

Received March 13, 2020, accepted April 11, 2020, date of publication May 6, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992683

# Towards Imbalanced Image Classification: A Generative Adversarial Network Ensemble Learning Method

YANGRU HUANG<sup>1</sup>, YI JIN<sup>1</sup>, (Member, IEEE), YIDONG LI<sup>1</sup>,  
AND ZHIPING LIN<sup>2</sup>, (Senior Member, IEEE)

<sup>1</sup>School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

<sup>2</sup>School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

Corresponding author: Yi Jin (yjin@bjtu.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2020JBM019, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61972030.

**ABSTRACT** Learning from minority class has been a significant and challenging task which has many potential applications. Weather classification is such a case of imbalanced label distribution. This is because in places like Beijing, some types of weather, such as rain and snow, are relatively rare compared to sunny and haze days. Existing methods are primary to classify the weather conditions relying on expensive sensors or human assistance, which however usually are expensive and time-consuming. In this paper, we propose a new ensemble framework based on the advanced generative adversarial network and an effective data cleaning way to address the class imbalance problem for weather classification. The proposed method not only generates new and reliable samples for the minority class to restore balance, but also filters those generated samples which are unreliable. Experiments show that our approach outperforms the state-of-the-art methods by a huge margin for imbalanced weather classification on several benchmark data sets.

**INDEX TERMS** Imbalanced learning, GAN, data augmentation, data cleaning, boosting, weather classification.

## I. INTRODUCTION

Weather conditions not only strongly influence our daily lives in many ways but also have many potential applications ranging from solar energy systems to outdoor intelligent transportation. While weather classification [1], [2] techniques have been pretty accurate, most of them relied on expensive sensors or human assistance to identify the weather conditions. If we can exploit existing surveillance, which captures thousands of weather images in each camera everyday, it may be possible to turn an expensive sensor detection system into a powerful and cost-effective computer vision application.

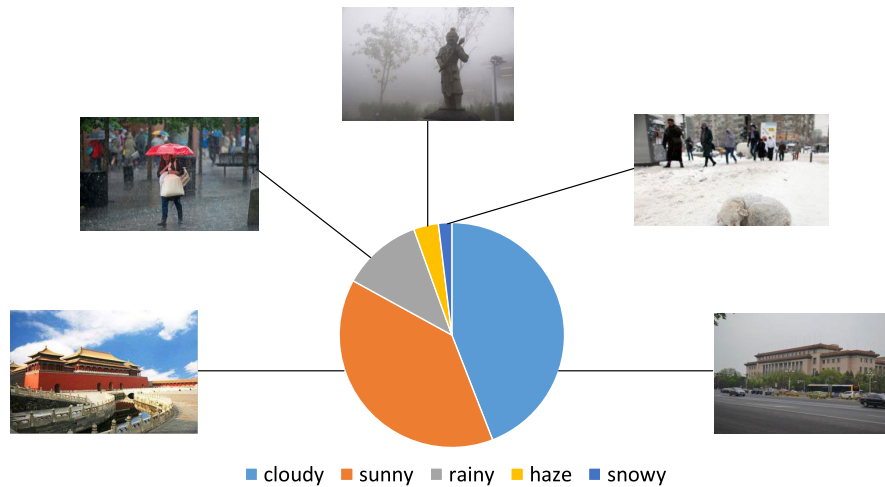
In the paper, we formulate the weather classification as an image classification problem based on computer vision. Weather image classification is influenced by various factors, such as complex lighting changes, shadows and object occlusion. In particular, although the previous works [3]–[5] have provided interesting solutions for image classification, most of them only pay attention to the progress of feature extraction

and matching [6] without considering the imbalanced data. That means these methods are based on the clear and balanced data. However, as the weather distribution table of 365 days a year in Beijing shown in Figure 1, more than 300 days a year are cloudy and sunny, while the percentage of snowy weather or hazy weather a year is just one to two percents. Overall, the data of weather are extremely unbalanced in real life.

Existing classification methods tend to underperform on minority classes. This is because their goal is to optimize the overall accuracy regardless of the relative distribution of each class [7]. When a small number of samples (such as snow) is combined with most types of samples (such as sunny days) to train a model, most of machine learning classification algorithms often struggle to learn the characteristics of snowy days. To address the problem, in the paper, we take measures from two aspects: constructing a balanced dataset and designing a stronger classifier.

To make the weather data balanced, we can increase more samples to the minority class as supplementary data or remove samples from the majority class. However, only creating copies of the existing samples may cause over-fitting

The associate editor coordinating the review of this manuscript and approving it for publication was Yuyu Yin <sup>1</sup>.



**FIGURE 1.** Weather data for 365 days in Beijing in 2018. As we have observed, the weather data are extremely imbalanced. There are many sunny and cloudy days in a year, and almost no snow days.

and the reduction of the number of samples may cause loss of useful information by removing significant patterns [8]. Thus it is important to figure out a valid and reliable generative model for mimicking real minority class of weather in order to augment and balance the training dataset in classification.

