AUTOMATIC MARTIAN DUST STORM DETECTION VIA DECISION LEVEL FUSION BASED ON DEEP EXTREME LEARNING MACHINE

Keisuke Maeda, Takahiro Ogawa and Miki Haseyama

Graduate School of Information Science and Technology, Hokkaido University
N-14, W-9, Kita-ku, Sapporo, Hokkaido, 060-0814, Japan
E-mail: {maeda, ogawa}@lmd.ist.hokudai.ac.jp, miki@ist.hokudai.ac.jp

ABSTRACT

This paper presents an automatic Martian dust storm detection via decision level fusion (DLF) based on deep extreme learning machine (DELM). Since Martian images are taken in multi-wavelength bands, DLF techniques which output a final classification result by integrating multiple classification results are necessary. Furthermore, since the number of Martian images taken by satellites is different for each region, the number of the classification results to be integrated is different. Thus, we present a new DLF framework based on confidence values of the classification results. Specifically, we generate multiple extreme learning machines with kernel classifiers to obtain their classification results. Moreover, we monitor the classification results as confidence values and select the same number of the classification results with high confidence for each region. Finally, these selected results can be integrated by using a DLF based on DELM, which is a multilayered ELM. This integration framework is the biggest contribution of our method. Experimental results show the effectiveness of the DLF based on DELM.

Index Terms— Mars, dust storm, deep extreme learning machine, decision level fusion.

1. INTRODUCTION

Recently, the human race has researched some planets in order to pursue the potential for extraterrestrial life and emigrate to other planets [1–3]. Especially, since the environment on Mars is similar to that of Earth, a project which transplants the human race to Mars is being carried out [4]. From such a background, many researchers have studied the climate on Mars [5–8]. On Mars, there exists the phenomenon named "dust storm" that has a great influence on the Martian environment [6]. Although analysis of dust storms has attracted wide attention, detection of dust storms from a large amount of satellite images has been conducted manually. Therefore, since researchers have spent a lot of time [9], an automatic dust storm detection method is necessary.

The dust storm detection can be realized by classifying satellite images into two classes, existence and non-existence [10,11]. Therefore, image classification methods based on classifiers obtained by machine learning techniques [12–14] can be used[1]. Furthermore, since Martian images are taken in multi-wavelength bands, decision level fusion (DLF) techniques which output a final classification result by integrating multiple classification results are necessary. Thus, various DLF techniques have been proposed so far [15–18]. Note that in general DLF methods, the number of classification results to be integrated is predetermined. However, since the number of Martian images taken by satellites is different for each region, the number of the classification results to be integrated is different. Therefore, a novel DLF scheme which can integrate the different number of the classification results is necessary.

In order to overcome the above problem, we propose a novel DLF method in this paper. First, the proposed method generates multiple extreme learning machines with kernel (KELM) [19] classifiers corresponding to multi-wavelength images taken by satellites for each region on Martian surface and obtains multiple classification results. Then by monitoring confidence values of these results, a selection of the same number of the classification results with higher confidence becomes feasible for each region. Finally, a DLF based on deep extreme learning machine (DELM) [20], which is one of deep learning techniques [21–25] and has excellent classification performance, is performed. Therefore, by using the above non-conventional framework, the final classification result can be successfully obtained from the multiple classification results. Experimental results show the effectiveness of the proposed method by comparing with previously reported DLF methods.

2. DUST STORM DETECTION VIA DLF BASED ON DELM

In this section, we explain the automatic dust storm detection via the DLF based on DELM. The proposed method consists of three procedures as shown in Fig. 1. First, we generate multiple KELM classifiers corresponding to multi-wavelength bands in **2.1**. A novel DLF framework based on confidence values of the classification results is explained in **2.2**. Finally, the integration of the multiple classification results via the DLF based on DELM is explained in **2.3**.

