	91.58	89.05	90.62	91.44
8	66.83	28.87	90.9	94.58	27.92	90.32
9	84.92	86.23	91.21	90.97	93.65	91.09
10	85.3	64.31	90.2	81.9	94.53	87.1
11	38.8	89.61	89.58	70.22	90.28	79.36
12	86.22	89.58	81.32	94.17	66.02	**95.7**

Table 5.5. Accuracy (%) of the *h* × *k*-FFT when *h* and *k* are varied (*k* = 7 – 12).

***h* value**	***k* value**					
	7	**8**	**9**	**10**	**11**	**12**
1	85.6	85.48	33.33	86.56	85.76	64.41
2	88.6	71.09	84.84	42.66	88.84	86.72
3	89.25	91.12	89.66	91.3	88.47	92.12
4	90.44	90.31	94.46	47.16	65.16	78.95
5	89.09	84.65	91.01	88.53	93.38	45.22
6	92.57	93.65	95.16	65.8	46.76	90.68
7	91.54	92.01	91.83	90.96	66.28	95.22
8	92.19	67.82	94.96	70.41	92.6	88.75
9	90.88	90.48	93.47	67.29	69.6	75.45
10	89.69	89.17	68.5	91.45	93.49	85.31
11	83.22	91.38	91.98	93.02	90.63	90.39
12	88.46	90.77	89.85	72.49	94.92	89.08

The optimization of *k* value for the first FFT feature, the modified *k*-FFTPX, is performed by Algorithm 8. Figure 5.4 shows the result of finding the optimized *k* value when *k* is varied from 1 to 30. The accuracy of this feature on S channel reaches the peak of 88.06% when *k* is 19. The *k* value for the second FFT feature, the half *k*-FFTPX, is also optimized by Algorithm 8. Figure 5.5 shows that the proper setting of *k* value for this feature is 16 which gives the accuracy of 87.20%. The optimization of *h* and *k* values for the third FFT feature, the *h* × *k*-FFT, is performed by Algorithm 9. Table 5.4 to 5.5 show that when *h* is equal to 12 and *k* is equal to 6, it yields the highest accuracy at 95.70%. This accuracy is more than the results of the modified *k*-FFTPX and the half *k*-FFTPX. Even with a smaller size (14 dimensions) of feature vector; *h* is equal to 7 and *k* is equal to 2, the method still provides 90.38% accuracy (see Table 5.4 to 5.5). Consequently, the *h* × *k*-FFT outperforms the two previous FFT features. At this level of classification tree, the best feature for distinguishing three forms of clouds has been identified as the *h* × *k*-FFT. The advantage of this approach is that users are not required to manually specify *h* and *k* values. Our algorithm will choose the optimized values for *h* and *k* automatically.

Table 5.6 shows the confusion matrix obtained from Algorithm 10. Class 1, 2 and 3 are cirriform, cumuliform and stratiform, respectively. The classification results of Class 2 and Class 3 are close to 100%. However, there is some misclassification of Class 1 as Class 2. This is because cirriform often occurs with cirrocumulus which belongs to Class 2. Nevertheless, the accuracy of Class 1 classification is still above

90%. The average accuracy of classification at this level is 95.70%.

Table 5.6. Confusion matrix classifying three forms of clouds using 12 × 6-FFT.

True class	**Classified as**		
	1	**2**	**3**
1	**90.95**	8.64	0.41
2	1.20	**98.80**	0
3	1.06	1.59	**97.35**

5.3.3 Level 3 - Shapes of Cumuliform Clouds

Considering lumpy appearances of cumuliform clouds, they can be rounded, patchy, puffy, and mushroom-like. The aim of this level is to classify four classes of cumuliform clouds which are high cumuliform, stratocumulus, cumulus, and cumulonimbus. Three features based on Fourier transform are tested in the same way as in Section 5.2.1. However, the best performed feature is also appended to a suitable minimal set of texture features computed by Algorithm 7.

Table 5.7. Accuracy (%) of the *h* × *k*-FFT when *h* and *k* are varied (*k* = 1 – 6).

***h* value**	***k* value**					
	1	**2**	**3**	**4**	**5**	**6**
1	44.06	57.31	58.5	63.15	30.4	65.98
2	44.83	57.47	62.63	63.89	66.96	67.08
3	56.19	69.61	74.75	76.63	78.41	78.89
4	48.8	71.95	59.46	75.73	78.64	71.98
5	57.27	73.28	78.61	78.18	25.57	29.04
6	64.69	75.37	79.75	63.74	39.05	68.3
7	68.98	54.44	77.81	75.84	79.96	71.57
8	70.7	75.59	61.01	35.2	78.13	84.15
9	69.14	76.15	77.71	80.55	81.1	81.54
10	71.1	77.91	46.05	77.46	24.81	87
11	60.48	77.5	78.65	79.53	81.85	85.38
12	70.23	53.86	82.71	84.07	78.34	84.04

Table 5.8. Accuracy (%) of the *h* × *k*-FFT when *h* and *k* are varied (*k* = 7 – 12).

***h* value**	***k* value**					
	7	**8**	**9**	**10**	**11**	**12**
1	67.57	69.55	69.36	62.09	56.83	62.1
2	57.7	68.15	69.08	69	70.18	70.07
3	77.04	78.06	82.78	47.93	83.4	78.44
4	71.68	53.41	63.72	25.95	77.11	79.95
5	78.91	63.24	80.99	62.15	80.22	84.33
6	52.87	76.14	83.14	72.48	81.09	87.53
7	66.54	80.23	80.76	80.52	84.97	51.77
8	82.07	82.51	82.28	88.31	84.23	84.09
9	84.19	82.83	53.11	**89**	87.45	85.16
10	79.41	77.2	85.07	85.24	88.41	84.21
11	81.97	83.27	85.05	88.17	86.61	84.96
12	71.49	84.42	88.89	88.73	85.95	87.45

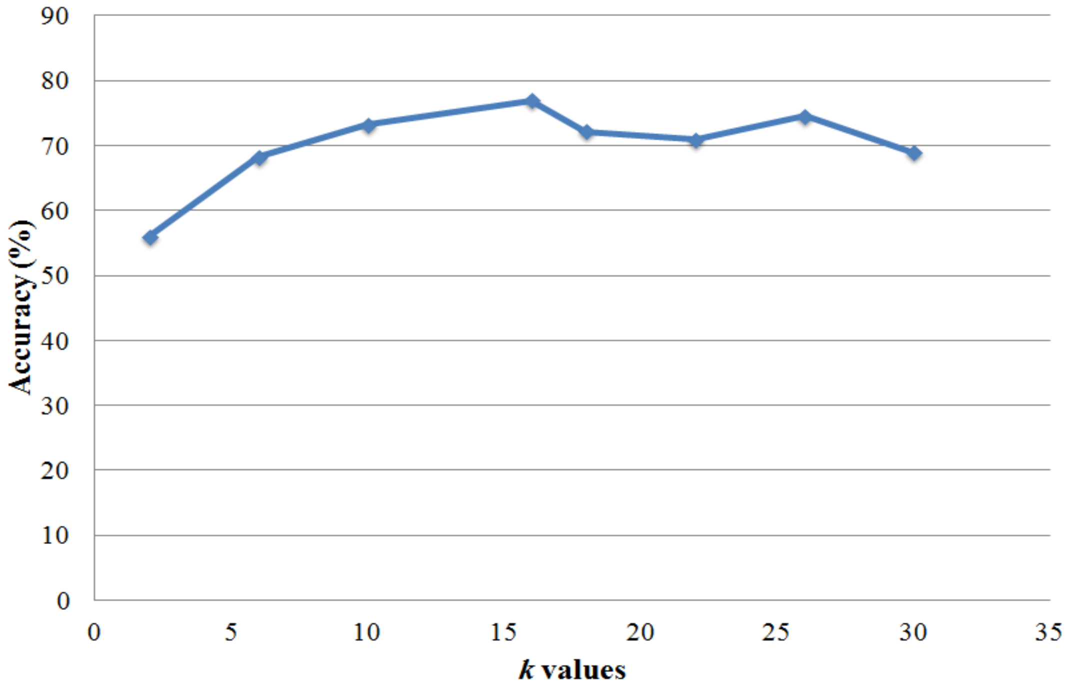


Figure 5.6. Accuracy of the modified *k*-FFTPX on S channel when *k* is varied.

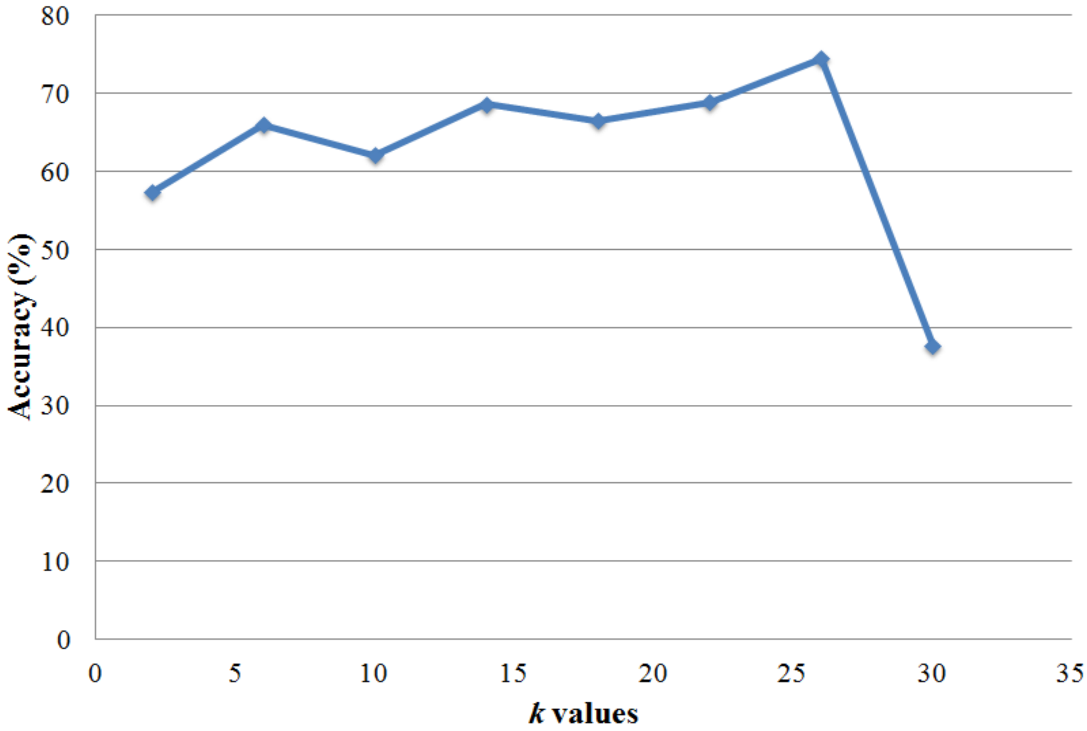


Figure 5.7. Accuracy of the half *k*-FFTPX on S channel when *k* is varied.

Figure 5.6 shows the result of finding the suitable *k* value for the modified *k*-FFTPX using Algorithm 8. The best accuracy is at 76.95% for *k* equal to 16. The result of finding the suitable *k* value for the half *k*-FFTPX is shown in Figure 5.7. The best accuracy here is slightly decreased to 74.55% for *k* equal to 26. Table 5.7 to 5.8 show the result of finding the suitable *h* and *k* values for the *h* × *k*-FFT using Algorithm 9. When *h* is equal to 9 and *k* is equal to 10, the feature yields the highest accuracy at 89% which is about 12 – 14% higher than the two previous FFT features. Thus, at this level of classification tree, the 9 × 10-FFT is served as the best performed FFT feature.

Table 5.9. Accuracy (%) in selection of suitable texture features for classifying cumuliform clouds.

Features	**Iteration**	
	1	**2**
ME (R)	57.35	91.79
ME (G)	37.45	91.04
ME (B)	28.20	90.82
D (R-G)	54.45	91.42
D (R-B)	65.98	92.84

Features	**Iteration**	
	1	**2**
D (G-B)	73.98	**96.29**
U	82.70	92.96
CON	42.73	94.16
HOM	82.63	94.48
EN	81.38	92.42
ENT (R)	25.19	91.52
ENT (G)	83.30	93.90
ENT (B)	86.75	91.97
ENT (S)	88.88	95.53
EE	**93.24**	**-**
SE	91.57	93.75
SC	28.55	96.21
EG	25.73	92.10

The 18 texture features and the 9 × 10-FFT are passed to Algorithm 7 for the additional process of selecting a minimal set of texture features. Table 5.9 shows the two-iteration results where the selected texture features are *EE* and *D* (*G* - *B*), respectively. By concatenating the two suitable texture features with the 9 × 10-FFT, we obtain the confusion matrix as shown in Table 5.10.

Table 5. 10. Confusion matrix classifying four cumuliform clouds.

True class	**Classified as**			
	1	**2**	**3**	**4**
1	**97.32**	1.19	1.49	0
2	2.59	**95.69**	1.29	0.43
3	0	0	**99.36**	0.64
4	0	3.20	4	**92.80**

Class 1, 2, 3 and 4 are high cumuliform, stratocumulus, cumulus, and cumulonimbus, respectively. The classification results of Class 1 and Class 2 are well above 95%. The accuracy of Class 3 classification is almost 100%. However, there are slight misclassifications of Class 4 as Class 2 and Class 4 as Class 3. This is because when cloudlets are very close together, stratocumulus appears similar to cumulonimbus. Likewise, when cumulus is expanded high into the sky, it appears similar to cumulonimbus. Nevertheless, the average accuracy of classification at this level is still high at 96.29%.

5.3.4 All level - A Complete Classification Tree

In this experiment, all ANN models from Section 5.3.1 to 5.3.3 are combined for hierarchy classification as per the proposed tree in Figure 5.3.

Table 5. 11. Confusion matrix classifying seven cloud types.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**97.49**	0	0	0	2.06	0.45	0
2	0.68	**98.18**	0.63	0.24	0.27	0	0
3	0	0.73	**98.06**	0.86	0.34	0	0
4	0.74	0.23	0.68	**98.10**	0.26	0	0
5	2.32	0	0.16	0.96	**96.40**	98.31	0
6	1.11	0	0	0	0.58	**98.31**	0
7	0	0	0	0	0	0	**100**

When Algorithm 10 is executed, the confusion matrix is obtained as shown in Table 5.11. Class 1 to Class 7 are cirriform, high cumuliform, stratocumulus, cumulus, cumulonimbus, stratiform, and clear sky, respectively. The classification results of seven classes are all above 96%. Five of the seven classes have the accuracy higher than 98%. The accuracy of clear sky classification reaches 100%. The overall accuracy of classification is 98.08%.

5.4 Running Time Analysis

We analyze the performance of the five main algorithms by considering their running times. The results are shown in Table 5.12.

Table 5.12. The approximate running times of the five main algorithms.

Algorithms	**Settings**	**Number of images**	**Approx. running time**
The modified *k*-FFTPX	*k*=10	1	31.8 ms
	k=15		31.9 ms
	k=20		32.5 ms
The half *k*-FFTPX	*k*=10	1	30.2 ms
	k=15		30.6 ms

Algorithms	**Settings**	**Number of images**	**Approx. running time**
	k=20		30.7 ms
The *h* × *k*-FFT	*h*=5, *k*=4	1	32.8 ms
	h=12, *k*=6		33.2 ms
	h=9, *k*=10		33.4 ms
Building ANN models	Finding suitable features and training	1,660	20 days
	Training		23.4 min
Cloud Classification Tree Algorithm (CCTA)	LOOCV	1,660	1.6 hr
	Test 1 level	1	701.5 ms
	Test 2 levels		759.5 ms
	Test 3 levels		817.1 ms

The running times of three new FFT features are rather similar. The running time of the modified *k*-FFTPX is around 31 ms which is slightly increased when the *k* value is raised.

However, this running time is clearly less than the running time of the *k*-FFTPX around 20 milliseconds (see Table 4.12). The running time of the half *k*-FFTPX is slightly decreased compared to the modified *k*-FFTPX while the running time of the *h* × *k*-FFT is around 33 ms which is certainly faster than the original *k*-FFTPX. Although the building of ANN models by finding suitable features and training takes a long time (around 20 days), the process is only performed once. The running time of testing CCTA when considering no more than two levels is less than the running time of testing cloud classification algorithm in Chapter 4. While testing CCTA for three levels is slightly slower than the cloud classification algorithm around 3 milliseconds (see Table 4.12). Note that, both of the building ANN models and the CCTA use the same settings as Section 5.3.1 – 5.3.3

5.5 Chapter Summary

To achieve high accuracy of cloud classification, we designed hierarchical classification (tree structure) based on forms and shapes of clouds. The design was composed of three levels of tree, with the aim to classify seven sky conditions. Three new FFT features were proposed to use in the classification process, namely, the modified *k*-FFTPX, the half *k*-FFTPX, and the *h* × *k*-FFT. Three ANN classifiers were trained separately on each level of the tree. Unlike other previous works, we used ANN to classify up to four classes while others used SVM to classify only two classes. The classification result of Level 1 yields the accuracy of 100% using texture features. The accuracy of classification for Level 2 is 95.70% based on a selection of FFT features. The classification result of Level 3 is 96.29% using texture features and the best performed FFT feature. Overall, a complete classification tree provides a high accuracy of 98.08%. This is because the arrangement of classification tree helps reduce the number of competitions among the classes. The number of features used in the algorithms was also selected at minimal sufficient but still gave satisfied results at less computational time. Consider the past papers listed in Table 1.3, although some of the accuracies are higher than 90%, the number of classified cloud classes are limited to fewer classes. In this chapter, the result is better than our two previous experiments (Chapter 3 and 4). Thus, it confirms that the hierarchical classification performs better than a single classification.

Our method accepts any input images from the ten standard cloud types shown in Figure 2.1. Although, some cloud types are rare naturally-occurring types especially for Phuket, the method can be extended further to classify eleven cloud classes (ten standard cloud types and one clear sky) if application requires. In addition, the advantage of classification tree over other methods is that the classification result at each level of the tree is known and available in hierarchical structure; therefore, the classification tree can be easily reconfigured or rearranged to suit user needs. For example, the tree can be used to classify only cloud or no-cloud or even lesser number of cloud types. With this benefit, our method can provide a wider range of applications. However, the drawback of our method is that the errors from the first level of the tree may be carried on to the second level and so on. Hence, in the design of the tree we must place low-misclassified classes before the high-misclassified classes. Furthermore, our method cannot deal with simultaneous appearances of more than one cloud class on the same image. We suggest solving this problem using modified sub-images or considering clouds as objects.

In conclusion, our three main contributions are the new cloud classification method called CCTA, the three novel FFT features, and the presentation of our eight algorithms to readers who prefer the implementation of the method. Our algorithms can also be adapted to suit user requirements. Users can define their own accuracy to gain higher speed in calculation. In the next chapter, we will fine-tune our algorithms to fit Phuket sky conditions and to implement a low-cost cloud monitoring station for cloud monitoring on mobile devices.

CHAPTER 6

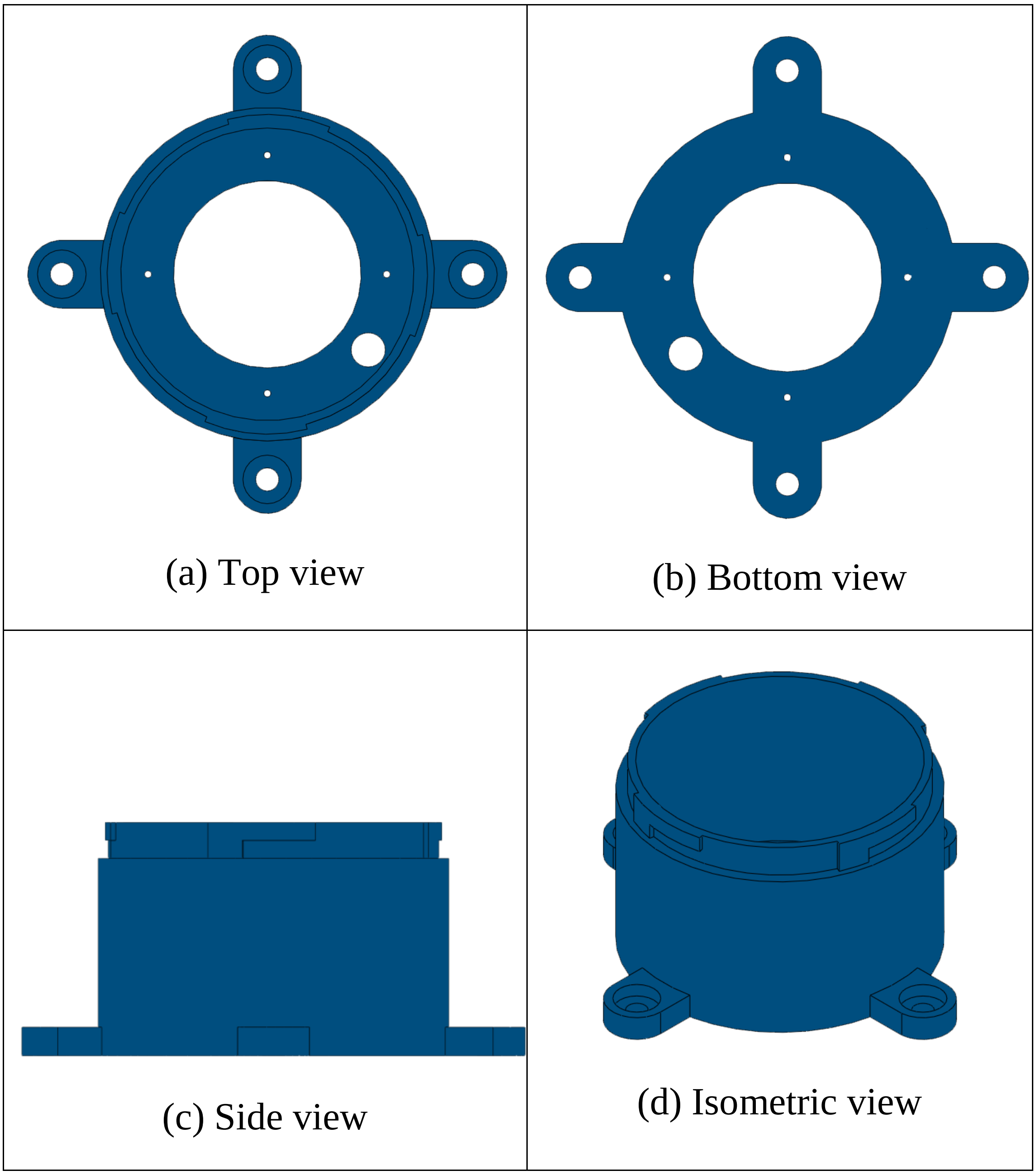
SYSTEM INTEGRATION TEST AND HARDWARE INSTALLATION

In this chapter, we describe each component of our hardware implementation. A cloud monitoring station equipped with a fisheye lens camera is built. It retrieves cloud information in conjunction with other meteorological sensors. The test using images captured from our cloud monitoring station shows that the classification accuracy in practice is as high as 99.82%. The final cloud monitoring system can report live cloud conditions and display them on a mobile application.

6.1 Hardware Implementation

Our hardware is implemented according to the design shown in Figure 1.1. The hardware is composed of four compartments. Each compartment is equipped with a different device. To withstand sun and rain, a plastic material is used to construct the body part using a 3D printer. We describe different compartment of a cloud monitoring station in more details as follows.

Figure 6.1 shows different views of mounting a cooling device (an exhaustive fan) compartment. Inside of this component, there is a large ventilation hole in the middle (See Figure 6.1 (a)). In the bottom, there are four small outer holes for mounting with the holding plate while the top of this compartment has a twist connector for connecting with a control device compartment.



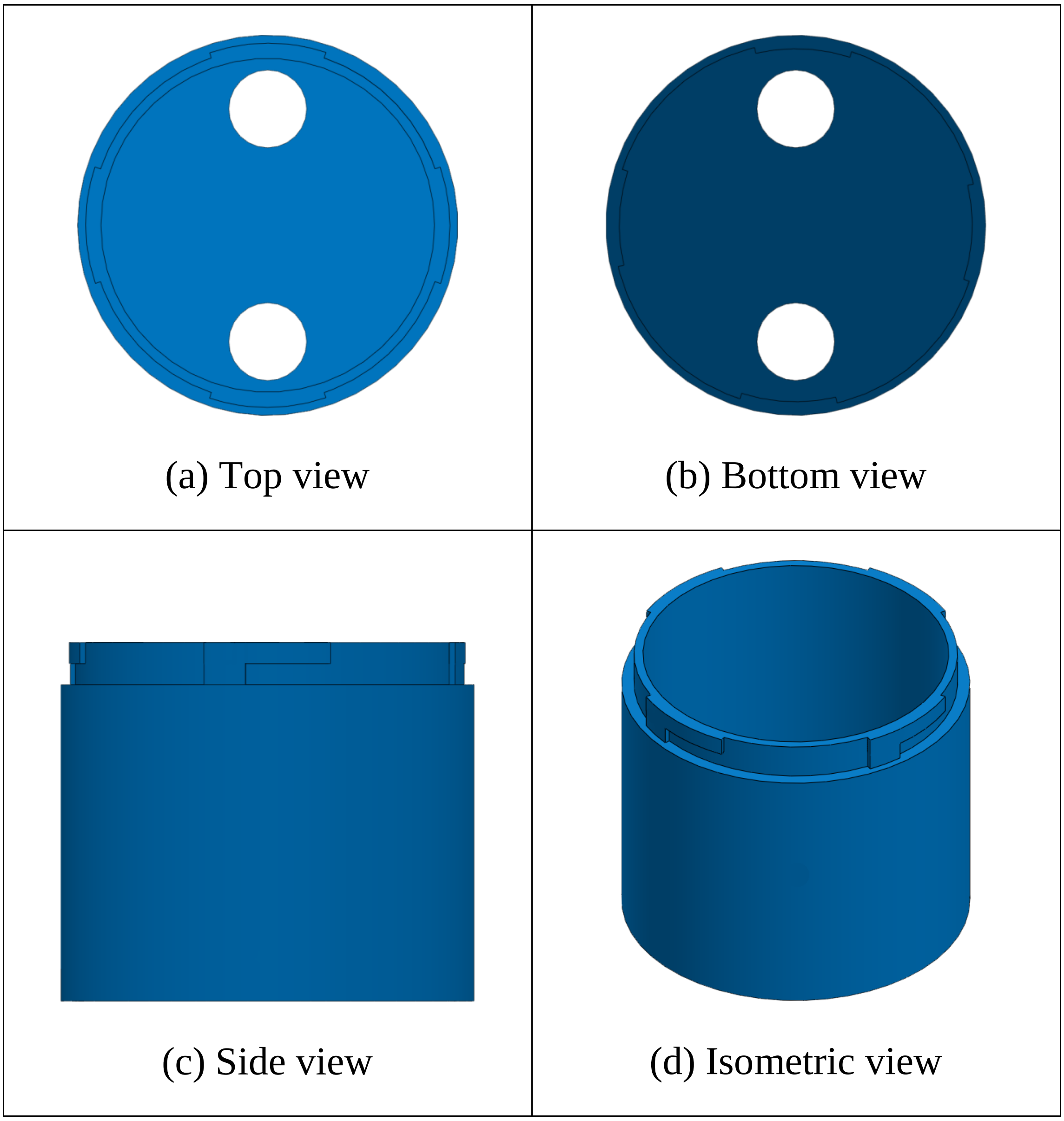
(a) Top view

(b) Bottom view

(c) Side view

(d) Isometric view

Figure 6.1. The mounting of a cooling device compartment.



(a) Top view

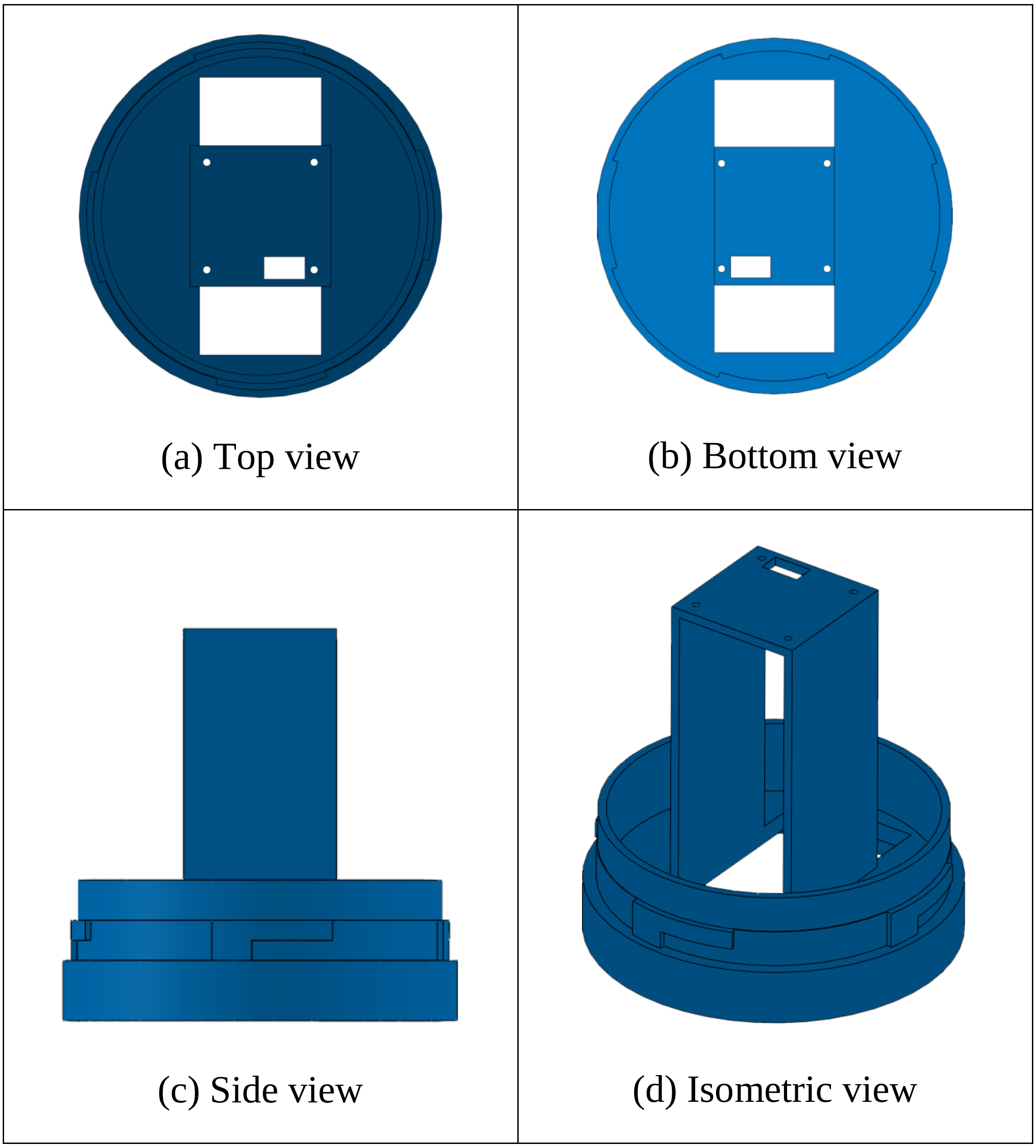
(b) Bottom view

(c) Side view

(d) Isometric view

Figure 6.2. The mounting of a control device compartment.

Figure 6.2 shows different views of mounting a control device (Raspberry Pi) compartment. Inside of this component, there are a Raspberry Pi board and a DHT22 sensor. There are two holes at the bottom for air passage. Two twist connectors are at the top and bottom for connecting with the previous and the next compartment.



(a) Top view

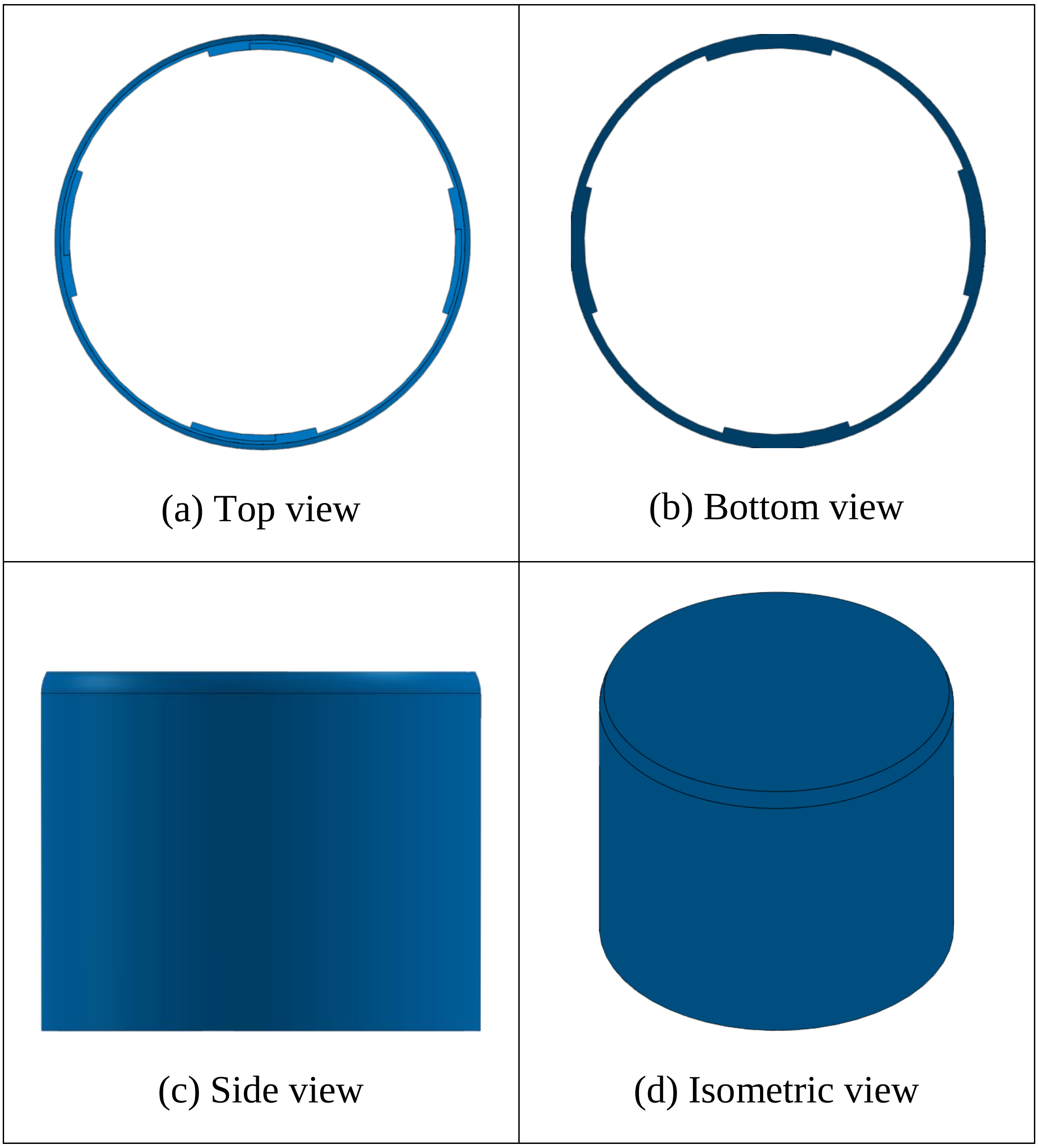
(b) Bottom view

(c) Side view

(d) Isometric view

Figure 6.3. The mounting of a digital camera compartment.

Figure 6.3 shows different views of mounting a digital camera compartment. There is a square hole for air passage coming from the first compartment and there is an extended column to hold a digital camera in the center. A twist connecter is on the side and will be joined with a glass dome.



(a) Top view

(b) Bottom view

(c) Side view

(d) Isometric view

Figure 6.4. The mounting of a glass dome compartment.

Figure 6.4 shows different views of mounting a glass dome compartment. This compartment has a glass dome on the top. At the bottom, there is a twist connector to connect a glass dome compartment with a digital camera compartment. Finally, all four compartments are connected as shown in Figure 1.1.

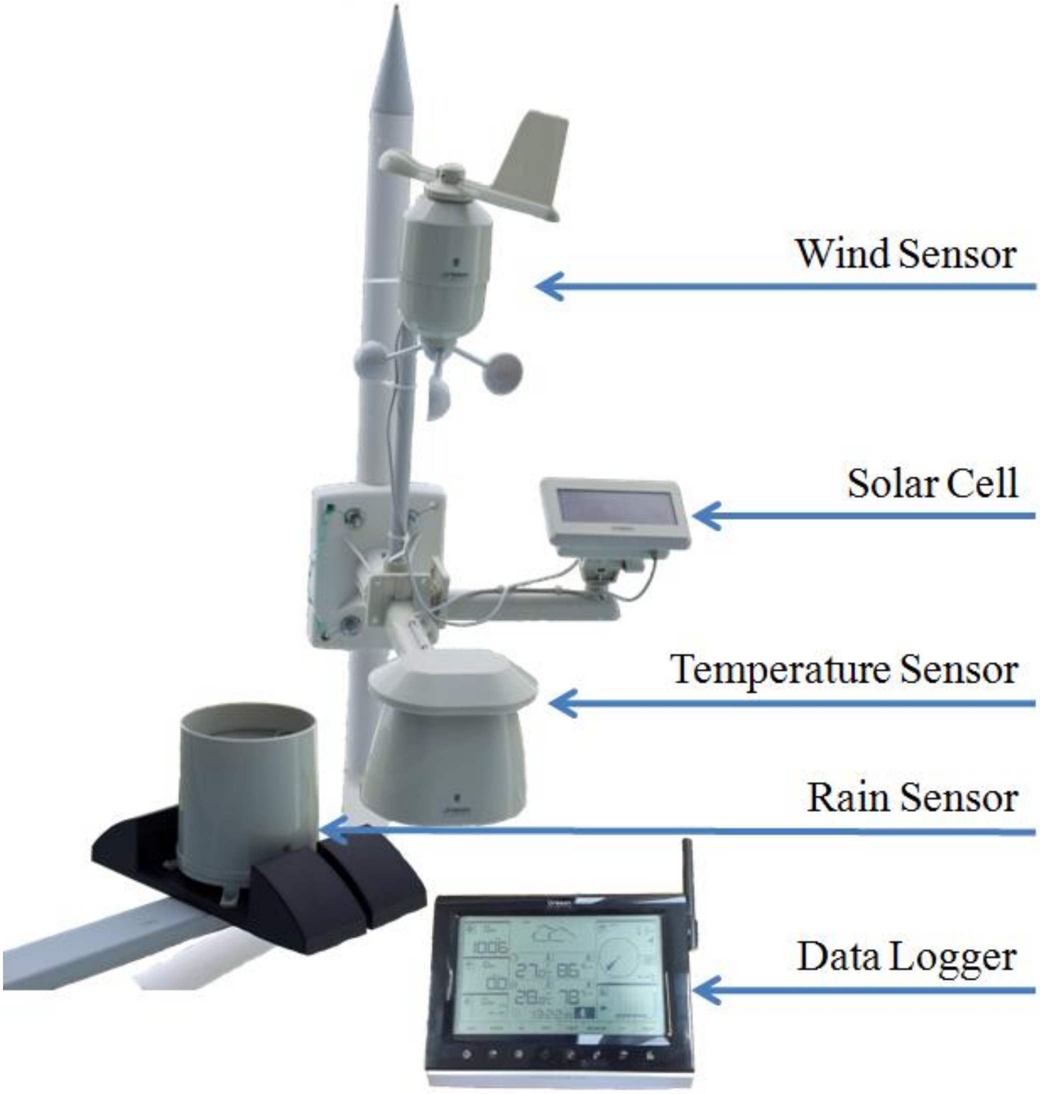


Figure 6.5. Weather station.