Motivated by the remarkable successes of Generative Adversarial Nets (GAN) [9] in computer vision and machine learning, we adopt GAN to increase the number of samples on the minority class. On the one hand, GAN can leverage the practically unlimited amount of unlabeled images to learn good intermediate representations [10]. On the other hand, instead of simply copying the content and style of the original image, the images produced by GAN are with diversity [11]. And it does not require the data to follow any specific assumptions when modeling the complex data even with some implicit distributions. Therefore, GAN is naturally suitable for the task of data generation in order to augment and balance the training dataset in classification. Further, VGG is introduced to extract the weather feature based on the augmented weather dataset that consists of both the original and the generated data. Although some generated weather data by using GAN are severely distorted, almost no studies try to filter the generated weather data. In the paper, we introduce the data clearing method Edited Nearest Neighbours (ENN) [12] to remove noisy instances from the training set whose class label differs from the class of at least half of its  $k$  nearest neighbors. The accuracy of classification model not only depends on the balanced training data but also the applied classifiers. We thus present an ensemble learning method by jointly applying the  $k$ -nearest neighbors (KNN), support vector machine (SVM), random forest (RF) and the boosting method. The framework is shown in Figure 2.

To summarize, our main contributions are as follows:

- A framework is proposed for data augmentation by using GAN to generate supplementary data in weather classification task, which aims to address the weather classification problem from an imbalanced dataset.

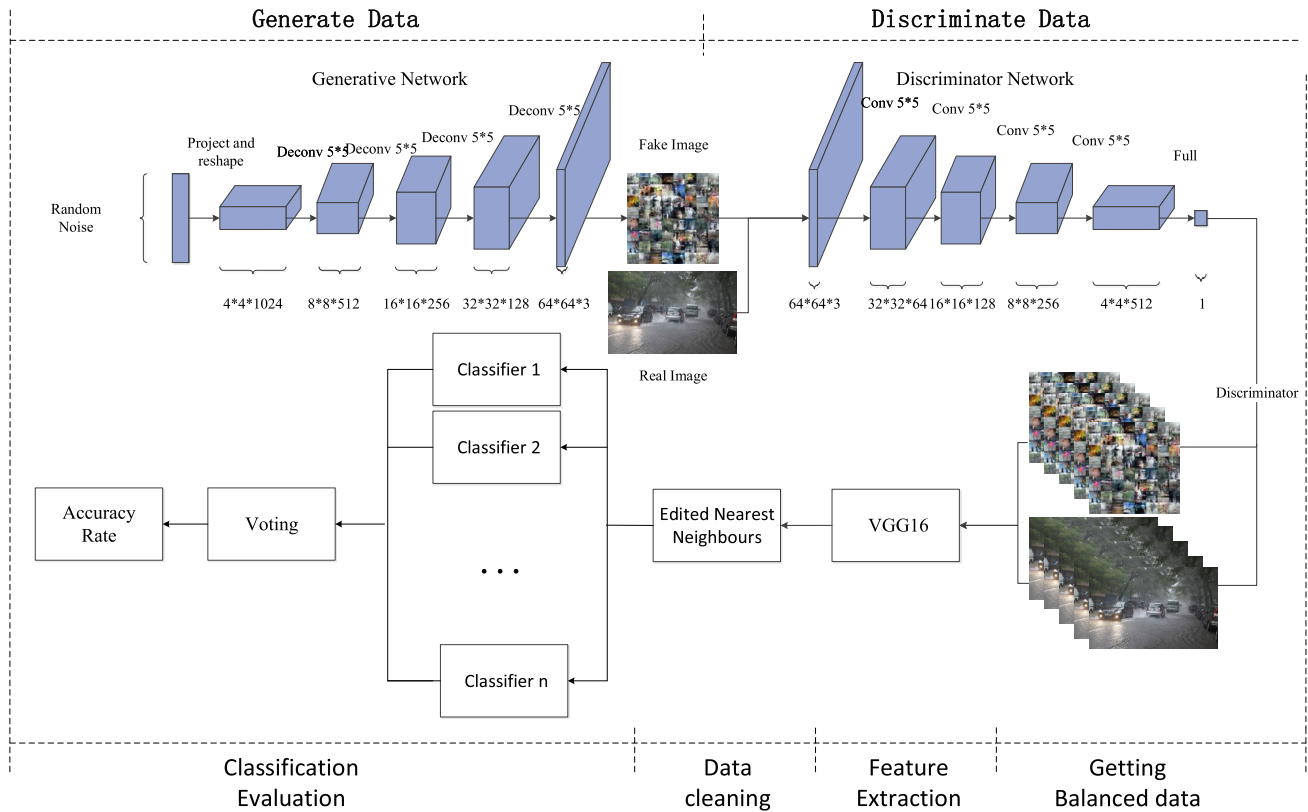
- To get high-quality data, data cleaning methods are introduced to select more reliably generated samples. In addition, we apply the ensemble learning method using KNN, SVM, RF and Adaboost jointly.
- Extensive experiments have been conducted and the results demonstrate that the proposed method is effective and performs superiorly than training the model based on unbalanced data directly.