2.1. Generation of Multiple KELM Classifiers

Given a training image I_i ($i = 1, 2, ..., N$; N being the number of images), we take difference between I_i and other images taken on the same region in order to extract dust storms' characteristics. The details of these subtraction images are explained in **3.1**. Consequently, we obtain a subtraction image I_{ψ_i} ($\psi_i = 1, 2, ..., \Psi_i$; Ψ_i being the number of subtraction images for I_i) and extract their SIFT-BoF [26] feature vectors. Thus, we obtain training feature vectors from all training images and effective features are selected via a feature selection by the minimum-Redundancy-Maximum-Relevance feature selection algorithm [27]. Furthermore, Martian images are imbalanced, i.e., the number of images belonging to an existence class is smaller than that of images belonging to a non-existence class. Thus,

In this research, we utilized Martian data that was kindly provided by Japan Aerospace eXploration Agency (JAXA). We would also like to sincerely thank Kazunori Ogohara and JAXA, for providing Martian data as well as invaluable advice. This work was partly supported by JSPS KAKENHI Grant Number JP17H01744.

[1]Since there are various kinds of machine learning methods, we cite only survey articles due to the limitation of space.

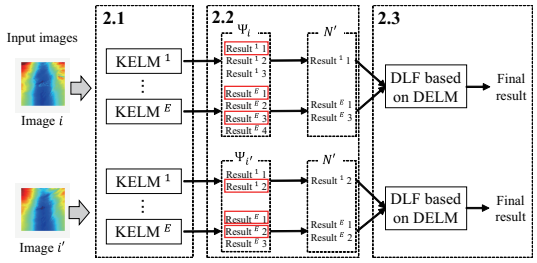


Fig. 1. An overview of the proposed method. We generate E kinds of KELM. Especially, in **2.2**, the number of classification results obtained from each KELM for each region is different. Thus, we select the same number of the classification results based on confidence values of these classification results. For example, in case of image i, KELM^1 outputs three results, but in case of image i', KELM^1 outputs two results. In such the case, the general DLF scheme cannot integrate these different number of results. Therefore, in **2.2**, we select the same number of results based on confidence values.

we assume that the $\tilde{\Psi}$ feature vectors of the minority class are generated via the adaptive synthetic sampling algorithm [28]. Finally, we obtain D-dimensional feature vectors $\boldsymbol{x}_j \in \mathbb{R}^D$ $(j = 1, 2, ..., J; J = \Psi_1 + \Psi_2 + \cdots + \Psi_N + \tilde{\Psi})$.

We generate multiple KELM classifiers from the above training feature vectors. Figure 2 shows the model of KELM with D input nodes, L hidden nodes, M output nodes (M-class problem) and a hidden layer's activation function. The KELM output function $\boldsymbol{f}(\boldsymbol{x}_j) = [f_1(\boldsymbol{x}_j), f_2(\boldsymbol{x}_j), ..., f_M(\boldsymbol{x}_j)]$ is defined as

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}_j) &= \sum_{l=1}^{L} \boldsymbol{\beta}_l \boldsymbol{h}_l(\boldsymbol{x}_j) \\ &= \boldsymbol{h}(\boldsymbol{x}_j)\boldsymbol{\beta}, \end{aligned} \tag{1}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_L]^\top$ is a matrix of the output weights between the hidden layer and the output layer and $\boldsymbol{h}(\boldsymbol{x}_j)$ is an output vector of the hidden layer.

$$\begin{aligned} \boldsymbol{h}(\boldsymbol{x}_j) &= [\boldsymbol{h}_1(\boldsymbol{x}_j), \boldsymbol{h}_2(\boldsymbol{x}_j), ..., \boldsymbol{h}_L(\boldsymbol{x}_j)] \\ &= [G(\boldsymbol{w}_1, b_1, \boldsymbol{x}_j), G(\boldsymbol{w}_2, b_2, \boldsymbol{x}_j), ..., G(\boldsymbol{w}_L, b_L, \boldsymbol{x}_j)], \end{aligned} \tag{2}$$

where $\boldsymbol{h}(\boldsymbol{x}_j)$ actually maps the features from the D-dimensional input space to the L-dimensional hidden feature space, and $G(\boldsymbol{w}_l, b_l, \boldsymbol{x}_j)$, which is a hidden activation function. In the proposed method, we used a gaussian function $K(\boldsymbol{x}_j, \boldsymbol{w}_l, b_l) = \exp(-b_l \|\boldsymbol{x}_j - \boldsymbol{w}_l\|^2)$, where the calculation of $\boldsymbol{w}_l$ and b_l are shown in [19].