Figure 6.5 shows the Oregon scientific professional WMR200A weather station which is composed of several meteorological sensors such as a temperature sensor, a humidity sensor, a wind sensor, a rain gauge, and a solar panel. A device can capture over ten weather measurements such as indoor / outdoor temperature and humidity, wind speed and direction, wind chill, dew point, heat index, barometric pressure and rainfall data. All of these data are sent to the central data logger.

6.2 System Integration

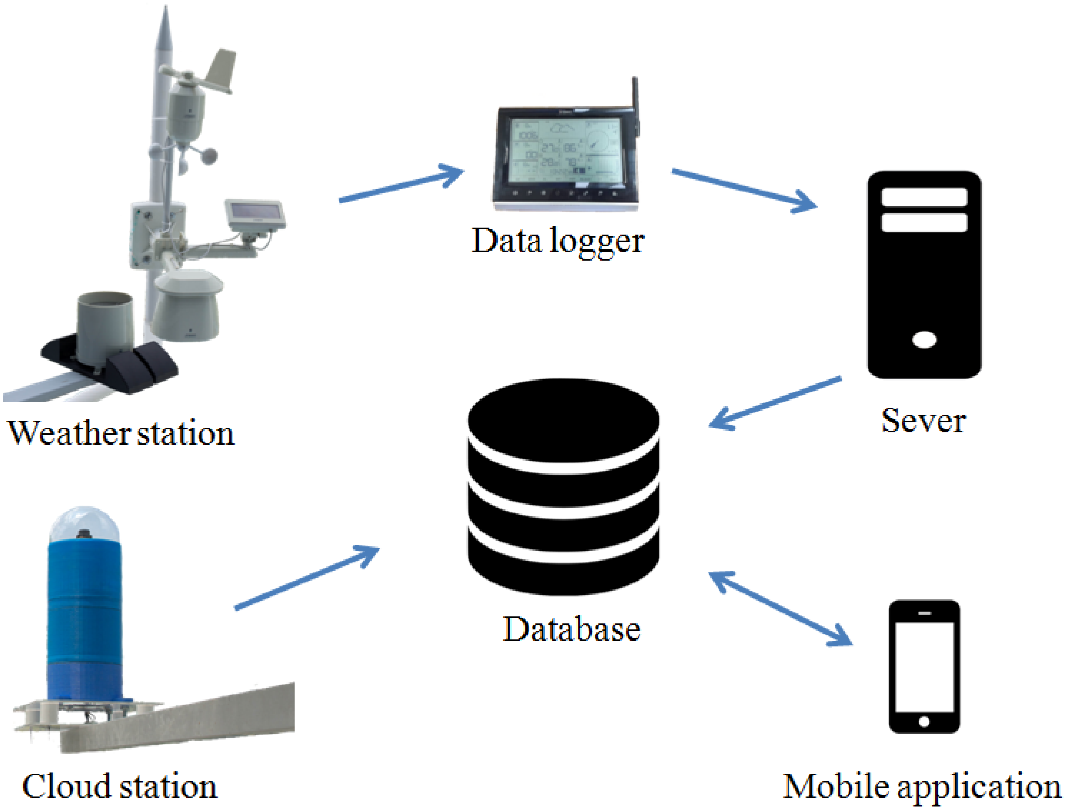


Figure 6.6. Overview of a local cloud monitoring system.

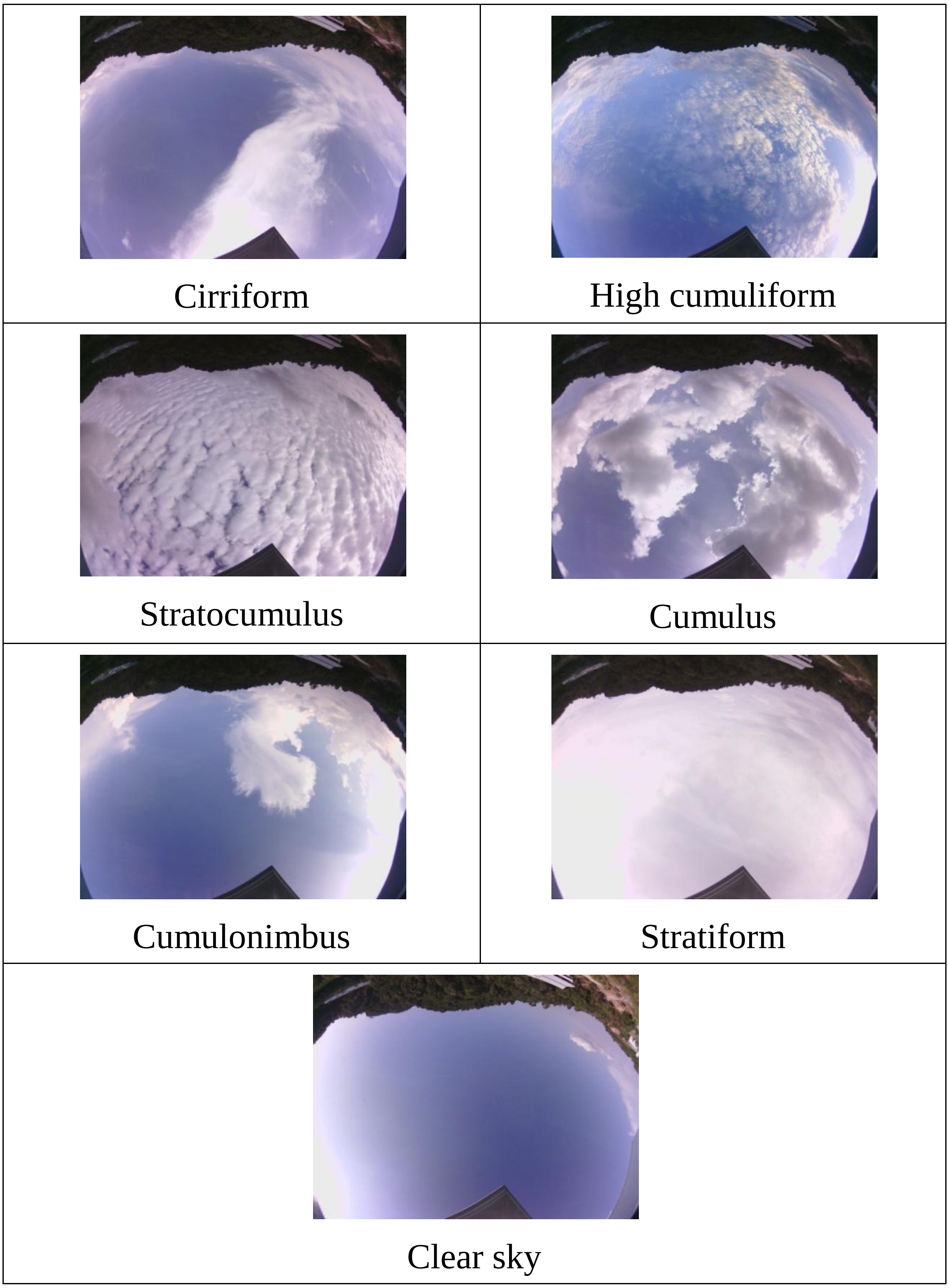
Figure 6.6 shows an overview of a local cloud monitoring system. Two data sources are used to provide the information for our database. The first source is the whole sky images captured from the cloud station every 5 minutes. The second source comes from meteorological sensors such as a temperature and humidity sensor, a wind sensor, a rain gauge and so on. These meteorological data are sent every 5 minutes to the data logger before forwarding to the server and kept in the database. Mobile application is developed to retrieve the images and the meteorological data for online classification purposes.

Mobile users can also view live cloud images and live meteorological data from our station.

The cost of our cloud station is around 142 USD while the cost of WMR200A weather station is around 350 USD. In total, our estimated budget for building a complete system is 492 USD which is cheaper than using TSI-880 (around 30,000-35,000 USD) and WSC (about 2,500 USD) [62].

6.3 A Test on Local Cloud Classification

In this experiment, we use 1,045 whole sky images from our cloud station. The sample images are shown in Figure 6.7. Due to there are unwanted parts of buildings and trees on the sides, the segmentation process is performed before the feature extraction process. A binary mask technique is used for eliminating the unwanted parts (see Figure 6.8). All algorithms in Chapter 5 with the same settings of ANN are used again to classify the seven cloud types. In the feature extraction process, ANN model for Level 1 and Level 2 also use the same features as in Chapter 5. There are five texture textures for classifying cloud and no-cloud and 12 × 6-FFT for classifying three forms of clouds. However, in ANN model for Level 3, two texture features (*SE* and *EE*) with 12 × 6-FFT are used in the training of the model.



Cirriform

High cumuliform

Stratocumulus

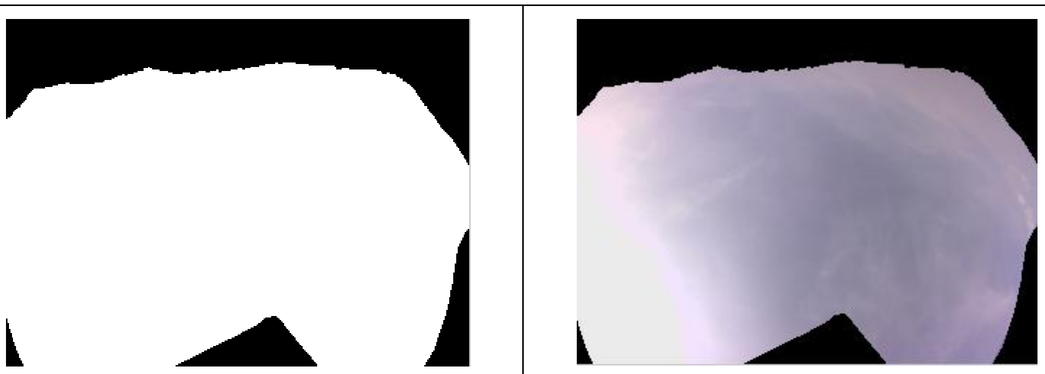
Cumulus

Cumulonimbus

Stratiform

Clear sky

Figure 6.7. Seven cloud types from our cloud monitoring station.



(a) A binary mask image

(a) A result of segmentation

Figure 6.8. Segmentation of a typical whole sky image.

Table 6.1. Confusion matrix for classifying the local whole sky images into seven cloud classes.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**99.92**	0	0	0	0.04	0	0.04
2	0.13	**99.38**	0	0	0.25	0.25	0
3	0.45	0	**99.55**	0	0	0	0
4	0	0	0	**100**	0	0	0
5	0	0	0	0	**100**	0	0
6	0	0	0	0	0.06	**99.79**	0.15
7	0	0	0	0	0	0	**100**

Table 6.1 shows the confusion matrix for classifying our whole sky images into seven cloud classes. Class 1 to Class 7 are cirriform, high cumuliform, stratocumulus, cumulus, cumulonimbus, stratiform, and clear sky, respectively. The correctly classified instances of all class are now higher than 99.30%. Among these results, three classes (cumulus, cumulonimbus, and clear sky) have the classification accuracy as high as 100%. The average accuracy of classification is 99.80% which is higher than the previous results presented in Chapter 5. This is because when the camera is installed at a fixed position, the images obtained from the cloud station are less affected by viewpoint and zooming.

6.4 Integrating with Meteorological Data

It is interesting to see the performance of our cloud monitoring system if we combine cloud image information with meteorological data. Therefore, in this final experiment, we will integrate sixteen meteorological data obtained from the WMR200A weather station. These data are local pressure, sea level pressure, pressure trend, weather status, rainfall rate, current-hour rainfall, last-24-hour rainfall, temperature, temperature trend, humidity, humidity trend, comfort zone, dew point, heat index, gust wind, and the average wind. Using Algorithm 7, we select only a suitable set of meteorological data as a new feature vector and we concatenate it with the 12 × 6-FFT. This new feature is introduced in ANN model for Level 3. In ANN model for Level 1 and Level 2, the settings are the same as before.

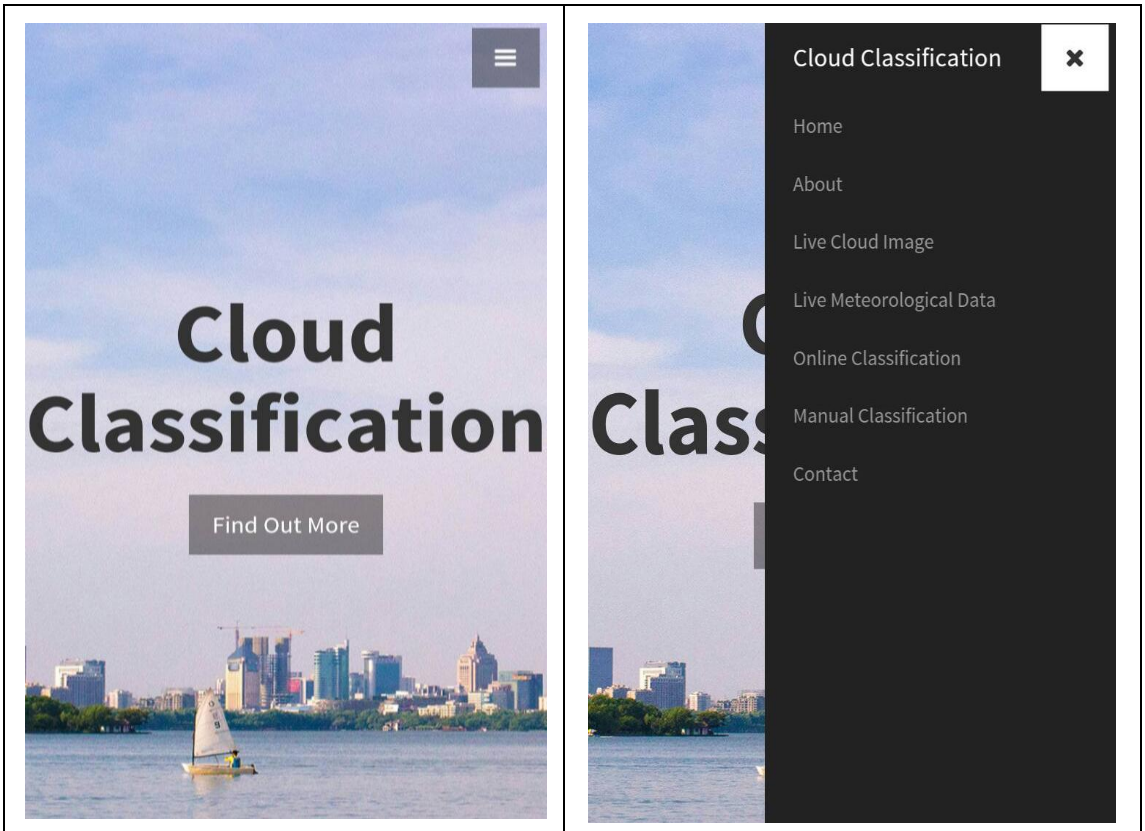
Table 6.2 shows the confusion matrix after a temperature data is chosen to concatenate with 12 × 6-FFT in the third level of the tree. There is a slight misclassification of Class 2. However, the overall accuracy of this experiment is 99.82%. This result is marginally better than the previous experiment. Furthermore, the computation time is slightly reduced because texture features was no longer needed in the third level of the classification tree.

Table 6.2. Confusion matrix after adding meteorological data.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**100**	0	0	0	0	0	0
2	0	**98.75**	0	0	0	1.25	0
3	0	0	**100**	0	0	0	0
4	0	0	0	**100**	0	0	0
5	0	0	0	0	**100**	0	0
6	0	0	0	0	0	**100**	0
7	0	0	0	0	0	0	**100**

6.5 Mobile Application

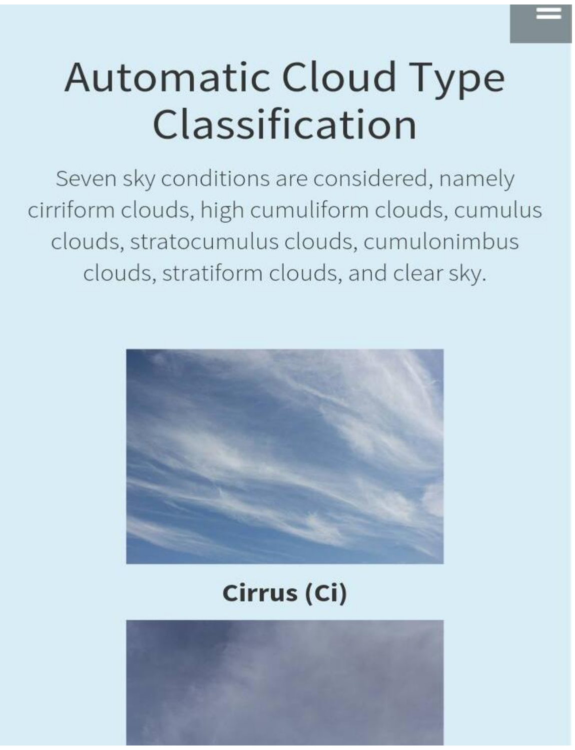
The approach explained in Section 6.4 is used to develop a mobile application for online cloud classification. Figure 6.9 shows the first page of our mobile application. The main menu is on the top right corner of the page. The important menus are such as live cloud image, live meteorological data, online classification and manual classification. The basic descriptions of cloud types and their appearances are available in the about-page for users to study before using the application (see Figure 6.10). Figure 6.11 shows the example of live cloud images from our cloud station. The cloud image is automatically updated every 5 minutes. The classification result and related weather conditions (see Table 6.3) of each cloud image is available along with the image. It is shown under of the cloud image.



(a) Main page

(b) Main menu

Figure 6.9. The first page of our mobile application.

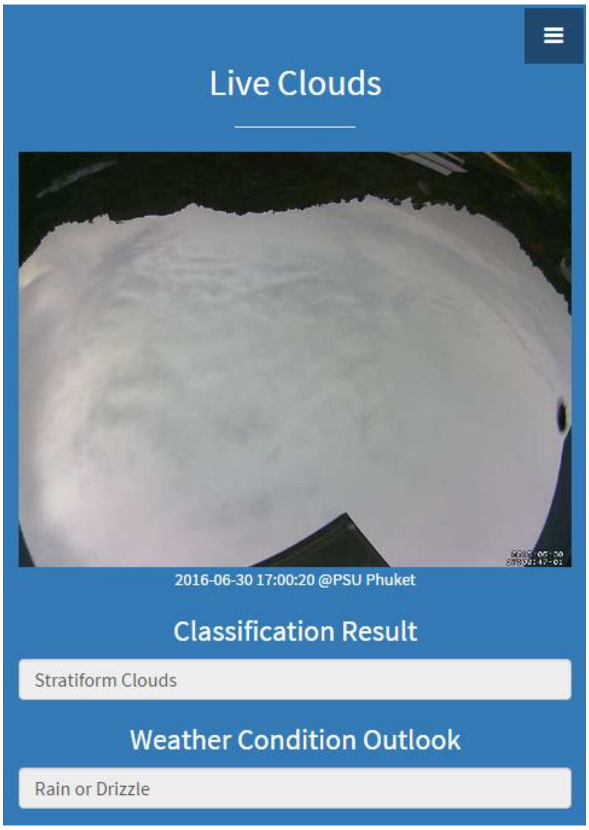


Automatic Cloud Type Classification

Seven sky conditions are considered, namely cirriform clouds, high cumuliform clouds, cumulus clouds, stratocumulus clouds, cumulonimbus clouds, stratiform clouds, and clear sky.

Cirrus (Ci)

Figure 6.10. The about-page for basic cloud-type descriptions.



Live Clouds

2016-06-30 17:00:20 @PSU Phuket

Classification Result

Stratiform Clouds

Weather Condition Outlook

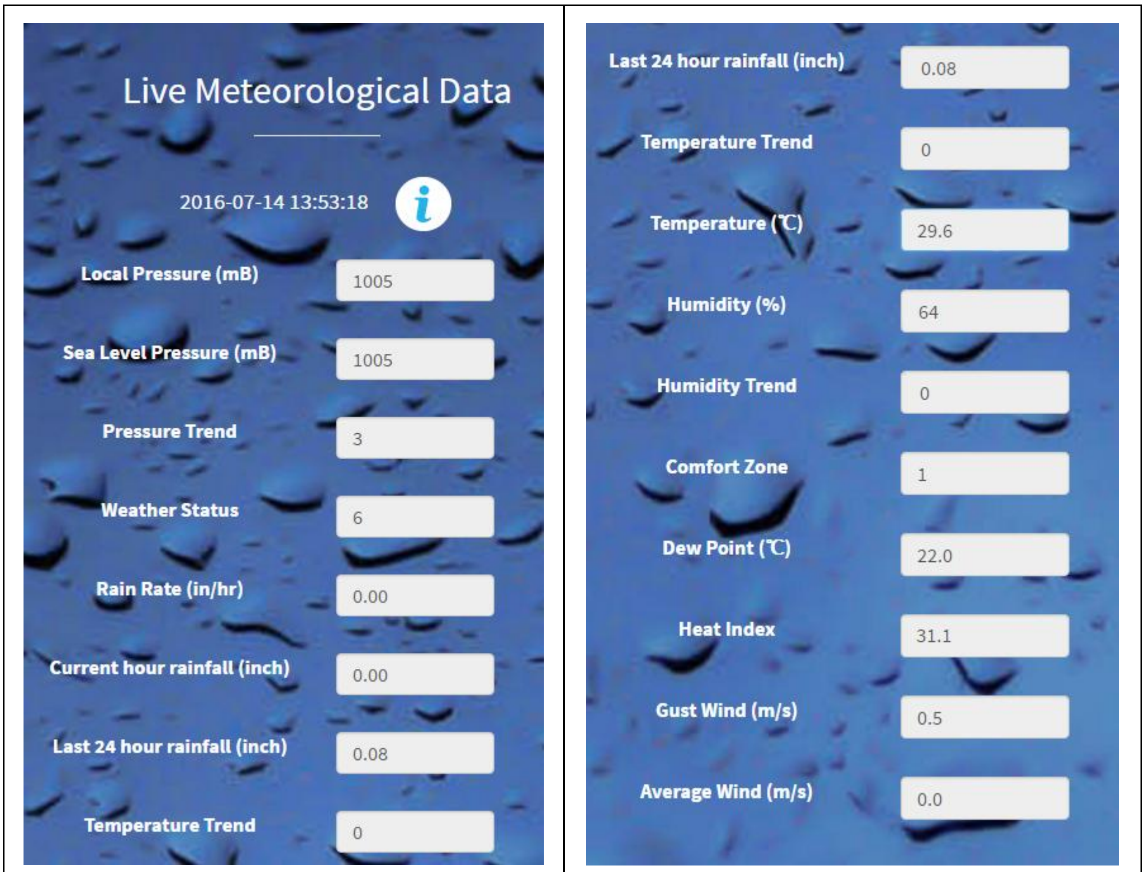
Rain or Drizzle

Figure 6.11. Live cloud image from our cloud station.

Table 6.3. The relationship between each cloud type and weather conditions.

Cloud types	Weather conditions
Cirriform	Fair weather, changing in the weather
High cumuliform	Fair weather, indicator of bad weather
Stratocumulus	Overcast or clearing
Cumulus	Fair weather, sunny day
Cumulonimbus	Thunderstorms
Stratiform	Rain or drizzle
Clear sky	No moisture in the air

Figure 6.12 shows the menu of live meteorological data from the weather station. These data are, local pressure (mB), rainfall rate (in/hr), temperature (°C), humidity (%), dew point (°C), gust wind (m/s), and so on. The meteorological data is automatically updated every 5 minutes. The information button is placed next to the time stamp. It explains the meanings of each meteorological data as shown in Figure 6.13.



Live Meteorological Data

2016-07-14 13:53:18

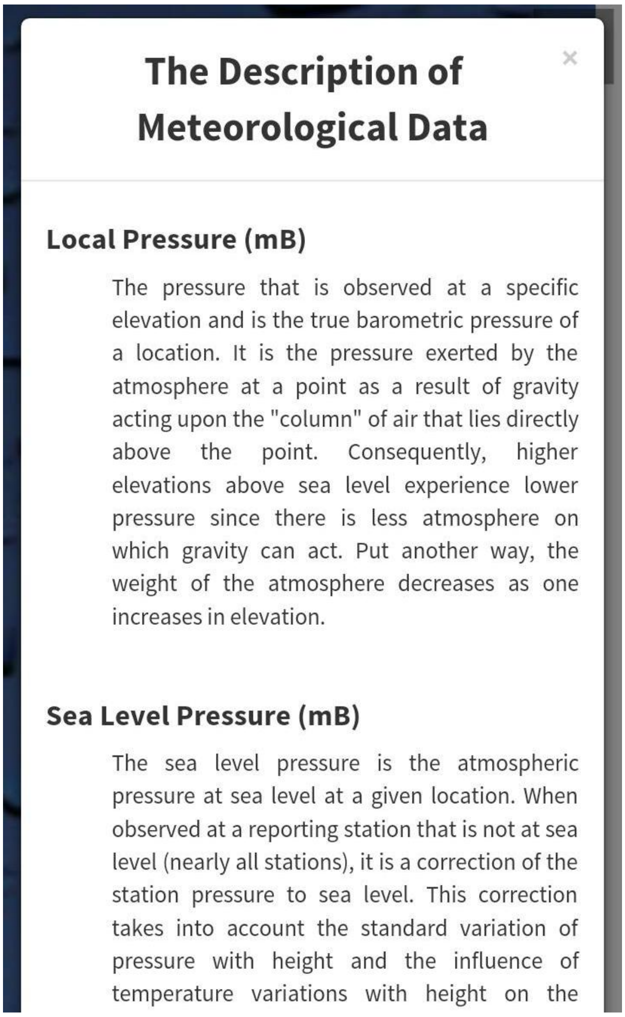
Local Pressure (mB)	1005
Sea Level Pressure (mB)	1005
Pressure Trend	3
Weather Status	6
Rain Rate (in/hr)	0.00
Current hour rainfall (inch)	0.00
Last 24 hour rainfall (inch)	0.08
Temperature Trend	0

(a) Live data

Last 24 hour rainfall (inch)	0.08
Temperature Trend	0
Temperature (℃)	29.6
Humidity (%)	64
Humidity Trend	0
Comfort Zone	1
Dew Point (℃)	22.0
Heat Index	31.1
Gust Wind (m/s)	0.5
Average Wind (m/s)	0.0

(b) Live data (cont.)

Figure 6.12. Live meteorological data menu.



The Description of Meteorological Data

×

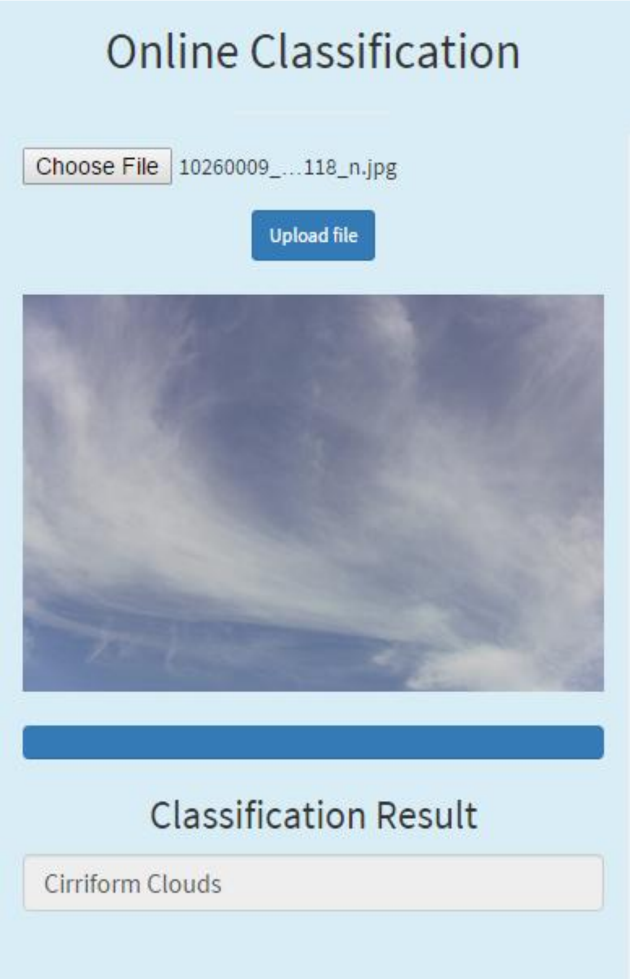
Local Pressure (mB)

The pressure that is observed at a specific elevation and is the true barometric pressure of a location. It is the pressure exerted by the atmosphere at a point as a result of gravity acting upon the "column" of air that lies directly above the point. Consequently, higher elevations above sea level experience lower pressure since there is less atmosphere on which gravity can act. Put another way, the weight of the atmosphere decreases as one increases in elevation.

Sea Level Pressure (mB)

The sea level pressure is the atmospheric pressure at sea level at a given location. When observed at a reporting station that is not at sea level (nearly all stations), it is a correction of the station pressure to sea level. This correction takes into account the standard variation of pressure with height and the influence of temperature variations with height on the

Figure 6.13. The description of meteorological data.


Online Classification

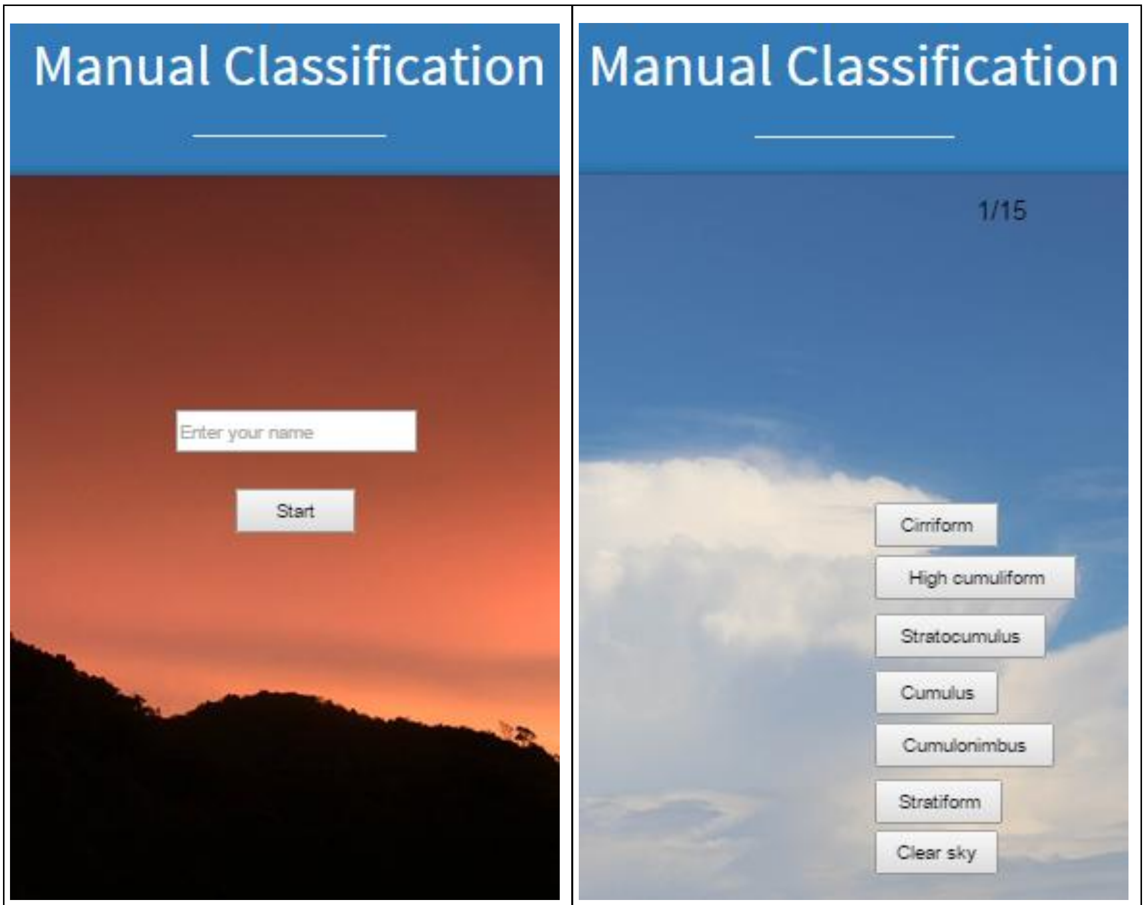
10260009_...118_n.jpg

Classification Result

Cirriform Clouds

Figure 6.14. Online classification menu.

Figure 6.14 shows the menu for online classification. In this page, users can submit a cloud image from anywhere, then click upload file. Our system will classify the cloud image into one of the seven cloud types. The classification result is shown on the bottom of the page.



(a) Login menu

(b) Manual classification

Figure 6.15. Manual classification menu.

Figure 6.15 shows the menu for manual classification. In this page, users can manually view the cloud picture and choose one preferred type from the list, then send the answer to our system for verifying the result. These results reflect different perspectives of how people view the clouds; therefore, the results will be collected in the database for improving our classification process in the future.

6.6 Benefits of Weather Station

The meteorological data from the weather station is used to concatenate with cloud image information as a feature vector for cloud classification. The accuracy of classification is slightly improved from using cloud images alone. Moreover, the computation time is reduced by replacing meteorological data with texture features which is less cumbersome when developing mobile application. However, our current algorithms are still complicated and cannot make use of all useful meteorological data. In the future, the system may be simplified by using key meteorological data from the weather station to partially classify some easy cloud types first before sending to the normal cloud classification process.

6.7 Chapter Summary

A cloud station was designed and implemented. A weather station provided additional useful meteorological information. Together we have a low-cost local cloud monitoring system installed at Prince of Songkla University, Phuket campus. The performance test of local cloud classification using our whole sky images yielded 99.80% accuracy which is higher than the results in Chapter 5. Then, we modified a feature vector in Level 3 of ANN model to include meteorological data. The last result gave 99.82% accuracy which is slightly higher. This approach was then used to develop online classification for mobile users. Live cloud image and live meteorological data can be viewed through mobile application from anywhere in the world. In the future, the past cloud images and the historical meteorological data can be further analyzed for forecasting clouds and weather conditions. Moreover, if our cloud monitoring system is installed in many areas, it may be useful for improving results of the weather forecast in the wider area.

CHAPTER 7

CONCLUSIONS

In this chapter, we summarize and discuss all developments of cloud classification system. Then, we conclude all contributions and open problems.

7.1 Discussion

We will discuss whether this research has met all of the objectives in Chapter 1. Referring to the first objective, it is to select a suitable classifier from the two commonly used classifiers, namely, *k*-NN and ANN. Our result showed that the well-performed classifier is ANN. Thus, we selected ANN as the preferred classifier in our system design and implementation.

The second objective is to select a suitable feature extraction technique that is effective for cloud type classification. Our result showed the most effective feature extraction technique is our proposed feature based on FFT called the *k*-FFTPX. However, the textual contents of clouds were also shown as useful information as the shapes of clouds. Hence, we recommended keeping texture features (textual information) while applying the new FFT feature (shape information).

The third objective is to propose novel features and algorithms that are different from the previous works. We proposed four new features, namely, the *k*-FFTPX, the modified *k*-FFTPX, the half *k*-FFTPX, and the *h* × *k*-FFT. Our new algorithm was based on a hierarchy classification tree and was proven to gain higher accuracy in the classification.

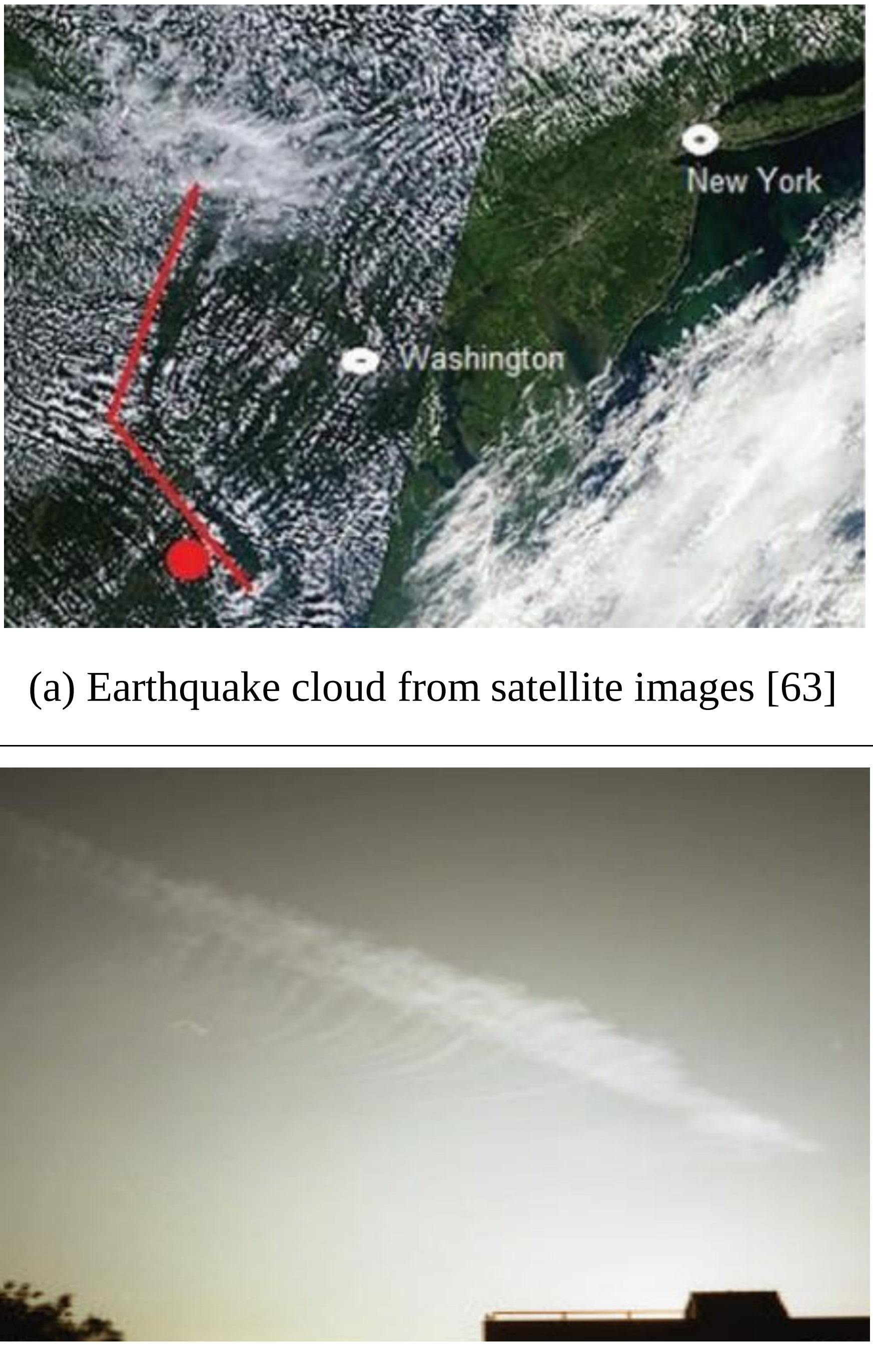
Table 7.1**.** Summary of cloud classification methods and their accuracies from all experiments.

No.	**Proposed Method**	**Accuracy (%)**
1	The 18 texture features with *k*-NN	72.99
2	The 18 texture features with ANN	86.93
3	The most effective 7 texture features with ANN	77.44
4	Texture features	80.76
5	Texture features with 2D-moments	82.66
6	Texture features with Abs-FFT	85.44
7	Texture features with Log-FFT	86.11
8	Texture features with *k*-FFTPX	90.40
9	Classification of cloud and no-cloud	100
10	Classification of three forms of cloud	95.70
11	Classification of fours shapes of cumuliform clouds	96.29
12	A hierarchical classification tree	98.08
13	A hierarchical classification tree using whole sky images	99.80
14	A hierarchical classification tree using whole sky images together with meteorological data	99.82

The fourth objective is to propose a complete cloud classification system for classifying seven cloud types using ground-based images which has accuracy higher than 95%. Table 7.1 summarizes cloud classification methods and their accuracies from all experiments performed in this thesis. The highest accuracy for classifying seven cloud types is 99.82% which is a result of a hierarchical classification tree technique integrating with the meteorological data from our cloud monitoring station. We also obtained a by-product from the tree for classifying two cloud types, cloud and no-cloud, the performance here is as high as 100%.