## II. RELATED WORK

### A. LEARNING WITH IMBALANCED DATASETS

Weather classification has attracted much attention in computer vision recently. Current techniques related to weather classification mainly focus on balanced datasets [3], [4]. However, as we known, weather classification is such a case of imbalanced distribution of labels. Hence, learning with imbalanced weather datasets is of great significance.

Data-level methods for addressing class imbalance include over-sampling and under-sampling. Chawla *et al.* [13] introduce the Synthetic Minority Over-sampling Technique (SMOTE) which produces artificial minority samples by interpolating between existing minority samples and their nearest minority neighbors. Majority weighted minority over-sampling technique (MWMOTE) [14] generates the synthetic minority class samples according to Euclidean distance. In imbalanced data sets, the noisy and borderline examples might create problems. To address the problem, some methods make efforts to develop re-sampling method with some filtering techniques. Borderline-SMOTE [15] improves upon the original algorithm by also taking majority class neighbors into consideration by limiting over-sampling to the samples near class borders. SMOTE-IPF [16] addresses the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. However, the broadened decision regions of the SMOTE variants [15], [16] are still error-prone by synthesizing noisy and borderline examples [17]. Therefore, although facing a risk



**FIGURE 2.** An illustration of the proposed framework of using DCGAN for data augmentation and classification using multi-classifiers.

of less fitting, under-sampling the larger class data to make it similar in number to the smaller class is often preferred to over-sampling [18]. In [19], the air pollutant data is used to identify the weather quality also in a extremely imbalanced setting and a prior duplication strategy is applied to improve the forecast of minority class. Different from these works, we aim to resolve this problem by generating weather data of minority classes based on GAN, which does not require the data to follow any specific assumptions when modeling the complex data even with some implicit distributions [20].

### B. IMBALANCED LEARNING VIA GAN

Generative Adversarial Networks (GANs) are powerful generative models which have achieved impressive results in many computer vision tasks such as image generation [21], super resolution [22] and image-to-image translation [23]–[27]. GANs formulate generative modeling as a game between two competing networks: a generator network produces synthetic data given some input noise and a discriminator network distinguishes between the generator's output and true data. The game between the generator  $G$  and the discriminator  $D$  has the minmax objective. Unlike GANs which learn a mapping from a random noise vector to an output image, conditional GANs (cGANs) [10] learn a mapping from a random noise vector to an output image conditioning on additional information. Pix2pix is a generic image-to-image translation algorithm using cGANs [10].

It can produce reasonable results on a wide variety of problems. Given a training set which contains pairs of related images, pix2pix learns how to convert an image of one type into an image of another type, or vice versa. Cycle-consistent GANs (Cycle GANs) [24] learn the image translation without paired examples. By inverting the mapping of a cGAN [10], i.e., mapping a real image into a latent space and a conditional representation. More recently, Star GAN [27] is proposed to perform multi-domain image translation using a single network conditioned on the target domain label. It learns the mappings among multiple domains using only a single generator and a discriminator. Different from StarGAN, which learns all domain transformations within a single model, we train different simple composable translation networks for different attributes.

A well known problem of generative adversarial models is that while they learn to fool the discriminator they may end up drawing one or few foolish examples. In this work, our aim is to augment an imbalanced image dataset to restore its balance. It is of paramount importance that the augmented dataset is variable enough and does not include a continuously repeating example. Thus we need to avoid mode collapse.

### III. THE PROPOSED METHOD

The overall architecture of our method can be seen in Fig 2. In the following, we will focus on introducing the

implementations of the method and describe it in the form of an algorithm.

### A. RESAMPLING WITH GENERATIVE ADVERSARIAL NETWORKS

As the classification results are more biased towards the conventional majority in the classification process, some minority categories are seriously ignored. In this paper, Generative Adversarial Network (GAN) is introduced as a data enhancement approach to effectively increase the number of minority class samples to solve the weather's imbalanced problem rather than by over-sampling with replacement like rotation and skew which only operates in data space. GAN has the ability to synthesize realistic images with better quality through implicitly modeling high-dimensional distributions of data [28], [29].

We mainly focus on feature space by reusing parts of the generator and discriminator networks as feature extractors. The generator which takes a uniform noise distribution as the input tries to find the best image to fool the discriminator and allows us to generate visually appealing samples. Note that, only the minority class is over-sampled by taking each minority class sample. In our experiments, only if the number of some class samples is less than half the number of samples in another class will be displayed as the minority class and then used as the input of discriminator to assist the model to generate more images with the same property.

#### 1) ORIGINAL GENERATIVE ADVERSARIAL NETWORKS (GANs)

GANs includes two models: a generative model  $G$  that captures the data distribution, and a discriminative model  $D$  that estimates the probability that a sample came from the training data rather than  $G$  [9].