In the training of KELM, we try to minimize the training error $\boldsymbol{\xi}_j = [\xi_{j,1}, \xi_{j,2}, ..., \xi_{j,M}]^\top$ as well as the output weights $\boldsymbol{\beta}$.

$$\begin{aligned} &\min_{\boldsymbol{\beta} \in \mathbb{R}^{L \times M}} R = \frac{1}{2}\|\boldsymbol{\beta}\|_F^2 + \frac{C}{2}\sum_{j=1}^{J}\|\boldsymbol{\xi}_j\|^2, \\ &\text{s.t. } \boldsymbol{h}(\boldsymbol{x}_j)\boldsymbol{\beta} = \boldsymbol{t}_j^\top - \boldsymbol{\xi}_j^\top, \ j = 1, ..., J, \end{aligned} \tag{3}$$

where $\boldsymbol{t}_j = [t_{j,1}, t_{j,2}, ..., t_{j,M}]^\top$ is a vector whose m-th element is one, while the rest of elements are zeros if the original true class label is m. Furthermore, C is a regularization parameter. Based on the KKT theorem, Eq. (3) is equivalent to solving the following optimization problem:

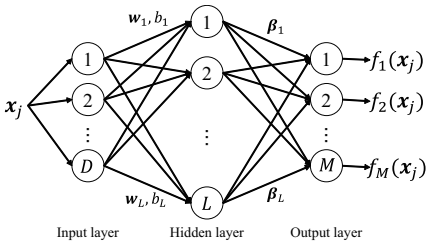


Fig. 2. An overview of a KELM classifier. KELM used in our method is an extended version of ELM which is an efficient learning method of single layer feed-forward neural networks. Given input vectors $\boldsymbol{x}_j \in \mathbb{R}^D$, KELM outputs M classification results.

$$\begin{aligned} \min_{\boldsymbol{\beta} \in \mathbb{R}^{L \times M}} \tilde{R} = \frac{1}{2}\|\boldsymbol{\beta}\|_F^2 + \frac{C}{2}\sum_{j=1}^{J}\|\boldsymbol{\xi}_j\|^2 \\ - \sum_{j=1}^{J}\sum_{m=1}^{M}\alpha_{j,m}(\boldsymbol{h}(\boldsymbol{x}_j)\boldsymbol{\gamma}_m - t_{j,m} + \xi_{j,m}), \end{aligned} \tag{4}$$

where $\boldsymbol{\gamma}_m = [\boldsymbol{\beta}_{1,m}, \boldsymbol{\beta}_{2,m}, ..., \boldsymbol{\beta}_{L,m}]^\top$ is a vector of the weights linking the hidden layer to the m-th output node. By taking derivatives with respect to $\boldsymbol{\gamma}_m$, $\boldsymbol{\xi}_j$ and $\boldsymbol{\alpha}_j$, where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_J]^\top$ and $\boldsymbol{\alpha}_j = [\alpha_{j,1}, \alpha_{j,2}, ..., \alpha_{j,M}]^\top$, the optimal solution of $\boldsymbol{\beta}$ can be obtained as

$$\boldsymbol{\beta} = \boldsymbol{H}^\top\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^\top\right)^{-1}\boldsymbol{T}, \tag{5}$$

where

$$\begin{aligned} \boldsymbol{H} &= [\boldsymbol{h}(\boldsymbol{x}_1)^\top, \boldsymbol{h}(\boldsymbol{x}_2)^\top, ..., \boldsymbol{h}(\boldsymbol{x}_J)^\top]^\top \\ &= \begin{bmatrix} G(\boldsymbol{w}_1, b_1, \boldsymbol{x}_1) & \cdots & G(\boldsymbol{w}_L, b_L, \boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{w}_1, b_1, \boldsymbol{x}_J) & \cdots & G(\boldsymbol{w}_L, b_L, \boldsymbol{x}_J) \end{bmatrix}, \end{aligned} \tag{6}$$

and $\boldsymbol{T} = [\boldsymbol{t}_1^\top, \boldsymbol{t}_2^\top, ..., \boldsymbol{t}_J^\top]^\top$. From the obtained $\boldsymbol{\beta}$, Eq. (1) is rewritten as follows:

$$\boldsymbol{f}(\boldsymbol{x}_j) = \boldsymbol{h}(\boldsymbol{x}_j)\boldsymbol{H}^\top\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^\top\right)^{-1}\boldsymbol{T}. \tag{7}$$