The final objective is to implement hardware system for capturing cloud images and to build a low-cost local cloud monitoring station. We implemented the system. The low-cost local cloud monitoring station was built and installed at Prince of Songkla University, Phuket Campus. The mobile application was developed for online classification. It is active and operational. Users can monitor live images and live meteorological data from our station.

Our research has opened up new directions for those interested in the cloud classification. There are many areas where improvements are needed; for example, how to classify many types of clouds that appear on the same image, how to design the algorithm that recognizes clouds in the night time, how to enhance the accuracy of the system even closer to 100%. Other feature extraction techniques such as point detection, object detection, shape detection, gradient operators, and fractals may be worth exploring because shapes of clouds are useful information. Other classifiers such as SVM perhaps should be looked at and compared. Finally, more cloud types in nature may be considered, for example, earthquake clouds.



(a) Earthquake cloud from satellite images [63]

(b) Earthquake cloud from a digital camera [29]

Figure 7.1. Earthquake clouds.

Figure 7.1 shows some earthquake clouds that are caused by thermal effects intensifying from the ground. These clouds appeared a few days before the earthquake. In Figure 7.1 (a), the earthquake epicenter is shown by the red circle while the red line shows cloud anomalies which occurred a day before the Virginia earthquake in 2011 [63]. Figure 7.2 (b) shows the earthquake clouds which appeared a day before the earthquake in Northern California in 1994 [29]. However, the challenge ahead requires researchers to investigate more closely on the true unknown origin of earthquake clouds and the feasibility study of them for earthquake prediction.

7.2 Research Contribution

The research has proposed different methods of feature extraction techniques and several classification algorithms to improve accuracy of cloud type classification. All contributions in different aspects are summarized below.

7.2.1 Feature Extraction

Five different combinations of feature extraction techniques were presented and the accuracies are 80.76%, 82.66%, 85.44%, 86.11%, and 90.40%, respectively. We showed that the most effective feature extraction technique is based on our proposed FFT features. We, therefore, proposed more features based on FFT. There are in total of four new FFT features in the thesis. However, the main contribution is the half-FFT features which refer to the half *k*-FFTPX and the *h* × *k*-FFT.

7.2.1 Classifier

The parameter settings of *k*-NN and ANN were recommended. The performances of the two classifiers were compared. We also discussed advantages and limitations of both classifiers. The experiment revealed that the well-performed classifier is ANN.

7.2.2 Algorithms

Two algorithms were presented in Chapter 4. Eight algorithms were presented in Chapter 5. There are in total of ten proposed algorithms in the thesis.

7.2.3 Performances

The final classification accuracy is 99.82%. This result is higher than the previous studies on cloud classification using digital camera and ANN method. The number of cloud classes being classified in those studies was also less than in this research.

7.2.4 Textual Contents

The textual contents of clouds were also shown as useful information as the shapes of clouds. Hence, we recommended using texture features together with other features. For texture features, we thoroughly tested 18 texture features. The most effective texture features were also recommended for general use in a more simple system of cloud classification.

7.2.5 Hierarchical Classification

The proposed Cloud Classification Tree Algorithm (CCTA) used the new technique called hierarchical classification which is designed to reduce the number of competitions among the cloud classes. The result has confirmed that the hierarchical classification performs better than a single classification.

7.2.6 Flexibility

According to Table 7.1, users have flexibility to choose any of the 12 proposed methods to suit their preferred accuracy. The results suggest a variety of practical applications from the simple to the sophisticated ones. In our classification algorithms, users also have freedom to reduce their expected accuracy to gain higher speed in calculation.

7.2.7 By-Product

By-product from the hierarchical classification tree is that the tree can also be used to classify lesser number of cloud types. Classification of two cloud classes; cloud and no-cloud; gave the accuracy as high as 100%. Classification of three cloud classes; cirriform, cumuliform and stratiform; yielded the accuracy of 95.70%. Classification of four cloud classes; high cumuliform, stratocumulus, cumulus, and cumulonimbus; returned the accuracy of 96.29%.

7.2.8 Real Life Application

The low-cost local cloud monitoring station was built. It is now working in conjunction with other inexpensive meteorological sensors to report weather conditions and display them through a mobile application.

7.3 Open Problems

There are still fruitful opportunities to advance our field of research in automatic cloud classification using ground-based images. We suggest open problems as follows.

1) To classify more cloud classes such as eleven sky conditions.

2) To design algorithms for classifying clouds in the night time using images from infrared camera.

3) To apply less dimensional feature vectors but still gain high accuracy using more efficient features or dimension reduction techniques such as principal component analysis (PCA), linear discriminant analysis (LDA), or independent component analysis (ICA).

4) To consider other features extraction techniques such as point detection, object detection, shape detection, gradient operators, and fractals

5) To consider other new classifiers such as SVM, and MSVM. We believe these classifiers may prudently avoid an overfitting problem that often occurs with ANN.

6) To classify more than one type of clouds that appeared on the same image by looking at sub-images or considering clouds as objects.

7) To design less complicated and low computation time algorithm while maintaining high accuracy by exploiting meteorological data more cleverly.

8) To develop a system for weather prediction or other applications related with clouds such as earthquake clouds.

REFERENCES

[1] H. S. R. T. Force, “Hurricane Sandy rebuilding strategy : stronger communities, a resilient region,” US Department of Housing and Urban Development, Washington DC, 2013.

[2] J. Lee, R. C. Weger, S. K. Sengupta, and R. M. Welch, “A neural network approach to cloud classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 5, pp. 846–855, Sept. 1990.

[3] U. Heinzmann, “Cloud classification on the basis of NOAA-APT data using a fuzzy logic approach,” *International Archives of Photogrammetry and Remote Sensing*, vol. 29, pp. 908–913, Aug. 1992.

[4] R. L. Bankert, “Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network,” *Journal of Applied Meteorology*, vol. 33, no. 8, pp. 909–918, Aug. 1994.

[5] D. W. Aha and R. L. Bankert, “Feature selection for case-based classification of cloud types: An empirical comparison,” in *Proceedings of the AAAI workshop on case-based reasoning*, 1994, pp. 106–112.

[6] K. A. Buch, C.-H. Sun, and L. R. Thorne, “Cloud classification using whole-sky imager data,” in *Proceedings of the Fifth Atmospheric Radiation Measurement Science Team Meeting*, 1995, vol. 7, pp. 19–23.

[7] Y. Fan, L. Changsheng, and C. Weimin, “Man–computer interactive method on cloud classification based on bispectral satellite imagery,” *Advances in Atmospheric Sciences*, vol. 14, no. 3, pp. 389–398, Sept. 1997.

[8] C. Ambroise, G. Sèze, F. Badran, and S. Thiria, “Hierarchical clustering of self-organizing maps for cloud classification,” *Neurocomputing*, vol. 30, no. 1, pp. 47–52, Jan. 2000.

[9] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 67–81, Mar. 2004.

[10] M. Singh and M. Glennen, "Automated ground-based cloud recognition," *Pattern Analysis and Applications*, vol. 8, no. 3, pp. 258–271, Dec. 2005.

[11] M. P. Souza-Echer, E. B. Pereira, L. S. Bins, and M. A. R. Andrade, "A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera," *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 3, pp. 437–447, Mar. 2006.

[12] W. Shangguan, Y. Hao, Z. Lu, and P. Wu, "The research of satellite cloud image recognition base on variational method and texture feature analysis," in *The Second IEEE Conference on Industrial Electronics and Applications*, 2007, pp. 2816–2820.

[13] J. Calbó and J. Sabburg, "Feature extraction from whole-sky ground-based images for cloud-type recognition," *Journal of Atmospheric and Oceanic Technology*, vol. 25, no. 1, pp. 3–14, Jan. 2008.

[14] R. Kaur and A. Ganju, "Cloud classification in NOAA AVHRR imageries using spectral and textural features," *Journal of the Indian Society of Remote Sensing*, vol. 36, no. 2, pp. 167–174, June 2008.

[15] A. Heinle, A. Macke, and A. Srivastav, "Automatic cloud classification of whole sky images," *Atmospheric Measurement Techniques Discussions*, vol. 3, no. 1, pp. 269–299, May 2010.

[16] M. Martínez-Chico, F. J. Batlles, and J. L. Bosch, "Cloud classification in a mediterranean location using radiation data and sky images," *Energy*, vol. 36, no. 7, pp. 4055–4062, July 2011.

[17] J. Parikh, “A comparative study of cloud classification techniques,” *Remote Sensing of Environment*, vol. 6, no. 2, pp. 67–81, 1977.

[18] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pagés, “Retrieving cloud characteristics from ground-based daytime color all-sky images,” *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 5, pp. 633–652, May 2006.

[19] A. Kazantzidis, P. Tzoumanikas, A. Bais, S. Fotopoulos, and G. Economou, “Cloud detection and classification with the use of whole-sky ground-based images,” *Atmospheric Research*, vol. 113, pp. 80–88, Sept 2012.

[20] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, “Salient local binary pattern for ground-based cloud classification,” *Acta Meteorologica Sinica*, vol. 27, no. 2, pp. 211–220, Apr. 2013.

[21] A. Taravat, F. Del Frate, C. Cornaro, and S. Vergari, “Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 3, pp. 666–670, Mar. 2015.

[22] H.-Y. Cheng and C.-C. Yu, “Block-based cloud classification with statistical features and distribution of local texture features,” *Atmospheric Measurement Techniques*, vol. 8, no. 3, pp. 1173–1182, Mar. 2015.

[23] Q. Li, Z. Zhang, W. Lu, J. Yang, Y. Ma, and W. Yao, “From pixels to patches: a cloud classification method based on a bag of micro-structures,” *Atmospheric Measurement Techniques*, vol. 9, no. 2, pp. 753–764, Mar. 2016.

[24] W. Zhuo, Z. Cao, and Y. Xiao, “Cloud classification of ground-based images using texture–structure features,” *Journal of Atmospheric and Oceanic Technology*, vol. 31, no. 1, pp. 79–92, Jan. 2014.

[25] M. Xia, W. Lu, J. Yang, Y. Ma, W. Yao, and Z. Zheng, “A hybrid method based on extreme learning machine and *k*-nearest neighbor for cloud classification of ground-based

visible cloud image," *Neurocomputing*, vol. 160, pp. 238–249, July 2015.

[26] Z. Wu, X. Xu, M. Xia, M. Ma, and L. Li, "Ground-based vision cloud image classification based on extreme learning machine," *Open Cybernetics and Systemics Journal*, vol. 9, pp. 2877–2885, Nov. 2015.

[27] G. Gup and G. Xie, "Earthquake cloud over Japan detected by satellite," *International Journal of Remote Sensing*, vol. 28, no. 23, pp. 5375–5376, Nov. 2007.

[28] G. Guangmeng and Y. Jie, "Three attempts of earthquake prediction with satellite cloud images," *Natural Hazards and Earth System Science*, vol. 13, no. 1, pp. 91–95, Jan. 2013.

[29] D. Harrington and Z. Shou, "Bam earthquake prediction & space technology," *United Nations Programme on Space Applications*, vol. 16, p. 39–63, 2005.

[30] Z. Shou, "Earthquake vapor, a reliable precursor," *Earthquake Prediction*, pp. 21–51, 2006.

[31] G. Guo and B. Wang, "Cloud anomaly before Iran earthquake," *International Journal of Remote Sensing*, vol. 29, no. 7, pp. 1921–1928, Apr. 2008.

[32] H. V. Alvan and F. H. Azad, "Satellite remote sensing in earthquake prediction. A review," in *National Postgraduate Conference (NPC)*, 2011, pp. 1–5.

[33] X. Liu, "Conjecture on imminent earthquake prediction—from shaving foam to cloud patterns," *arXiv:1206.3010*, Apr. 2013.

[34] Y. Jie and G. Guangmeng, "An attempt to predict earthquakes with satellite data," *Disaster Advances*, vol. 6, no. 1, pp. 35–37, 2013.

[35] F. R. Met Office, "Cloud names and classifications," [Online]. Available: http://www.metoffice.gov.uk/learning/clouds/cloud-names-classifications. [Accessed: 21-Jun-2013].

[36] WMO, *International cloud atlas.* World Meteorological Organization Geneva, Switzerland, 1956.

[37] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, Nov. 1973.

[38] A. Khotanzad and Y. H. Hong, “Invariant image recognition by Zernike moments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, May 1990.

[39] A. Ajanki, “*k*-nearest neighbors algorithm,” *Wikipedia, the free encyclopedia*. 02-Jul-2013.

[40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Wiley, New York, 2001.

[41] K. Chaowanawatee and A. Heednacram, “Improved cuckoo search in RBF neural network with gaussian distribution,” in *Proceedings of the IASTED International Conference*, 2013, vol. 201, pp. 379–386.

[42] A. Heednacram and T. Samitalampa, “Suspended sediment forecast of Khlong Bang Yai, Phuket,” *International Journal of Engineering and Technology*, vol. 6, no. 4, p. 338–345, Aug. 2014.

[43] O. Veksler, “Nonparametric density estimation nearest neighbors, KNN,” [Online]. Available: http://www.cs.haifa.ac.il/~rita/ml_course/lectures/KNN.pdf. [Accessed: 21-Jun-2013].

[44] I. H. Witten and E. Frank, *Data Mining: practical machine learning tools and techniques, Second Edition*. Morgan Kaufmann, San Francisco, 2005.

[45] I. Bonet, A. Rodríguez, R. Grau, M. M. García, Y. Saez, and A. Nowé, “Comparing distance measures with visual methods,” in *MICAI 2008: Advances in Artificial Intelligence*, 2008, pp. 90–99.

[46] B. Karlik and A. V. Olgac, “Performance analysis of various activation functions in generalized MLP

architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, Aug. 2011.

[47] A. Kasapis, "MLPs and pose, expression classification," UniS, Longyearbyen, Norway, 2003.

[48] S. Bauer, "Cloud atlas - cloud classification, cloud pictures." [Online]. Available: http://www.cloudsonline.com/. [Accessed: 28-May-2016].

[49] J. Deguara, "Weather stock photography Australian severe weather." [Online]. Available: http://australiasevereweather.com/photographs/index.php. [Accessed: 28-May-2016].

[50] O. Daowieng, B. Wongkittisuksa, S. Tanthanuch, and W. Permsirivanich, "Command system design of speaker dependent isolated word recognition for stroke patients," in *The second ECTI-Conference on Application Research and Development*, 2010, pp. 19–24.

[51] G. Chen, "Automatic EEG seizure detection using dual-tree complex wavelet-Fourier features," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2391–2394, Apr. 2014.

[52] W. Ben Soltana, A. Porebski, N. Vandenbroucke, A. Ahmad, and D. Hamad, "Texture analysis of lace images using histogram and local binary patterns under rotation variation," in *2014 First International Image Processing, Applications and Systems Conference*, 2014, pp. 1–5.

[53] W. J. Stępniowski, M. Michalska-Domańska, M. Norek, and T. Czujko, "Fast Fourier transform based arrangement analysis of poorly organized alumina nanopores formed via self-organized anodization in chromic acid," *Materials Letters*, vol. 117, pp. 69–73, Feb. 2014.

[54] J. Heaton, *Introduction to neural networks with Java*. Heaton Research Inc., Missouri, United States, 2008.

[55] M. Hansen, R. DeFries, J. R. Townshend, and R. Sohlberg, "Global land cover classification at 1 km spatial resolution

using a classification tree approach," *International Journal of Remote Sensing*, vol. 21, no. 6–7, pp. 1331–1364, 2000.

[56] K. Polat and S. Güneş, "Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1017–1026, Apr. 2007.

[57] S. Pang, D. Kim, and S. Y. Bang, "Face membership authentication using SVM classification tree generated by membership-based LLE data partition," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 436–446, Mar. 2005.

[58] M. Azimi-Sadjadi and S. Zekavat, "Cloud classification using support vector machines," in *Geoscience and Remote Sensing Symposium*, 2000, vol. 2, pp. 669–671.

[59] A. R. Smith, "Color gamut transform pairs," *ACM Siggraph Computer Graphics*, vol. 12, no. 3, pp. 12–19, Aug. 1978.

[60] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[61] J. Tian, L. Chen, L. Ma, and W. Yu, "Multi-focus image fusion using a bilateral gradient-based sharpness criterion," *Optics Communications*, vol. 284, no. 1, pp. 80–87, Jan. 2011.

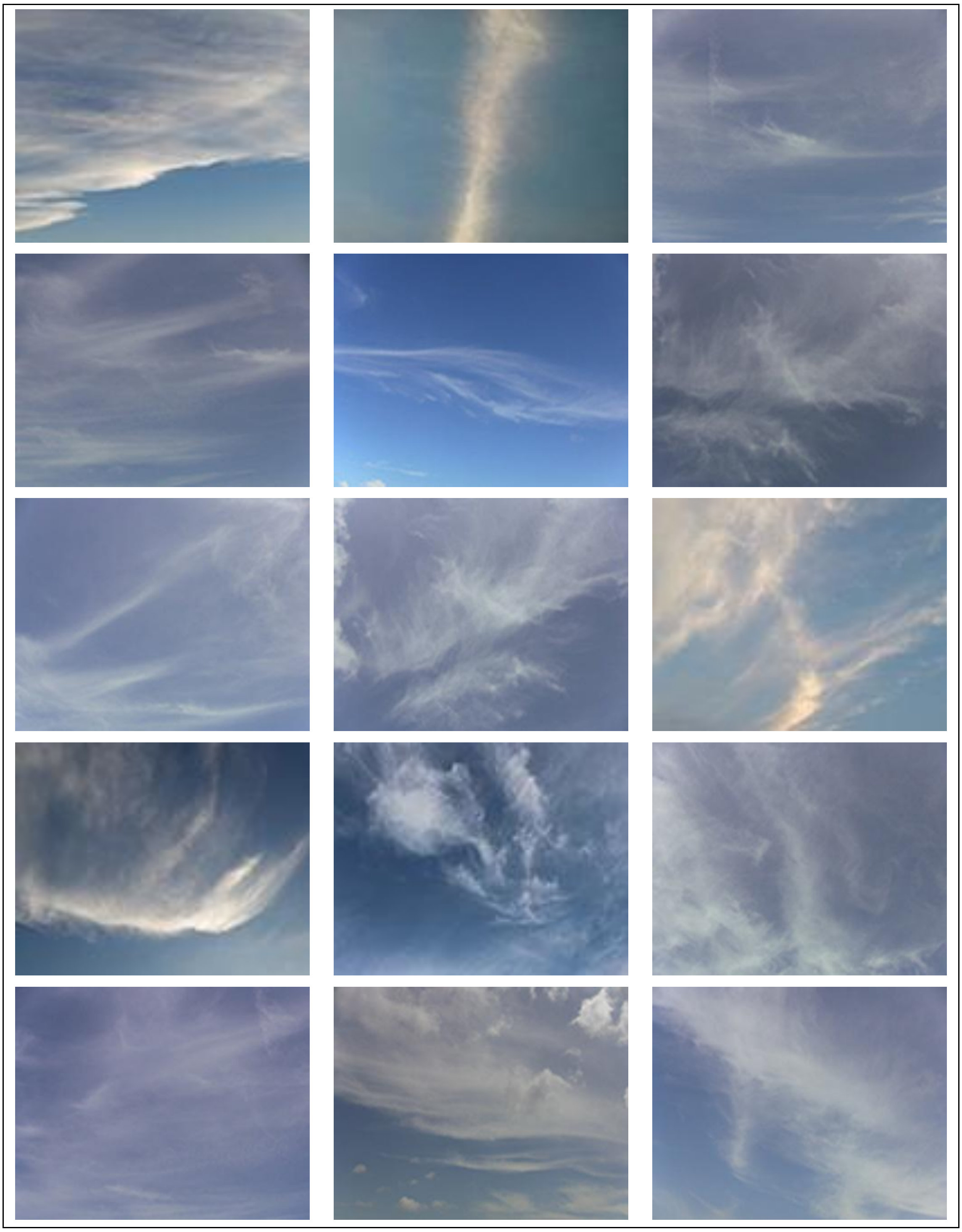
[62] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "WAHRSIS: A low-cost high-resolution whole sky imager with near-infrared capabilities," in *Proceedings of SPIE Infrared Imaging Systems*, 2014, vol. 9071.

[63] S. A. Pulinets, L. I. Morozova, and I. A. Yudin, "Synchronization of atmospheric indicators at the last stage of earthquake preparation cycle," *Research in Geophysics*, vol. 4, no. 1, Jan. 2015.