- Generator( $G$ ). The generator takes in random numbers with distribution  $p_z(z)$ , and returns an image by making the generated distribution  $p_g(x)$  as close as to the target data distribution  $p_{data}(x)$  as possible.

$$L_G = E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (1)$$

- Discriminator( $D$ ). The Discriminator takes in both real and fake images and returns probabilities to tell the difference between  $p_{data}(x)$  and  $p_g(x)$ .

$$L_D = -E_{x \sim p_{data}} [\log(D(x))] - E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (2)$$

#### 2) DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS (DCGANs)

GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs, hence deep convolutional generative adversarial networks (DCGANs) [30] is introduced in the paper. Similar to GANs, the structure of DCGANs also includes a discriminative network and a generated network. However, DCGANs have certain architectural constraints, and is demonstrated that they

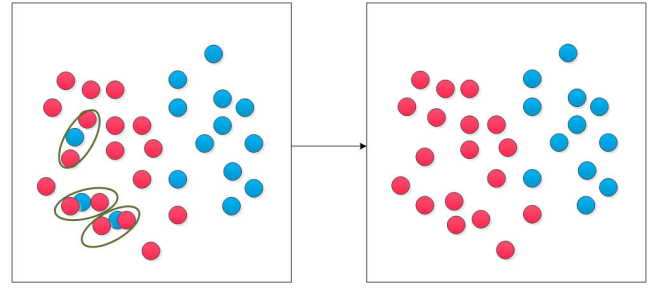


FIGURE 3. The idea of Edited Nearest Neighbor.

are a strong candidate for unsupervised learning. Specially, DCGANs have the following features:

- All pooling layers are cancelled in DCGAN. In the generator, fractional-strided convolution instead of the pooling layer is used for the down sampling. In the discriminator network, stride convolution is used instead of the pool to perform down sampling.
- Batch normalization (BN) is used for all layers of the network except the output layer of the generator and the input layer of the discriminator, and the use of BN enables stable learning and helps to deal with training problems caused by poor initialization.
- The fully connected layers are removed and the convolutional layer is used directly to connect the input and output layers of the generator and discriminator.
- Use the Tanh activation function at the output layer of the generator and ReLU at other layers, and use the leaky ReLU at the discriminator.

### B. DATA CLEANING

Besides sampling methods, another pre-processing step that is gaining popularity in class imbalance classification is data cleaning. Most generated image looks real, but there are still some poorly generated images. To overcome this limitation, data cleaning method Edited Nearest Neighbor (ENN) Rule is utilized to eliminate the outliers samples which generated by GAN method.

As is shown in Figure 3, ENN uses the existing sample set to edit itself and screen the samples at the junction of different weather categories in an appropriate way, which can achieve the dual purpose of reducing the number of samples and improving the correct recognition rate. First, for each  $X_i$ , find the nearest sample  $Y_i(X_i)$  in different categories. If  $Y_i$  and  $X_i$  do not belong to the same category,  $X_i$  is deleted from the current weather class, and finally a clipped sample set is obtained to replace the original sample set and classify the samples.

### C. ENSEMBLE LEARNING

In weather classification, the importance of image classifier cannot be understated. Because of the multiple distribution nature of diversity for the weather image, a single supervised classifier is probably insufficient to catch the diverse representations of weather data. Thus, in this paper, an ensemble



learning method is designed to improve the generalization ability of the overall classifier. First, two single effective classifier k-nearest neighbors (KNN) and Support vector machine (SVM) are introduced. KNN classifies data based on the distance metric whereas SVM need a proper phase of training. KNN is less computationally intensive than SVM. However, due to the optimal nature of SVM, it is guaranteed that the separated data would be optimally separated [31]. Second, two famous ensemble learning methods Random forest (RF) and AdaBoost are combined. RF is able to handle high dimensional data without feature selection but there can possibly be a problem of high bias if not modeled properly, and Adaboost can be less susceptible to the overfitting problem than most learning algorithms but it is sensitive to noisy data and outliers [32]. Finally, the predicted class label for a particular sample is the class label that represents the majority of the class labels predicted by each individual classifier (KNN, SVM, RF and AdaBoost). For example, if KNN, SVM, RF predict some sample belongs to sunny, and only AdaBoost predict this sample belongs to snow, we set the sample label is sunny.

#### 1) k-NEAREST NEIGHBORS (KNN) [33]

KNN classifies samples by measuring the distance between different features. If most of the  $k$  closest training examples in the feature space (that is, the most adjacent samples in the feature space) belong to a certain category, the sample also belongs to this category, where  $k$  is usually an integer not greater than 20. This method only determines the category of the samples to be divided according to the category of the nearest one or several samples. First, the distance between the test sample and all training data is calculated. Then, the distances should be sorted in ascending order. And the  $K$  points with the smallest distance are selected. Counting the occurrence times of the category of the first  $K$  points, finally, the category with the highest occurrence in the first  $K$  points is returned as the prediction  $P_{knn}$ .