2.2. Classification Confidence-based DLF Framework

In this subsection, a novel DLF framework which integrates the different number of classification results is explained. In subsection **2.1**, given an image I_i, we generate a subtraction image I_{ψ_i} and extract feature vectors $\boldsymbol{x}_{\psi_i}$ and obtain Ψ_i classification results from E kinds of KELM classifiers, where the details of E is shown in **3.1**. Thus, we obtain the final classification result for I_i by integrating Ψ_i classification results. However, since the number of Martian images taken by satellites is different for each region, the number of the obtained classification results for I_i and $I_{i'} (i \neq i')$ is different $(\Psi_i \neq \Psi_{i'})$. Therefore, the proposed method selects the same number of classification results based on their confidence to integrate these classification results.

Given $\boldsymbol{F} = [\boldsymbol{f}(\boldsymbol{x}_1), ..., \boldsymbol{f}(\boldsymbol{x}_{\psi_i}), ..., \boldsymbol{f}(\boldsymbol{x}_{\Psi_i})]$ obtained in **2.1**, each element of $\boldsymbol{F}$ is the output of KELM. Then we monitor $\boldsymbol{f}(\boldsymbol{x}_{\psi_i})$ as confidence values. The higher value $\boldsymbol{f}(\boldsymbol{x}_{\psi_i})$ has, the more classification confidence is. Thus, we select N' classification results which have higher classification confidence for each region, and obtain input vectors $\boldsymbol{u}_i^0$ including the selected confidence values. Consequently, we can obtain the same number of the classification results for each image I_i and train a classifier for DLF in the following subsection.

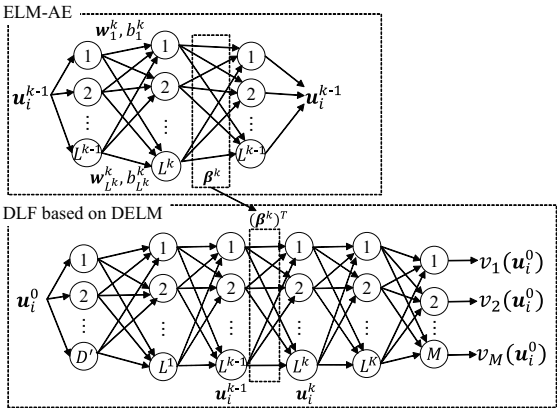


Fig. 3. An overview of the DLF based on DELM. The parameters between DELM's hidden layers are determined by output weights of ELM-AE.

2.3. Decision Level Fusion based on DELM

In this subsection, we generate the DLF based on DELM in order to integrate the obtained N' classification results. The model of the DLF based on DELM is shown in Fig. 3. Given feature vectors $\boldsymbol{u}_i^0 \in \mathbb{R}^{D'}$ ($D' = N' \times M$), $\boldsymbol{u}_i^0$ are input into the DLF based on DELM. Note that since each result has M outputs, the dimensions of an input vector for DELM are $N' \times M$. Furthermore, the final classification result $label(\boldsymbol{u}_i^0)$ for an image I_i is obtained. The predicted class label $label(\boldsymbol{u}_i^0)$ of input features $\boldsymbol{u}_i^0$ is the index of the output node which has the highest output value of M outputs.

$$label(\boldsymbol{u}_i^0) = \arg \max_{m \in \{1,...,M\}} v_m(\boldsymbol{u}_i^0). \tag{8}$$

In DELM, the relationship between the outputs of k-th hidden layer and the outputs of $(k-1)$-th hidden layer can be expressed as

$$\boldsymbol{U}^k = G\Big(\boldsymbol{U}^{k-1}(\boldsymbol{\beta}^k)^\top\Big), \tag{9}$$

where $\boldsymbol{U}^k = [\boldsymbol{u}_1^k, \boldsymbol{u}_2^k, ..., \boldsymbol{u}_N^k]^\top$ is an output matrix of k-th hidden layer of DELM. If $k-1=0$, the corresponding layer is the input layer. Note that $(\boldsymbol{\beta}^k)^\top$ is an output weight matrix of k-th hidden layer of DELM. The output weight matrix of the connections between the last hidden layer and the output layer can be analytically calculated using regularized least squares [29].