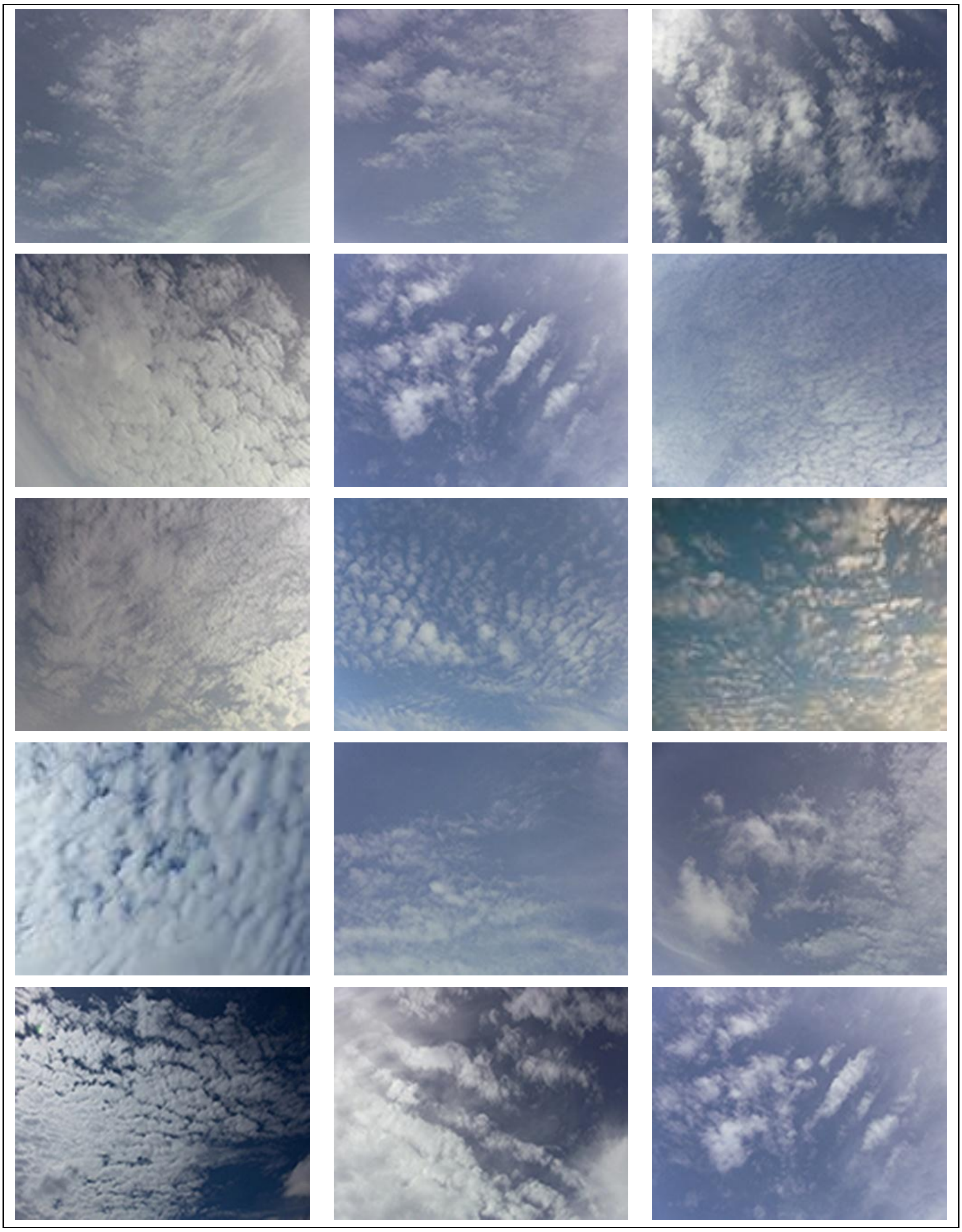
APPENDIX A

SAMPLE CLOUD IMAGES

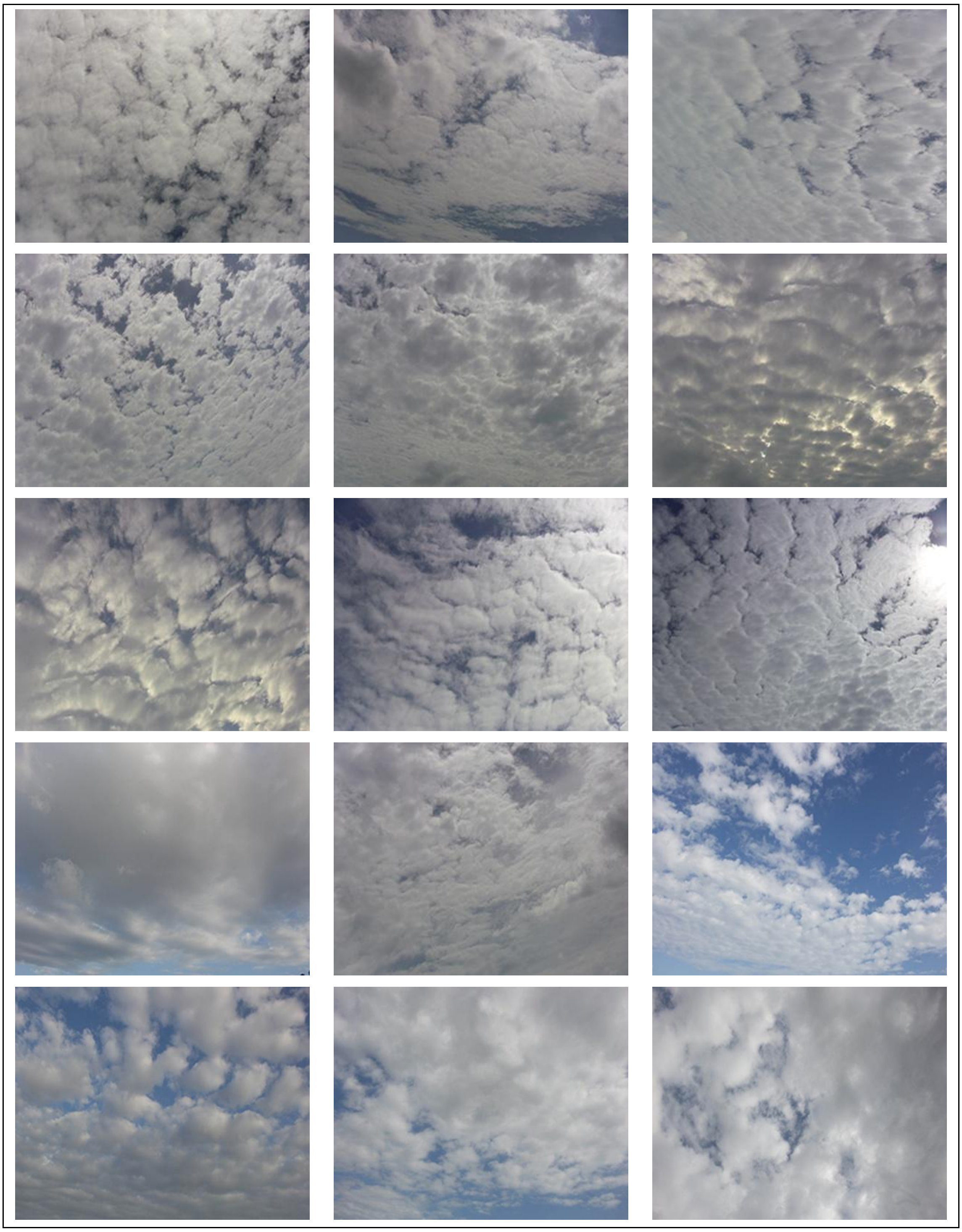
a. Class 1: Cirriform Clouds



b. Class 2: High Cumuliform Clouds

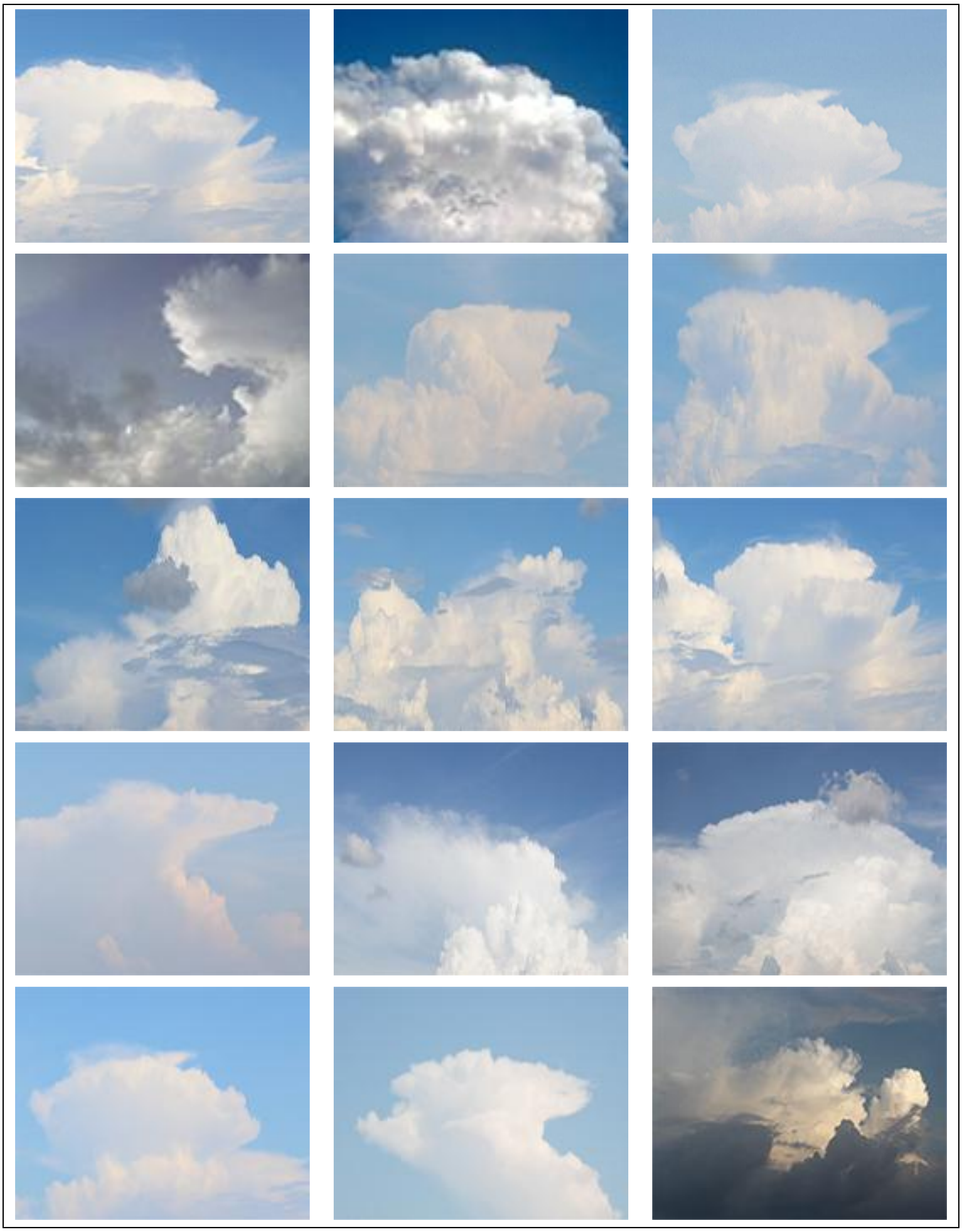


c. Class 3: Stratocumulus Clouds

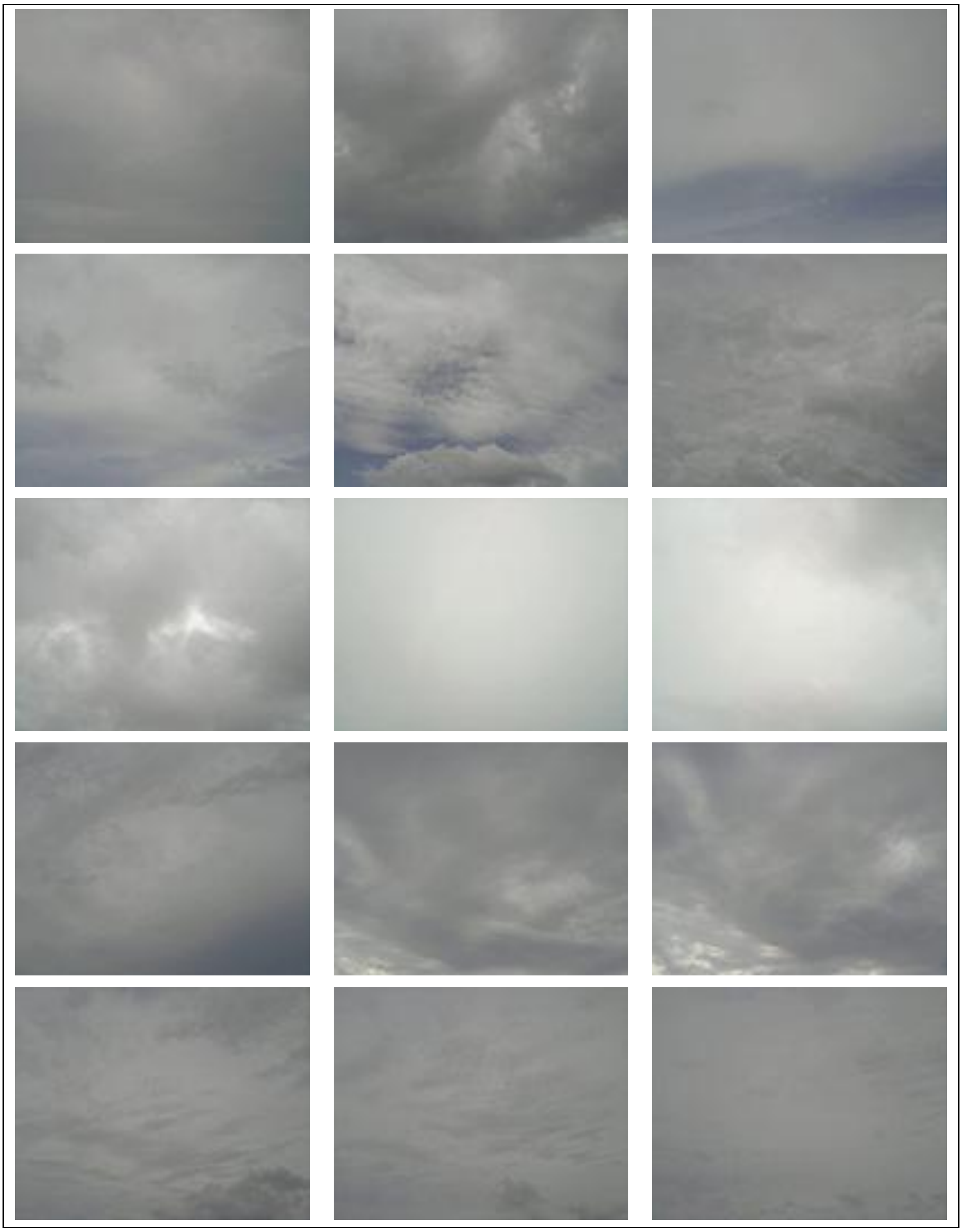


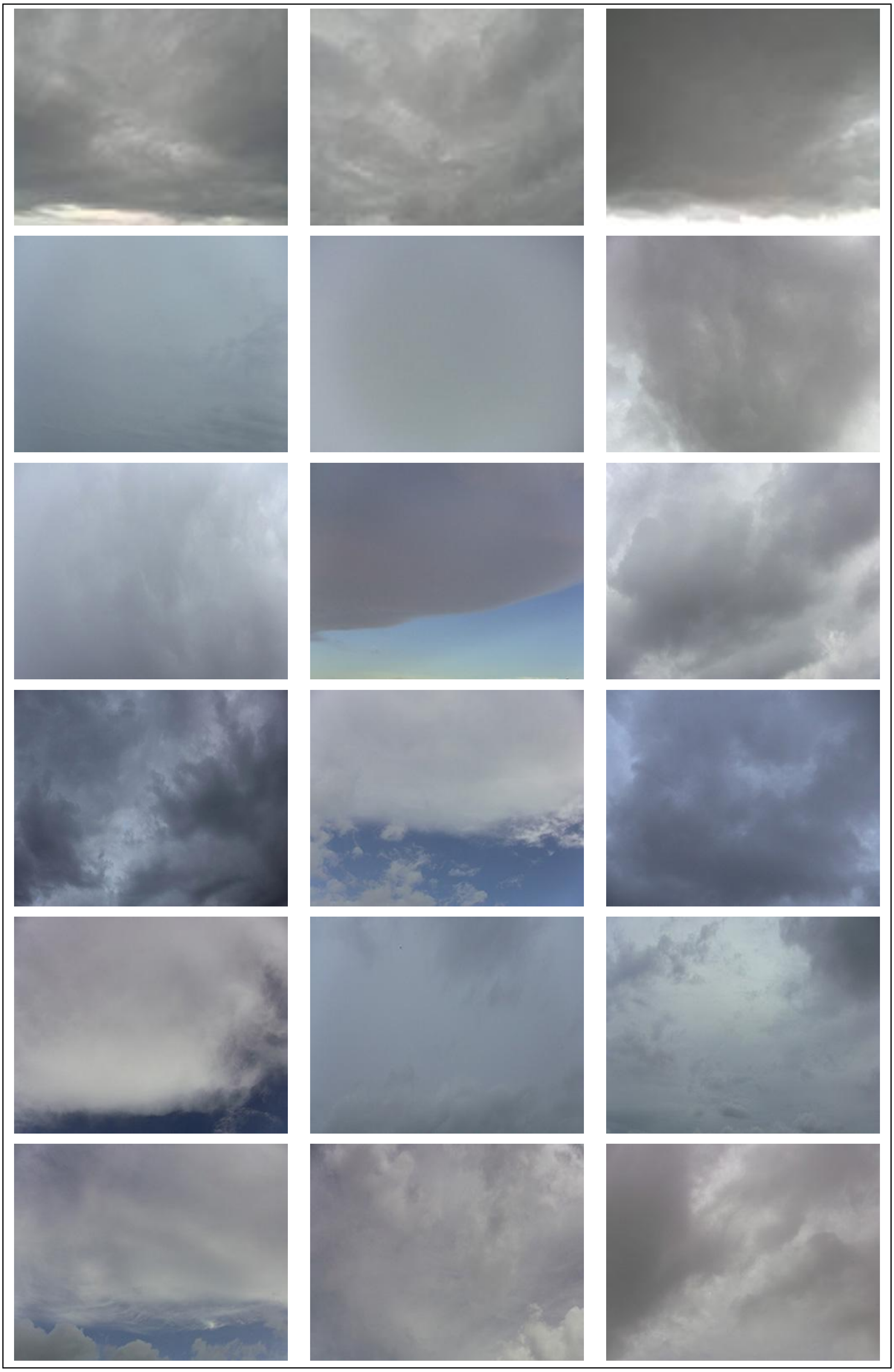
d. Class 4: Cumulus Clouds

e. Class 5: Cumulonimbus Clouds

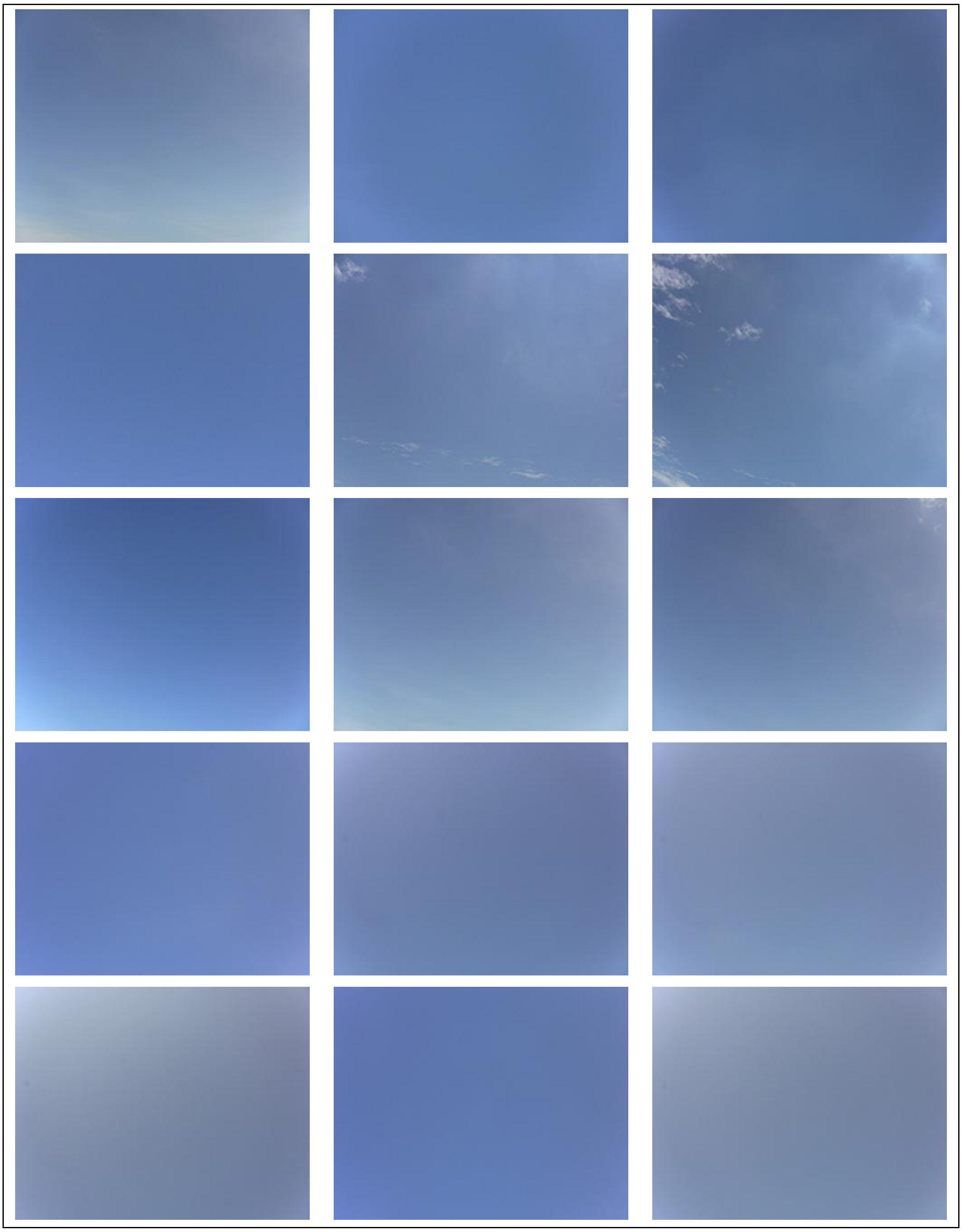


f. Class 6: Stratifom Clouds



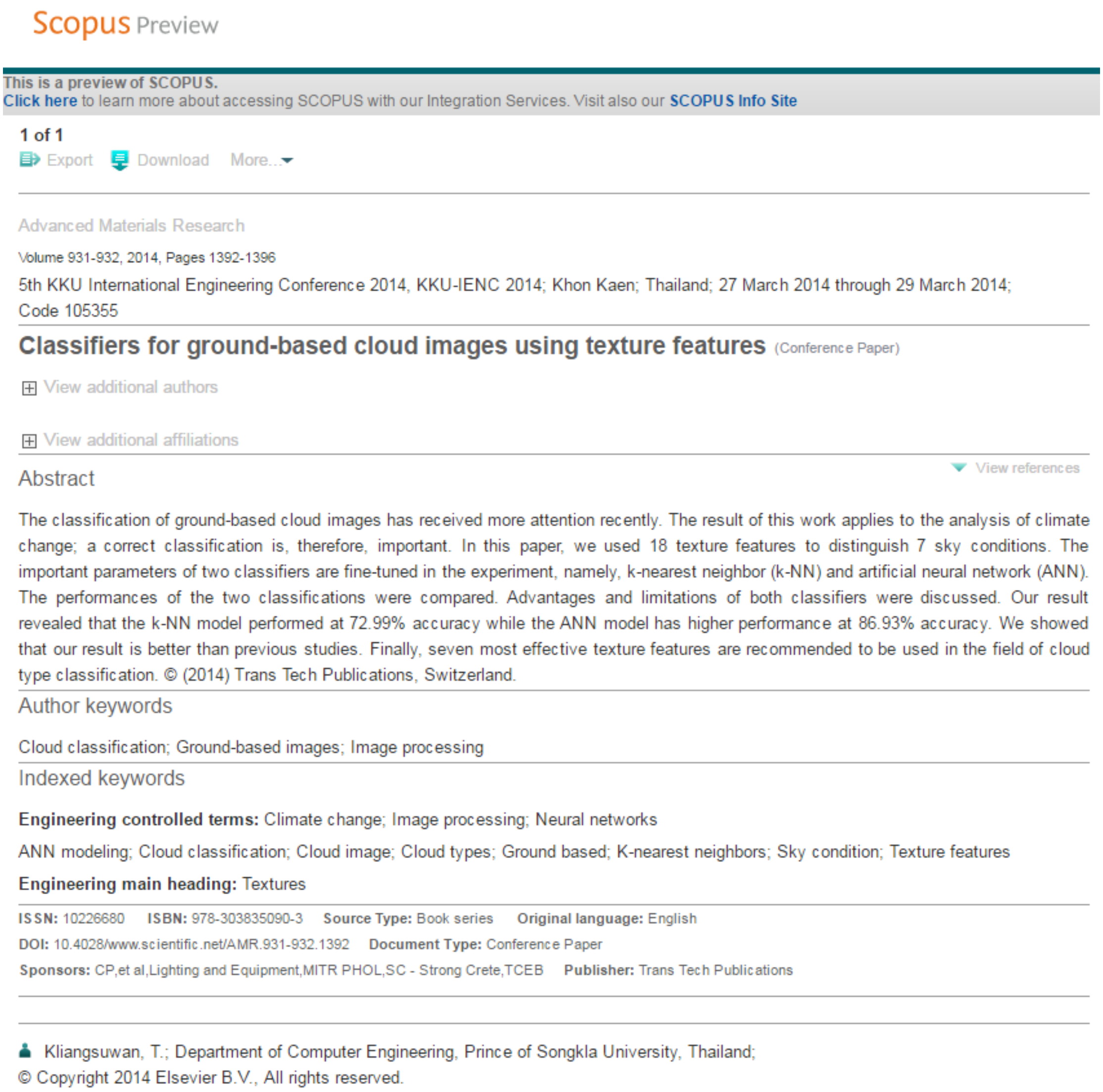


g. Class 7: Clear Sky



APPENDIX B

LIST OF PUBLICATIONS



This is a preview of SCOPUS.
Click here to learn more about accessing SCOPUS with our Integration Services. Visit also our SCOPUS Info Site

Advanced Materials Research

Volume 931-932, 2014, Pages 1392-1396

5th KKU International Engineering Conference 2014, KKU-IENC 2014; Khon Kaen; Thailand; 27 March 2014 through 29 March 2014; Code 105355

Classifiers for ground-based cloud images using texture features (Conference Paper)

Abstract

The classification of ground-based cloud images has received more attention recently. The result of this work applies to the analysis of climate change; a correct classification is, therefore, important. In this paper, we used 18 texture features to distinguish 7 sky conditions. The important parameters of two classifiers are fine-tuned in the experiment, namely, k-nearest neighbor (k-NN) and artificial neural network (ANN). The performances of the two classifications were compared. Advantages and limitations of both classifiers were discussed. Our result revealed that the k-NN model performed at 72.99% accuracy while the ANN model has higher performance at 86.93% accuracy. We showed that our result is better than previous studies. Finally, seven most effective texture features are recommended to be used in the field of cloud type classification. © (2014) Trans Tech Publications, Switzerland.

Author keywords

Cloud classification; Ground-based images; Image processing

Indexed keywords

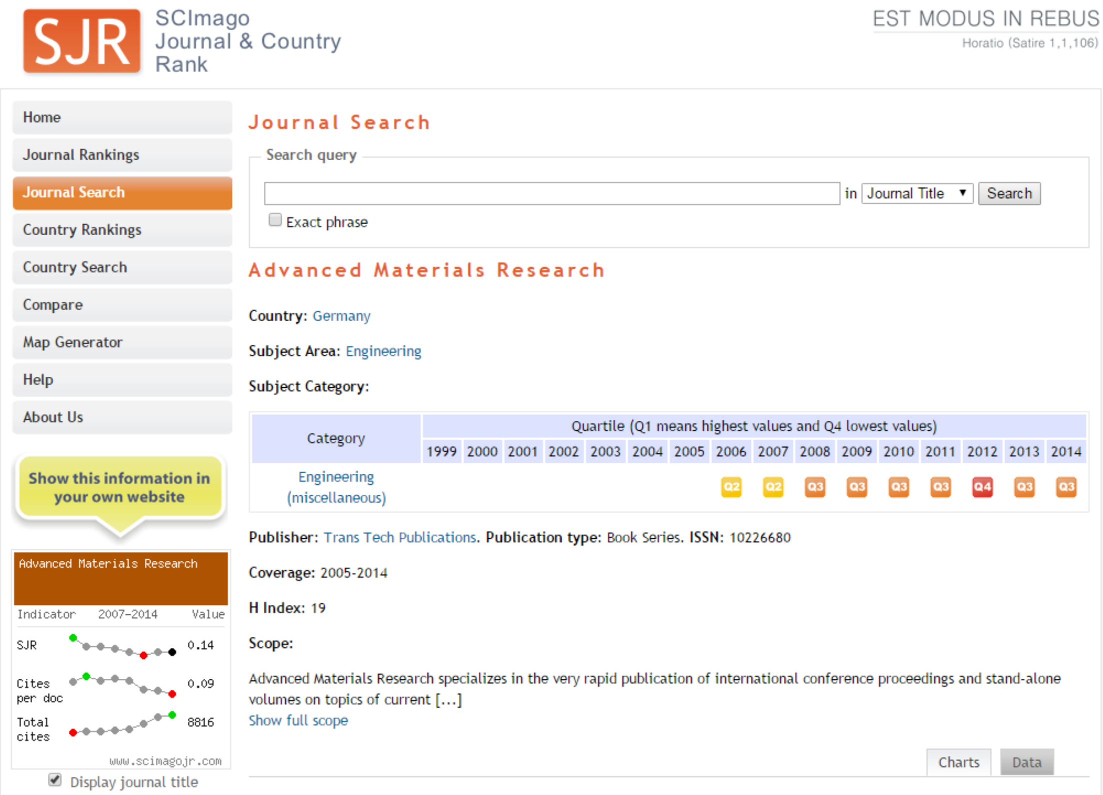
Engineering controlled terms: Climate change; Image processing; Neural networks

ANN modeling; Cloud classification; Cloud image; Cloud types; Ground based; K-nearest neighbors; Sky condition; Texture features

Engineering main heading: Textures

ISSN: 10226680 **ISBN:** 978-303835090-3 **Source Type:** Book series **Original language:** English
DOI: 10.4028/www.scientific.net/AMR.931-932.1392 **Document Type:** Conference Paper
Sponsors: CP,et al,Lighting and Equipment,MITR PHOL,SC - Strong Crete,TCEB **Publisher:** Trans Tech Publications

Kliangsuwan, T.; Department of Computer Engineering, Prince of Songkla University, Thailand;



SJR SCImago Journal & Country Rank

EST MODUS IN REBUS
Horatio (Satire 1,1,106)

Advanced Materials Research

Indicator	2007-2014	Value
SJR		0.14
Cites per doc		0.09
Total cites		8816

www.scimagojr.com

☑ Display journal title

Journal Search

Search query

in Journal Title ▼ Search

□ Exact phrase

Advanced Materials Research

Country: Germany

Subject Area: Engineering

Subject Category:

Category	Quartile (Q1 means highest values and Q4 lowest values)															
	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Engineering (miscellaneous)								Q2	Q2	Q3	Q3	Q3	Q3	Q4	Q3	Q3

Publisher: Trans Tech Publications. **Publication type:** Book Series. **ISSN:** 10226680

Coverage: 2005-2014

H Index: 19

Scope:

Advanced Materials Research specializes in the very rapid publication of international conference proceedings and stand-alone volumes on topics of current [...]

Show full scope

Advanced Materials Research Vols. 931-932 (2014) pp 1392-1396
© (2014) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMR.931-932.1392

Classifiers for Ground-Based Cloud Images Using Texture Features

Thitinan Kliangsuwan[1,a *] and Apichat Heednacram[1,b *]

[1]Department of Computer Engineering, Faculty of Engineering

Prince of Songkla University, Thailand

[a]thitinan.kl@gmail.com, [b]apichat@coe.phuket.psu.ac.th

Keywords: Image Processing, Cloud Classification, Ground-Based Images

Abstract. The classification of ground-based cloud images has received more attention recently. The result of this work applies to the analysis of climate change; a correct classification is, therefore, important. In this paper, we used 18 texture features to distinguish 7 sky conditions. The important parameters of two classifiers are fine-tuned in the experiment, namely, *k*-nearest neighbor (*k*-NN) and artificial neural network (ANN). The performances of the two classifications were compared. Advantages and limitations of both classifiers were discussed. Our result revealed that the *k*-NN model performed at 72.99% accuracy while the ANN model has higher performance at 86.93% accuracy. We showed that our result is better than previous studies. Finally, seven most effective texture features are recommended to be used in the field of cloud type classification.

Introduction

Clouds are part of a water cycle that controls the temperature of the earth by reflecting the radiation of the sun and from the ground. Clouds have different shapes and colors in different weather conditions. In Thailand, floodings and storms in recent years have caused damages that affected the population of 12.8 million which 813 people are dead and the damage value is as high as 1.44 trillion baht [1]. Thus, cloud classification is important because it is related to meteorology in the analysis of weather condition and prediction. Applications for cloud type classification include earthquake prediction [2], air traffic control [3], radiative forcing [4] and global warming [5].

In pattern recognition, cloud classification is difficult than general object classification because cloud has unsettled shape and changes its shape over time. Satellite images are used in cloud classification since 1970s [6]. However, they provide limited details of clouds in the specific area [7, 8, 9] and data access is sometimes restricted. A few years ago, ground-based images have been increasingly used in cloud classification. Many ground-based imagers such as the total sky imager (TSI), the whole sky camera (WSC), and the all sky imager (ASI) are widely used [10, 11]. Ground-based images have the advantage over satellite images in terms of lower cost, and more localized information. Souza-Echeret al. [12] proposed the supervised parallelepiped method for classifying clear sky images with 94% accuracy using a digital camera. Singh and Glennen [3] used co-occurrence features for classifying five sky conditions with 59.5% accuracy using *k*-nearest neighbor classifier and with 64% accuracy using neural network classifier.

Texture features are used to describe characteristics of cloud. However, the accuracy depends on how careful we choose features to use in feature extraction methods. In this paper, we develop an automatic cloud classification using ground-based sky images from a digital camera. We combine several texture features and we select the 18 suitable ones for classifying images into seven sky conditions namely, cirrus, cirro and altocumulus, stratocumulus, cumulus, cumulonimbus, stratus, and clear sky (see Fig. 1). In classification part, we use two classifiers namely, *k*-nearest neighbor and artificial neural network. Then, we compare the performance of both classifiers in terms of percentage of accuracy.

We organize the paper as follows. First, we introduce the technical background for cloud classification. Then, the experimental results are presented. Lastly, we conclude our contribution and suggest the future work.

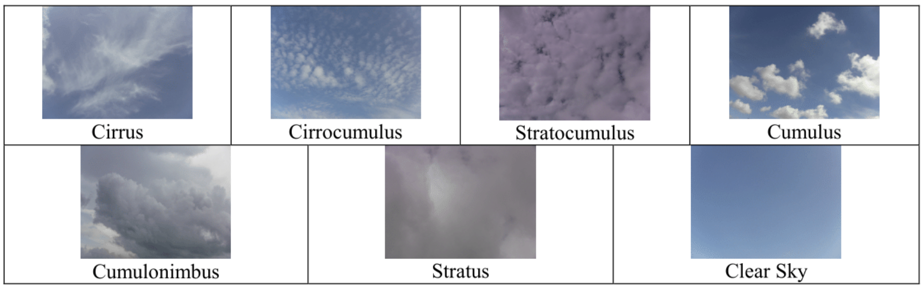


Figure 1. Ground-Based Cloud Images of Seven Sky Conditions.

Technical Background

We do the feature extraction, before we apply a classifier to compute a class type based on the given features. Features are extracted from grayscale image by splitting channels of the image into Red (R), Green (G), or Blue (B) channel. There are several feature extraction methods. We use texture features because they are commonly used in cloud classification [3, 7, 8].

Texture Features. Texture features describe texture of image region. This paper uses two sources of texture features in the calculation. The first type of texture features is extracted from image directly. These are mean of red channel (*ME*), standard deviation of blue channel (*SD*) and difference of mean between each channel (D_{ij}). All equations are defined below [7], [8]. The annotation *N* is a number of pixels in image, x_i is a value of pixel *i*-th, and $p(x_i)$ is a probability of x_i.

Texture Feature 1: $$ME = \sum_{i=0}^{N-1} x_i p(x_i) \quad (1)$$

Texture Feature 2: $$SD = \left(\sum_{i=0}^{N-1} (x_i - ME)^2 p(x_i)\right)^{1/2} \quad (2)$$

Texture Feature 3 – 5: $$D_{ij} = ME_i - ME_j \;\; for \;\; i, j \in \{R,\ G,\ B\} \; where \; i \neq j \quad (3)$$

The second type of texture features is computed from Grey Level Co-occurrence Matrix (GLCM) which is a square matrix with a number of columns equal a number of grey levels. Each element in the matrix is the frequency that two pixels occur ($P^{\Delta}(a,b)$) [7]. We use 11 out of 14 Haralick texture features [13] excluding the following 3 features; correlation f_3, information measures of correlation f_{13}, and maximal correlation coefficient f_{14}. We compute Texture Feature 6 – 16 from the 11 features on the red channel and also compute on the blue channel of contrast f_2 and homogeneity f_5 [13] become Texture Feature 17 – 18.

Two commonly used classifiers [3], [7], pattern recognition tools for distinguishes classes, for cloud classification are *k*-nearest neighbor and artificial neural network.

***k*-Nearest Neighbor.** The *k*-nearest neighbor (*k*-NN) is a popular technique in cloud classification because of low computation and its ability to solve complex problem. Most of all, it is simple to implement [3], [7]. A brief detail of *k*-NN algorithm is that it works by considering *k* neighbors of a sample before classifying that sample as a class by a majority vote of its neighbors; that is the sample is assigned to the class most frequent among its *k* nearest neighbors. In theory, the larger *k* is more suitable for infinite number of samples. However, this is impossible in practice [14]. Hence, we are required to choose the optimal *k* within the available samples.

Artificial Neural Network. When a problem is complex, artificial neural network (ANN) is often used [3]. ANN is composed of three main layers namely, input layer, hidden layer, and output layer. It can be used to solve nonlinear problems without considering the relationship between input and output. However, the classifier requires the tuning of multiple parameters in the model [15] for the best performance. Therefore, to achieve high accuracy we will examine key parameters of ANN in the experimental section. In this paper, multilayer feed forward neural network is used.

Experimental Results

The experiment uses 696 ground-based cloud images taken from a digital camera. These images are scaled to 320x240 pixels for all experiments. The 18 texture features explained earlier are implemented and the two classifiers used are ANN and *k*-NN. First, we tuned the parameters of two classifiers to achieve high accuracy. Then, we compared and discussed the performance of the two classifiers. In all experiments, we fixed a learning rate of ANN to 0.01 and a momentum to 0.9 and used the leave-one-out cross-validation (LOOCV) [15] to evaluate the results.

Selection of *k* Value for *k*-NN. We decide a suitable *k* as shown in Table 1. The accuracy [%] values are the average percentages of all correctly classified instances.

Table 1. The *k* Value of *k*-NN Model between 1 – 14.

k-value	1	2	3	4	5	6	7
Accuracy [%]	60.20	50.00	49.86	50.14	51.72	50.72	52.01
k-value	8	9	10	11	12	13	14
Accuracy [%]	51.29	51.72	49.86	48.42	49.71	49.86	50.00

The *k* values from 1 to 14 were tested with a distance measure set to the Euclidean distance. In practice, *k* should not be too large as it can lead to over-smoothed boundaries [14]. According to Table 1, when *k* = 1 the accuracy is highest among others. For this problem, *k* = 1 can separate noises from a correct sample better than other *k* values. Therefore, the remaining of this paper will apply *k* = 1 as the parameter of *k*-NN model.

Selection of Distance Measures for *k*-NN. Distance function is an important parameter of *k*-NN classifier for determining nearest neighbor of the test data. There are many distance measures for *k*-NN and choosing a suitable measure is an important problem in supervised learning techniques [16]. We used 7 distance measures as shown in Table 2.

Table 2. Accuracy When Using Seven Different Distance Measures.

Distance Measure	City Block	Chebychev	Correlation	Cosine	Euclidean	Min-kowski	Standardized Euclidean
Accuracy [%]	61.06	61.49	59.34	59.91	60.20	60.20	72.99

From Table 2, each distance measure provides similar results except that Standardized Euclidean distance has around 10% higher accuracy than others. We, therefore, resolve that Standardized Euclidean distance measure is compatible with a form of our training data distribution.

Selection of Activation Function for ANN. ANN model has many parameters that we can vary. The activation function of the hidden layer is examined here. Note that the activation function of the input layer is fixed as linear to keep the same input values throughout the model. For the output layer, we fixed as hyperbolic tangent because this function can generate binary outputs [17].

Table 3. Different Activation Functions in Hidden Layer.

Activation Function	Radial basis	Tangent	Linear	Log-sigmoid
Accuracy [%]	80.45	82.04	77.87	80.32

From Table 3, there are 5 hidden nodes in the hidden layer. The result shows that the hyperbolic tangent function outperforms the other 3 functions because the hyperbolic tangent can provide binary outputs which are suitable for classification problems. However, it provides a good result if this function is used in both hidden and output layers [17].

Selection of Number of Hidden Nodes for ANN. The number of hidden nodes in a hidden layer is varied as 3, 5, 7, and 9. The activation function is set to hyperbolic tangent.

Table 4. The Number of Hidden Nodes is Chosen as 3, 5, 7, and 9.

Number of Hidden Nodes	3	5	7	9
Accuracy [%]	77.44	82.04	86.93	82.90

From Table 4, the number of hidden nodes equal to 7 gives the best result. The number of hidden nodes affects the generalization error. When there are fewer hidden nodes such as 3 hidden nodes, this might cause underfitting and high statistical bias. In contrast, if we design our hidden layer with many hidden nodes such as 9 hidden nodes, it might cause overfitting and high variance [18].

Comparison of *k*-NN and ANN. In the past experiments, it may seem that ANN classifier is cumbersome to use than *k*-NN due to the fine-tuning of various parameters. However, if we can tune the parameters appropriately, it can lead to a more accurate classification over *k*-NN.

Table 5. The Accuracy Comparison of *k*-NN and ANN.

Classifiers	*k*-NN	ANN
Accuracy [%]	72.99	86.93

$$\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & \mathbf{58.77} & 7.89 & 1.75 & 17.54 & 11.40 & 0.88 & 1.75 \\
2 & 13.98 & \mathbf{64.52} & 13.98 & 5.38 & 2.15 & 0.00 & 0.00 \\
3 & 1.20 & 7.23 & \mathbf{75.90} & 6.02 & 3.61 & 6.02 & 0.00 \\
4 & 17.12 & 0.90 & 1.80 & \mathbf{69.37} & 10.81 & 0.00 & 0.00 \\
5 & 8.94 & 4.07 & 6.50 & 13.82 & \mathbf{65.04} & 1.63 & 0.00 \\
6 & 1.11 & 0.00 & 5.56 & 0.00 & 0.00 & \mathbf{92.22} & 1.11 \\
7 & 2.44 & 1.22 & 0.00 & 0.00 & 0.00 & 1.22 & \mathbf{95.12}
\end{array}$$

(a)

$$\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & \mathbf{79.82} & 4.39 & 2.63 & 10.53 & 1.75 & 0.00 & 0.88 \\
2 & 4.30 & \mathbf{86.02} & 5.38 & 4.30 & 0.00 & 0.00 & 0.00 \\
3 & 3.61 & 14.46 & \mathbf{72.29} & 4.82 & 2.41 & 2.41 & 0.00 \\
4 & 8.11 & 0.00 & 0.00 & \mathbf{82.88} & 8.11 & 0.00 & 0.90 \\
5 & 1.63 & 0.00 & 0.00 & 3.25 & \mathbf{95.12} & 0.00 & 0.00 \\
6 & 2.22 & 0.00 & 1.11 & 0.00 & 1.11 & \mathbf{95.56} & 0.00 \\
7 & 0.00 & 0.00 & 0.00 & 1.22 & 1.22 & 1.22 & \mathbf{96.34}
\end{array}$$

(b)

Figure 2. Confusion Matrix for the Ground-Based Cloud Image Using Two Different Classifiers (a) *k*-NN Classifer (b) ANN Classifier.

Table 5 shows that the accuracy of ANN is about 14% higher than *k*-NN. While the result of Singh and Glennen is 64% accurate for a classification of five sky conditions using ANN, we obtained 86.93% accuracy for a classification of seven sky conditions using also ANN. Nevertheless, our *k*-NN did not perform badly in the comparison with Singh and Glennen. For the *k*-NN, we received 72.99% accuracy while Singh and Glennen have 59.5% accuracy for a classification of less sky conditions. Although Souza-Echeret al. have 94% accuracy, their classification is only for clear sky condition. If we classify only this class, we would get 96.34%. The percentage of the correct classification for each class is described in Fig. 2, using confusion matrix. According to Fig. 2, each row represents the percentage of the instances in a true class and each column represents the percentage of the instances in a predicted class. Note that Class 1 refers to cirrus and Class 7 refers to clear sky which is in the same order as explained in the introduction section. In Fig. 2(a), four classes have less than 75% accuracy leading to a poorer result when using *k*-NN. On the other hand, in Fig. 2(b), the majority of correctly classified instances for ANN are over 80%, some are even as high as 96.34%. To further improve this result, we could try to find more effective features for distinguishing cirrus and stratocumulus from the rest of the other clouds.

Recommended Texture Features. The 18 texture features used in the experiment did not produce the same impact. Therefore, we rank the most effective texture features using a knock-out method. The results of ranking are as follows: 1) deviation of blue channel, 2) difference of mean red and mean blue channel, 3) homogeneity of blue channel, 4) angular second moment f_1 of red channel, 5) variance f_4 of red channel, 6) entropy f_9 of red channel, and 7) information measures of correlation f_{12} of red channel. With merely 7 texture features above, we get the accuracy as high as 77.44% using ANN with 7 hidden nodes.

Conclusion

The seven texture features in recognizing each sky condition were recommended. If the 18 texture features were combined, this would give even better results. Although, ANN was difficult to fine-tune, ANN provided a better accuracy than *k*-NN with 86.93% of instances are correctly classified. This result is about 23% higher than the previous studies which were using also a digital camera and ANN method but the number of classes in those studies was only for five sky conditions, in our case we have seven sky conditions. In particular, our classification for clear sky condition returned 96.34% accuracy which is higher than the previous studies. In the future, we will experiment with not only texture features but also other kinds of features such as a 2 dimension Fast Fourier Transform to achieve higher percentage of accuracy.

References

[1] W. C. K. Anuroj, Case Study of Stress from Thai Flood 2012, in School Age Student, Pathumthani Province, Thailand, R. Thai Airf. Med. Gaz., 58(1) (2012) 34–38.

[2] G. Guangmeng, and Y. Jie, Three Attempts of Earthquake Prediction with Satellite Cloud Images, Nat Hazards Earth Syst Sci, 13(1) (2013) 91–95.

[3] M. Singh, and M. Glennen, Automated Ground-Based Cloud Recognition, Pattern Anal. Appl., 8(3) (2005) 258–271.

[4] J. Calbó, and J.-A. González, Empirical Studies of Cloud Effects on UV Radiation: A Review, Rev. Geophys., 43(2) (2005).

[5] J. T. Houghton, Y. Ding, D. J. Griggs, M. Noguer, P. J. van der LINDEN, X. Dai, K. Maskell, and C. A. Johnson, Climate Change 2001: The Scientific Basis, Cambridge Uni. Press, 2001.

[6] J. A. Parikh, Automatic Cloud Classification and Segmentation, Ph.D. dissertation, University of Maryland at College Park, College Park, MD, USA, 1977.

[7] A. Heinle, A. Macke, and A. Srivastav, Automatic Cloud Classification of Whole Sky Images, Atmospheric Meas. Tech. Discuss., 3(1) (2010) 269–299.

[8] J. Calbo, and J. Sabburg, Feature Extraction from Whole-Sky Ground-Based Images for Cloud-Type Recognition, J. Atmospheric Ocean. Technol., 25(1) (2008) 3–14.

[9] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, Salient Local Binary Pattern for Ground-Based Cloud Classification, Acta Meteorol. Sin., 27 (2013) 211–220.

[10] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pages, Retrieving Cloud Characteristics from Ground-Based Daytime Color All-Sky Images, JTECH, 23(5) (2006) 633–652.

[11] K. B. Widener, and C. N. Long, All Sky Imager, U.S. Patent Application 10/377, 042. (2003)

[12] M. P. Souza-Echer, E. B. Pereira, L. S. Bins, and M. A. R. Andrade, A Simple Method for the Assessment of the Cloud Cover State in High-Latitude Regions by a Ground-Based Digital Camera, J. Atmospheric Ocean. Technol., 23(3) (2006) 437–447.

[13] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, Textural Features for Image Classification, IEEE Trans. on Syst. Man Cybern., (6) (1973) 610–621.

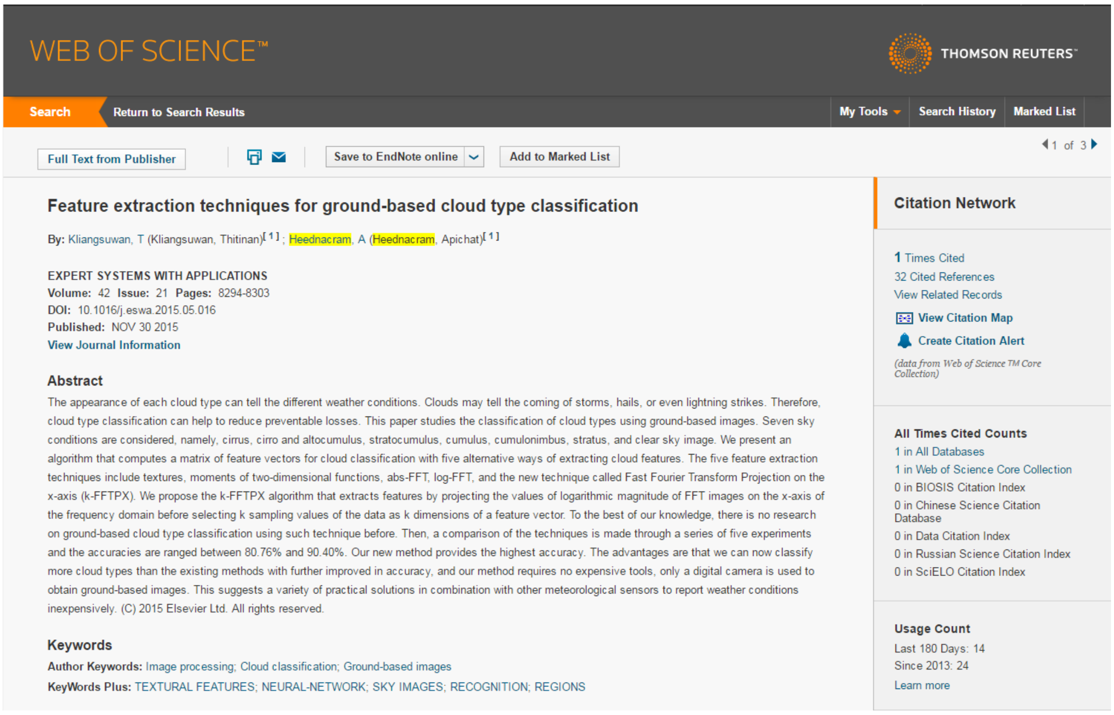
[14] O. Veksler, Nonparametric Density Estimation Nearest Neighbors, KNN [Online], Available: http://www.cs.haifa.ac.il/~rita/ml_course/lectures/KNN.pdf (2013)

[15] I. H. Witten, and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann, San Francisco, 2005.

[16] I. Bonet, A. Rodríguez, R. Grau, M. M. García, Y. Saez, and A. Nowé, Comparing Distance Measures with Visual Methods, MICAI, AAI, Springer, California, 2008, pp. 90–99.

[17] B. Karlik, and A. V. Olgac, Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks, IJAE, 1(4) (2011) 111–122.

[18] A. Kasapis, MLPs and Pose, Expression Classification, in Proc. of UNiS Report, (2003) 1-87.



Feature extraction techniques for ground-based cloud type classification

By: Kliangsuwan, T (Kliangsuwan, Thitinan)[1] ; Heednacram, A (Heednacram, Apichat)[1]

EXPERT SYSTEMS WITH APPLICATIONS
Volume: 42 **Issue:** 21 **Pages:** 8294-8303
DOI: 10.1016/j.eswa.2015.05.016
Published: NOV 30 2015
View Journal Information

Abstract

The appearance of each cloud type can tell the different weather conditions. Clouds may tell the coming of storms, hails, or even lightning strikes. Therefore, cloud type classification can help to reduce preventable losses. This paper studies the classification of cloud types using ground-based images. Seven sky conditions are considered, namely, cirrus, cirro and altocumulus, stratocumulus, cumulus, cumulonimbus, stratus, and clear sky image. We present an algorithm that computes a matrix of feature vectors for cloud classification with five alternative ways of extracting cloud features. The five feature extraction techniques include textures, moments of two-dimensional functions, abs-FFT, log-FFT, and the new technique called Fast Fourier Transform Projection on the x-axis (k-FFTPX). We propose the k-FFTPX algorithm that extracts features by projecting the values of logarithmic magnitude of FFT images on the x-axis of the frequency domain before selecting k sampling values of the data as k dimensions of a feature vector. To the best of our knowledge, there is no research on ground-based cloud type classification using such technique before. Then, a comparison of the techniques is made through a series of five experiments and the accuracies are ranged between 80.76% and 90.40%. Our new method provides the highest accuracy. The advantages are that we can now classify more cloud types than the existing methods with further improved in accuracy, and our method requires no expensive tools, only a digital camera is used to obtain ground-based images. This suggests a variety of practical solutions in combination with other meteorological sensors to report weather conditions inexpensively. (C) 2015 Elsevier Ltd. All rights reserved.

Keywords

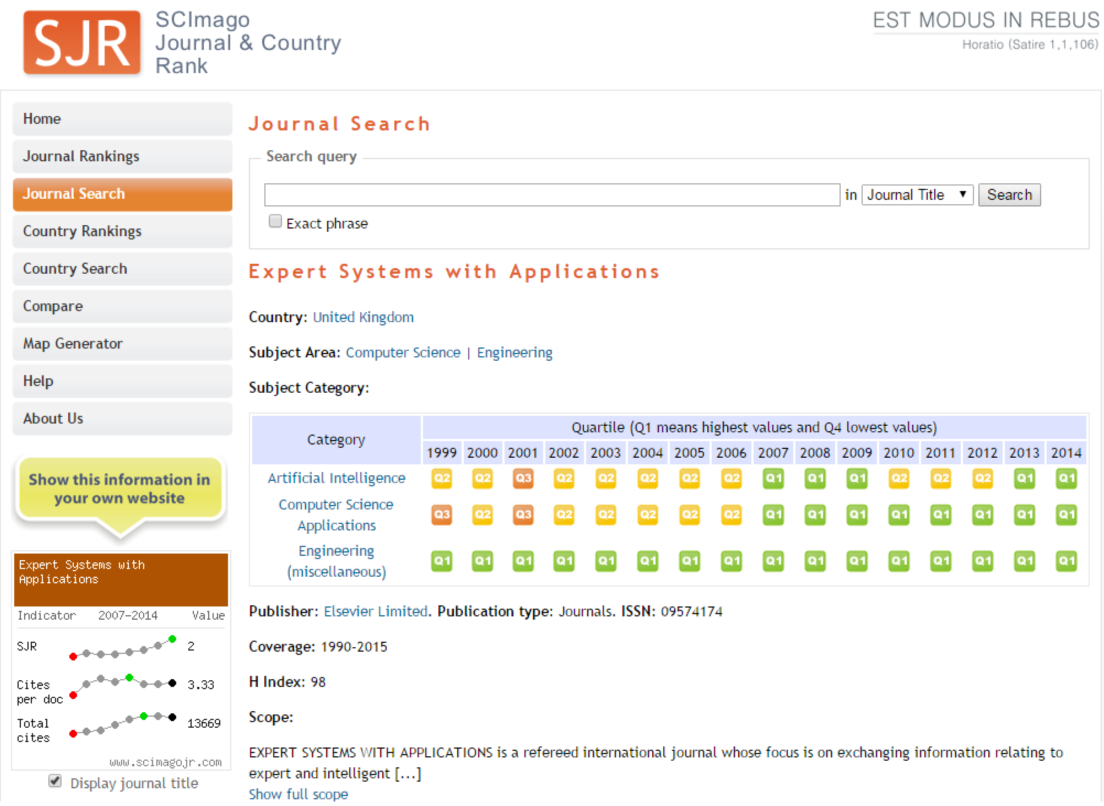
Author Keywords: Image processing; Cloud classification; Ground-based images
KeyWords Plus: TEXTURAL FEATURES; NEURAL-NETWORK; SKY IMAGES; RECOGNITION; REGIONS

Citation Network

1 Times Cited
32 Cited References
View Related Records
View Citation Map
Create Citation Alert
(data from Web of Science TM Core Collection)

All Times Cited Counts
1 in All Databases
1 in Web of Science Core Collection
0 in BIOSIS Citation Index
0 in Chinese Science Citation Database
0 in Data Citation Index
0 in Russian Science Citation Index
0 in SciELO Citation Index

Usage Count
Last 180 Days: 14
Since 2013: 24
Learn more



Journal Search

Search query
in Journal Title ▾ Search
□ Exact phrase

Expert Systems with Applications

Country: United Kingdom

Subject Area: Computer Science | Engineering

Subject Category:

Category	Quartile (Q1 means highest values and Q4 lowest values)															
	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Artificial Intelligence	Q2	Q2	Q3	Q2	Q2	Q2	Q2	Q2	Q1	Q1	Q1	Q2	Q2	Q2	Q1	Q1
Computer Science Applications	Q3	Q2	Q3	Q2	Q2	Q2	Q2	Q2	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1
Engineering (miscellaneous)	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1	Q1

Publisher: Elsevier Limited. **Publication type:** Journals. **ISSN:** 09574174

Coverage: 1990-2015

H Index: 98

Scope:

EXPERT SYSTEMS WITH APPLICATIONS is a refereed international journal whose focus is on exchanging information relating to expert and intelligent [...]
Show full scope

Expert Systems with Applications

Indicator	2007-2014	Value
SJR		2
Cites per doc		3.33
Total cites		13669

☑ Display journal title

Expert Systems with Applications 42 (2015) 8294–8303

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Feature extraction techniques for ground-based cloud type classification

Thitinan Kliangsuwan, Apichat Heednacram

Department of Computer Engineering, Prince of Songkla University, Phuket, Thailand

ARTICLE INFO

Article history:
Available online 22 May 2015

Keywords:
Image processing
Cloud classification
Ground-based images

ABSTRACT

The appearance of each cloud type can tell the different weather conditions. Clouds may tell the coming of storms, hails, or even lightning strikes. Therefore, cloud type classification can help to reduce preventable losses. This paper studies the classification of cloud types using ground-based images. Seven sky conditions are considered, namely, cirrus, cirro and altocumulus, stratocumulus, cumulus, cumulonimbus, stratus, and clear sky image. We present an algorithm that computes a matrix of feature vectors for cloud classification with five alternative ways of extracting cloud features. The five feature extraction techniques include textures, moments of two-dimensional functions, abs-FFT, log-FFT, and the new technique called Fast Fourier Transform Projection on the *x*-axis (*k*-FFTPX). We propose the *k*-FFTPX algorithm that extracts features by projecting the values of logarithmic magnitude of FFT images on the *x*-axis of the frequency domain before selecting *k* sampling values of the data as *k* dimensions of a feature vector. To the best of our knowledge, there is no research on ground-based cloud type classification using such technique before. Then, a comparison of the techniques is made through a series of five experiments and the accuracies are ranged between 80.76% and 90.40%. Our new method provides the highest accuracy. The advantages are that we can now classify more cloud types than the existing methods with further improved in accuracy, and our method requires no expensive tools, only a digital camera is used to obtain ground-based images. This suggests a variety of practical solutions in combination with other meteorological sensors to report weather conditions inexpensively.