#### 2) SUPPORT VECTOR MACHINE (SVM) [34]

The fundamental objective of SVM is to find a hyperplane to separate the points of different classes. However, it is clear that sometimes a full separation of two different objects would require a curve which is more complex than a line. We need to use a set of mathematical functions, known as kernels, to map the original objects to the higher dimensional feature space to find an optimal line that can separate different objects. In short, SVM performs classification tasks by mapping a vector of predictors into higher dimensional plane that separates cases of two different class labels. The model can be written:

$$f(x) = w^T \Phi(x) + b \quad (3)$$

To find a linear separating hyper-plane, minimization of the error function is written:

$$\min \frac{1}{2} \|w\|^2 \quad (4)$$

subject to the constraints:

$$s.t. y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, \dots, m \quad (5)$$

where  $w$  is the vector of coefficients,  $b$  is a constant. The index  $i$  labels the  $N$  training cases. Note that  $y \in \pm 1$  represents the class labels and  $x_i$  represents the independent variables. The kernel  $\Phi$  is used to transform data from the input (independent) to the feature space.

#### 3) RANDOM FOREST (RF) [35]

RF takes a bootstrap sample from the data and fit a classification or regression tree. When constructing the decision tree, random samples of  $m$  predictors are selected from the full set of  $p$  predictors as split candidates every time the split is considered.

Each tree is built according to the following algorithm: 1.  $Y$  represents the number of training samples, and  $X$  represents the number of features. 2. Input feature number  $x$  is used to determine the decision result of a node in the decision tree; Where  $x$  should be much less than  $X$ . 3. The  $Y$  training samples are sampled with replacement  $N$  times to form a training set (bootstrap sampling), and the undrawn sample are used to make predictions and evaluate the errors. 4. For each node,  $x$  features are randomly selected, and the decision of each node in the decision tree is determined based on these features. 5. Every tree grows whole without pruning.

$$f = \frac{1}{B} \sum_{b=1}^B f_b(x) \quad (6)$$

The function  $f_b(x)$  represents the classifier of a single random forest.  $B$  is an adjustable parameter, and the best  $B$  is selected by cross-validation.

#### 4) AdaBoost [36]

AdaBoost is an ensemble method for improving the model predictions of any given learning algorithm which utilizes a set of weak learners to create a single strong learner. AdaBoost trains different classifiers through the operation of the sample set. It changes the sample weight by updating the distribution weight vector, that is, to improve the weight of high error samples, and focuses on training the error samples.

Firstly, the weight distribution of training data is initialized. If there are  $N$  samples, each training sample is initially given the same weight  $\frac{1}{N}$ .

$$D_1 = (w_{11}, w_{22}, \dots, w_{1i}, \dots, w_{1N}), \\ w_{1i} = \frac{1}{N}, \quad i = 1, 2, \dots, N \quad (7)$$

The second step is to train the weak classifier. Perform multiple iterations, using  $m = 1, 2, \dots, M$  represents the number of rounds of iteration. Learn from the training data set with weight distribution  $D_m$  to obtain the basic classifier (select the threshold with the lowest error rate to design the basic classifier)  $G_m(x) : \chi \rightarrow \{-1, +1\}$ . The classification

error rate of  $G_m(x)$  on the training data set is calculated

$$e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^N w_{mi} I(G_m(x_i) \neq y_i) \quad (8)$$

In the training process, if a sample point has been accurately classified, its weight will be reduced in the construction of the next training set. On the contrary, if a sample is not accurately classified, its weight will be improved. The formulation is:

$$\alpha_m = \frac{1}{2} \log \frac{1 - e_m}{e_m} \quad (9)$$

Then, the sample set with updated weights is used to train the next classifier, and the whole training process proceeds in such an iterative manner.

$$D_{m+1} = (w_{m+1,1}, w_{m+1,2}, \dots, w_{m+1,i}, \dots, w_{m+1,N}) \quad (10)$$

$$w_{m+1,i} = w_{mi} \exp(-\alpha_m y_i G_m(x_i)), \quad i = 1, 2, \dots, N \quad (11)$$

Finally, the weak classifiers are combined into strong classifiers.

$$f(x) = \sum_{m=1}^M \alpha_m G_m(x) \quad (12)$$

$$G(x) = \text{sign}(f(x)) = \text{sign}\left(\sum_{m=1}^M \alpha_m G_m(x)\right) \quad (13)$$

After the training of each weak classifier, the weight of the weak classifier with small error rate is increased to play a greater decisive role in the final classification function, while the weight of the weak classifier with large error rate is reduced to play a smaller decisive role in the final classification function. In other words, the weak classifier with low error rate has a large weight in the final classifier, otherwise it is small.