In training of DELM, calculation of the optimal output weight matrix $(\boldsymbol{\beta}^k)^\top$ is necessary. Although general deep learning methods determine parameters of their networks by using back propagation, DELM determines parameters of the network by using ELM-auto encoder (ELM-AE) which is a layer-by-layer unsupervised learning algorithm. Thus, the training of DELM is faster and can be realized from lower training examples. The model of ELM-AE with L^{k-1} input layer nodes, L^k hidden layer nodes and L^{k-1} output layer nodes is also shown in Fig. 3. Given an input vector $\boldsymbol{u}_i^{k-1}$ for the ELM-AE, the outputs of ELM-AE's hidden layer and these from the output layer can be respectively obtained as

$$\boldsymbol{y}_i^k = G(\boldsymbol{W}^k \cdot \boldsymbol{u}_i^{k-1} + \boldsymbol{b}^k), \tag{10}$$

where G is the hidden layer's activation function which is a sigmoid function. Generally, ELM-AE has the orthogonal random weights $\boldsymbol{W}^k = [\boldsymbol{w}_1^k, \boldsymbol{w}_2^k, ..., \boldsymbol{w}_{L^k}^k]^\top$ and the orthogonal random bias $\boldsymbol{b}^k = [b_1^k, b_2^k, ..., b_{L^k}^k]^\top$ of hidden nodes. Thus, $(\boldsymbol{W}^k)^\top \cdot \boldsymbol{W}^k = \boldsymbol{I}$, $(\boldsymbol{b}^k)^\top \cdot \boldsymbol{b}^k = 1$.

Table 1. The details of Martian images used in this experiment. N^s is the number of the images taken in region s, and ${}_{N^s}\mathrm{C}_2$ is the number of subtraction images of region s.

Region	Original images			Subtraction images			
s	existence	non-existence	N^s	Pattern A	Pattern B	Pattern C	${}_{N^s}\mathrm{C}_2$
1	4	15	19	${}_4\mathrm{C}_2$	4×15	${}_{15}\mathrm{C}_2$	${}_{19}\mathrm{C}_2$
2	5	13	18	${}_5\mathrm{C}_2$	5×13	${}_{13}\mathrm{C}_2$	${}_{18}\mathrm{C}_2$
3	2	19	21	${}_2\mathrm{C}_2$	2×19	${}_{19}\mathrm{C}_2$	${}_{21}\mathrm{C}_2$
4	2	16	18	${}_2\mathrm{C}_2$	2×16	${}_{16}\mathrm{C}_2$	${}_{18}\mathrm{C}_2$
5	2	14	16	${}_2\mathrm{C}_2$	2×14	${}_{14}\mathrm{C}_2$	${}_{16}\mathrm{C}_2$
6	3	14	17	${}_3\mathrm{C}_2$	3×14	${}_{14}\mathrm{C}_2$	${}_{17}\mathrm{C}_2$
7	5	14	19	${}_5\mathrm{C}_2$	5×14	${}_{14}\mathrm{C}_2$	${}_{19}\mathrm{C}_2$
Total	23	105	$N = 128$	32	335	747	1114

By using $\boldsymbol{Y}^k = [\boldsymbol{y}_1^k, \boldsymbol{y}_2^k, ..., \boldsymbol{y}_N^k]^\top$ which is a hidden layer output matrix of ELM-AE, and $\boldsymbol{U}^{k-1}$ which is a its input matrix, the calculation of the output weight $\boldsymbol{\beta}^k$ of ELM-AE can be divided into the following two patterns.