1. Introduction

Weather conditions affect human life greatly in terms of daily life since some occupations depend on these weather conditions, for example, farmers and fishermen. In 2009, a total of 385 natural disasters have caused the damages that affected over the population of 217 million which more than 297,000 people worldwide are dead and the damage value of economic is as high as US$ 123.9 billion (Guha-Sapir, Vos, Below, & Ponserre, 2011). Therefore, the knowledge of recognition and understanding weather conditions is important for preventing the unexpected losses. One area of weather element recognition is cloud type classification since different cloud types can lead to different weather conditions.

Traditionally, classification of cloud types requires specialists to do it manually. However, it does not appear to have many specialists in this area. Moreover, the speed of manual classification is limited and human errors are sometimes introduced into the system. Each individual's experiences are also different. Hence, there have been many attempts to develop an automatic cloud type classification system (Aha & Bankert, 1994; Ambroise, Sèze, Badran, & Thiria, 2000; Bankert, 1994; Buch, Sun, & Thorne, 1995; Calbó & Sabburg, 2008; Fan, Changsheng, & Weimin, 1997; Heinle, Macke, & Srivastav, 2010; Heinzmann, 1993; Kaur & Ganju, 2008; Lee, Weger, Sengupta, & Welch, 1990; Lee, Lin, & Wahba, 2004; Martínez-Chico, Batlles, & Bosch, 2011; Shangguan, Hao, Lu, & Wu, 2007; Singh & Glennen, 2005; Souza-Echer, Pereira, Bins, & Andrade, 2006).

Since 1977 researchers have begun to use satellite images as the input (Parikh, 1977). However, this solution is expensive, and the images are sometimes restricted for public access. Furthermore, the satellite images are not suitable for the specific area of interests because of their lack of local details (Calbó & Sabburg, 2008; Singh & Glennen, 2005). Later, ground based imager devices were introduced (Long, Sabburg, Calbó, & Pagès, 2006). There are two types of imagers, namely the total sky imager (TSI) and the whole sky camera (WSC). Both of the imagers are expensive. Therefore, using digital camera is more suitable for smaller research groups and independent study. Moreover, the digital camera provides specific information, low cost, and less cumbersome than others.

In this paper, we will develop an automatic cloud type classification system for ground-based digital camera using image processing and pattern recognition. Seven different cloud types for

E-mail addresses: thitinan.kl@gmail.com (T. Kliangsuwan), apichat@coe.psu.ac.th (A. Heednacram)

http://dx.doi.org/10.1016/j.eswa.2015.05.016
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

our recognition are cirrus, cirro and altocumulus, stratocumulus, cumulus, cumulonimbus, stratus, and clear sky. We extract texture features from cloud images and use these information in the training process of the classification. Artificial neural network is then used for classifying instances. Moreover, we add three types of features based on Fourier transform. The first two types use logarithmic and absolute magnitudes for extracting texture features of Fast Fourier Transform (FFT) images. The last type uses logarithmic magnitude but we project these values on the *x*-axis. Our main contribution is a novel feature that uses a projection of logarithmic magnitude of the FFT onto the *x*-axis. We call this feature *k*-FFTPX.

We organize the paper as follows. In Section 2, we give the literature review of cloud type classification. In Section 3, various features used in the classification are explained. In Section 4, we describe artificial neural network which is the main classifier used in the experiment. In Section 5, the methodology for cloud type classification is given. In Section 6, we apply various features for the classification and present the experimental results. Finally, in Section 7 we summarize the results.

2. Literature review

Most of automatic cloud type classifications use a set of satellite images as an input. Heinzmann (1993) provided fuzzy logic approach for classifying four cloud classes. Lee et al. (2004) performed multi-category support vector machine (MSVM) to classify each pixel into three cloud classes. Lee et al. (1990) used a neural network with texture features to classify sub-regions into one of three cloud types. Bankert (1994) exploited a probabilistic neural network (PNN) to classify each area into one of ten cloud classes which give 79.80% correctly classified. Aha and Bankert (1994) introduced feature selection algorithms for classifying ten cloud classes. Forward sequential selection combined with IB1 gives the best accuracy of 88%. Fan et al. (1997) used a bispectral cloud classification method based on man-computer interactive way to classify land, water, and six types of clouds. The method has an accuracy of 87.10%. Ambroise et al. (2000) presented Probabilistic Self-Organizing Maps for classifying nine cloud types. The accuracy of the classification is 63%. Shangguan et al. (2007) proposed texture feature analysis combined with Variational theory to extract texture features. Kaur and Ganju (2008) used singular value decomposition (SVD) to extract the salient spectral and textural features to classify clouds as low, medium or high clouds. This technique gives an accuracy of 70%–90%.

Recently, ground-based images are used more in cloud type classification. Martínez-Chico et al. (2011) classified clouds according to their heights by using radiation data and images from total sky imager (TSI). The result is presented as the frequency of occurrence for each class. Buch et al. (1995) used images from two whole-sky imager (WSC) to produce three-dimensional volume. In the classification process they used the binary decision trees with three groups of features (texture measures, position information, and pixel brightness) to classify each pixel in the cloud scene as either one of the five sky conditions. The accuracy of system is 61%. Calbó and Sabburg (2008) developed a system using images from total sky imager and whole sky imager. They classified cloud types using parallelepiped technique with features that are calculated from texture, Fourier transform, and cloudy pixels. The classification accuracy is 62% when eight sky conditions are considered and increases to 76% when five different sky conditions are considered. Heinle et al. (2010) developed real-time classification cloud types using whole sky images. There are 12 features from spectral features and textural features. The *k*-nearest neighbor classifier is used to classify seven different sky conditions. The accuracy of the classification is as high as 97% when it is based on the Leave-One-Out Cross-Validation (LOOCV) but in the general case with unseen data, the accuracy is 75%–88%. Based on these results, Tzoumanikas, Kazantzidis, Bais, Fotopoulos, and Economou (2013) improved *k*-nearest neighbor classifier by considering multi-color criterion where the accuracy is increased to 78%–95%. Liu, Wang, Xiao, Zhang, and Shao (2013) developed the new feature method called salient local binary pattern based on the previous work of Heinle et al. (2010). Their accuracy classified by the nearest neighborhood using chi-square metric is at 93.65%, the best result so far for images from WSC. Taravat, Del Frate, Cornaro, and Vergari (2014) used pixel values of red, green, and blue bands of the whole-sky images for classified pixels in terms of cloud coverage or others. The overall accuracies of 95.07% using multilayer perceptron (MLP) neural networks. Cheng and Yu (2015) used block-based classification on all-sky images. Each block is extract statistical texture features and local binary pattern for six sky conditions. Then, the features are classified with Bayesian classifier which give the accuracy of 90%.

Moreover, there are several researchers started to use the input images captured from digital cameras. Souza-Echer et al. (2006) showed their new algorithm that classifies each pixel based on a criteria decision process on Illuminant-Hue-Saturate (IHS) space using images from the digital camera. The output yields accuracy of 94% for the classification of only a clear sky. Singh and Glennen (2005) used five different feature extraction methods with the *k*-nearest neighbor and neural network classifiers for identifying five sky conditions. The best of their classification has the accuracy of 64%. Xia et al. (2015) used texture features, color features and shape features with *k*-nearest neighbor for classifying four sky conditions. The average accuracy is 84.82%.

From the summary of literature survey in Table 1, the texture feature is still a popular extraction technique. However, its accuracy has been somewhat limited to around 90%. Although some authors showed more than 90% accuracy, their output classes are limited to two. In this paper, the output will be seven classes of cloud types and the input images will be from digital camera and not satellite or TSI/WSC images. Therefore, the accuracy will be compared among those digital camera images. Moreover, our new approach will incorporate the strength of texture analysis into the new technique of FFT feature extraction that focuses more on the shape of cloud. In the most recent survey, Table 2 shows various uses of FFT techniques, some are incorporated with other methods. Calbó and Sabburg (2008) used features based on Fourier transform to discriminate cloud shapes. They extracted the characteristics of the spectral power image using correlation with clear (CC) and spectral intensity (SI). Daowieng, Wongkittisuksa, Tanthanuch, and Permsirivanich (2010) used FFT and discrete wavelet transform (DWT) for word recognition. More recently, Chen (2014) extracted dual-tree complex wavelet (DTCWT) features from EEG signals and perform the FFT to the DTCWT features subbands. Soltana, Porebski, Vandenbroucke, Ahmad, and Hamad (2014) applied FFT with Local Binary Patterns (LBP) histogram to calculate features from lace images. Stępniowski, Michalska-Domańska, Norek, and Czujko (2014) calculated the radial average of FFT for arrangement analysis of the aluminum nanopores. To the best of our knowledge, there is no research on ground-based cloud type classification that performs feature extraction by projecting the values of logarithmic magnitude of FFT images on the *x*-axis of the frequency domain. Furthermore, the new idea of introducing the *k*-sampling and sorting techniques in the settings of feature vector will be incorporated into our proposed algorithms. These techniques will be explained later.

3. Features

We use a grayscale image which is computed by splitting channels of image as R, G, and B channels for extracting features. There

Table 1
Summary of literature survey on existing methods.

Input images	Year	Proposed method	No. class	Accuracy (%)	References
Satellite	1990	NN with texture features to classify subregions	3	93	Lee et al. (1990)
	1994	PNN to classify each area into one of ten cloud classes	10	79.80	Bankert (1994)
	1994	Feature selection algorithms	10	88	Aha and Bankert (1994)
	1997	Bispectral cloud classification method	8	87.10	Fan et al. (1997)
	2000	Probabilistic self-organizing maps	9	63	Ambroise et al. (2000)
	2004	MSVM classifier	3	90.11	Lee et al. (2004)
	2008	SVD to extract the salient spectral and textural features	3	70–90	Kaur and Ganju (2008)
TSI/WSC	1995	Texture measures, position information, and pixel brightness	5	61	Buch et al. (1995)
	2010	Spectral and textural features with k-NN	7	75–88	Heinle et al. (2010)
	2013	Improved k-NN by multi-color criterion	7	78–95	Tzoumanikas et al. (2013)
	2013	Salient local binary pattern feature	7	93.65	Liu et al. (2013)
	2014	Pixel values with MLP NN	2	95.07	Taravat et al. (2014)
	2015	Block-based classification	6	90	Cheng and Yu (2015)
Digital camera	2005	Five feature extraction methods with k-NN and NN	5	64	Singh and Glennen (2005)
	2006	Classifying each pixel based on a criteria decision process on IHS	2	94	Souza-Echer et al. (2006)
	2015	Texture, color and shape features with k-NN	4	84.82	Xia et al. (2015)

Table 2
Related works on Fast Fourier Transform techniques.

Year	Proposed method	Application	References
2008	Extracting CC and SI from FFT	Cloud classification	Calbó and Sabburg (2008)
2010	DWT with FFT	Word recognition	Daowieng et al. (2010)
2014	DTCWT with FFT	EEG analysis	Chen (2014)
2014	LBP and FFT with k-NN	Analysis of lace images	Soltana et al. (2014)
2014	Radial average of FFT	Aluminum nanopores	Stępniowski (2014)

are three groups of features which are used in the experiments, namely, the texture feature, the moments of two-dimensional, and the features based on Fourier transform.

3.1. Texture feature

Texture feature is used to describe the texture of image region. There are two types of texture features which are based on data sources used in the calculation.

The first type of texture feature is extracted from images directly. These are mean of red channel (ME), standard deviation of blue channel (SD) and difference of mean between each channel (D_{ij}) defined by the equations below (Calbó & Sabburg, 2008; Heinle et al., 2010).

Mean-R

$$ME = \sum_{i=0}^{N-1} x_i p(x_i) \tag{1}$$

Standard Deviation-B

$$SD = \left(\sum_{i=0}^{N-1} (x_i - ME)^2 p(x_i) \right)^{1/2} \tag{2}$$

Difference R–G, R–B, and G–B

$$D_{ij} = ME_i - ME_j \tag{3}$$

The annotation N is a number of pixels in image, x_i is a value of pixel i-th, and $p(x_i)$ is a probability of x_i.

The second type of texture feature is computed from Gray Level Co-occurrence Matrices (GLCM). This is a square matrix where a number of columns equals a number of gray level. Each element in the matrix is the frequency that two pixels occur ($P^{\Delta}(a,b)$) (Heinle et al., 2010). We will use the following Haralick texture features (Haralick, Shanmugam, & Dinstein, 1973).

Contrast-R and Contrast-B

$$CON = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} (a-b)^2 P^{\Delta}(a,b) \tag{4}$$

Homogeneity-B

$$HOM = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} \frac{P^{\Delta}(a,b)}{1+|a-b|} \tag{5}$$

Energy-R

$$EN = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} P^{\Delta}(a,b)^2 \tag{6}$$

Variance-R

$$V = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} (a-\mu)^2 P^{\Delta}(a,b) \tag{7}$$

Inverse Difference Moment-R

$$IDM = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} \frac{P^{\Delta}(a,b)}{1+(a-b)^2} \tag{8}$$

Sum Average-R

$$SA = \sum_{a=1}^{2(G-1)} aP_{x+y}(a) \tag{9}$$

Sum Variance-R

$$SV = \sum_{a=1}^{2(G-1)} (a - SE)^2 P_{x+y}(a) \tag{10}$$

Sum Entropy-R

$$SE = -\sum_{a=1}^{2(G-1)} P_{x+y}(a) \log P_{x+y}(a) \tag{11}$$

Entropy-R

$$EN = -\sum_{a=0}^{G-1} \sum_{b=0}^{G-1} P^{\Delta}(a,b) \log P^{\Delta}(a,b) \tag{12}$$

Difference Variance-R

$$DV = variance\ of\ P_{x+y} \tag{13}$$

Difference Entropy-R

$$DEN = -\sum_{a=0}^{G-1} P_{x-y}(a) \log P_{x-y}(a) \tag{14}$$

Information Measures of Correlation-R

$$C = \frac{EN - HXY}{max\{HX, HY\}} \tag{15}$$

$$HXY = -\sum_{a=0}^{G-1} \sum_{b=0}^{G-1} P^{\Delta}(a,b) \log\{P_x(a)P_y(b)\} \tag{16}$$

The size of GLCM matrix is defined by G where HX and HY are entropies of summing row and column of GLCM matrix, respectively, and HXY is the entropy of multiplying row and column together. $P_{x+y}(a)$ is a sum of GLCM element where row plus column equals a. In contrast, $P_{x-y}(a)$ is a sum of GLCM element where row minus column equals a.

3.2. Moments of two-dimensional functions

Moments of two-dimensional functions are used because of their resistance to any transformation. That is, the value of moments after transformation is not varied too much. The Zernike moments is shown in equation below (Khotanzad & Hong, 1990).

$$A_{nl} = \frac{n+1}{\Pi} \sum_{x} \sum_{y} P_{xy} [V_{nl}(x,y)]^* \tag{17}$$

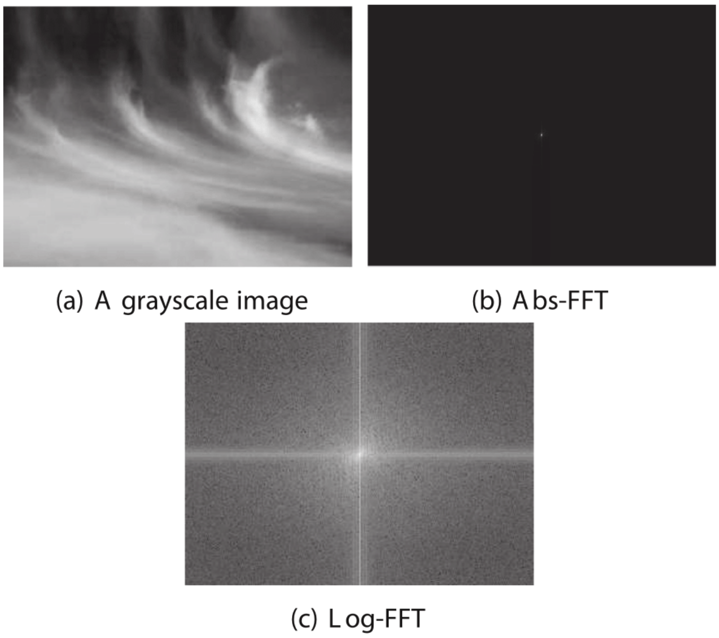
Note that n is an order with repetition l for a digital image, P_{xy} is the current pixel, $V_{nl}(x,y)$ is the Zernike polynomial and $*$ is the complex conjugate (Khotanzad & Hong, 1990).

3.3. Feature based on Fourier transform

We use a grayscale image (R channel) to transform pixels into frequency domain by two dimensional Fast Fourier Transform (2D-FFT) and use FFT shift to move the low frequency pixels into the center of the image (see Fig. 1). There are three types of features based-on Fourier transform that we exploit in the experiment namely, abs-FFT, log-FFT, and k-FFTPX.

Abs-FFT

Abs-FFT is an absolute-magnitude of Fourier transform image. Fig. 1(a) is a grayscale image, we use 2D-FFT and FFT shift to derive abs-FFT (see Fig. 1(b)) by calculating an absolute value of each pixel. The transformed image in Fig. 1(b) is later on to the texture feature extraction process.



(a) A grayscale image

(b) Abs-FFT

(c) Log-FFT

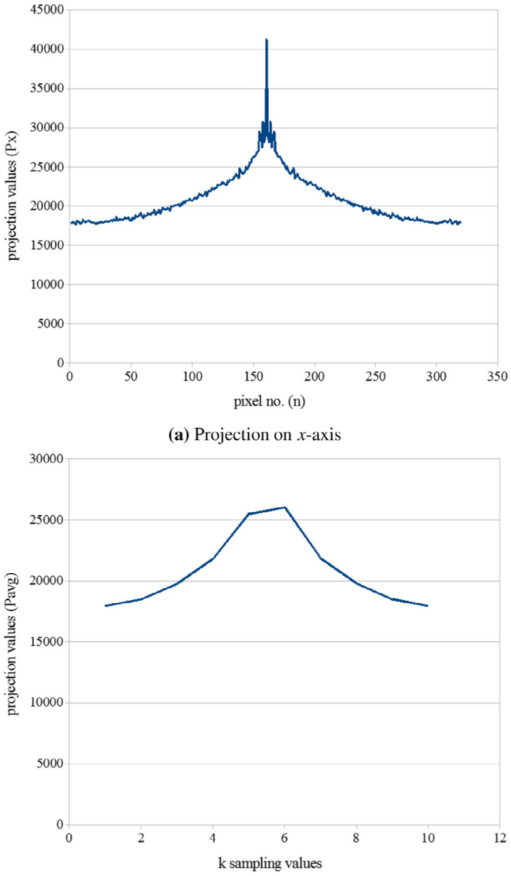
Fig. 1. Fast Fourier Transform of a grayscale image.

Log-FFT

We use a grayscale image in Fig. 1(a) followed by 2D-FFT and FFT shift. Then, we calculate the logarithmic of each pixel in frequency domain which is a logarithmic magnitude of Fourier transform image (log-FFT) as shown in Fig. 1(c). The log-FFT image is then used in a calculation of texture feature extraction.

***k*-FFTPX**

The proposed feature extraction method called *k*-FFTPX is named after the process of projecting the log-FFT values of an image onto the *x*-axis (see Fig. 2(a)) before selecting *k* sampling values of the data as *k* dimensions of a feature vector as depicted in Fig. 2(b). The *k*-FFTPX is based on FFT technique which has more sub methods inside. First, DWT is used to extract the key characteristics by frequency separation of an image. Second, the image is transformed to the frequency domain by FFT which is useful in distinguishing shapes of clouds and reduces the effects of unequal brightness of the image. Third, the characteristics and key features are extracted from the logarithmic magnitude of FFT image and the values are projected on the *x*-axis. Then, these projection values are split into *k* blocks in order to reduce the dimension of the feature vector. Each block is then represented by an average value. After that, values in the feature vectors are sorted in descending order to increase the performance because we found that the sorting technique works well in practice. The algorithm is explained in full in the methodology section.



(a) Projection on *x*-axis

(b) *k*-FFTPX when $k = 10$, $n = 320$

Fig. 2. *k*-FFTPX projection and sampling values.

4. Classifier

A classifier is a function that assigns the input images to a desired output class. We discuss below the most commonly used classifiers, that is artificial neural network.

Artificial neural network (ANN) technique has appeared in many cloud type classification research (Bankert, 1994; Lee et al., 1990; Singh & Glennen, 2005). It is a nonlinear classifier which is suitable for solving complex issues. Back propagation algorithm which applies gradient descent is often used to train the network for the best-fit value (Duda, Hart, & Stork, 2001).

There are three layers of neural network which are composed of input layer, hidden layer, and output layer (Duda et al., 2001) as depicted in Fig. 3.

Each node uses a net activation equation which is a sum of weights and inputs as shown in the equation below.

$$net_j = \sum_{i=0}^{n} x_i w_{ji} \tag{18}$$

In the equation above, the subscript *i* is the index of input layer, where *j* is the index of hidden layer; w_{ji} is the weight from layer *i* to *j* (Duda et al., 2001).

The output of each node uses a nonlinear function known as the activation function. There are many activation functions with difference properties. For example, the equation below shows a hyperbolic tangent function (Duda et al., 2001).

$$f(net) = sgn(net) \equiv \frac{1 - e^{-net}}{1 + e^{-net}} \tag{19}$$

5. Methodology

Algorithm 1. Cloud Classification Algorithm

```
Feed all color images
for each image $i$ do
   Resize to $n \times m$ resolution
   Transform the image $i$ with DWT
   Select coefficient approximated image from DWT
   Split the image into R, G, B channels
   Calculate a vector of texture features $F$
   Sort $F$ in descending order
   Sort a feature vector $H$ in descending order
      /* options for $H$ are 2D-moments, abs-FFT, log-FFT,
      $k$-FFTPX($n, m$), or none */
   Append $H$ to $F$
end for
Build a matrix of feature vectors $V$ for all images
for each image $i$ do
   Record answer into a targeted matrix $T$ by visual
 inspection
end for
Train classifier $C$ using $V$ and $T$
Build a matrix of feature vectors $U$ for all test data
Pass each feature in $U$ to $C$ for classification
Build a confusion matrix
```

This section, we explain two algorithms used in our experiments. We use a set of digital camera images with no more than 36 degree field of view as the input. Algorithm 1 explains our methodology for cloud classification starting from preprocessing the input, extracting dominant features, training the classifier, classifying the instances and returning the confusion matrix as the answer.

In the preprocessing stage of Algorithm 1, we scale down each image to the resolution of $n \times m$ size. By this process we achieve a much smaller computational time. For each image we perform the feature extraction method using discrete wavelet transform (DWT). This transformation gives four images which are coefficient approximated (CA), coefficient horizontal edge (CH), coefficient vertical edge (CV), and coefficient diagonal edge (CD). The size of four images is reduced by half. We select only CA image for a calculation of our feature vector.

In feature extraction stage, we split channels into R, G, B channels as grayscale images before calculating a vector of texture features *F*. The vector *F* is a based feature in our algorithm which is calculated by Eq. (1)–(16). Note that the difference of means in Eq. (3) are computed in three features (R–G, R–B and G–B). Hence, there are a total of 18 texture features implemented in the algorithm. However, to improve the accuracy of the classification we propose to add one of the four following feature extractions, 2D-moments, abs-FFT, log-FFT, or *k*-FFTPX. We call this additional feature vector, *H*. The moments of two-dimensional functions extract eight features using Zernike moments in Eq. (17) of order $n = 0$ to $n = 7$ and repetition $m = 2$. The features based on Fourier transform are referred to abs-FFT, log-FFT, and our proposed feature (also operated on frequency domain) called *k*-FFTPX. We test and evaluate each feature separately and discuss the performance in the experimental section. Note that the method of computing *k*-FFTPX are given in Algorithm 2 which we will explain later.

In the training stage of classifier, we build a matrix of feature vectors *V* for holding the trained features. We label the answer for each feature vector and call it a targeted matrix *T*. Based on these information, we train classifier *C* using *V* and *T*.

In the classification stage, we build a matrix of feature vectors *U* for holding the tested features. We pass each feature in *U* to *C* for the classification process before building confusion matrix for final results.

Algorithm 2. *k*-FFTPX(n, m)

```
for each grayscale image g do
    $\Delta$ = FFT(FFT(g))
    $\Theta$ = FFT_Shift($\Delta$)
    $M$ = log | $\Theta$ |
    /* Compute the projection of $M$ on the $x$-axis */
    $P_x = \sum_{y=1}^{m} M(x, y), x \in \{1, 2, \ldots, n\}, y \in \{1, 2, \ldots, m\}$
    /* Split $P_x$ in $k$ blocks and calculate $P_{avg}(s)$ */
    $P_{avg}(s) = \frac{k}{n} \sum_{j=s(n/k)+1}^{(n/k)(s+1)} P_x(j), \ s = 0, 1, \ldots, k-1$
    Compute a feature vector $H_1 = \bigcup_{s=0}^{k-1} P_{avg}(s)$
    Transform the image $g$ with DWT
    Select coefficient approximated image from DWT
    Calculate $\Delta, \ \Theta, \ M, \ P_x$, and $P_{avg}(s)$ again
    Compute a feature vector $H_2 = \bigcup_{s=0}^{k-1} P_{avg}(s)$
    Output $H = [H_1 H_2]$
end for
```

Our proposed *k*-FFTPX is shown in Algorithm 2. We extract the *k*-FFTPX feature from 2 levels which are DWT level 1 (DWT1) and DWT level 2 (DWT2) using R channel of the CA image *g*. After the transformation of DWT1 (in Algorithm 1), we first transform the image *g* into a frequency domain using 2D-FFT to get Δ as presented in Algorithm 2. After that, Θ is obtained by shifting 2D-FFT of Δ. Then, we calculate the logarithmic magnitude of Θ to find *M*. We project the magnitude *M* on the *x*-axis of the transformed image and called it a vector P_x. We split P_x into *k* blocks and take the average of the magnitude in each block to get *k* projection values. A feature vector H_1 is constructed by joining the *k* projection values of $P_{avg}(s)$ in order from $s = 0$ until $s = k - 1$. The process is repeated with DWT2 to obtain the second feature vector H_2. The output of *k*-FFTPX algorithm is H_1 concatenating with H_2.

The overall feature extraction process is summarized in Fig. 4. According to Fig. 4, two features vectors, F_1 and H_1, are extracted from DWT1. The feature vector F_1 contains 18 texture features derived from Algorithm 1. The feature vector H_1 is a *k* dimensional FFTPX feature. The feature vector H_2 is extracted from DWT2. This feature vector is also a *k* dimensional FFTPX feature. Finally, we concatenate all vectors to get a feature vector *F* which will be the input of our classifier in Algorithm 1. Each *F* represents one image. In the training stage of the classifier, we then build a matrix of feature vectors called *V* which comes from all vectors *F* combined.

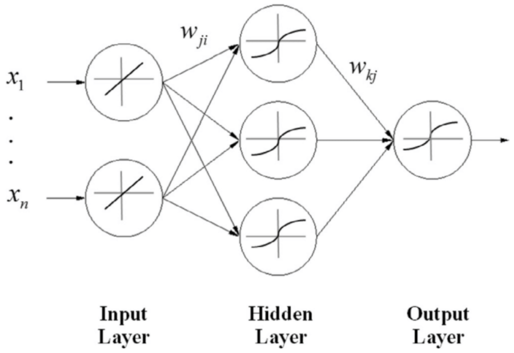


Fig. 3. Multilayer feed forward neural network (Chaowanawatee & Heednacram, 2013).

6. Experimental results

In the experiment, we used over 353 ground-based images from our digital camera for both training and testing purposes. Each image has a resolution of 640 × 480 pixels. The methodology used for all experiments is as per depicted in Algorithm 1 except in the last experiment Algorithm 1 and 2 are both used. LOOCV is used to evaluate the accuracy of the classification. Our classifier is a multilayer feed forward neural network with a single hidden layer for classifying seven sky conditions. The hyperbolic tangent function is used as the activation function in hidden layer and in output layer as shown in Fig. 3. The number of input nodes is equal to the size of each feature vector in each experiment. The number of output nodes is seven. The suitable number of hidden nodes will be determined experimentally. Other parameters of ANN are fixed with a learning rate of 0.01 and a momentum of 0.9. There are five experiments based on different features being tested. In the first experiment, we test the performance of correctly classification with our 18 chosen texture features. Experiment 2–5, the 18 texture features are used in conjunction with 2D-moments, abs-FFT, log-FFT, and *k*-FFTPX, respectively. Furthermore, we will analyze the strengths and weaknesses of each feature used in each experiment.

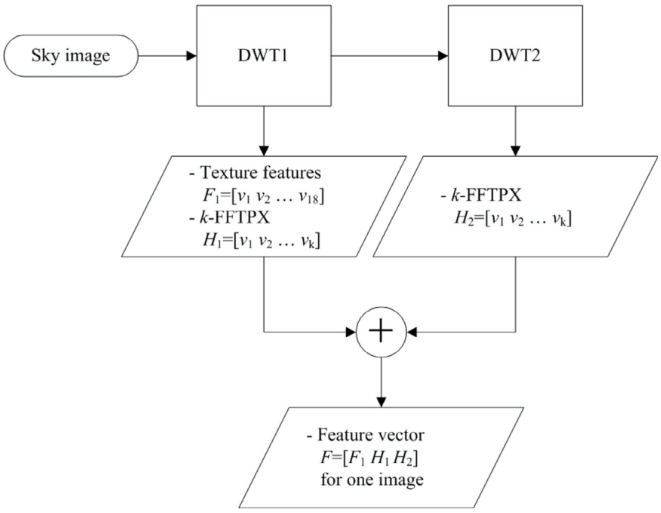


Fig. 4. Feature extraction process for *k*-FFTPX.

6.1. Experiment 1 - Texture features

There are many features that we used in the classification of cloud types. Therefore, we must find a way to choose the suitable features for classification. We select the features by observing a distribution of each feature where the overlapping is minimized between each class.

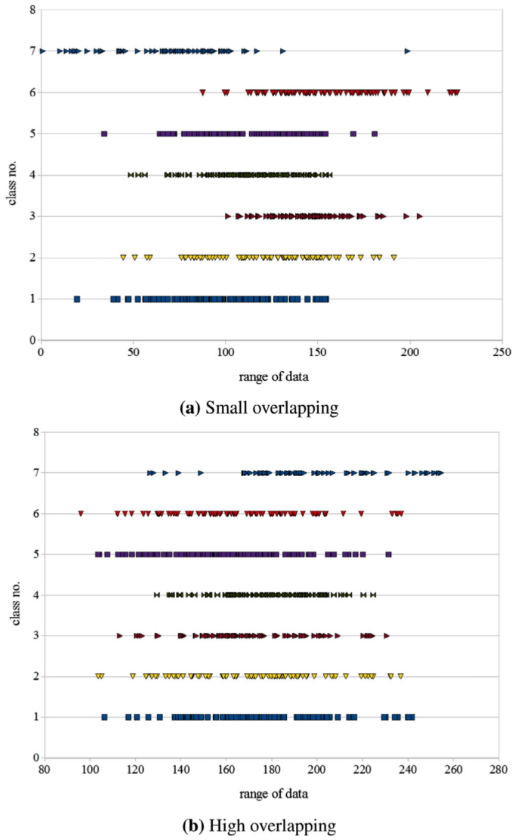
Fig. 5 shows a plot of the mean distribution of the two considered features. The horizontal axis displays a range of distribution of a chosen feature whereas the vertical axis is a class number identifying the cloud types. In Fig. 5(a) we give an example of the suitable feature (mean-R) where the distribution has small overlapping among classes. Fig. 5(b) is an example of the unsuitable feature (mean-B) where the distribution has high overlapping among classes. This latter feature cannot distinguish a class from each other. Although, the mean-R feature from Fig. 5(a) is suitable we still cannot use one feature solely for cloud classification because the distribution cannot distinguish all classes at once. Hence, based on the above technique we checked the distributions of 39 texture features appeared in Heinle et al. (2010), Calbó and Sabburg (2008), and Haralick et al. (1973) and selected the best 18 texture features such that the distributions can separate classes most effectively.

The chosen 18 texture features are implemented in the experiment using Eq. (1) to Eq. (16) where Eq. (3) has three features (R–G, R–B and G–B). The 18 texture features are used in the training of ANN classifier. The ANN model is constructed and fine-tuned by varying the number of hidden nodes. According to Heaton (2008), this value is usually lied between the number of input nodes and output nodes. We conduct separate experiments to heuristically find which number of hidden nodes optimizes the performance of our ANN model. For this experiment (Experiment 1 – using 18 texture features) the suitable number of hidden nodes is set to 15. Note that the number of hidden nodes for the remaining four experiments will be optimized in the same manner. This value will be varied experimentally based on the size of feature vector. From Table 3, the result of classification using 18 texture features is presented in the form of confusion matrix. Each row of the confusion matrix is a true class while each column represents the output class given by our classifier. For example, the element in the first row and third column is a percentage of the accuracy classified as Class 3 while in fact, it is Class 1. The diagonal of the matrix is, therefore, a correct classification for each class. From the table, the classification accuracies of Class 2, 6 and 7 are more than 80%, however, class 4 and 5 have quite low accuracy in comparison with the other classes since the misclassification of Class 4 as Class 1 and Class 5 as Class 6 are still high. On the average, the accuracy of a classification using 18 texture features is 80.76%.

6.2. Experiment 2 - Moments of two-dimensional functions

Moments of two-dimensional functions or 2D-moments are used in the experiment because their ability to tolerate any transformation changes of images. They are calculated from CA images of DWT level 1 using Eq. (17). We split this experiment into 3 sub-experiments. First, we add in the feature vector each n-th order for $n = 0, 1, \ldots, 7$ one by one to the existing 18 texture features. Second, only 0-th order to 7-th order are in the feature vector. And third, the 18 texture features combined with all n-th orders are in the feature vector.

Table 4 shows the result of the first sub-experiment where the 18 texture features are used in combination with each n-th order. The accuracy of classification is peaked at 82.58% when using texture features with the 4-th order of Zernike moments. However, in the second sub-experiment when the texture features were removed, the accuracy is reduced to 63.03%. Therefore, we have learnt that 2D-moments work better in the presence of 18 texture features. In the third sub-experiment when the 18 texture features and all 2D-moments are combined, we obtained more satisfiable result with the accuracy of 82.66%. This is better than using 18 texture features alone (see Experiment 1). The confusion matrix for this sub-experiment is presented in Table 5. In particular, the accuracy of Class 7 is increased to 95.64%. However, the accuracy of Class 5 is reduced to 66.96%. Therefore, we will explore another technique to improve these results in the next experiments.



(a) Small overlapping

(b) High overlapping

Fig. 5. Distribution of chosen features.

Table 3
Confusion matrix classified using 18 texture features.

True class	Classified as						
	1	2	3	4	5	6	7
1	**79.14**	6.30	2.47	7.41	0.86	1.23	2.59
2	6.50	**84.25**	4.25	2.13	1.25	1.63	0
3	1.92	13.46	**79.42**	1.92	1.35	1.92	0
4	15.59	9.15	1.19	**70.00**	1.19	0	2.88
5	8.70	0	0	4.35	**73.91**	13.04	0
6	5.26	0	5.26	0	0	**89.47**	0
7	1.79	0	0	0	2.56	0.77	**94.87**

Table 4
Texture features with one additional Zernike moments of order *n*.

n-th order	0	1	2	3
Accuracy (%)	81.66	82.15	81.73	82.04
n-th order	4	5	6	7
Accuracy (%)	82.58	80.52	78.56	81.78

Table 5
Confusion matrix classified using textures features and 2D-moments.

True class	Classified as						
	1	2	3	4	5	6	7
1	**81.73**	8.27	2.59	4.07	0	0.99	2.35
2	6.25	**85.13**	2.00	4.25	0	1.25	1.25
3	8.08	4.04	**84.62**	0.19	0	3.08	0
4	16.95	4.24	1.19	**73.39**	2.71	0.85	0.68
5	4.35	11.30	1.30	5.22	**66.96**	10.87	0
6	2.63	0	1.58	3.68	0	**92.11**	0
7	1.79	2.56	0	0	0	0	**95.64**

Table 6
Texture features with one additional abs-FFT feature.

Additional feature	Max	Average	Energy	Variance
Accuracy (%)	81.30	85.44	80.74	82.55

6.3. Experiment 3 - Absolute FFT

Absolute FFT or abs-FFT computes texture features using the absolute magnitude of FFT. There are two sub-experiments. First, we select four best texture features which operate on this magnitude of FFT and add each of these features one by one to the existing 18 texture features. Second, the 18 texture features combined with all four abs-FFT features are in the feature vector.

Table 6 yields the result of the first sub-experiment. The best result is at 85.44% when using average-features in conjunction with the 18 texture features and this is superior than the result in Experiment 2. Therefore, it is still worth to use 18 texture features with our test features as the performance has been improved by 2.78%. The second sub-experiment gives 85.06% of the accuracy. This result is close to using the average-feature. However, the average-feature yields slightly higher percentage. The confusion matrix of the average-feature is, therefore, shown in Table 7. The accuracies in most classes are improved from Experiment 2. Especially, Class 4 and 5 we had problems before in Experiment 2, now Class 4 is improved by 1.02% while Class 5 is improved by 9.13%. The accuracy of Class 7 is now climbed up to 97.18%, and the average accuracy of Table 7 is 85.44%.

6.4. Experiment 4 - Logarithmic FFT

Logarithmic FFT or log-FFT closely resembles the abs-FFT but log-FFT uses the logarithmic magnitude of FFT. There are two sub-experiments. First, we use one additional log-FFT feature with 18 texture features. Second, we use all log-FFT features together with 18 textures features in order to compare which sub-experiments give better results.

The result of the first sub-experiment is given in Table 8. The average of log-FFT feature outperforms the other three features. The best accuracy is at 86.11% which is higher than the abs-FFT average-feature in Experiment 3. In the second sub-experiment when we use all log-FFT features with 18 texture features, the overall accuracy is down to 78.47%. We believe that the

Table 7
Confusion matrix classified using texture features and the abs-FFT average-feature.