## 5) VOTING CLASSIFIER

To achieve better results, the classification results of these classifiers are combined to determine the classification results. For each sample, the prediction result of most classifiers is the final prediction label of the sample. We define the decision of  $t^{\text{th}}$  classifier as  $d_{t,j} \in \{0, 1\}$ ,  $t = 1, 2, 3, 4$  denote KNN, SVM, RF and AdaBoost respectively, and  $j = 1, \dots, C$ , where  $C$  is the number of classes. For each weather class, if  $t_{\text{th}}$  classifier chooses the class, then  $d_{t,j} = 1$ , otherwise,  $d_{t,j} = 0$ .

$$\sum_{t=1}^T d_{t,j} = \max_{j=1}^C \sum_{t=1}^T d_{t,j} \quad (14)$$

## IV. EXPERIMENTS

### A. DATASETS

We use three weather datasets, Multi-class Weather Image(MWI) [37], Multi-class weather dataset(MWD) [38] and self-made dataset, for performance evaluations. Each of datasets has at least four different weather conditions,

as shown in Fig 4. The following is a brief description of these three datasets:

- Multi-class weather image (MWI). MWI dataset contains 20K images obtained from many web albums and film, such as Flickr, Picasa, MojiWeather, Poco, Fengniao. Weather conditions are divided into four types, including sunny, rainy, hazy and snowy.
- Multi-class weather dataset (MWD). MWD contains 65,000 images from six common categories, including sunny, cloudy, rainy, snowy, haze and thunder weather.
- Our self-made dataset: the dataset belongs to the laboratory self-made dataset. The weather dataset includes 4 weather categories, sunny, rainy, hazy and snowy. There are 2,858 images of sunny days, 1,142 images of rainy days, 1,586 images of hazy days and 750 images of snowy days.

Note that, in the original datasets, the first dataset is imbalanced dataset, but the second and the third dataset are balanced dataset. To keep the setting from the balanced to unbalanced, we need to manually change the amount of each type of weather.

### B. EXPERIMENTAL SETTINGS

All compared methods are implemented using TensorFlow in Python. The batch size is set to 64 for GAN and VGG models. During image generation process, all the input images are resized to  $108 * 108$ , and output images are  $128 * 128$ . For KNN classifier, the number of nearest neighbors is set to 4. Euclidean distance is adopted to measure the distance. The other settings are selected by default. The number of trees set by Ntree is 50, and the other settings are default values. For AdaBoost, the number of trees for  $n$  estimators is 100.

### C. EXPERIMENTAL EVALUATIONS

The proposed method contains three novel ingredients: 1) the GAN model is able to effectively augment and balance the training dataset in classification, 2) the Edited Nearest Neighbor is utilized to eliminate the outliers samples generated by GAN method, 3) a ensemble learning method is designed to improve the generalization ability of the overall classifier. To reveal how each ingredient contributes to the performance improvement, we implemented the following four variants of the proposed method:

*Variant1 (Denoted as Unbalance)*: Three unbalanced datasets are used directly to classify without any dispose.

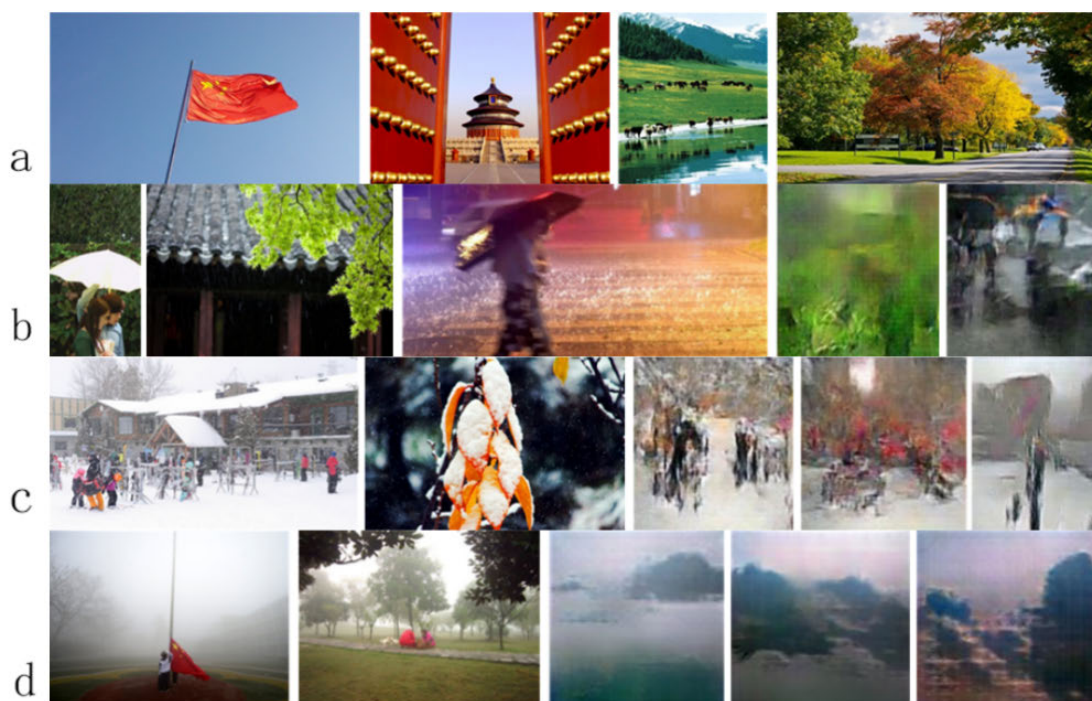
*Variant2 (Denoted as +GAN)*: We use GAN to generate data of the minority to adjust the imbalanced dataset to a new balanced dataset.