(i) $L^k = L^{k-1}$

This means that the number of nodes of an input layer of ELM-AE is equal to the number of nodes of a hidden layer of ELM-AE. The output weight $\boldsymbol{\beta}^k$ is simply obtained as

$$\boldsymbol{\beta}^k = (\boldsymbol{Y}^k)^{-1}\boldsymbol{U}^{k-1}, \tag{11}$$

where $(\boldsymbol{\beta}^k)^\top \boldsymbol{\beta}^k = \boldsymbol{I}$.

(ii) $L^k \neq L^{k-1}$

This means sparse or compressed representation. The output weight $\boldsymbol{\beta}^k$ is obtained as follows:

$$\boldsymbol{\beta}^k = \left(\frac{\boldsymbol{I}}{C'} \sum_{l^k=1}^{L^k} \mathrm{KL}(\rho \| \hat{\rho}_{l^k}) + (\boldsymbol{Y}^k)^\top \boldsymbol{Y}^k\right)^{-1} (\boldsymbol{Y}^k)^\top \boldsymbol{U}^{k-1}, \tag{12}$$

where $\mathrm{KL}(\rho \| \hat{\rho}_{l^k}) = \rho \log \frac{\rho}{\hat{\rho}_{l^k}} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_{l^k}}$ is the KL divergence, ρ is a sparsity parameter ($\rho = 0.05$), and $\hat{\rho}_{l^k}$ is the average activation of each hidden layer's node l^k of ELM-AE.

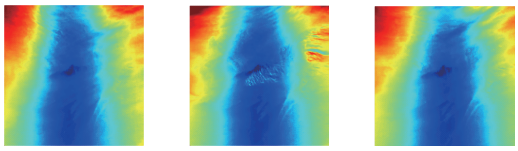
In the proposed method, since the number of the input layer nodes of DELM is very low, we obtain $\boldsymbol{\beta}^1$ by calculating Eq. (12). In other cases, we obtain $\boldsymbol{\beta}^k$ by calculating Eq. (11). Consequently, construction of DELM network is realized by using output weights based on ELM-AE. In this way, we can integrate the different number of the classification results by the selection based on the confidence of the classification results in **2.2**. Furthermore, the DLF based on DELM improves the classification performance of the previously reported DLF methods.

3. EXPERIMENTAL RESULTS

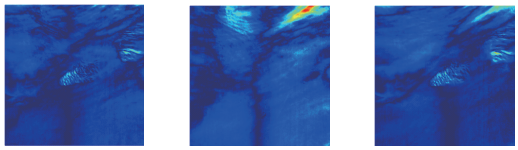
In this section, we explain the details of Martian images used in our experiment in **3.1**. Furthermore, experimental results are shown to verify the effectiveness of the proposed method in **3.2**.

3.1. Characteristics of Martian Images

In studies on Mars, many researchers have used the reflectance images taken by Mars Orbiter Camera (MOC) on Mars Global Surveyor (MGS). Since the MOC has daily global maps of Martian surface in two wavelength bands : BLUE (400nm – 450nm) and RED (575nm – 625nm), we use both of the reflectance images. Examples of RED reflectance images are shown in Figs. 4 (a)-(c). These images are taken in the same region at different times.



(a) Non-existence image at time $T-1$ (b) Existence image at time T (c) Non-existence image at time $T+1$



(d) Difference between (a) and (b) (e) Difference between (a) and (c) (f) Difference between (b) and (c)

Fig. 4. (a)-(c) original RED reflectance images taken in the same region, (d)-(f) subtraction images obtained from (a)-(c). In this example, (d) and (f) belong to pattern B, and (e) belongs to pattern C in Table 2.

Table 2. Generation of two kinds of classifiers for each wavelength. Given an unknown image, since we take the subtraction between the unknown image and training images, we can classify existence or non-existence based on these classified patterns.