True class	Classified as						
	1	2	3	4	5	6	7
1	**85.41**	3.70	0.37	5.43	0.62	0.86	1.61
2	7.5	**85.75**	3.63	1.63	0.25	1.25	0
3	2.69	5.00	**90.00**	1.15	0.58	0.58	0
4	14.24	6.78	2.20	**74.41**	1.53	0.17	0.68
5	4.35	0.43	3.04	9.13	**76.09**	6.96	0
6	8.42	0	1.58	0.53	4.21	**84.74**	0.53
7	1.54	0	0	0.51	0.51	0.26	**97.18**

max-feature and energy-feature are the plausible causes since their accuracies are not too high as shown before in Table 8. Therefore, only average-feature is recommended with the 18 texture features. The confusion matrix when using texture features and the log-FFT average-feature is presented in Table 9. The result reveals that most of the classes have the accuracies higher than Experiment 3 with the slight drop of Class 1 and Class 4 performances. However, Class 5–7 give rather excellent results, all are above 90%. The performances of Class 5 and Class 6 are enhanced by 15.21% and 10%, respectively. Moreover, Class 7 accuracy is now 100%.

Until now, the accuracy of Class 4 is still less than 75%. In the next experiment, we will show how our new technique of feature extraction can lead to a significant improvement of Class 4 and the rest of the remaining classes. We expect to have no less than 87% accuracy for every class.

6.5. Experiment 5 – FFT projection on *x* axis

The experiment uses the logarithmic magnitude of FFT presented in Algorithm 2 (*k*-FFTPX) along with the 18 texture features that are obtained from Algorithm 1. Algorithm 2 is called by Algorithm 1 when the step of appending optional features *H* to *F* is reached.

There are two sub-experiments. First, we investigate the suitable value of *k* in the *k*-FFTPX algorithm. Table 10 shows the accuracy of the *k*-FFTPX algorithm when $k = 1$, 5, 10, 15, and 20, respectively. The accuracy increases as the *k* value increases until it reaches a peak at $k = 10$ and the accuracy begins to decrease when $k > 10$.

In the second sub-experiment, we apply $k = 10$ and derive the confusion matrix as shown in Table 11.

Most of the correctly classified instances are now over 87% which are better than the previous four experiments. Moreover, the correctly classified instances of all the classes are higher than 80%. Furthermore, the accuracy of Class 4 is improved by 8.99%. The overall (average) accuracy is at 90.40% which is better than the results of the previous four experiments.

Table 8
Texture features with one additional log-FFT feature.

Additional feature	Max	Average	Energy	Variance
Accuracy (%)	62.46	86.11	79.96	84.99

Table 9
Confusion matrix classified using texture features and the log-FFT average-feature.

True class	Classified as						
	1	2	3	4	5	6	7
1	**82.72**	4.94	1.23	4.94	2.47	1.23	2.47
2	6.25	**87.50**	3.75	1.25	0	1.25	0
3	3.85	5.77	**90.38**	0	0	0	0
4	10.17	8.47	3.39	**71.18**	3.39	0	3.39
5	0	0	0	0	**91.30**	8.70	0
6	0	0	5.26	0	0	**94.74**	0
7	0	0	0	0	0	0	**100**

Table 10
Performance of *k*-FFTPX when *k* is varied.

k values	*k* = 1	*k* = 5	*k* = 10	*k* = 15	*k* = 20
Accuracy (%)	71.84%	89.94%	90.40%	30.03%	32.44%

Table 11
Confusion matrix classified using texture features and 10-FFTPX.

True class	Classified as						
	1	2	3	4	5	6	7
1	**89.26**	1.85	0.12	7.28	0.12	0	13.58
2	2.38	**93.88**	1.23	1.25	0.12	1.25	0
3	2.88	4.42	**92.69**	0	0	0	0
4	11.53	6.78	0.68	**80.17**	0.85	0	0
5	1.74	0	4.35	1.74	**87.83**	4.35	0
6	6.32	0	0	0	0	**93.68**	0
7	0	0	0	2.05	0	0	**97.95**

Table 12
Each feature extraction method and its accuracy.

Feature	Accuracy (%)
Texture + 2D moments	82.66
Texture + Abs-FFT	85.44
Texture + Log-FFT	86.11
Texture + *k*-FFTPX	90.40

6.6. Comparison of each feature extraction method

Table 12 shows the best results from Experiment 2–5 where four different feature extraction techniques are used in conjunction with the 18 texture features. First, when we append 2D-moments to the 18 texture features, we obtain the accuracy of 82.66% which comes from using textures features combined with eight features of Zernike moments.

Later, the abs-FFT average-feature improves the accuracy to 85.44%. Then, the log-FFT average-feature increases the result even further to 86.11%. Finally, the highest accuracy arises from using the 18 texture features with *k*-FFTPX where the confusion matrix shows the overall accuracy of 90.40%. This suggests that the 18 texture features with *k*-FFTPX is the most effective among those feature extraction techniques we presented. In addition, when the magnitude of FFT is plotted in the logarithmic scale, the magnitude differences are far more prominent than the scale of the absolute magnitude (see Fig. 1(b) and (c), for example). Hence, the projection of log-FFT image on the *x*-axis works very well. Note that the suitable *k* value may require tuning and it can be varied from problems to problems. That is why we stated in the algorithm as *k*-FFTPX.

7. Conclusion

Two algorithms were presented. The first algorithm computes a matrix of feature vectors for all images before using this information in the process of cloud classification. The second algorithm is provoked by the first one if the feature extraction technique is set to use *k*-FFTPX. None of the literatures in Table 2 has provided such algorithms before. Besides, our *k*-FFTPX is different from other FFT techniques in many aspects. We project the values of logarithmic magnitude of FFT images on the *x*-axis of the frequency domain and split the projection values into *k* blocks and take the average of the magnitude in each block to get *k* projection values whereas the past research has calculated the characteristics of FFT using CC, SI, or the radial average. We also add a sorting technique to sort values in the feature vector for a better performance. In Expert and Intelligent Systems, there were research using DWT with FFT for word recognition and using DTCWT with FFT for EEG analysis; the merit of this research is that we used DWT with texture features in conjunction with the new *k*-FFTPX features for cloud type classification. By using merely a digital camera available anywhere on the market today, our method is inexpensive. Our next contribution is the increase of cloud types to seven different cloud types and yet a good result, 90.40% accuracy, was obtained. Among digital camera images used in the literatures (see Table 1), there was a result showing more than 90% accuracy; however, their output classes were limited to two cloud types. In this paper, we also delivered more than one extraction technique; in fact, five different combinations of feature extraction techniques were presented and the accuracies are 80.76%, 82.66%, 85.44%, 86.11%, and 90.40%, respectively. These results, therefore, suggest a variety of practical solutions from the simple to the sophisticated functionality that requires no satellite images or expensive tools. Note that our approach will be developed until it can combine with other inexpensive meteorological sensors to report weather conditions and display them on a smart phone.

In addition to the practical advantages above, our algorithm is rather simple to implement while the accuracy of our proposed method was improved from the conventional methods with the capability of distinguishing cloud types of up to seven classes. By using simply a digital camera with a resolution of at least 640 × 480 pixels, our method is less cumbersome and less expensive than those using TSI/WSI imagers. However, our method cannot classify more than one type of clouds that appear on the same image. Also, the *k*-FFTPX algorithm has not yet been tested with the night time images. The algorithm must be used in conjunction with the texture features to achieve the best performance.

Our research has opened up new directions for those interested in the cloud classification. There are many areas where improvements are needed; for example, how to classify many types of clouds that appear on the same image, how to design the algorithm that recognizes clouds in the night time, how to enhance the accuracy of the system to at least 95%. Other feature extraction techniques such as point detection, gradient operators, and fractals may be worth exploring because shapes of clouds are useful information. Other classifiers such as decision tree and SVM perhaps should be looked at and compared. Finally, more cloud types in nature may be considered, for example, earthquake clouds. However, this requires researchers to investigate more closely on the unknown mystery of earthquake clouds and the feasibility study of earthquake prediction.

Acknowledgment

This work is funded by Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Thailand. We also thank INFormation engineering for Andaman Region (INFAR) research group for providing working space and several useful resources.

References

Aha, D. W., & Bankert, R. L. (1994). Feature selection for case-based classification of cloud types: An empirical comparison. In *Proceedings of the 1994 AAAI workshop on case-based reasoning* (pp. 106–112). Menlo Park, California: AAAI Press.

Ambroise, C., Sèze, G., Badran, F., & Thiria, S. (2000). Hierarchical clustering of self-organizing maps for cloud classification. *Neurocomputing, 30*(1), 47–52.

Bankert, R. L. (1994). Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network. *Journal of Applied Meteorology, 33*(8), 909–918.

Buch, K. A., Sun, C. H., & Thorne, L. R. (1995). Cloud classification using whole-sky imager data. In *Proceedings of the fifth atmospheric radiation measurement science team meeting* (pp. 19–23). San Diego, California: Department of Energy.

Calbó, J., & Sabburg, J. (2008). Feature extraction from whole-sky ground-based images for cloud-type recognition. *Journal of Atmospheric and Oceanic Technology, 25*(1), 3–14.

Chaowanawatee, K., & Heednacram, A. (2013). Improved cuckoo search in RBF neural network with gaussian distribution. In *Proceedings of the IASTED international conference* (Vol. 201, pp. 379–386). Alberta: ACTA Press.

Chen, G. (2014). Automatic EEG seizure detection using dual-tree complex wavelet-Fourier features. *Expert Systems with Applications, 41*(5), 2391–2394.

Cheng, H.-Y., & Yu, C.-C. (2015). Block-based cloud classification with statistical features and distribution of local texture features. *Atmospheric Measurement Techniques, 8*(3), 1173–1182.

Daowieng, O., Wongkittisuksa, B., Tanthanuch, S., & Permsirivanich, W. (2010). Command system design of speaker dependent isolated word recognition for stroke patients. In *The second ECTI-conference on application research and development* (pp. 19–24). Thailand: ECTI.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. Upper Saddle River, New Jersey: Prentice Hall.

Fan, Y., Changsheng, L., & Weimin, C. (1997). Man–computer interactive method on cloud classification based on bispectral satellite imagery. *Advances in Atmospheric Sciences, 14*(3), 389–398.

Guha-Sapir, D., Vos, F., Below, R., & Ponserre, S. (2011). Annual disaster statistical review 2010. *The numbers and trends*, Brussels: CRED.

Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE transactions on systems, man and cybernetics*, (pp. 610–621). IEEE.

Heaton, J. (2008). *Introduction to neural networks with java*. Chesterfield, Missouri, United States: Heaton Research, Inc.

Heinle, A., Macke, A., & Srivastav, A. (2010). Automatic cloud classification of whole sky images. *Atmospheric Measurement Techniques Discussions, 3*(1), 269–299.

Heinzmann, U. (1993). Cloud classification on the basis of NOAA-APT data using a fuzzy logic approach. *International Archives of Photogrammetry and Remote Sensing, 29*, 908–913.

Kaur, R., & Ganju, A. (2008). Cloud classification in NOAA AVHRR imageries using spectral and textural features. *Journal of the Indian Society of Remote Sensing, 36*(2), 167–174.

Khotanzad, A., & Hong, Y. H. (1990). Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(5), 489–497.

Lee, Y., Lin, Y., & Wahba, G. (2004). Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association, 99*(465), 67–81.

Lee, J., Weger, R. C., Sengupta, S. K., & Welch, R. M. (1990). A neural network approach to cloud classification. *IEEE Transactions on Geoscience and Remote Sensing, 28*(5), 846–855.

Liu, S., Wang, C., Xiao, B., Zhang, Z., & Shao, Y. (2013). Salient local binary pattern for ground-based cloud classification. *Acta Meteorologica Sinica, 27*(2), 211–220.

Long, C. N., Sabburg, J. M., Calbó, J., & Pagès, D. (2006). Retrieving cloud characteristics from ground-based daytime color all-sky images. *Journal of Atmospheric and Oceanic Technology, 23*(5), 633–652.

Martínez-Chico, M., Batlles, F. J., & Bosch, J. L. (2011). Cloud classification in a mediterranean location using radiation data and sky images. *Energy, 36*(7), 4055–4062.

Parikh, J. (1977). A comparative study of cloud classification techniques. *Remote Sensing of Environment, 6*(2), 67–81.

Shangguan, W., Hao, Y., Lu, Z., & Wu, P. (2007). The research of satellite cloud image recognition base on variational method and texture feature analysis. In *The second IEEE conference on industrial electronics and applications (ICIEA)* (pp. 2816–2820). IEEE.

Singh, M., & Glennen, M. (2005). Automated ground-based cloud recognition. *Pattern Analysis and Applications, 8*(3), 258–271.

Soltana, W. B., Porebski, A., Vandenbroucke, N., Ahmad, A., & Hamad, D. (2014). Texture analysis of lace images using histogram and local binary patterns under rotation variation. In *2014 First international image processing, applications and systems conference (IPAS)* (pp. 1–5). IEEE.

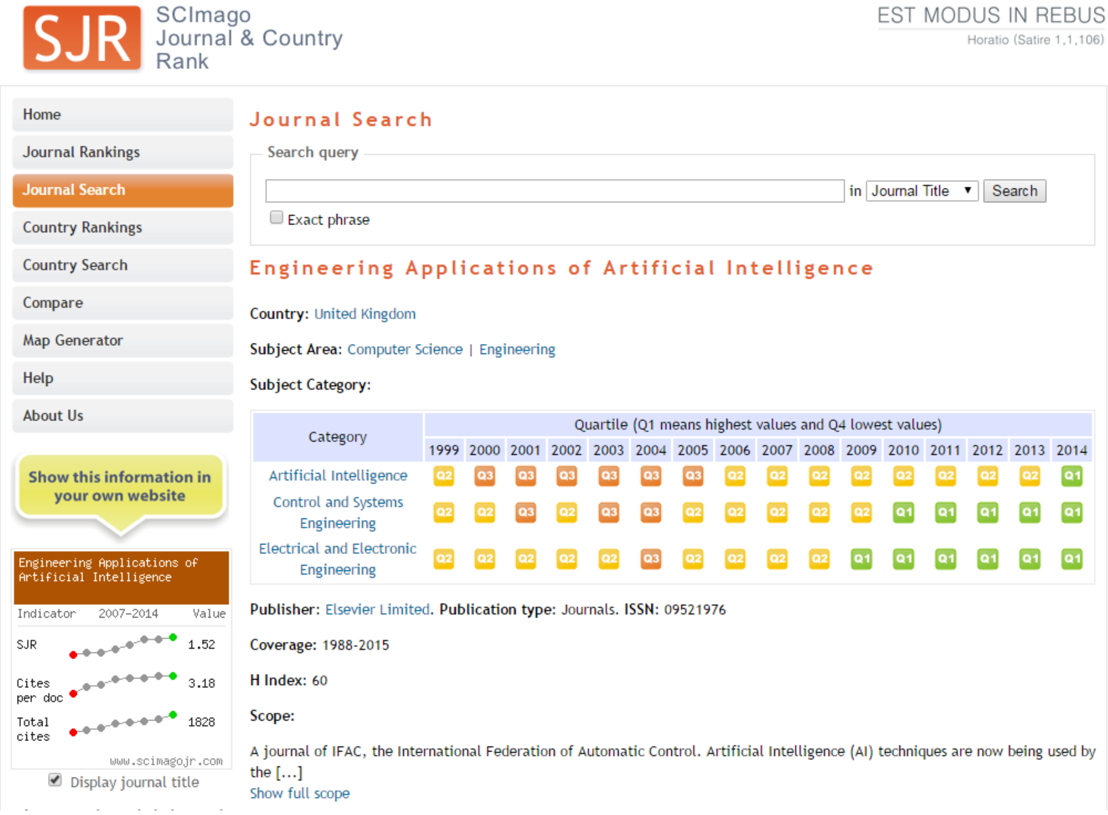
Souza-Echer, M. P., Pereira, E. B., Bins, L. S., & Andrade, M. A. R. (2006). A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera. *Journal of Atmospheric and Oceanic Technology, 23*(3), 437–447.

Stępniowski, W. J., Michalska-Domańska, M., Norek, M., & Czujko, T. (2014). Fast Fourier transform based arrangement analysis of poorly organized alumina nanopores formed via self-organized anodization in chromic acid. *Materials Letters, 117*, 69–73.

Taravat, A., Del Frate, F., Cornaro, C., & Vergari, S. (2014). Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images. *IEEE Geoscience and Remote Sensing Letters, 12*(3), 666–670.

Tzoumanikas, P., Kazantzidis, A., Bais, A., Fotopoulos, S., & Economou, G. (2013). Cloud detection and classification with the use of whole-sky ground-based images. In C. G. Helmis & P. T. Nastos (Eds.), *Advances in meteorology, climatology and atmospheric physics* (pp. 349–354). Berlin: Springer.

Xia, M., Lu, W., Yang, J., Ma, Y., Yao, W., & Zheng, Z. (2015). A hybrid method based on extreme learning machine and k-nearest neighbor for cloud classification of ground-based visible cloud image. *Neurocomputing, 160*, 238–249.



EST MODUS IN REBUS
Horatio (Satire 1,1,106)

Engineering Applications of Artificial Intelligence

Indicator	2007-2014	Value
SJR		1.52
Cites per doc		3.18
Total cites		1828

www.scimagojr.com

☑ Display journal title

Journal Search

Search query

in Journal Title ▾ Search

□ Exact phrase

Engineering Applications of Artificial Intelligence

Country: United Kingdom

Subject Area: Computer Science | Engineering

Subject Category:

Category	Quartile (Q1 means highest values and Q4 lowest values)															
	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Artificial Intelligence	Q2	Q3	Q3	Q3	Q3	Q3	Q3	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q1
Control and Systems Engineering	Q2	Q2	Q3	Q2	Q3	Q3	Q2	Q2	Q2	Q2	Q2	Q1	Q1	Q1	Q1	Q1
Electrical and Electronic Engineering	Q2	Q2	Q2	Q2	Q2	Q3	Q2	Q2	Q2	Q2	Q1	Q1	Q1	Q1	Q1	Q1

Publisher: Elsevier Limited. **Publication type:** Journals. **ISSN:** 09521976

Coverage: 1988-2015

H Index: 60

Scope:

A journal of IFAC, the International Federation of Automatic Control. Artificial Intelligence (AI) techniques are now being used by the [...]

Show full scope

FFT features and hierarchical classification algorithms for cloud images

Thitinan Kliangsuwan*, Apichat Heednacram*

Department of Computer Engineering, Prince of Songkla University, Phuket, Thailand

Abstract

The formation of different cloud types is one of the important components of weather recognition. Cloud-type recognition is useful in preventing losses caused by adverse weather conditions. This paper performs cloud classification on seven cloud types, namely cirriform, high cumuliform, cumulus, stratocumulus, cumulonimbus, stratiform, and clear sky. We present eight algorithms that are used in automatic cloud classification with ground-based images as input. Three new fast Fourier transform (FFT) features are introduced in the feature extraction process, namely the modified k-FFTPX, the half k-FFTPX, and the $h \times k$-FFT. The proposed Cloud Classification Tree Algorithm (CCTA) uses the technique called hierarchical classification which is composed of three levels of tree. The design of our tree helps reduce the number of competitions among the cloud classes. We show that this method provides the highest accuracy at 98.08% through a series of four experiments. The result confirms that the hierarchical classification performs better than a single classification. In addition, the tree can be adapted to classify lesser number of cloud types. Our experiment reveals that the accuracy for classifying two classes, cloud and no-cloud, is high as 100%. Moreover, users have freedom to specify their expected accuracy to gain higher speed in calculation. Finally, the hardware implementation of a low-cost cloud monitoring station is briefly discussed in the paper.

Keywords: Pattern recognition; Cloud classification; Ground-based images.

1. Introduction

Weather conditions have received increasing attention because of climate change and global warming which affects many areas of the world. Studying weather conditions is vital in help preparing for incoming natural disasters and extreme weather changes such as storm, lightning, heavy rain, landslides, and flash flood. In 2013, super typhoon Haiyan in Philippines have caused severe damages, 822 people died and 35 people disappeared (Carine et al., 2015). Weather also relates to occupations in agriculture and fisheries which need to rely on suitable weather conditions. Analyzing weather conditions involves essential elements of celestial phenomena and clouds. The appearance of each cloud type changes upon the weather conditions at the time and in the near future. Therefore, we can recognize weather conditions using cloud classification. Nowadays, with the help of image processing, weather conditions can be monitored without any use of sophisticated instruments, even in the remote or rural areas. Hence, cloud classification is worth exploring.

Automatic cloud classification has been highly appreciated over the last decades (Lee et al., 1990; Heinzmann, 1993; Aha & Bankert, 1994; Bankert, 1994; Buch et al., 1995; Fan et al., 1997; Ambroise et al., 2000; Lee et al., 2004; Singh & Glennen, 2005; Souza-Echer et al., 2006; Shangguan et al., 2007; Calbó & Sabburg, 2008; Kaur & Ganju, 2008; Heinle et al., 2010; Martínez-Chico et al., 2011) because the traditional techniques of using human for manual classification can take a long time and human errors may occur. Unique shape and disorder of clouds can lead to different results for different observers. Moreover, the observation of cloud requires experts and there is limited number of experts in the area. Hence, automatic cloud classification has been developed since 1977 (Parikh, 1977) to serve the purpose of auto-recognition.

Satellite images have been used in the analysis of cloud classification. It can provide information and overview, but is unable to provide local details on a specific area. Furthermore, it is an expensive solution and the access to image data is somewhat limited (Singh & Glennen, 2005; Calbó & Sabburg, 2008). Consequently, ground-based images which obtained by imager devices have been utilized more extensively. Two types of imagers are common (Long et al., 2006), namely the total sky imager (TSI) and the whole sky camera (WSC). Both of them can take an image with wide angle of more than 160° field of view (FOV) that can cover large area of sky. However, these devices are rather expensive and may not be suitable for small research groups. For this reason, a normal digital camera is used. Although, this type of camera provides a fraction of sky image, it has more specific information needed, easier acquisition, and lower cost.

In this paper, we proposed automatic ground-based cloud classification system using a digital camera. Seven sky conditions are considered, namely cirriform clouds, high cumuliform clouds, cumulus clouds, stratocumulus clouds, cumulonimbus clouds, stratiform clouds, and clear sky. Texture and Fourier

*Corresponding author

Email addresses: thitinan.kl@gmail.com (Thitinan Kliangsuwan), apichat@coe.psu.ac.th (Apichat Heednacram)

Table 1
Summary of literature survey on existing methods.

Year	Feature/Method	Classifier	No. class	Accuracy (%)	Reference
	Satellite				
1990	Texture features	Neural network	3	93	(Lee et al., 1990)
1994	Spectral, textural, and physical measures	PNN	10	79.80	(Bankert, 1994)
1994	Feature selection algorithms	IB1	10	88	(Aha & Bankert, 1994)
1997	Bispectral cloud classification method	Look up table	8	87.10	(Fan et al., 1997)
2000	Distribution of pixels	Hierarchical clustering	9	63	(Ambroise et al., 2000)
2004	Radiance profiles	MSVM	3	90.11	(Lee et al., 2004)
2008	SVD	Unclear	3	70-90	(Kaur & Ganju, 2008)
	TSI/WSC				
1995	Texture, position info, and pixel brightness	Binary decision trees	5	61	(Buch et al., 1995)
2008	Fourier transform, texture, and cloudy pixels	Parallelepiped technique	5	76	(Calbó & Sabburg, 2008)
2010	Spectral and textural features	*k*-NN	7	75-88	(Heinle et al., 2010)
2012	Existence of raindrops	*k*-NN	7	87.90	(Kazantzidis et al., 2012)
2013	SLBP feature	Nearest neighbor	7	93.65	(Liu et al., 2013)
2015	Pixel values	MLP neural network	2	95.07	(Taravat et al., 2015)
2015	Block-based classification	Bayesian	6	90	(Cheng & Yu, 2015)
2016	BoMs	SVM	5	90.90	(Li et al., 2016)
	Digital camera				
2005	Five feature extraction methods	Neural network and *k*-NN	5	64	(Singh & Glennen, 2005)
2006	Criteria decision process on IHS	Parallelepiped method	2	94-99	(Souza-Echer et al., 2006)
2014	CCT and automatic block assignment	SVM	6	81.17	(Zhuo et al., 2014)
2014	Texture features	ANN and *k*-NN	7	86.93	(Kliangsuwan & Heednacram, 2014)
2015	Texture, color, and shape features	ELM and k-NN	4	84.82	(Xia et al., 2015)
2015	Texture, color, and SIFT features	ELM	4	86.64	(Wu et al., 2015)
2015	Texture and Fourier transform	ANN	7	90.40	(Kliangsuwan & Heednacram, 2015)

Fourier transform features are extracted from images. Then, artificial neural network (ANN) is used for training and classifying instances based on Cloud Classification Tree Algorithm (CCTA). Our classification tree splits sky conditions into clouded sky and clear sky before further separating clouded sky into three forms of clouds which are cirriform, cumuliform, and stratiform. Then, cumuliform clouds with four different shapes are classified last. Along with the classification tree, we also proposed three new features which are the modified *k*-FFTPX, the half *k*-FFTPX, and the $h \times k$-FFT. Eight algorithms are introduced and their validated results are presented in our four experiments.

We organize the paper as follows. In Section 2, the literature review of cloud type classification are presented. In Section 3, we introduce the input cloud types and describe characteristics of each cloud type. In Section 4, various features used in the experiment section are explained. In Section 5, ANN and its main parameters used in the experiment are described. In Section 6, the methodology and our proposed algorithms for cloud type classification are given. In Section 7, we discuss experimental results. In Section 8, we briefly mention hardware implementation and future work. Finally, in Section 9 we summarize the results and our contributions.

2. Literature review

Several researchers use satellite images as an input for automatic cloud type classification. Lee et al. (1990) presented a classification of sub-regions into one of three cloud types using a neural network with texture features. The overall accuracy of classification is 93%. Heinzmann (1993) used fuzzy logic approach to classify four cloud classes in form of cloud cover percentage. Bankert (1994) suggested a probabilistic neural network (PNN) with spectral, textural, and physical measures for classifying each region into one of ten cloud classes which achieve an accuracy of 79.80%. Aha & Bankert (1994) provided feature selection algorithms for classifying ten cloud classes using IB1. The best accuracy here is 88%. Fan et al. (1997) used a bispectral cloud classification method based on man-computer interactive way to classify six types of clouds, land, and water through looking up table. This method provides an accuracy of 87.10%. Ambroise et al. (2000) used distribution of pixels with hierarchical clustering to classify nine cloud types. The method achieves an accuracy of 63%. Lee et al. (2004) applied multi-category support vector machine (MSVM) to classify radiance profiles as one of three cloud classes. The result of classification is 90.11%. Shangguan et al. (2007) studied texture feature analysis combined with Variational theory to extract texture features. Kaur & Ganju (2008) used singular value decomposition (SVD) to extract the salient spectral and textural features to classify clouds based on their heights. This method has an accuracy of 70-90%. All the papers mentioned above used satellite images.

Recently, ground-based images have been used increasingly in cloud classification. Buch et al. (1995) used images from two whole-sky imagers (WSIs) to create three-dimensional volume. Binary decision trees with three groups of features (texture measures, position information, and pixel brightness) are used to classify each pixel into one of five sky conditions. The accuracy of classification is 61%. Calbó & Sabburg (2008) used images from two ground-based imagers, namely, TSI and WSC. They applied parallelepiped technique with three types of features (texture, Fourier transform, and cloudy pixels). When classifying eight sky conditions, it yields accuracy of 62%

Table 2
Related works on classification tree and decision tree.

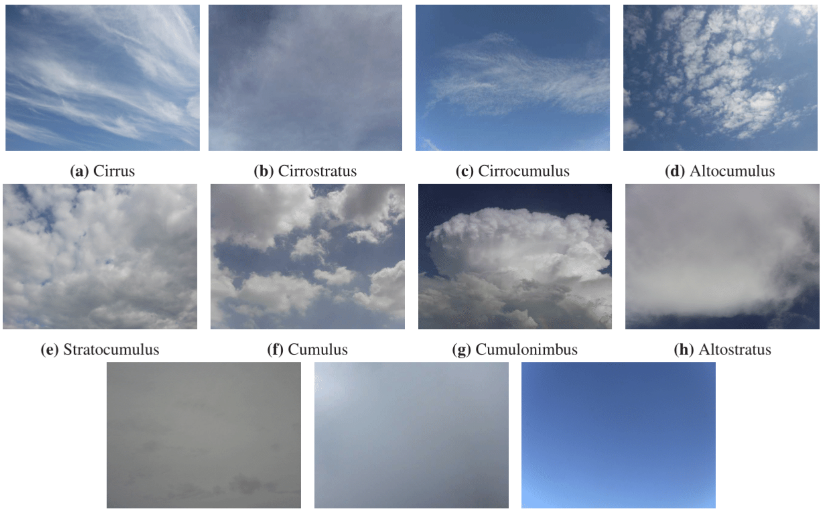
Year	Proposed method	Application	Reference
2000	Hierarchical tree structure	Land cover classification	(Hansen et al., 2000)
2000	Hierarchical SVM structure	Cloud classification	(Azimi-Sadjadi & Zekavat, 2000)
2005	SVM classification tree	Face authentication	(Pang et al., 2005)
2007	Decision tree and FFT	Detection epileptic seizure	(Polat & Güneş, 2007)

and is increased to 76% when classifying five sky conditions. Heinle et al. (2010) used whole sky images as an input for cloud type classification. Their *k*-nearest neighbor (*k*-NN) classifier applied 12 features from spectral features and textural features to classify seven sky conditions. The accuracy classification is as high as 97% but in general case with unseen data, the accuracy is between 75% and 88%. Martínez-Chico et al. (2011) used radiation data and images from TSI to classify clouds according to their heights. The result was shown as the frequency of occurrence for each class. Afterwards, there were two papers based on the previous work of Heinle et al. (2010). First, Kazantzidis et al. (2012) proposed method to detect raindrops for feature extraction. The average of classification is 87.9%. Second, Liu et al. (2013) proposed the new feature method called salient local binary pattern (SLBP). Their accuracy classified by the nearest neighborhood using chi-square metric is 93.65%. Taravat et al. (2015) used pixel values of the whole-sky images to classify pixels into either cloud coverage or others (cloud-free and sun). They used Multilayer perceptron (MLP) neural networks in the classification and the overall accuracy is 95.07%. Cheng & Yu (2015) used block-based classification on all-sky images to classify six sky conditions. In each block, statistical texture features and local binary pattern are extracted. Then, these features are classified using Bayesian classifier which gives the accuracy of 90%. Li et al. (2016) proposed new method named bag of micro-structures (BoMs) for classifying five sky conditions using s support vector machine (SVM) classifier. Their result yields an accuracy of 90.9%.

The input images captured from digital cameras have received high attention from several researchers since around 2005. Singh & Glennen (2005) classified five sky conditions using *k*-NN and neural network classifiers with five different feature extraction methods. The best result has an accuracy of 64%. Souza-Echer et al. (2006) presented the new algorithm based on a criteria decision process on Illuminant-Hue-Saturate (IHS) space to classify each pixel as clear sky, cloudy or undefined using parallelepiped method. This method estimates the percentage of sky cloud coverage, the output yields better than 94% for classifying clear sky and better than 99% for classifying cloudy sky. Most recently, Zhuo et al. (2014) introduced color census transform (CCT) and automatic block assignment method. Then, texture and structure information are extracted in every block and are concatenated as a feature vector. SVM classifier is used to classify six sky conditions and the result is 81.17%. Kliangsuwan & Heednacram (2014) selected 18 texture features to classify seven sky conditions using ANN and *k*-NN. The best classification result is 86.93%. Xia et al. (2015) presented texture features, color features, and shape features which performed with a hybrid method based on *k*-NN and extreme learning machine (ELM). The overall accuracy of classifying four sky conditions is 84.82%. Wu et al. (2015) used ELM classifier to classify four sky conditions. By combining three features, namely, texture features, color features, and SIFT features, it provides accuracy is 86.64%.

From the literature survey in Table 1, the texture features are commonly used for feature extraction. However, the classification accuracy using only texture features is not very high; therefore, they must be combined with other features to achieve higher accuracy. Although some result using a digital camera shows the high accuracy of 94%, the number of cloud classes being considered is only two. Our previous work (Kliangsuwan & Heednacram, 2015) proposed the new technique of FFT feature extraction called *k*-FFTPX. It used 18 texture features for classifying seven cloud classes. Using ANN as a classifier, our last accuracy was 90.04%. In this paper work, we will enhance the accuracy of our algorithm by improving techniques used in the classification process. We will incorporate ANN classifier based on hierarchy classification in a tree-like fashion. Moreover, we will propose three new FFT features.

In the most recent survey, Table 2 shows various uses of hierarchy classification. Hansen et al. (2000) used hierarchical tree structure, a decision tree that applies training data to generate tree structure using pruning method to separate data into two sets. One data set is used to grow the tree and the other is used to prune errors. This method classifies satellite images (AVHRR) into one of twelve classes for land cover classification. Polat & Güneş (2007) used a decision tree classifier and fast Fourier transform (FFT) based on Welch method to classify EEG signals as either patient or normal. Their method is used to detect epileptic seizure. Pang et al. (2005) introduced a binary classification tree algorithm for face membership authentication. Their classification tree is constructed by clustering one data set from the root node into two subsets. Then, clustering procedure is repeated on both child nodes. The procedure terminates when all nodes remain only either membership or nonmembership data. The result from each node of tree is trained by SVM and all SVM classifiers are combined into SVM classification tree. From what mentioned above, the tree structure is generated automatically from training data. However, some researchers prefer to construct tree based on characteristics of each class. Azimi-Sadjadi & Zekavat (2000) used satellite images (GOES 8) IR channel to classify areas into ten classes which composed of six cloud classes and four no-cloud classes. They used SVM classifier together with mean and standard deviation features to classify each block of image. Hierarchical SVM structure was then formed to classify at most two classes



(a) Cirrus **(b)** Cirrostratus **(c)** Cirrocumulus **(d)** Altocumulus

(e) Stratocumulus **(f)** Cumulus **(g)** Cumulonimbus **(h)** Altostratus

(i) Nimbostatus **(j)** Stratus **(k)** Clear sky

Fig. 1. Ten standard cloud types by WMO.

Table 3
Seven sky conditions and their descriptions.

Class No.	Cloud types	Description	Subtypes
1	Cirriform	Hair-like, milky whitening of the blue, thin, white.	Cirrus and cirrostratus
2	High cumuliform	Rounded, patched clouds of small cloudlets, white or gray.	Cirrocumulus and altocumulus
3	Cumulus	Puffy with flat bases, white or light-gray.	Cumulus
4	Stratocumulus	Patches of clouds, white or gray.	Stratocumulus
5	Cumulonimbus	Mushroom-like, dark base, gray.	Cumulonimbus
6	Stratiform	Layer of cloud, uniform, usually overcast, gray or dark.	Altostratus, nimbostratus and stratus
7	Clear sky	No cloud or a very few of clouds, blue.	Clear sky

on each level of hierarchy. Moreover, Parikh (1977) suggested that hierarchical classification leads to better results than using solely a single classification. To the best of our knowledge, apart from SVM classifiers there is no research on hierarchy classification using ANN classifiers. In this paper, we will design tree structure and use ANN to classify at most four classes on each level of hierarchy. Other research on SVM used only binary classes (two classes) per hierarchical level.

3. Cloud types

Cloud classification can have multiple criteria. However, there are two main classifications based on shapes and their heights. In general case, meteorologists classify cloud by shapes into one of the three classes, namely, cirriform, cumuliform, and stratiform. According to the World Meteorological Organization (WMO), clouds are classified into one of the ten genera (classes). Each class has its own species. There are fourteen species that describe shape, size or inside structure. In addition, many varieties can appear in each cloud type which can be used to describe transparency or spreading or arrangement of the cloud. Moreover, for each cloud type it may appear together with accessory clouds and supplementary features. Each cloud type will change shape over time and can be changed to other types (WMO, 1956). Fig. 1 shows the example of ten cloud types and clear sky. These clouds are grouped into seven sky conditions as discussed in Table 3. In this table, some cloud types are combined together because they have similar characteristics. For example, cirrus and cirrostratus are grouped into one class because cirrostratus rarely occurs in nature and it usually appears together with cirrus. Similarly, cirrocumulus is hardly found in nature. Hence, it is grouped with altocumulus and being classified under the name of high cumuliform cloud.

4. Features

Two color models, namely, RGB and HSV (hue, saturation, and value) are used for feature extraction. In this paper, digital images are in RGB color while HSV color codes are computed by equations below (Smith, 1978).

Table 4
The texture features used in the image feature extraction.

Features	Equations	
Mean	$ME = \sum_{i=0}^{N-1} x_i p(x_i)$	(1)
Difference of mean	$D_{ij} = ME_i - ME_j \quad where\, i, j \in \{R, G, B\}\, and\, i \neq j$	(2)
Uniformity	$U = \sum_{i=0}^{N-1} p^2(x_i)$	(3)
Contrast	$CON = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} (a-b)^2 P^{\Delta}(a,b)$	(4)
Homogeneity	$HOM = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} \frac{P^{\Delta}(a,b)}{1+\lvert a-b \rvert}$	(5)
Energy	$EN = \sum_{a=0}^{G-1} \sum_{b=0}^{G-1} [P^{\Delta}(a,b)]^2$	(6)
Entropy	$ENT = -\sum_{a=0}^{G-1} \sum_{b=0}^{G-1} P^{\Delta}(a,b) \log [P^{\Delta}(a,b)]$	(7)
Sum of edge pixels	$SE = \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} e(i,j)$	(8)
Sum of cloud pixels	$SC = \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} c(i,j)$	(9)
Energy of image gradient	$EG = \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} (I_x^2(i,j) + I_y^2(i,j))$	(10)

$$V = max(R', G', B') \tag{11}$$

$$X = min(R', G', B') \tag{12}$$

$$S = \begin{cases} 0 & if\; V = 0 \\ \frac{V-X}{V} & if\; V \neq 0 \end{cases} \tag{13}$$

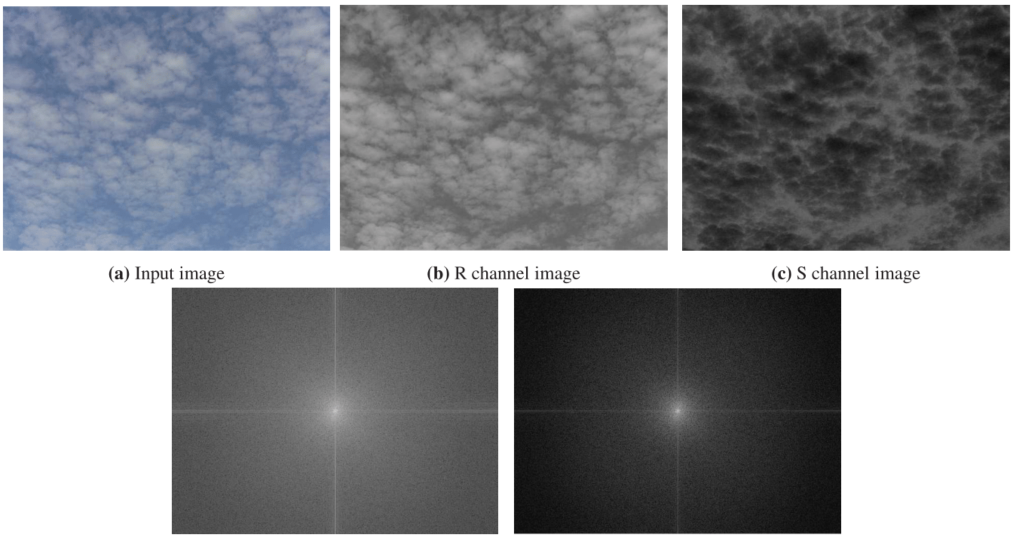
$$H = \begin{cases} 0 & if\; V = X \\ 5 + \frac{V-B'}{V-X} & if\; V = R'\; and\, X = G' \\ 1 - \frac{V-G'}{V-X} & if\; V = R'\; and\, X \neq G' \\ 1 + \frac{V-R'}{V-X} & if\; V = G'\; and\, X = B' \\ 3 - \frac{V-B'}{V-X} & if\; V = G'\; and\, X \neq B' \\ 3 + \frac{V-G'}{V-X} & if\; V = B'\; and\, X = R' \\ 5 - \frac{V-R'}{V-X} & if\; V = B'\; and\, X \neq R' \end{cases} \tag{14}$$

The notation R', G' and B' are R, G, B which are scaled to [0, 1]. The H, S, and V values are all varied in a range of [0, 1]. The H values corresponds to colors varying from red through yellow, green, cyan, blue, magenta, and back to red, so at 0 and 1 they are both red. The S values are varied so that the corresponding hue colors change from unsaturated to fully saturated (no white component) whereas the V values are brightness values in scale of [0,1].