*Variant3 (Denoted as +GAN+ENN)*: Considering some data generated by GAN are unreliable, ENN is adopted to clean the data, which utilises the nearest neighbor algorithm to edit data sets and finds out those samples that are not friendly to neighbors and remove them.

*Variant4 (Denoted as +GAN+TL)*: Another method of data cleaning Tomek Link (TL) is compared with ENN.



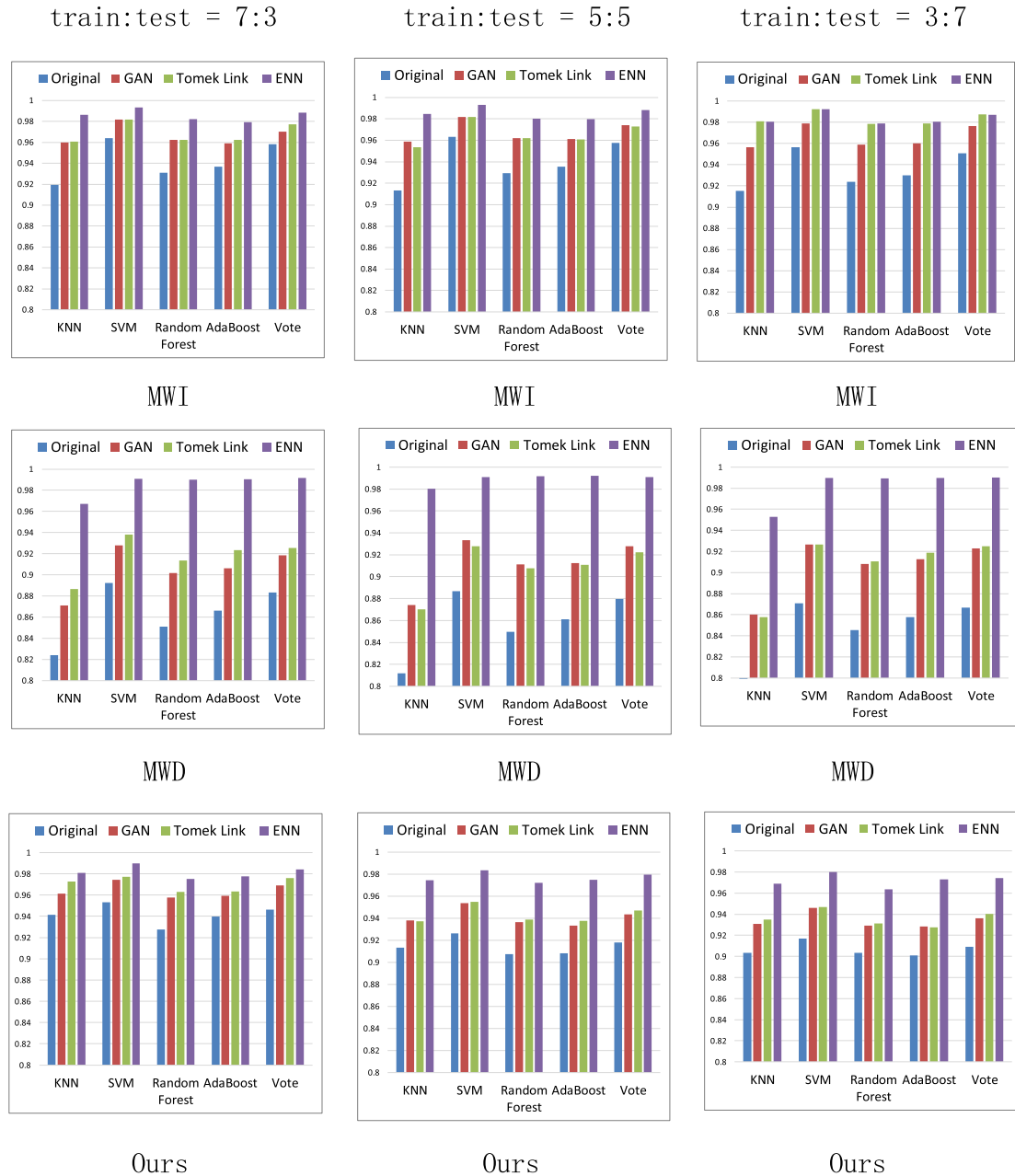
**FIGURE 4.** Tiananmen Square in different weather conditions.



**FIGURE 5.** Classification results. (a) Sunny images. The four pictures in this diagram are the original datasets. (b) Rainy images. The first three pictures in the diagram are the original dataset, and the last two pictures are generated by DCGAN. (c) Snowy images. The first two pictures in the diagram are the original pictures of the dataset, and the last three pictures are generated by DCGAN. (d) Haze images. The first two pictures in the diagram are the original pictures of the dataset, and the last three pictures are generated by DCGAN.

For TL, sample  $x$  and sample  $y$  come from different categories and satisfy the following conditions, they are called Tomek Links: There is not any sample  $z$  could make  $d(x, z) < d(x, y)$

or  $d(y, z) < d(x, y)$ . And the Tomek Links will be removed because sample  $x$  or sample  $y$  is likely to be noisy data, or the location of two samples near the category boundary.