	Positive-class	Negative-class
KELM_T	Pattern A	Pattern B
KELM_F	Pattern B	Pattern C

In our study, the original Martian images are preprocessed for accurate classification of existence or non-existence images, i.e., $M = 2$. Since dust storms occur by flinging up sand and dust on Martian surface, the discrimination between dust storms and Martian surface is difficult by using only original images. Thus, we generate subtraction images from the original Martian images. Since the MOC takes the same region on Martian surface every few days, the subtraction images are generated by taking difference between images taken at different times. Thus, when there exist N^s images on region s, we can obtain ${}_{N^s}\text{C}_2$ subtraction images as shown in Table 1. Each subtraction image belongs to at least one of the following three patterns:

· Pattern A : Difference between existence images

· Pattern B : Difference between existence image and non-existence image

· Pattern C : Difference between non-existence images

Then we generated two kinds of classifiers ($\text{KELM}_T, \text{KELM}_F$) shown in Table 2. Note that there is no dust storm in pattern C. Furthermore, since we generated KELM_T and KELM_F for each wavelength, four kinds of classifiers ($E = 4$) are obtained as shown in **2.1**. Given an unknown image, we obtain two kinds of subtraction images which are differences between the unknown image and training images with existence or non-existence. Next, these obtained subtraction images are input into appropriate KELM classifiers, and we obtain the classification results of these subtraction images.

3.2. Performance Evaluation

As shown in Table 1, we used 128 images including 23 existence images and 105 nonexistence images in the experiment. Then the number of the subtraction images of patterns A, B and C are 32, 335 and 747, respectively.

In our method, since the number of DELM's hidden layers K is three, we constructed a five layer deep network. The number of DELM's hidden layer nodes L^k was ten in the proposed method. The verification method was three-fold cross-validation. Furthermore, each image has a resolution of 300×300 pixels. For the evaluation, we utilized Recall, Precision and F-measure. In order to verify the effectiveness of the proposed method, we compare its classification performance with other DLF methods shown in Table 3. Since the method proposed by Raykar et al. [16] is one of the most famous benchmarking methods of supervised decision level fusion, we used this method.

Table 3. Comparison of classification performance of our method and comparative methods.

Methods	Recall	Precision	F-measure
Majority voting	0.82	0.95	0.88
ELM-based DLF	0.87	0.94	0.89
KELM-based DLF	0.92	**0.96**	0.93
Raykar et al. [16]	0.93	0.92	0.92
Proposed method	**0.99**	0.93	**0.96**

Table 4. Classification performance of four kinds of KELM classifiers which classify subtraction images into the patterns shown in Table 2.

Classifiers		Recall	Precision	F-measure
RED	KELM_T	0.67	0.37	0.47
RED	KELM_F	0.82	0.56	0.67
BLUE	KELM_T	0.65	0.74	0.68
BLUE	KELM_F	0.56	0.53	0.54

The experimental results are shown in Tables 3 and 4. Table 4 shows the classification performance of four kinds of KELM classifiers. These classifiers assign subtraction images to the patterns shown in Table 2. Although from Table 4, the classification performance of subtraction images is low, as shown in Table 3, by integrating selected results based on the confidence of classification results, DLF performance of all methods becomes high. Therefore, our confidence-based DLF framework realizes the integration of the different number of classification results with high performance.

Furthermore, in the proposed method, Table 3 shows Recall and F-measure, which is the harmonic mean of Recall and Precision are the highest of all comparative methods. Therefore, the DLF based on DELM outperforms the other DLF methods. Moreover, the computation time for the training and the test of DELM for DLF are 2.1×10^{-3} sec and 0.9×10^{-3} sec [2], respectively. Thus, compared to general deep learning methods, computation costs of DELM are extremely lower. In addition, the DLF based on DELM needs less training examples for the accurate classification, which is also a unique characteristic of our method. Thus, we focus on the ELM series. Since it is difficult to prepare sufficient training examples for each region, the above characteristic is effective. From the above discussion, the effectiveness of our confidence-based DLF framework is verified, and the DLF based on DELM realizes not only accurate but also high-speed dust storm classification.

4. CONCLUSIONS

In this paper, we have presented a novel DLF framework based on the confidence of the classification results. In order to train the classifier for DLF, we have selected some classification results with higher confidence based on the outputs of KELM. Furthermore, we have obtained the final classification result by using the DLF based on DELM which integrated these selected classification results. The experimental results verified the effectiveness of the proposed method.

[2]Note that this experiment was performed on a personal computer using Intel (R) Core (TM) i7-3770 CPU 3.40 GHz with 16.0 Gbytes RAM.