We will use a grayscale image which is computed by splitting channels of image as R, G, B, H, S or V channels in feature extraction. There are two groups of features used in the experimental section which are the texture features and the new features based on Fourier transform. These features are used together because no single feature extraction method is best suited for recognizing all classes (Singh & Glennen, 2005). Each method has its own merits.

4.1. Texture feature

We use texture feature to represent the textual contents of image. There are five types of texture features depending on data sources used in the calculation. The first type is when texture feature is extracted straight from images. These features are mean (ME) of R, G and B, difference of mean (D_{ij}) between each channel, and uniformity (U) of R defined by Eq. 1 to Eq. 3 (Calbó & Sabburg, 2008; Heinle et al., 2010) in Table 4. The notation N in the equation is the number of pixels in image, x_i is a value of pixel i-th, and $p(x_i)$ is a probability of



(a) Input image **(b)** R channel image **(c)** S channel image

(d) FFT image of R channel **(e)** FFT image of S channel

Fig. 2. A comparison of Fourier transform image of R and S channels.

x_i. This type of texture features is used to distinguish between dark clouds and white clouds as well as to separate thin clouds like cirrus from others. However, some clouds have the same color tone such as cumulus and stratocumulus; hence we cannot separate them by these features alone (Heinle et al., 2010). Therefore, other texture features will be combined to solve this problem.

The second type is when texture feature is calculated from Grey Level Co-occurrence Matrices (GLCM) which is a square matrix where the number of columns equal the number of grey levels. An element in the matrix refers to the frequency that two pixels occurred ($P^{\Delta}(a, b)$) (Heinle et al., 2010). We will use four of Haralick texture features (Haralick et al., 1973). The GLCM matrix size is denoted by G and a row and a column of G are represented by a and b, respectively. In Table 4, these type of features are contrast (CON) of B, homogeneity (HOM) of B, energy (EN) of B and entropy (ENT) of R, G, B, and S as depicted by Eq. 4 to Eq. 7.

The third type of texture feature is computed from edge of image which is calculated by canny edge detection (Canny, 1986) on R channel. The number of edge pixels is different for each cloud class; hence it can be used to distinguish clear sky and stratiform clouds from other cloud classes. The ENT (Eq. 7) is calculated again, but on edge of image, this texture feature is called entropy of edge image (EE). The sum of edge pixels (SE) is calculated by Eq. 8. The annotation $e(i, j)$ is a pixel value from the i-th row and the j-th column while A and B are the size of edge image $A \times B$ pixels, width and height, respectively.

The fourth type of texture feature is computed from binary image or threshold image on S channel. Threshold algorithm is used, if pixel value is less than threshold value then the pixel is set to one otherwise it is set to zero. The remaining pixels in the image are cloud pixels and we can use them to separate cloud types by considering the number of cloud pixels for each type. The sum of cloud pixels (SC) using threshold image is calculated by Eq. 9. The annotation $c(i, j)$ is a pixel value from the i-th row and the j-th column.

The fifth type of texture feature is computed from two gradient images on S channel using Eq. 10 called energy of image gradient (EG). The notation I_x and I_y describe image gradients of row and column directions, respectively (Tian et al., 2011). This feature is used to measure sharpness of grayscale image.

4.2. *Feature based on Fourier transform*

From Fig. 2, we take input RGB image (Fig. 2(a)) and transform it to grayscale image by splitting channel into R, G, or B channel. The sample of grayscale image from red channel is shown in Fig. 2(b). The channel splitting transforms HSV image to H, S or V image in grayscale. The S channel image is shown in Fig. 2(c). We transform pixels of the grayscale image into frequency domain by two dimensional Fast Fourier Transform (2D-FFT) and we use FFT shift to move the low frequency pixels into the center of the image (see Fig. 2(d) and Fig. 2(e)). In Fig. 2(b) the white color in cloud areas means that the areas have high red value while Fig. 2(c) shows the complement of the image which presents a purity of pixels. The pixels with the highest purity have the highest values and are represented as white. The non-white color in cloud areas corresponds to a mixture of colors. Furthermore, the S image provides more details or better contrast than the R image. When both images are transformed to FFT images, the FFT image of S channel

(Fig. 2(e)) also has better contrast than the FFT image of R channel (Fig. 2(d)). Therefore, we will use the FFT image of S channel and the verification of results will be presented later in the experimental section.

Algorithm 1 The modified k-FFTPX

```
Input: $k$, a grayscale image $g$
Output: $H$
$\Delta = \text{FFT}(\text{FFT}(g))$
$\Theta = \text{FFT\_Shift}(\Delta)$
$M = \log|\Theta|$
/* Compute the projection of $M$ on the $x$-axis */
$P_x = \sum_{y=1}^{m} M(x,y),\ x \in \{1,2,...,n\},\ y \in \{1,2,...,m\}$
/* Split $P_x$ in $k$ blocks and calculate $P_{avg}(s)$ */
$P_{avg}(s) = \frac{k}{n} \sum_{j=s(n/k)+1}^{(n/k)(s+1)} P_x(j), \quad s = 0,1,...,k-1$
Compute a feature vector $H = \bigcup_{s=0}^{k-1} P_{avg}(s)$
return $H$
```

Algorithm 2 The half k-FFTPX

```
Input: $k$, a grayscale image $g$
Output: $H$
$\Delta = \text{FFT}(\text{FFT}(g))$
$\Theta = \text{FFT\_Shift}(\Delta)$
$M = \log|\Theta|$
/* Compute the half projection of $M$ on the $x$-axis */
$P_x = \sum_{y=1}^{m} M(x,y),\ x \in \{1,2,...,n/2\},\ y \in \{1,2,...,m\}$
/* Split $P_x$ in $k$ blocks and calculate $P_{avg}(s)$ */
$P_{avg}(s) = \frac{2k}{n} \sum_{j=s(n/2k)+1}^{(n/2k)(s+1)} P_x(j), \quad s = 0,1,...,k-1$
Compute a feature vector $H = \bigcup_{s=0}^{k-1} P_{avg}(s)$
return $H$
```

Algorithm 3 The $h \times k$-FFT

```
Input: $h$, $k$, a grayscale image $g$
Output: $H$
$\Delta = \text{FFT}(\text{FFT}(g))$
$\Theta = \text{FFT\_Shift}(\Delta)$
$M = \log|\Theta|$
$r \in I^+,\ t \in I^+$
$d(r,t) = (s \times r/t + 1)mod(r)$
$u(r,t) = (r/t)(s+1)mod(r)$
$y' = \begin{cases} m & if\ u(m,h) = 0 \\ u(m,h) & if\ u(m,h) \neq 0 \end{cases}$
$x' = \begin{cases} n/2 & if\ u(n/2,k) = 0 \\ u(n/2,k) & if\ u(n/2,k) \neq 0 \end{cases}$
/* Split $M$ in $h \times k$ blocks and calculate $P_{sum}(s)$ */
$x \in \{1,2,...,n/2\},\ y \in \{1,2,...,m\}$
$P_{sum}(s) = \sum_{y=d(m,h)}^{y'} \sum_{x=d(n/2,k)}^{x'} M(x,y),\ s = 0,1,...,(h \times k) - 1$
Compute a feature vector $H = \bigcup_{s=0}^{k-1} P_{sum}(s)$
return $H$
```

The shape of clouds cannot be explained by texture features alone. Therefore, we introduced features based on Fourier transform for differentiating the shape of clouds. Moreover, it helps reduce the effect of unequal brightness in cloud images. There are three types of features based on Fourier transform that we exploit in the experiment, namely, the modified k-FFTPX, the half k-FFTPX, and the $h \times k$-FFT.

The modified k-FFTPX

The original k-FFTPX (Kliangsuwan & Heednacram, 2015) used coefficient approximated image from discrete wavelet transform (DWT) to transform the image to frequency domain using FFT. The projection of logarithmic magnitude of Fourier transform image on the x-axis was used. Then, we chose k uniform sampling values of the projection data as k dimensions of a feature vector. All steps are repeated twice and the second feature vector found was concatenated to the first one. To reduce computation time, the modified k-FFTPX is proposed (see Algorithm 3). The DWT is no longer used in the algorithm, a simple grayscale image is used instead. Moreover, it does not repeat the step of finding the second k dimensions of a feature vector. Only the first feature vector is required.

The half k-FFTPX

This feature is similar to the modified k-FFTPX, but half k-FFTPX operates with a half projection of logarithmic magnitude of Fourier transform image on the x-axis to reduce calculation and processing time even further (See Algorithm 4).

The $h \times k$-FFT

The $h \times k$-FFT is used to describe the shape of clouds in greater detail than the above two FFT features because it uses a sampling block technique on the FFT image. After logarithmic magnitude of Fourier transform image is calculated, we take half of the image and split it into h rows and k columns. This becomes the sub-images as depicted in Fig. 3. The sum of pixel values in each sub-image is then calculated. Then, each sum value is concatenated as a feature vector $V = [v1, v2, v3, ..., V_{h \times k}]$ (See Algorithm 5).

Both the half k-FFTPX and the $h \times k$-FFT used the left half of logarithmic magnitude of Fourier transform image for feature extraction. Since Fourier transform is the process of signal transformation into series of sine and cosine which are odd and even function, respectively; therefore, we can use only half image for calculated features.

5. Classifier

A classifier is used to assign a feature vector of input image to a correct output class. Learning algorithm is used to teach classifier using training data before it can be used to classify the image feature vector.

ANN classifier is commonly used for automatic cloud classification (Lee et al., 1990; Bankert, 1994; Singh & Glennen, 2005). It is a nonlinear classifier which can solve complex problems, and gives relatively high accuracy. Three main layers in

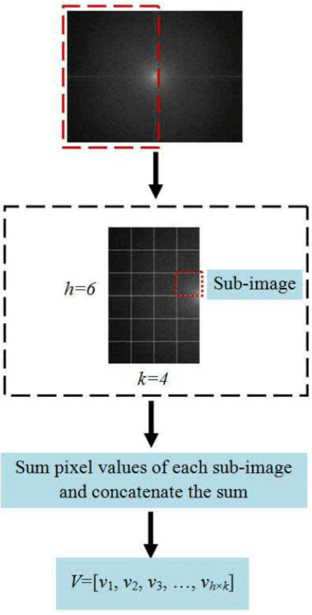

h=6
Sub-image
k=4
Sum pixel values of each sub-image and concatenate the sum
$V=[v_1, v_2, v_3, \ldots, v_{h\times k}]$

Fig. 3. The $h \times k$-FFT diagram.

ANN are input, hidden, and output layers. ANN is trained using back propagation algorithm based on gradient descent technique that maps inputs to outputs by adjusting weights (Duda et al., 2001; Saechai et al., 2013; Heednacram & Werapun, 2014). Input of each node is calculated by net activation equation and its output is calculated using activation function. There are many activation functions, each with difference characteristics. In our experiments, a hyperbolic tangent function is set as the activation function for both hidden and output layers. The activation function of input layer is set to linear so the input remains unchanged. The number of hidden layers is one. In general, a single hidden layer can approximate a solution to any problem, if the number of hidden nodes is enough (Heednacram & Samitalampa, 2014). The number of hidden nodes is set between the number of input nodes and the number of output nodes. We set the number of hidden nodes to 9 for the first level of the tree and 11 for the second and third level. The number of input nodes is equal to the size of a feature vector and the number of output nodes is equal to the number of classes which is seven. Learning rate is set to 0.01. If it is set to a smaller value, learning process will consume more time to construct a correct model. On the other hand, if it is set to a larger value, learning process will use less time to construct a model but the error may be high. Therefore, learning rate should be set to a reasonable small value. Momentum is set to 0.9 for avoiding local minimum in the learning process. Stopping criteria is a criterion to stop learning process. We set stopping criteria when error in the training process reaches 0.001. All weights are initially fixed for our first three experiments. In the final experiment, random weights are used to achieve better classification performance.

6. Methodology

This section, we introduce the algorithm for cloud type classification based on classification tree. Fig. 4 shows a hierarchy classification of clouds in a tree-like fashion. There are three levels of classification. In each level of classification, instances are classified at the internal node (black circle in Fig. 4) using ANN classifier. Different features are used depending on the types of clouds being classified. For seven types of cloud, their abbreviations and their descriptions we recommend readers to revisit Table 3. Level 1, clouded sky and clear sky are separated by the texture features. At this level, we select the most suitable five features introduced in Section 4.1. Level 2 deals with three different forms of clouds which are cirriform, stratiform, and cumuliform. At this level, the modified k-FFTPX, the half k-FFTPX, and the $h \times k$-FFT are tested and the best performed feature is used to distinguish the three forms of clouds. Level 3, cumuliform clouds with different shapes are classified using two texture features and one of the best performed feature from the modified k-FFTPX, the half k-FFTPX or the $h\times k$-FFT. Note that leaves of the tree are the result of classification.

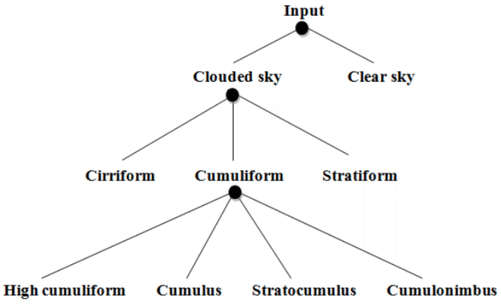


Fig. 4. Cloud classification tree.

There are two main algorithms which are used to construct ANN models and to classify instances based on the classification tree. To construct ANN models, Algorithm 6 is presented. The color ground-based images are scaled to 240 × 320 pixels before transforming to grayscale images for feature extraction. The texture features described in Section 4.1 are selected by Algorithm 7 and ANN model for Level 1 is then constructed. After testing the three FFT features, ANN model for Level 2 is trained by the best performed FFT feature. Note that the optimization of k value for the first two FFT features uses Algorithm 8 while the optimization of h and k values for the $h\times k$-FFT uses Algorithm 9. The three FFT feature vectors are computed using Algorithm 3, 4, and 5, respectively. To construct ANN model for Level 3, the best performed FFT feature is appended to the suitable set of texture features. The set of texture

Algorithm 4 Building ANN models

```
Input: all color images
Output: $C_1$, $C_2$, and $C_3$
Resize all color images to $n \times m$ resolution
/* Create a model to classify clouded and clear sky */
for each image $i$ do
    Calculate a vector of texture features $F_1$
end for
Build a matrix $F_{N1}$ from feature vector $F_1$
for each image $i$ do
    Put answer into a matrix $T_1$ by visual inspection
end for
Use $F_{N1}$ to find suitable texture features $F_s$
Train classifier $C_1$ using $F_s$ and $T_1$
/* Create a model to classify three forms of clouds */
Feed all clouded images
$K_{M1}$ = optimization of $k$ value for modified-$k$-FFTPX
$K_{H1}$ = optimization of $k$ value for half-$k$-FFTPX
$[H_1, K_1]$ = optimization of $h$ and $k$ values for $h \times k$-FFT
for each clouded image $i$ do
    $F_{a1}$ = call modified-$k$-FFTPX function using $K_{M1}$
    $F_{b1}$ = call half-$k$-FFTPX function using $K_{H1}$
    $F_{c1}$ = call $h \times k$-FFT function using $H_1$ and $K_1$
end for
Build a matrix $F_{K1}$ from feature vector $F_{a1}$
Build a matrix $F_{H1}$ from feature vector $F_{b1}$
Build a matrix $F_{HK1}$ from feature vector $F_{c1}$
for each clouded image $i$ do
    Put answer into a matrix $T_2$ by visual inspection
end for
$F_{M1}$ = testFeatures($F_K$, $F_H$, $F_{HK}$)
/* testFeatures return the best FFT feature */
Train classifier $C_2$ using $F_{M1}$ and $T_2$
/* Create a model to classify four cumuliform clouds */
Feed all cumuliform images
$K_{M2}$ = optimization of $k$ value for modified-$k$-FFTPX
$K_{H2}$ = optimization of $k$ value for half-$k$-FFTPX
$[H_2, K_2]$ = optimization of $h$ and $k$ values for $h \times k$-FFT
for each cumuliform image $i$ do
    $F_{a2}$ = call modified-$k$-FFTPX function using $K_{M2}$
    $F_{b2}$ = call half-$k$-FFTPX function using $K_{H2}$
    $F_{c2}$ = call $h \times k$-FFT function using $H_2$ and $K_2$
    Calculate a vector of texture features $F_2$
end for
Build a matrix $F_{K2}$ from feature vector $F_{a2}$
Build a matrix $F_{H3}$ from feature vector $F_{b2}$
Build a matrix $F_{HK2}$ from feature vector $F_{c2}$
Build a matrix $F_T$ from feature vector $F_2$
for each cumuliform image $i$ do
    Put answer into a matrix $T_3$ by visual inspection
end for
$F_{M2}$ = testFeatures($F_{K2}$, $F_{H2}$, $F_{HK2}$)
Use $F_T$ and $F_{M2}$ to find suitable texture features $F_{N2}$
$F_P$ = appendFeatures($F_{N2}$, $F_{M2}$)
Train classifier $C_3$ using $F_P$ and $T_3$
return $C_1$, $C_2$, and $C_3$
```

Algorithm 5 Selection of a suitable minimal set of features

```
Input: $F[1...n]$, $H[1...(h \times k)][1...N]$, Acc
Output: result, resultIdx, score
/* $F$ is a vector of $n$ texture features, $H$ is a matrix of feature
vectors, Acc is a user-defined accuracy */
result = an empty sequence
resultIdx = an empty sequence
score[1...$n$][1...$n$] initial with zero    /* record % accuracy */
temp = $F$, seq[1...$n$], $c$=$n$
for $i$=1; $i \le n$; $i$++ do
    seq[$i$]=$i$
end for
for $i$=1; $i \le n$; $i$++ do
    for $j$=1; $j \le c$; $j$++ do
        if $H$ is empty then
            score[$i$][$j$]=classify(result $\frown$ temp[$j$])
        else
            score[$i$][$j$]=classify(result $\frown$ temp[$j$],$H$)
        end if
    end for
    [maxVal, maxIdx]=max(score[$i$])
    result=result $\frown$ temp[maxIdx]
    resultIdx=resultIdx $\frown$ seq[maxIdx]
    Remove temp[maxIdx]
    Remove seq[maxIdx]
    $c$=$c$-1
    if maxVal $\ge$ Acc then
        return result, resultIdx, score
    end if
end for
return result, resultIdx, score
```

Algorithm 6 Optimization of k value

```
Input: $K$    /* user-defined maximum threshold */
Output: $k$, score
$k$=0, Acc=0
score[1...$K$] initial with zero
for $i$=1; $i \le K$; $i$++ do
    score[$i$]=classify($i$)
    if score[$i$] > Acc then
        Acc = score[$i$]
        $k$=$i$
    end if
end for
return $k$, score
```

features are derived from Algorithm 7 using the 18 texture features and the best FFT feature vectors as the inputs. Finally, to classify instances based on the classification tree, Cloud Classification Tree Algorithm (CCTA) is elaborated in Algorithm 10. The images are scaled and transformed in the same manner as described before in Algorithm 6. The same feature extraction methods which used to train ANN models are used to form feature vectors for classifying instances according to the classification tree in Fig. 4.

Algorithm 7 Optimization of $h \times k$ value

```
Input: $H$, $K$    /* user-defined maximum threshold */
Output: $h$, $k$, score
$h$=0, $k$=0, Acc=0
score[1...$H$][1...$K$] initial with zero
for $i$=1; $i \leq H$; $i$++ do
  for $j$=1; $j \leq K$ ; $j$++ do
    score[$i$][$j$]=classify($i$,$j$)
    if score[$i$][$j$] > Acc then
      Acc = score[$i$][$j$]
      $h$=$i$, $k$=$j$
    end if
  end for
end for
return $h$, $k$, score
```

Algorithm 8 Cloud Classification Tree Algorithm (CCTA)

```
/* Test ANN models using the test data */
Input: all test images, $C_1$, $C_2$, $C_3$
Output: $C_M$
for each test image $i$ do
  Extract a feature vector $F_1$ for model $C_1$
  $R_1$ = Classify($F_1$, $C_1$)
  if $R_1$ is clear sky then
    $CS$ = Class No. 7
  else
    /* Three forms of clouds */
    Extract a feature vector $F_2$ for model $C_2$
    $R_2$ = Classify($F_2$, $C_2$)
    if $R_2$ is cirriform then
      $CS$ = Class No. 1
    else if $R_2$ is stratiform then
      $CS$ = Class No. 6
    else
      /* Cumuliform clouds */
      Extract a feature vector $F_3$ for model $C_3$
      $R_3$ = Classify($F_3$, $C_3$)
      if $R_3$ is high cumuliform then
        $CS$ = Class No. 2
      else if $R_3$ is cumulus then
        $CS$ = Class No. 3
      else if $R_3$ is stratocumulus then
        $CS$ = Class No. 4
      else
        $CS$ = Class No. 5
      end if
    end if
  end if
end for
Build a confusion matrix $C_M$ from $CS$
return $C_M$
```

7. Experimental results

In the following experiments, we used 1,660 ground-based images from digital camera. Some cloud types are rare naturally-occurring types; hence the number of images per class is collected based on the frequency of cloud occurrence in nature. The classification of cloud images is very challenging because these images are taken from different views and come in different sizes but they all have at least 640 × 480 pixels in JPEG format. Therefore, we can scale all the images down to 320×240 pixels. Least-One-Out-Cross-Validation (LOOCV) is used for result evaluation. There are four experiments and the first three experiments match to the three levels of classification tree in Fig. 4. In the first experiment, we classify instances into two classes either cloud or no-cloud (clear sky). In the second experiment, we classify three groups of clouds by considering their forms. In the third experiment, we classify four cloud classes by considering lumpy appearances of cumuliform clouds. Finally, all levels in the classification tree are combined together for the final classification.

7.1. Level 1 - Cloud or no-cloud

The set of n texture features are sent to Algorithm 7. In this experiment, n is set to 18 which refers to the 18 texture features are described in Section 4.1. Algorithm 7 is used to select a suitable minimal set of features.

Table 5 shows the five-iteration results from Algorithm 7 which keep in the score matrix when Acc or expected accuracy value is set to 100%. Each iteration, one suitable feature is selected from the feature set (F) that has a maximum outcome of accuracy. On the next iteration, the previously selected feature will be concatenated with the remaining features for another classification round. The iteration is repeated until the accuracy reaches the expected value. As a result, we selected five texture features from Table 5 and referred to them as a suitable minimal set of features. These features in order of being added to the set (on each iteration) are EG, SE, $ENT(R)$, $D(G-B)$, and $ME(G)$, respectively. Note that on the final iteration, $ME(B)$ can also be chosen in place of $ME(G)$ as they both yielded the same result. The five features are used in the process of building ANN model for Level 1 to classify clouded sky and clear sky in Algorithm 6. After executing Algorithm 10, we obtain the result for classifying cloud and no-cloud. The accuracy is as high as 100%. The advantage of Algorithm 7 is the flexibility of user-defined accuracy (Acc). For example, if Acc is expected at 92.25%, the algorithm will run only two iterations and only two features are used to classify cloud and no-cloud. Our algorithm is designed to have freedom to trade-off accuracy with computation time.

7.2. Level 2 -Three forms of clouds

The aim of this level is to classify three forms of clouds which are cirriform, stratiform, and cumuliform. In this experiment, features based on Fourier transform are used, namely, the modified k-FFTPX, the half k-FFTPX, and the $h \times k$-FFT. However, only one of these features will be selected by Algorithm 6.

Table 5
Accuracy (%) in selection of suitable features for classifying between cloud and no-cloud.

Iteration	Features																	
	ME (R)	ME (G)	ME (B)	D (R-G)	D (R-B)	D (G-B)	U	CON	HOM	EN	ENT (R)	ENT (G)	ENT (B)	ENT (S)	EE	SE	SC	EG
1	50	50	54.94	52.03	82.75	81.81	69.83	60.50	50.23	50	73.47	50	53.76	49.97	85.65	85.44	50	**86.04**
2	50	50	50	50	50	50	49.97	49.37	50	86.75	50	50	50	50	50	**92.25**	92.17	-
3	97.71	97.71	96.93	99.03	98.84	98.65	98.28	97.57	96.93	97.38	**99.19**	97.76	96.55	97	95.99	-	99.06	-
4	96.29	98.39	94.73	99.48	50	**99.97**	92.91	98.37	89.72	95.86	-	50	50	96.41	50	-	50	-
5	99.97	**100**	100	99.77	99.74	-	99.77	99.77	99.77	99.77	-	99.77	99.77	99.77	99.77	-	99.77	-

Table 6
Accuracy of the modified 20-FFTPX when channel is varied.

Channel	R	G	B	H	S	V
Accuracy (%)	74.30	63.23	61.22	79.39	81.25	61.64

Table 6 shows the accuracy of classification using the modified k-FFTPX for k equal to 20. Each color channel is tested in the feature extraction process. The accuracy of S channel is higher than the five other channels. This result confirms that the use of FFT image of S channel has paid off. Hence, we will use the information obtained from this channel for the remaining experiments.

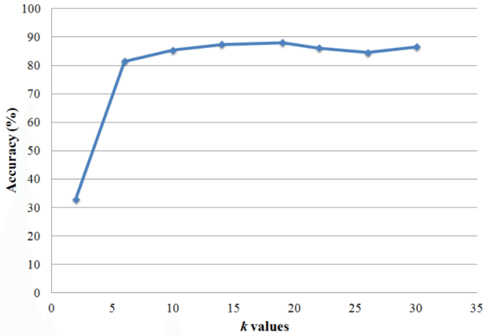


Fig. 5. Accuracy of the modified k-FFTPX on S channel when k is varied.

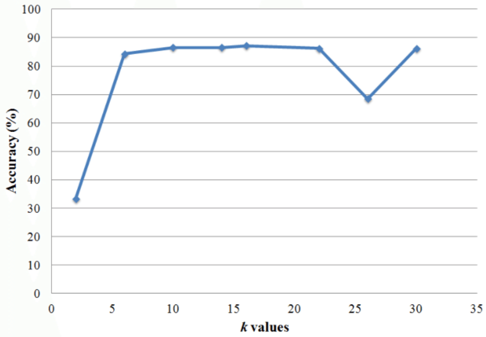


Fig. 6. Accuracy of the half k-FFTPX on S channel when k is varied.

The optimization of k value for the first FFT feature, the modified k-FFTPX, is performed by Algorithm 8. Fig. 5 shows the result of finding the optimized k value when k is varied from 1 to 30. The accuracy of this feature on S channel reaches the peak of 88.06% when k is 19. The k value for the second FFT feature, the half k-FFTPX, is also optimized by Algorithm 8. Fig. 6 shows that the proper setting of k value for this feature is 16 which gives the accuracy of 87.20%. The optimization of h and k values for the third FFT feature, the $h \times k$-FFT, is performed by Algorithm 9. Table 7 shows that when h is equal to 12 and k is equal to 6, it yields the highest accuracy at 95.70%. This accuracy is more than the results of the modified k-FFTPX and the half k-FFTPX. Even with a smaller size (14 dimensions) of feature vector; h is equal to 7 and k is equal to 2, the method still provides 90.38% accuracy (see Table 7). Consequently, the $h \times k$-FFT outperforms the two previous FFT features. At this level of classification tree, the best feature for distinguishing three forms of clouds has been identified as the $h \times k$-FFT. The advantage of this approach is that users are not required to manually specify h and k values. Our algorithm will choose the optimized values for h and k automatically.

Table 8
Confusion matrix classifying three forms of clouds using 12 × 6-FFT.

True class	Classified as		
	1	2	3
1	**90.95**	8.64	0.41
2	1.20	**98.80**	0
3	1.06	1.59	**97.35**

Table 8.shows the confusion matrix obtained from Algorithm 10. Class 1, 2 and 3 are cirriform, cumuliform and stratiform, respectively. The classification results of Class 2 and Class 3 are close to 100%. However, there is some misclassification of Class 1 as Class 2. This is because cirriform often occurs with cirrocumulus which belongs to Class 2. Nevertheless, the accuracy of Class 1 classification is still above 90%. The average accuracy of classification at this level is 95.70%.

7.3. Level 3 - Shapes of cumuliform clouds

Considering lumpy appearances of cumuliform clouds, they can be rounded, puffy, patchy, and mushroom-like. The aim of this level is to classify four classes of cumuliform clouds which are high cumuliform, cumulus, stratocumulus, and cumulonimbus. Three features based on Fourier transform are tested in the same way as in Section 7.2. However, the best performed feature is also appended to a suitable minimal set of texture features computed by Algorithm 7.

Table 7
Accuracy (%) of the $h \times k$-FFT when h and k are varied.

h **value**	***k* value**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	48.27	45.71	76.59	78.83	33.33	84.35	85.6	85.48	33.33	86.56	85.76	64.41
2	34.02	80.72	82.59	80.18	86.27	64.96	88.6	71.09	84.84	42.66	88.84	86.72
3	81.6	88.19	86.69	88.05	78.72	89.51	89.25	91.12	89.66	91.3	88.47	92.12
4	83.02	63.53	65.31	71.95	71.25	77.79	90.44	90.31	94.46	47.16	65.16	78.95
5	84.14	88.81	89.2	55.3	90.38	91.24	89.09	84.65	91.01	88.53	93.38	45.22
6	85.6	80.94	78.06	61.58	91.16	92.56	92.57	93.65	95.16	65.8	46.76	90.68
7	84.99	90.38	91.58	89.05	90.62	91.44	91.54	92.01	91.83	90.96	66.28	95.22
8	66.83	28.87	90.9	94.58	27.92	90.32	92.19	67.82	94.96	70.41	92.6	88.75
9	84.92	86.23	91.21	90.97	93.65	91.09	90.88	90.48	93.47	67.29	69.6	75.45
10	85.3	64.31	90.2	81.9	94.53	87.1	89.69	89.17	68.5	91.45	93.49	85.31
11	38.8	89.61	89.58	70.22	90.28	79.36	83.22	91.38	91.98	93.02	90.63	90.39
12	86.22	89.58	81.32	94.17	66.02	**95.7**	88.46	90.77	89.85	72.49	94.92	89.08

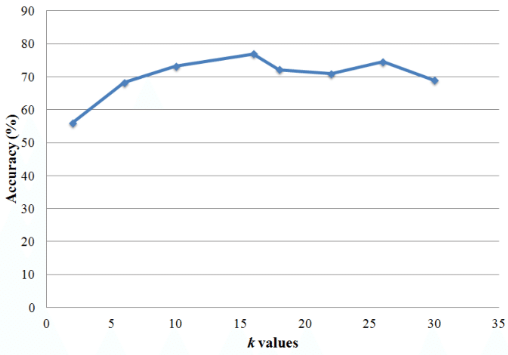


Fig. 7. Accuracy of the modified k-FFTPX on S channel when k is varied.

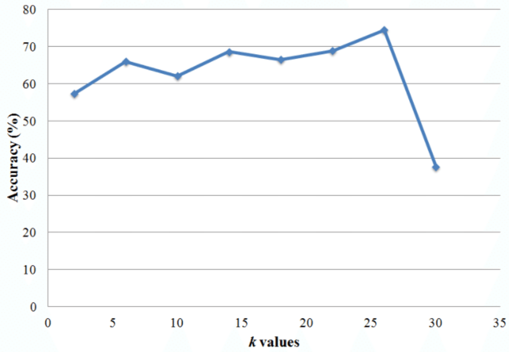


Fig. 8. Accuracy of the half k-FFTPX on S channel when k is varied.

Fig. 7 shows the result of finding the suitable k value for the modified k-FFTPX using Algorithm 8. The best accuracy is at 76.95% for k equal to 16. The result of finding the suitable k value for the half k-FFTPX is shown in Fig. 8. The best accuracy here is slightly decreased to 74.55% for k equal to 26. Table 9 shows the result of finding the suitable h and k values for the hxk-FFT using Algorithm 9. When h is equal to 9 and k is equal to 10, the feature yields the highest accuracy at 89% which is about 12-14% higher than the two previous FFT features. Thus, at this level of classification tree, the 9×10-FFT is served as the best performed FFT feature.

The 18 texture features and the 9×10-FFT are passed to Algorithm 7 for the additional process of selecting a minimal set of texture features. Table 10 shows the two-iteration results where the selected texture features are EE and D(G - B), respectively. By concatenating the two suitable texture features with the 9×10-FFT, we obtain the confusion matrix as shown in Table 11. Class 1, 2, 3 and 4 are high cumuliform, cumulus, stratocumulus, and cumulonimbus, respectively. The classification results of Class 1 and Class 2 are well above 95%. The accuracy of Class 3 classification is almost 100%. However, there are slight misclassifications of Class 4 as Class 2 and Class 4 as Class 3. This is because when cumulus is expanded high into the sky, it appears similar to cumulonimbus. Likewise, when cloudlets are very close together, stratocumulus appears similar to cumulonimbus. Nevertheless, the average accuracy of classification at this level is still high at 96.29%.

Table 11
Confusion matrix classifying four cumuliform clouds.

True class	**Classified as**			
	1	**2**	**3**	**4**
1	**97.32**	1.19	1.49	0
2	2.59	**95.69**	1.29	0.43
3	0	0	**99.36**	0.64
4	0	3.20	4	**92.80**

7.4. All level - A complete classificaion tree

In this experiment, all ANN models from Section 7.1 to 7.3 are combined for hierarchy classification as per the proposed tree in Fig. 4. When Algorithm 10 is executed, the confusion matrix is obtained as shown in Table 12. Class 1 to Class 7 are cirriform, high cumuliform, cumulus, stratocumulus, cumulonimbus, stratiform, and clear sky, respectively. The classification results of seven classes are all above 96%. Five of the seven classes have the accuracy higher than 98%. The accuracy of clear sky classification reaches 100%. The overall accuracy of classification is 98.08%.

Table 9
Accuracy (%) of the $h \times k$-FFT when h and k are varied.

h value	*k* value											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	44.06	57.31	58.5	63.15	30.4	65.98	67.57	69.55	69.36	62.09	56.83	62.1
2	44.83	57.47	62.63	63.89	66.96	67.08	57.7	68.15	69.08	69	70.18	70.07
3	56.19	69.61	74.75	76.63	78.41	78.89	77.04	78.06	82.78	47.93	83.4	78.44
4	48.8	71.95	59.46	75.73	78.64	71.98	71.68	53.41	63.72	25.95	77.11	79.95
5	57.27	73.28	78.61	78.18	25.57	29.04	78.91	63.24	80.99	62.15	80.22	84.33
6	64.69	75.37	79.75	63.74	39.05	68.3	52.87	76.14	83.14	72.48	81.09	87.53
7	68.98	54.44	77.81	75.84	79.96	71.57	66.54	80.23	80.76	80.52	84.97	51.77
8	70.7	75.59	61.01	35.2	78.13	84.15	82.07	82.51	82.28	88.31	84.23	84.09
9	69.14	76.15	77.71	80.55	81.1	81.54	84.19	82.83	53.11	**89**	87.45	85.16
10	71.1	77.91	46.05	77.46	24.81	87	79.41	77.2	85.07	85.24	88.41	84.21
11	60.48	77.5	78.65	79.53	81.85	85.38	81.97	83.27	85.05	88.17	86.61	84.96
12	70.23	53.86	82.71	84.07	78.34	84.04	71.49	84.42	88.89	88.73	85.95	87.45

Table 10
Accuracy (%) in selection of suitable texture features for classifying cumuliform clouds.

Iteration	Features																	
	ME (R)	**ME (G)**	**ME (B)**	**D (R-G)**	**D (R-B)**	**D (G-B)**	**U**	**CON**	**HOM**	**EN**	**ENT (R)**	**ENT (G)**	**ENT (B)**	**ENT (S)**	**EE**	**SE**	**SC**	**EG**
1	57.35	37.45	28.20	54.45	65.98	73.98	82.70	42.73	82.63	81.38	25.19	83.30	86.75	88.88	**93.24**	91.57	28.55	25.73
2	91.79	91.04	90.82	91.42	92.84	**96.29**	92.96	94.16	94.48	92.42	91.52	93.90	91.97	95.53	-	93.75	96.21	92.10

Table 12
Confusion matrix classifying seven cloud types.

True class	Classified as						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**97.49**	0	0	0	2.06	0.45	0
2	0.68	**98.18**	0.63	0.24	0.27	0	0
3	0	0.73	**98.06**	0.86	0.34	0	0
4	0.74	0.23	0.68	**98.10**	0.26	0	0
5	2.32	0	0.16	0.96	**96.40**	0.16	0
6	1.11	0	0	0	0.58	**98.31**	0
7	0	0	0	0	0	0	**100**

8. Hardware implementation and future work

We have recently built a low-cost cloud monitoring station at Prince of Songkla University, Phuket, Thailand. Our hardware is designed to capture whole sky images. The hardware has 5 main components as depicted in Fig. 9. The first component is a glass dome for protecting the hardware from the rain and dust particles. The second component consists of a digital camera with fish-eye lens of 170° FOV. This provides a color images with a resolution of 2048 x 1536 pixels in JPEG format. The third component is a Raspberry Pi 2 Model B processor for controlling camera and cooling device. The fourth component is a humidity and temperature sensor, DHT22. The final component is a cooling device that will be activated only if a temperature inside the glass dome is above 30° C. Fig. 10 shows the hardware installation and the sample image obtained from the station. The images from the station are being recorded every 5 minutes. Once we have sufficient number of images, our future work is to deploy the algorithms presented in this paper for cloud classification at this specific location. Our algorithms will be also further modified to have a small computational time so they can be used on a smart phone.

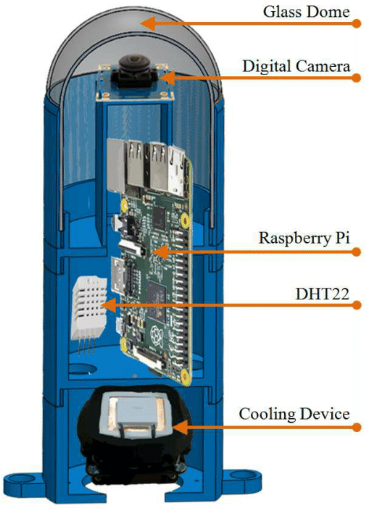
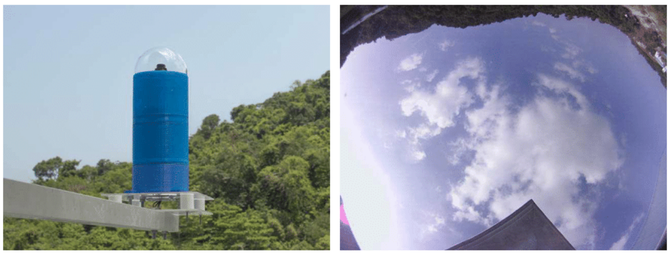


Fig. 9. Simple hardware design for a low-cost cloud monitoring station.

9. Conclusion

To achieve high accuracy of cloud classification, we designed hierarchical classification (tree structure) based on forms and shapes of clouds. The design was composed of three levels of tree, with the aim to classify seven sky conditions. Three new FFT features were proposed to use in the classification process, namely, the modified k-FFTPX, the half k-FFTPX, and the h×k-FFT. Three ANN classifiers were trained separately on each level of the tree. Unlike other previous works, we used ANN to classify up to four classes while others used SVM to clas-



(a) Hardware installation.

(b) A sample image.