5. REFERENCES

[1] A. D. D. Genio and J. M. Barbara, "An objective classification of saturn cloud features from cassini iss images," *Icarus*, vol. 271, pp. 222–236, 2016.

[2] E. Kardasis, J. H. Rogers, G. Orton, M. Delcroix, A. Christou, M. Foulkes, Y. F. Padma, M. Jacquesson, and G. Maravelias, "The need for professional-amateur collaborations in studies of jupiter and saturn," *arXiv preprint arXiv:1503.07878*, 2015.

[3] S. D. Guzewich, A. D. Toigo, L. Kulowski, and H. Wang, "Mars orbiter camera climatology of textured dust storms," *Icarus*, vol. 258, pp. 1–13, 2015.

[4] T. May and G. Lyles, "Nasa's space launch system program update," 2015.

[5] B. A. Cantor, P. B. James, M. Caplinger, and M. J. Wolff, "Martian dust storms: 1999 mars orbiter camera observations," *Journal of Geophysical Research: Planets*, vol. 106, no. E10, pp. 23653–23687, 2001.

[6] H. Wang, M. I. Richardson, R. J. Wilson, A. P. Ingersoll, A. D. Toigo, and R. W. Zurek, "Cyclones, tides, and the origin of a cross-equatorial dust storm on mars," *Geophysical Research Letters*, vol. 30, no. 9, 2003.

[7] S. W. Bougher, D. Pawlowski, J. M. Bell, S. Nelli, T. M. Dunn, J. R. Murphy, M. Chizek, and A. Ridley, "Mars global ionosphere-thermosphere model: Solar cycle, seasonal, and diurnal variations of the mars upper atmosphere," *Journal of Geophysical Research: Planets*, vol. 120, no. 2, pp. 311–342, 2015.

[8] K. Ogohara, T. Munetomo, Y. Hatanaka, and S. Okumura, "Automated detection of martian water ice clouds using support vector machine and simple feature vectors," *Planetary and Space Science*, vol. 134, pp. 64–70, 2016.

[9] H. Wang and M. I. Richardson, "The origin, evolution, and trajectory of large dust storms on mars during mars years 24–30 (1999–2011)," *Icarus*, vol. 251, pp. 112–127, 2015.

[10] K. Maeda, T. Ogawa, and M. Haseyama, "Automatic detection of martian dust storms from heterogeneous data based on decision level fusion," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2246–2250.

[11] K. Maeda, T. Ogawa, and M. Haseyama, "Automatic martian dust storm detection from multiple wavelength data based on decision level fusion," *Information and Media Technologies*, vol. 10, no. 3, pp. 473–477, 2015.

[12] W. Y. Lin, Y. H. Hu, and C. F. Tsai, "Machine learning in financial crisis prediction: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 421–436, 2012.

[13] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2013.

[14] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[15] J. Whitehill, T. F. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in neural information processing systems*, 2009, pp. 2035–2043.

[16] V. C. Raykar, S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy, "Supervised learning from multiple experts: whom to trust when everyone lies a bit," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 889–896.

[17] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The multidimensional wisdom of crowds," in *Advances in neural information processing systems*, 2010, pp. 2424–2432.

[18] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 469–478.

[19] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[20] E. Cambria, G. B. Huang, L. L. C. Kasun, H. Zhou, C. M. Vong, J. Lin, J. Yin, Z. Cai, Q. Liu, and K. Li et al., "Extreme learning machines [trends & controversies]," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 30–59, 2013.

[21] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems*, 2012, pp. 665–673.

[22] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2329–2336.

[23] Z. Liao and G. Carneiro, "The use of deep learning features in a hierarchical classifier learned with the minimization of a non-greedy loss function that delays gratification," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4540–4544.

[24] D. Held, S. Thrun, and S. Savarese, "Deep learning for single-view instance recognition," *arXiv preprint arXiv:1507.08286*, 2015.

[25] W. Li, M. Li, Z. Su, and Z. Zhu, "A deep-learning approach to facial expression recognition with candid images," in *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*. IEEE, 2015, pp. 279–282.

[26] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, 1999, vol. 2, pp. 1150–1157.

[27] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.

[28] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1322–1328.

[29] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*. IEEE, 2004, vol. 2, pp. 985–990.