Fig. 10. Cloud monitoring station at PSU Phuket.

sify only two classes. The classification result of Level 1 yields the accuracy of 100% using texture features. The accuracy of classification for Level 2 is 95.70% based on a selection of FFT features. The classification result of Level 3 is 96.29% using texture features and the best performed FFT feature. Overall, a complete classification tree provides a high accuracy of 98.08%. This is because the arrangement of classification tree helps reduce the number of competitions among the classes. The number of features used in the algorithms was also selected at minimal sufficient but still gave satisfied results at less computational time. Consider the past papers listed in Table 1, although some of the accuracies are higher than 90%, the number of classified cloud classes are limited to fewer classes. In this paper, the result is better than our two previous works (Kliangsuwan & Heednacram, 2014, 2015). Thus, it confirms that the hierarchical classification performs better than a single classification.

Our method accepts any input images from the ten standard cloud types shown in Fig. 1. Although, some cloud types are rare naturally-occurring types especially for Phuket, the method can be extended further to classify eleven cloud classes (ten standard cloud types and one clear sky) if application requires. In addition, the advantage of classification tree over other methods is that the classification result at each level of the tree is known and available in hierarchical structure; therefore, the classification tree can be easily reconfigured or rearranged to suit user needs. For example, the tree can be used to classify only cloud or no-cloud or even lesser number of cloud types. With this benefit, our method can provide a wider range of applications. However, the drawback of our method is that the errors from the first level of the tree may be carried on to the second level and so on. Hence, in the design of the tree we must place low-misclassified classes before the high-misclassified classes. Furthermore, our method cannot deal with simultaneous appearances of more than one cloud class on the same image. We suggest solving this problem using modified sub-images or considering clouds as objects.

In conclusion, our three main contributions are the new cloud classification method called CCTA, the three novel FFT features, and the presentation of our eight algorithms to readers who prefer the implementation of the method. Our algorithms can also be adapted to suit user requirements. Users can define their own accuracy to gain higher speed in calculation. The future work is to fine-tune our algorithms to fit Phuket sky conditions and to improve our low-cost cloud monitoring station so that it can produce an acceptable computational time for weather forecast on mobile devices.

Acknowledgment

This research is funded by Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University. The authors would like to thank INFormation engineering for Andaman Region (INFAR) research group for offering working space and many useful resources.

References

Aha, D. W., & Bankert, R. L. (1994). Feature selection for case-based classification of cloud types: An empirical comparison. In *Proceedings of the 1994 AAAI workshop on case-based reasoning* (pp. 106–112). Menlo Park, California: AAAI Press.

Ambroise, C., Sèze, G., Badran, F., & Thiria, S. (2000). Hierarchical clustering of self-organizing maps for cloud classification. *Neurocomputing*, *30*(1), 47–52.

Azimi-Sadjadi, M., & Zekavat, S. (2000). Cloud classification using support vector machines. In *Geoscience and Remote Sensing Symposium* (Vol. 2, pp. 669–671). IEEE.

Bankert, R. L. (1994). Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network. *Journal of Applied Meteorology*, *33*(8), 909–918.

Buch, K. A., Sun, C. H., & Thorne, L. R. (1995). Cloud classification using whole-sky imager data. In *Proceedings of the Fifth Atmospheric Radiation Measurement Science Team Meeting* (pp. 19–23). San Diego, California: Department of Energy.

Calbó, J., & Sabburg, J. (2008). Feature extraction from whole-sky ground-based images for cloud-type recognition. *Journal of Atmospheric and Oceanic Technology*, *25*(1), 3–14.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Carine, J. Y., Suppasri, A., Kure, S., Bricker, J. D., Mas, E., Quimpo, M., & Yasuda, M. (2015). Storm surge mapping of typhoon Haiyan and its impact in Tanauan, Leyte, Philippines. *International Journal of Disaster Risk Reduction*, *13*, 207–214.

Cheng, H.-Y., & Yu, C.-C. (2015). Block-based cloud classification with statistical features and distribution of local texture features. *Atmospheric Measurement Techniques*, *8*(3), 1173–1182.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. Upper Saddle River, New Jersey: Prentice Hall.

Fan, Y., Changsheng, L., & Weimin, C. (1997). Man–computer interactive method on cloud classification based on bispectral satellite imagery. *Advances in Atmospheric Sciences*, *14*(3), 389–398.

Hansen, M., DeFries, R., Townshend, J. R., & Sohlberg, R. (2000). Global land cover classification at 1 km spatial resolution using a classification tree approach. *International journal of remote sensing*, *21*(6-7), 1331–1364.

Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, *3*(6), 610–621.

Heednacram, A., & Samitalampa, T. (2014). Suspended sediment forecast of Khlong Bang Yai, Phuket. *International Journal of Engineering and Technology*, *6*(4), 338–345.

Heednacram, A., & Werapun, W. (2014). Java predictors for water level forecast based on daily gage height. *Advanced Materials Research*, *931*, 839–843.

Heinle, A., Macke, A., & Srivastav, A. (2010). Automatic cloud classification of whole sky images. *Atmospheric Measurement Techniques Discussions*, *3*(1), 269–299.

Heinzmann, U. (1993). Cloud classification on the basis of NOAA-APT data using a fuzzy logic approach. *International Archives of Photogrammetry and Remote Sensing*, *29*, 908–913.

Kaur, R., & Ganju, A. (2008). Cloud classification in NOAA AVHRR imageries using spectral and textural features. *Journal of the Indian Society of Remote Sensing*, *36*(2), 167–174.

Kazantzidis, A., Tzoumanikas, P., Bais, A., Fotopoulos, S., & Economou, G. (2012). Cloud detection and classification with the use of whole-sky ground-based images. *Atmospheric Research*, *113*, 80–88.

Kliangsuwan, T., & Heednacram, A. (2014). Classifiers for ground-based cloud images using texture features. *Advanced Materials Research*, *931*, 1392–1396.

Kliangsuwan, T., & Heednacram, A. (2015). Feature extraction techniques for ground-based cloud type classification. *Expert Systems with Applications*, *42*(21), 8294–8303.

Lee, J., Weger, R. C., Sengupta, S. K., & Welch, R. M. (1990). A neural network approach to cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*, *28*(5), 846–855.

Lee, Y., Lin, Y., & Wahba, G. (2004). Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, *99*(465), 67–81.

Li, Q., Zhang, Z., Lu, W., Yang, J., Ma, Y., & Yao, W. (2016). From pixels to patches: a cloud classification method based on a bag of micro-structures. *Atmospheric Measurement Techniques*, *9*(2), 753–764.

Liu, S., Wang, C., Xiao, B., Zhang, Z., & Shao, Y. (2013). Salient local binary pattern for ground-based cloud classification. *Acta Meteorologica Sinica*, *27*(2), 211–220.

Long, C. N., Sabburg, J. M., Calbó, J., & Pagés, D. (2006). Retrieving cloud characteristics from ground-based daytime color all-sky images. *Journal of Atmospheric and Oceanic Technology*, *23*(5), 633–652.

Martínez-Chico, M., Batlles, F. J., & Bosch, J. L. (2011). Cloud classification in a mediterranean location using radiation data and sky images. *Energy*, *36*(7), 4055–4062.

Pang, S., Kim, D., & Bang, S. Y. (2005). Face membership authentication using SVM classification tree generated by membership-based LLE data partition. *IEEE Transactions on Neural Networks*, *16*(2), 436–446.

Parikh, J. (1977). A comparative study of cloud classification techniques. *Remote Sensing of Environment*, *6*(2), 67–81.

Polat, K., & Güneş, S. (2007). Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform. *Applied Mathematics and Computation*, *187*(2), 1017–1026.

Saechai, S., Kusalanggoorawat, P., Kongprawechnon, W., & Sahamitmongkol, R. (2013). Test system for defect detection in cementitious material with artificial neural network. *Songkhanakarin International Journal of Science and Technology (SJST)*, *35*(2), 217–225.

Shangguan, W., Hao, Y., Lu, Z., & Wu, P. (2007). The research of satellite cloud image recognition base on variational method and texture feature analysis. In *the Second IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 2816–2820). IEEE.

Singh, M., & Glennen, M. (2005). Automated ground-based cloud recognition. *Pattern analysis and applications*, *8*(3), 258–271.

Smith, A. R. (1978). Color gamut transform pairs. *ACM Siggraph Computer Graphics*, *12*(3), 12–19.

Souza-Echer, M. P., Pereira, E. B., Bins, L. S., & Andrade, M. A. R. (2006). A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera. *Journal of Atmospheric and Oceanic Technology*, *23*(3), 437–447.

Taravat, A., Del Frate, F., Cornaro, C., & Vergari, S. (2015). Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images. *IEEE Geoscience and Remote Sensing Letters*, *12*(3), 666–670.

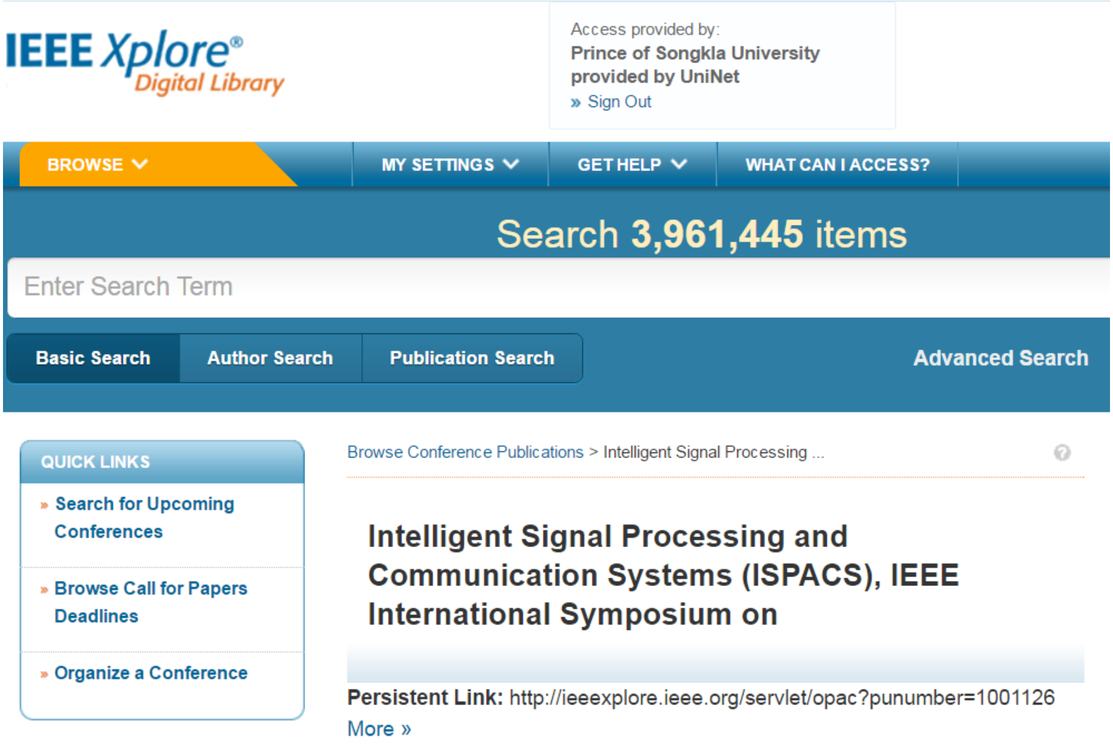
Tian, J., Chen, L., Ma, L., & Yu, W. (2011). Multi-focus image fusion using a bilateral gradient-based sharpness criterion. *Optics communications*, *284*(1), 80–87.

WMO (1956). International cloud atlas. Geneva, Switzerland: World Meteorological Organization.

Wu, Z., Xu, X., Xia, M., Ma, M., & Li, L. (2015). Ground-based vision cloud image classification based on extreme learning machine. *Open Cybernetics and Systemics Journal*, *9*, 2877–2885.

Xia, M., Lu, W., Yang, J., Ma, Y., Yao, W., & Zheng, Z. (2015). A hybrid method based on extreme learning machine and *k*-nearest neighbor for cloud classification of ground-based visible cloud image. *Neurocomputing*, *160*, 238–249.

Zhuo, W., Cao, Z., & Xiao, Y. (2014). Cloud classification of ground-based images using texture–structure features. *Journal of Atmospheric and Oceanic Technology*, *31*(1), 79–92.



Access provided by:
Prince of Songkla University provided by UniNet
» Sign Out

Browse Conference Publications > Intelligent Signal Processing ...

Intelligent Signal Processing and Communication Systems (ISPACS), IEEE International Symposium on

Persistent Link: http://ieeexplore.ieee.org/servlet/opac?punumber=1001126

More »

A Low-Cost Local Cloud Monitoring System

Thitinan Kliangsuwan
Department of Computer Engineering
Prince of Songkla University
Phuket, Thailand
Email: thitinan.kl@gmail.com

Apichat Heednacram
Department of Computer Engineering
Prince of Songkla University
Phuket, Thailand
Email: apichat@coe.psu.ac.th

***Abstract*—In this paper, a local cloud monitoring station equipped with a fisheye lens camera is proposed and installed at Prince of Songkla University, Phuket Campus. The station retrieves cloud images in conjunction with several meteorological sensors. The methodology and cloud classifier algorithm is proposed. The performance test using images captured from our cloud monitoring station shows that the cloud classification accuracy in practice is as high as 98.58%. The installed cloud monitoring system can report live cloud conditions and display them on a mobile application. Our complete system is inexpensive and suitable for local use of weather monitoring and alerting.**

***Index Terms*—Image processing, cloud classification, ground-based images.**

I. Introduction

Different cloud characteristics can distinguish different weather conditions. The study of cloud types is useful in weather prediction and the prevention of natural disasters. Automatic cloud classification has been introduced since 1977 [1]. In early stage, satellite images have been used as an input for automatic cloud classification. However, they are not suitable to be used for local cloud classification due to their lack of local details of clouds on a specific area [2]–[4]. Besides, this solution is expensive and the access to satellite images is sometimes restricted.

Recently, ground-based images have been increasingly used for cloud classification because they can provide more specific information and have lower cost than satellite images. To cover large area in the sky, various ground-based imagers were developed for capturing sky image with wide angle such as total sky imager (TSI) and the whole sky camera (WSC) [5]. Calbó and Sabburg [2] used input images from TSI and WSC. In their paper, texture, Fourier transform, and cloudy pixels features are extracted and parallelepiped technique is used to classify eight sky conditions which yields accuracy of 62%. It is increased to 76% when classifying only five sky conditions. Heinle et al. [3] used k-nearest neighbor (k-NN) classifier with 12 features from spectral features and textural features to classify seven sky conditions. The accuracy of classification is between 75% and 88%. Based on previous work of Heinle et al., Liu et al. [4] proposed the new feature extraction technique called salient local binary pattern (SLBP). The classification result using the nearest neighborhood is 93.65%. Cheng and Yu [6] applied block-based classification on all-sky images. In each block, statistical texture features and local binary pattern are extracted and used to classify six sky conditions using Bayesian classifier. The result yields the accuracy of 90%. Li et al. [7] proposed the new method called bag of micro-structures (BoMs). Five sky conditions are classified using support vector machine (SVM) classifier. The result gives an accuracy of 90.9%. All research mentioned above used ground-based imagers which are still rather expensive and not appropriate for our research group that has limited budget and has no imagers such as TSI or WSC.

In this paper, we use ground-based images from our low-cost cloud station to develop an automatic cloud classification system. Seven cloud types are considered, namely, cirriform clouds, high cumuliform clouds, stratocumulus clouds, cumulus clouds, cumulonimbus clouds, stratiform clouds, and clear sky. Fourier transform and texture features are used in the feature extraction process. Moreover, meteorological data from our weather station are combined together with a feature vector to improve the result. Artificial neural network classifier is used in the classification process. Finally, a low-cost cloud monitoring system is constructed and monitored via mobile application. We organize the paper as follows. First, we suggest the low-cost cloud monitoring station. Second, the system overview of our system is described. Third, we introduce features and classifier. Fourth, we explain our cloud classification algorithm. Fifth, the experimental results are presented and discussed. Sixth, we present our mobile application. Lastly, we conclude our contribution and suggest the future work.

II. Low-Cost Cloud Monitoring Station

Two stations from our low-cost cloud monitoring system are described. These are cloud station and weather station.

A. Cloud Station

Our cloud station is implemented according to the design shown in Fig. 1. It is composed of four compartments. Each compartment is equipped with a different device. To withstand sun and rain, a plastic material is used to construct the body part using a 3D printer. The first compartment is used for mounting a cooling device (an exhaustive fan) The cooling device will run when a temperature inside is more than 30°C. Inside of this component, there is a large ventilation hole in the middle. In the bottom, there are four small outer holes

978-1-5090-0629-8/16/$31.00 ©2016 IEEE

for mounting with the holding plate while the top of this compartment has a twist connector for connecting with a control device compartment. The second compartment is used for mounting a control device (Raspberry Pi 2 Model B). Inside of this component, there are a Raspberry Pi board to capture images and control cooling device and a DHT22 which is humidity and temperature sensor. There are two holes at the bottom for air passage. Two twist connectors are at the top and bottom for connecting with the previous and the next compartment. The third compartment is used for mounting a digital camera. The digital camera can capture color image 170° field of view (FOV) with a resolution of 2048×1536 pixels in JPEG format. There is a square hole for air passage coming from the first compartment and there is an extended column to hold a digital camera in the center. A twist connector is on the side and will be joined with a glass dome. The fourth compartment is used for mounting a glass dome which protects camera from the rain and other particles. This compartment has a glass dome on the top. At the bottom, there is a twist connector to connect a glass dome compartment with a digital camera compartment. Finally, all four compartments are connected as the cloud station (see Fig. 3)

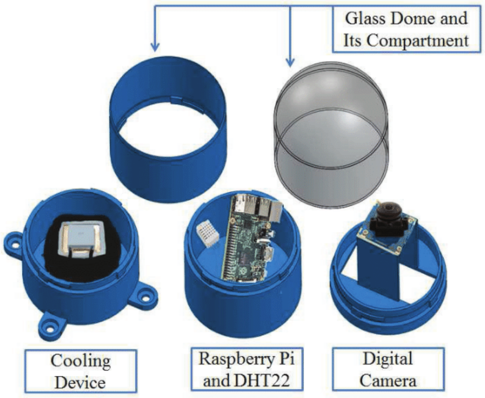


Fig. 1: Hardware architecture.

B. Weather Station

Fig. 2 shows the Oregon scientific professional WMR200A weather station which is composed of several meteorological sensors such as a temperature sensor, a humidity sensor, a wind sensor, a rain gauge, and a solar panel. A device can capture over ten weather measurements such as indoor / outdoor temperature and humidity, wind speed and direction, wind chill, dew point, heat index, barometric pressure and rainfall data. All of these data are sent to the central data logger.

The cost of our cloud station is around 142 USD while the cost of WMR200A weather station is around 350 USD. In total, our estimated budget for building a complete system is 492 USD which is cheaper than using TSI-880 (around 30,000-35,000 USD) and WSC (about 2,500 USD) [8].

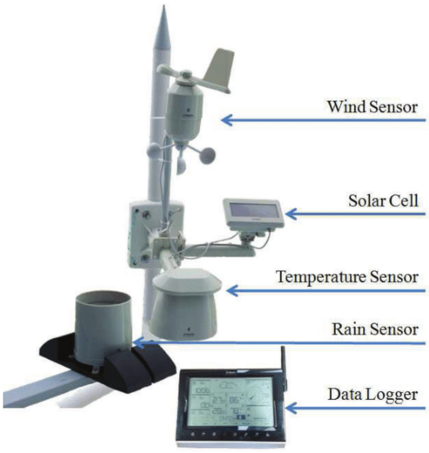


Fig. 2: Weather station.

III. System Overview

Fig. 3 shows an overview of a local cloud monitoring system. Two data sources are used to provide the information for our database. The first source is the whole sky images captured from the cloud station every 5 minutes. The second source comes from meteorological sensors such as a temperature and humidity sensor, a wind sensor, a rain gauge and so on. These meteorological data are sent every 5 minutes to the data logger before forwarding to the server and kept in the database. Mobile application is developed to retrieve the images and the meteorological data for real-time classification purposes. Mobile users can also view live cloud images and live meteorological data from our station.

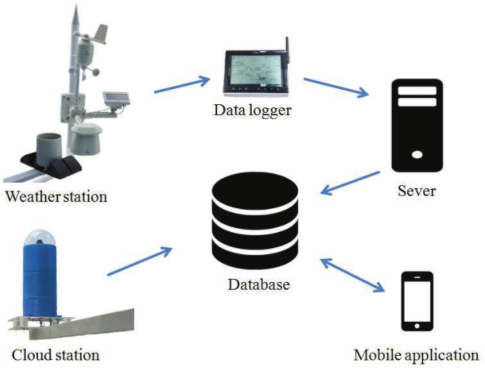


Fig. 3: Overview of a local cloud monitoring system.

IV. Feature Extraction and Classifier

Two feature extraction techniques are described, namely, Fourier transform feature and texture features. These features are extracted from grayscale image by splitting channels of the image into Red (R), Green (G), or Blue (B) channel. Moreover, we also split channels of Hue-Saturation-Value (HSV) color model into H, S, or V channel for grayscale image.

A. Texture Feature

Two types of texture features are used for feature extraction. The first type is extracted directly from image. These features are mean of R channel, standard deviation of B channel and difference of mean among the three channels [2], [3]. The second type is extracted from Grey Level Co-occurrence Matrix [3]. We use 11 Haralick texture features excluding f_3, f_{13}, and f_{14} [9]. The 11 features are computed only on R channel except that f_2 and f_5 are computed both on R and B channels. There are 5 features from the first type and 13 features from the second type, therefore, there are 18 texture features in total. These features were used in our previous works [10], [11].

B. Fourier Transform Feature

This feature is powerful in recognizing different shapes of clouds which cannot all recognize by texture features. Moreover, the feature can also reduce the effect of different luminance in cloud images. The $h \times k$-FFT is firstly introduced in our previous work [12]. It uses S channel obtained from splitting channel of HSV as grayscale image to transform the pixels into frequency domain using two dimensional Fast Fourier Transform (2D-FFT). Then, we use FFT shift to move the low frequency pixels into the center of image and calculate logarithmic magnitude of Fourier transform image. We use only half of the image and split it into h rows and k columns which becomes $h \times k$ sub-images. After that, we sum all pixels in each sub-image and concatenate these sum values as a feature vector.

C. Artificial Neural Network

Artificial neural network (ANN) classifier is often used to handle complex problems. Back propagation algorithm based on gradient descent technique is used to train ANN. The network is composed of three main layers which are input layer, hidden layer, and output layer [13]. Each layer is composed of one or more nodes. The output of each node is calculated using activation function. We use a hyperbolic tangent function as the activation function for hidden layer and output layer. The activation function of input layer is set to a linear function. The number of hidden layers is set to one because a single hidden layer can estimate a result of any problems, if the number of hidden nodes is sufficient [14]. We set the number of hidden nodes to 11. Learning rate is set to 0.01 and momentum is set to 0.9. Stopping criteria is set to 0.001, therefore, when the error in the training process reduces to 0.001, the process will stop.

V. Methodology

In Fig. 4, the input image to our system will be in RGB color. Then, the image will pass to cloud classification algorithm and return the output class which belongs to one of the seven cloud types. Each stage is explained in more details as follows.

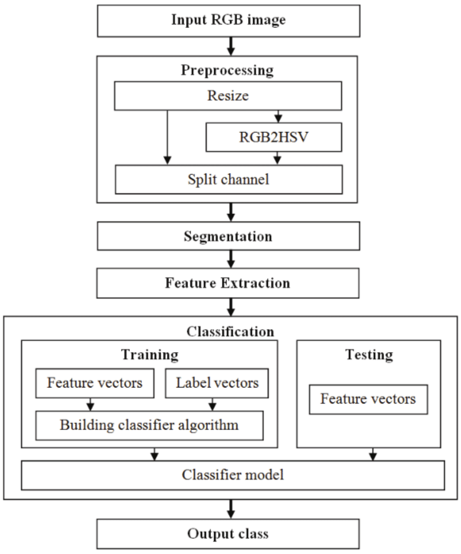


Fig. 4: System overview of proposed cloud classification.

In the preprocessing step, the image resolution is scaled for a suitable size. Channel splitting is used to convert an RGB image to a grayscale image. Moreover, the transformation of RGB to HSV color model (RGB2HSV) is used before channel splitting of HSV. Then, we use a binary mask image for segmentation. A segmentation of each pixel is made by multiplying a pixel value of image with a mask pixel in the same position; the result is set to zero if the pixel value is multiplied by zero, otherwise the result is equal to the pixel value itself if it is multiplied by one. By removing the unwanted parts such as buildings and trees, the output of segmentation will give only clouds and sky. Then, we extract texture feature and Fourier transform feature from the image. The classification is performed based on the extracted features. There are two parts involved in this step which are the training and the testing. The training process is used to construct a classifier model by building classifier algorithm (see Algorithm 1) using feature vectors together with their pre-defined classes (label vectors). Then, the classifier model is tested by utilizing feature vectors in the testing process. The automatic cloud classification system will be implemented and installed at Prince of Songkla University, Phuket Campus where local cloud classification will be performed every 5 minutes.

Algorithm 1 Building classifier algorithm

```
Input: all training grayscale $n$ images, label vectors $L$
Output: classifier model $C$
Compute a feature vector $HK$ from $h \times k$-FFT [12]
Compute $M$ as the second feature vector
/* In Experiment A, set $M$ to the 18 texture features */
/* In Experiment B, set $M$ to the 16 meteorological data */
Set $V = M^\frown HK$
Build a matrix $F$ from a vector $V$ for all $n$ images
Build a label matrix $T$ from label vectors $L$
Train classifier $C$ using $F$ and $T$
return $C$
```

VI. Experimental Result

In the experiment, we retrieve 1,045 whole sky images for seven cloud types from the database (see Fig. 3) and used them as the input. The number of images per class depends on the frequency appearance of each cloud type during the past 2 months of image collection. There are 244 cloud images for cirriform, 80 images for high cumuliform, 22 images for stratocumulus, 153 images for cumulus, 8 images for cumulonimbus, 332 images for stratiform, and 206 images for clear sky which represent various forms of each cloud type. The sample images of these cloud types are as shown in Fig. 5. Due to the unwanted parts of buildings and trees on the sides, the segmentation process is performed before the feature extraction process. A binary mask technique is used for eliminating the unwanted parts (see Fig. 6). We use Leave-One-Out-Cross-Validation (LOOCV) for evaluation of the ANN classifier. The process uses one image for testing and the remaining images for training. The test image is shifted and the training process is repeated 1,045 times. Finally, the errors of 1,045 judgments are averaged to yield the overall correctness. There are two experiments which involves different settings of two feature vectors in Algorithm 1. In the first experiment, we use the concatenation of 18 texture features and $h \times k$-FFT feature. In the second experiment, we replace the 18 texture features by the 16 meteorological data and the result is compared with the first experiment.

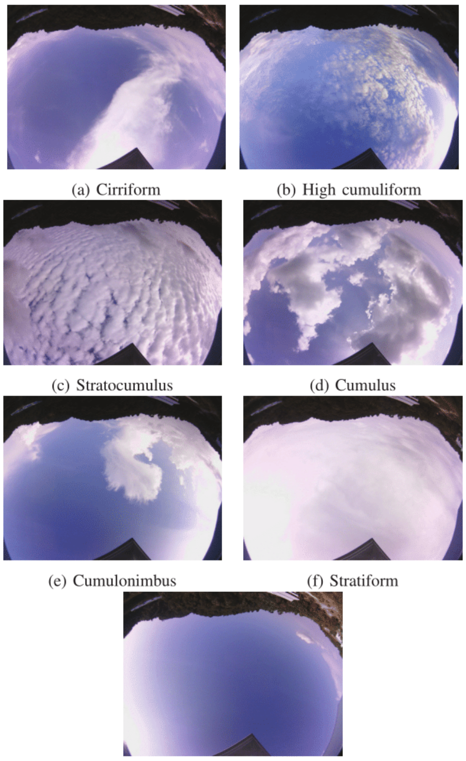

(a) Cirriform (b) High cumuliform (c) Stratocumulus (d) Cumulus (e) Cumulonimbus (f) Stratiform (g) Clear sky

Fig. 5: Seven cloud types from our cloud monitoring station.

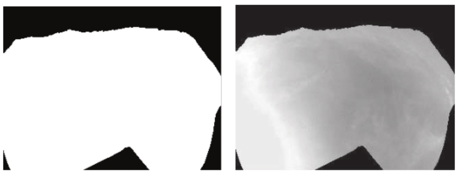

(a) A binary mask image (b) A result of segmentation

Fig. 6: Segmentation of a typical whole sky image.

A. Local Cloud Classification

In this first experiment, we concatenate 18 texture features and 12×6-FFT feature. Table I shows the confusion matrix for classifying our whole sky images into seven cloud classes. Class 1 to Class 7 are cirriform, high cumuliform, stratocumulus, cumulus, cumulonimbus, stratiform, and clear sky, respectively. The correctly classified instances of all class are higher than 87%. Among these results, two classes (cumulus and clear sky) have the classification accuracy as high as 100%. The average accuracy of classification is 96.28% which is higher than the previous study results [2]–[4], [6], [7].

TABLE I: Confusion matrix for classifying the local whole sky images into seven cloud classes.

True class	Classified as						
	1	2	3	4	5	6	7
1	**99.39**	0	0	0.61	0	0	0
2	0.63	**98.75**	0	0.63	0	0	0
3	0	0	**88.64**	11.36	0	0	0
4	0	0	0	**100**	0	0	0
5	6.25	6.25	0	0	**87.50**	0	0
6	0	0	0.15	0.15	0	**99.70**	0
7	0	0	0	0	0	0	**100**

B. Integrating with Meteorological Data

It is interesting to see the performance of our cloud monitoring system if we combine cloud image information with meteorological data. Therefore, in this second experiment, we integrate sixteen meteorological data obtained from the WMR200A weather station. These data are local pressure, sea level pressure, pressure trend, weather status, rainfall rate, current-hour rainfall, last-24-hour rainfall, temperature, temperature trend, humidity, humidity trend, comfort zone, dew point, heat index, gust wind, and the average wind. We set the 16 meteorological data as the first feature vector before we concatenate 12 × 6-FFT feature vector to the first one.

TABLE II: Confusion matrix after adding meteorological data.

True class	**Classified as**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**99.18**	0	0	0.82	0	0	0
2	0	**100**	0	0	0	0	0
3	0	0	**90.91**	4.55	0	4.55	0
4	0	0	0	**100**	0	0	0
5	0	0	0	0	**100**	0	0
6	0	0	0	0	0	**100**	0
7	0	0	0	0	0	0	**100**

Table II shows the confusion matrix after using the concatenation of 16 meteorological data and 12×6-FFT feature. There is some misclassification of Class 3. However, the correctly classified instances of this class are still over 90%. Five classes now give the accuracy of 100%. The average accuracy of classification for this experiment is 98.58%. This result is clearly better than the previous experiment. Furthermore, the computation time is reduced because the 18 texture features was no longer needed in the feature extraction process.

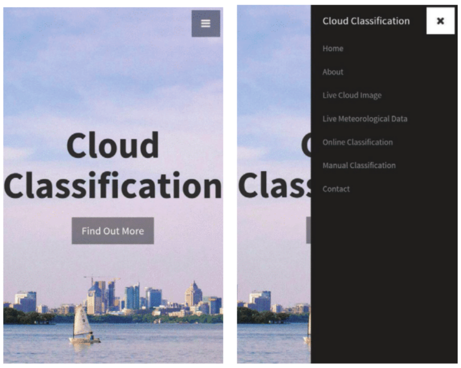
VII. Mobile Application

The approach explained in experiment section is used to develop a mobile application for real-time and online cloud classification. Fig. 7 shows the first page of our mobile application. The main menu is on the top right corner of the page. The important menus are live cloud image, live meteorological data, online classification and manual classification. In addition, the basic descriptions of cloud types and their appearances are available in the about-menu for users to study each cloud characteristic before using the application. Fig. 8 shows the example of live cloud images from our cloud station. The cloud image is automatically updated every 5 minutes. The classification result and the weather condition outlook of each cloud image are available under the image.

Fig. 9 shows the menu of live meteorological data from the weather station. These data are, local pressure (mB), rainfall rate (in/hr), temperature (°C), humidity (%), dew point (°C), gust wind (m/s), and so on. The meteorological data is automatically updated every 5 minutes.

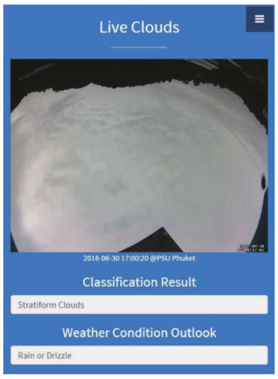
Fig. 10 shows the menu for online classification. In this page, users can submit a cloud image from anywhere, then click upload file. Our system will classify the cloud image into one of the seven cloud types. The classification result is shown on the bottom of the page.

Fig. 11 shows the menu for manual classification. In this page, users can manually view the cloud picture and choose one preferred type from the list, then send the answer to our system for verifying the result. These results reflect different perspectives of how people view the clouds; therefore, the results will be collected in the database for improving our classification process in the future.



(a) Main page (b) Main menu

Fig. 7: The first page of our mobile application.

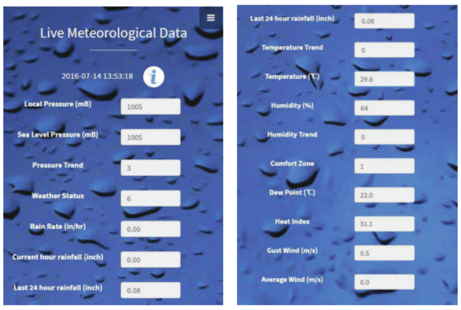


2016-06-30 17:00:20 @PSU Phuket

Stratiform Clouds

Rain or Drizzle

Fig. 8: Live cloud image from our cloud station.



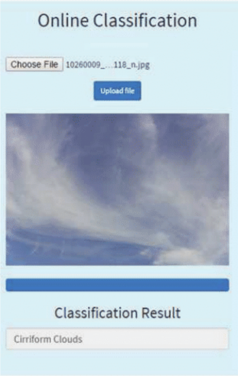
2016-07-14 13:53:18

Local Pressure (mB) 1005
Sea Level Pressure (mB) 1005
Pressure Trend 3
Weather Status 6
Rain Rate (in/hr) 0.00
Current hour rainfall (inch) 0.00
Last 24 hour rainfall (inch) 0.08

Last 24 hour rainfall (inch) 0.08
Temperature Trend 0
Temperature (℃) 29.6
Humidity (%) 64
Humidity Trend 0
Comfort Zone 1
Dew Point (℃) 22.0
Heat Index 31.1
Gust Wind (m/s) 0.5
Average Wind (m/s) 0.0

(a) Live data (b) Live data (cont.)

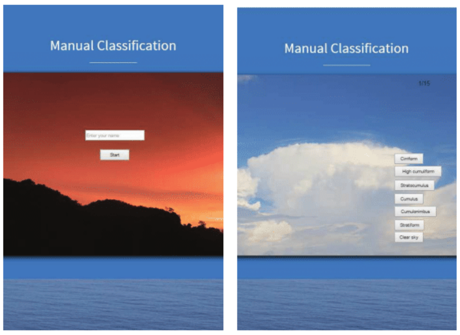
Fig. 9: Live meteorological data menu.



10260009_...118_n.jpg

Cirriform Clouds

Fig. 10: Online classification menu.



Enter your name

(a) Login menu (b) Manual classification

Fig. 11: Manual classification menu.

VIII. Conclusion

A cloud station was designed and implemented. A weather station provided additional useful meteorological information. Together we have a low-cost local cloud monitoring system installed at Prince of Songkla University, Phuket Campus. The performance test of the local cloud classification using our whole sky images yielded 96.28% accuracy which is higher than the previous study results. Then, we replaced the 18 texture features by the 16 meteorological data. The second result gave 98.58% accuracy which is better than relying solely on the whole sky images. This approach was then used to develop real-time and online classification for mobile users. Live cloud image and live meteorological data can be viewed through mobile application from anywhere in the world. Although our system can provide a high performance and has a practical use, it cannot classify clouds in the night-time or in very low light condition. Our approach cannot cover a large area as in the satellite approach. In the future, if our cloud monitoring system is installed in more locations, it may be useful for improving results of the weather forecast in the wider area. Moreover, the past cloud images and the historical meteorological data can be further analyzed for forecasting clouds and weather conditions.

Acknowledgment

This research receives the financial support from Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University. We thank INFAR (INFormation engineering for Andaman Region) research team for providing equipments and areas to test our cloud monitoring station.

References

[1] J. Parikh, "A comparative study of cloud classification techniques," *Remote Sensing of Environment*, vol. 6, no. 2, pp. 67–81, 1977.

[2] J. Calbó and J. Sabburg, "Feature extraction from whole-sky ground-based images for cloud-type recognition," *Journal of Atmospheric and Oceanic Technology*, vol. 25, no. 1, pp. 3–14, 2008.

[3] A. Heinle, A. Macke, and A. Srivastav, "Automatic cloud classification of whole sky images," *Atmospheric Measurement Techniques*, vol. 3, no. 3, pp. 557–567, 2010.

[4] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, "Salient local binary pattern for ground-based cloud classification," *Acta Meteorologica Sinica*, vol. 27, pp. 211–220, 2013.

[5] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pagès, "Retrieving cloud characteristics from ground-based daytime color all-sky images," *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 5, pp. 633–652, 2006.

[6] H.-Y. Cheng and C.-C. Yu, "Block-based cloud classification with statistical features and distribution of local texture features," *Atmospheric Measurement Techniques*, vol. 8, no. 3, pp. 1173–1182, 2015.

[7] Q. Li, Z. Zhang, W. Lu, J. Yang, Y. Ma, and W. Yao, "From pixels to patches: a cloud classification method based on a bag of micro-structures," *Atmospheric Measurement Techniques*, vol. 9, no. 2, pp. 753–764, 2016.

[8] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "WAHRSIS: A low-cost high-resolution whole sky imager with near-infrared capabilities," in *Proceedings of SPIE Infrared Imaging Systems*, vol. 9071, 2014.

[9] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[10] T. Kliangsuwan and A. Heednacram, "Classifiers for ground-based cloud images using texture features," *Advanced Materials Research*, vol. 931–932, pp. 1392–1396, 2014.

[11] T. Kliangsuwan and A. Heednacram, "Feature extraction techniques for ground-based cloud type classification," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8294–8303, 2015.

[12] T. Kliangsuwan and A. Heednacram, "FFT features and hierarchical classification algorithms for cloud images," *Engineering Applications of Artificial Intelligence*, 2016 (submitted).

[13] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2001.

[14] A. Heednacram and T. Samitalampa, "Suspended sediment forecast of khlong bang yai, phuket," *International Journal of Engineering and Technology*, vol. 6, no. 4, pp. 338–345, 2014.

VITAE

Name Mr. Thitinan Kliangsuwan

Student ID 5510130008

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering (Computer Engineering)	Prince of Songkla University	2012

Scholarship Awards during Enrolment

PSU Graduate Student Financial Support for Thesis, October 2013 – September 2015

Scholarships for Graduate Study of Faculty of Engineering, June 2012 – May 2016

Work – Position and Address

Teaching Assistant for 241-213 Mathematics for Computer Engineering, 242-402 Computer Engineering Project II (Image Processing related projects) at Department of Computer Engineering, Phuket Campus, June 2013 – April 2016

Web Administrator for International College of Arts and Sciences Phuket (ICASP), August 2014 – June 2015

List of Publication and Proceeding

T. Kliangsuwan and A. Heednacram, "Classifiers for ground-based cloud images using texture features," *Advanced Materials Research*, vol. 931-932, pp. 1392–1396, May 2014.

T. Kliangsuwan and A. Heednacram, "Feature extraction techniques for ground-based cloud type classification," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8294–8303, Nov. 2015.

T. Kliangsuwan and A. Heednacram, "FFT features and hierarchical classification algorithms for cloud images," *Engineering Applications of Artificial Intelligence* (Submitted).

T. Kliangsuwan and A. Heednacram, “A low-cost local cloud monitoring system,” in *Proceedings of the 2016 International Symposium on Intelligent Signal Processing and Communication Systems* (In press).