**FIGURE 6.** The quantitative results of the three data sets on the four classifiers and ensemble learning.

Each variant is evaluated by four single learning models including KNN, SVM, RandomForest, AdaBoost and one designed ensemble learning model. Table 1, 2 and 3 show the evaluation results on the three datasets. And for each dataset, the ratio of training set and test set is set to 7:3, 5:5, 3:7. In order to clearly show the role of each part, the histogram is displayed in Figure 6.

#### 1) THE EFFECTIVENESS OF GAN

To verify the quality of the generated image, we show the synthetic images in Figure 5. As we observed, the quality of fog and haze is very high, while the weather pictures of rainy and snowy days are somewhat similar and difficult to

distinguish. These results reveal that the proposed method could mostly capture the given weather photo information. What's more, compared to training the model with original unbalanced dataset, the model with GAN can effectively improve the accuracy by creating a balanced dataset. The results indicates that it is important to make an imbalance of weather dataset to become a balanced dataset to improve the classification accuracy.

#### 2) THE EFFECTIVENESS OF DATA CLEANING

Two different methods ENN and TL are adopted to clean data. For the generated dataset, ENN removing examples whose class label differs from the class of at least half of



**TABLE 1.** The results of MWI dataset.

Method		KNN	SVM	RF	AdaBoost	Voting
7 : 3	Unbalance	0.9195	0.9640	0.9311	0.9367	0.9582
	+GAN	0.9598	0.9819	0.9623	0.9592	0.9803
	+GAN+TL	0.9608	0.9819	0.9625	0.9623	0.9873
	+GAN+ENN	0.9865	0.9875	0.9823	0.9793	<b>0.9883</b>
5 : 5	Unbalance	0.9134	0.9632	0.9295	0.9354	0.9574
	+GAN	0.9586	0.9718	0.9618	0.9561	0.9742
	+GAN+TL	0.9535	0.9726	0.9618	0.9606	0.9757
	+GAN+ENN	0.9845	0.9832	0.9801	0.9736	<b>0.9881</b>
3 : 7	Unbalance	0.9054	0.9563	0.9239	0.9299	0.9587
	+GAN	0.9504	0.9788	0.9589	0.9518	0.9714
	+GAN+TL	0.9610	0.9822	0.9783	0.9789	<b>0.9870</b>
	+GAN+ENN	0.9805	0.9823	0.9787	0.9804	0.9869

**TABLE 2.** The results of MWD dataset.

Method		KNN	SVM	RF	AdaBoost	Voting
7 : 3	Unbalance	0.8239	0.8922	0.8508	0.8661	0.8932
	+GAN	0.8712	0.9277	0.9015	0.9060	0.9284
	+GAN+TL	0.8867	0.9380	0.9135	0.9233	0.9353
	+GAN+ENN	0.9670	0.9908	0.9901	0.9905	<b>0.9916</b>
5 : 5	Unbalance	0.8117	0.8867	0.8497	0.8610	0.8797
	+GAN	0.8742	0.9334	0.9114	0.9124	0.9380
	+GAN+TL	0.8701	0.9280	0.9076	0.9109	0.9294
	+GAN+ENN	0.9802	0.9910	0.9916	0.9921	<b>0.9927</b>
3 : 7	Unbalance	0.8100	0.8710	0.8454	0.8579	0.8668
	+GAN	0.8602	0.9267	0.9082	0.9127	0.9270
	+GAN+TL	0.8577	0.9267	0.9106	0.9187	0.9281
	+GAN+ENN	0.9530	0.9875	0.9874	0.9879	<b>0.9898</b>

**TABLE 3.** The results of our self-made dataset.

Method		KNN	SVM	RF	AdaBoost	Voting
7 : 3	Unbalance	0.9415	0.9530	0.9276	0.9396	0.9463
	+GAN	0.9614	0.9742	0.9577	0.9592	0.9762
	+GAN+TL	0.9727	0.9773	0.9629	0.9634	0.9771
	+GAN+ENN	0.9810	<b>0.9897</b>	0.9753	0.9774	0.9842
5 : 5	Unbalance	0.9131	0.9261	0.9071	0.9081	0.9278
	+GAN	0.9379	0.9534	0.9364	0.9332	0.9532
	+GAN+TL	0.9371	0.9548	0.9386	0.9375	0.9569
	+GAN+ENN	0.9745	0.9832	0.9719	0.9747	<b>0.9836</b>
3 : 7	Unbalance	0.9034	0.9167	0.9033	0.9010	0.9092
	+GAN	0.9310	0.9458	0.9291	0.9283	0.9362
	+GAN+TL	0.9350	0.9468	0.9313	0.9277	0.9471
	+GAN+ENN	0.9691	0.9789	0.9634	0.9728	<b>0.9794</b>

its k nearest neighbors can get better results than TL removing unwanted overlap between classes where majority class links are removed until all minimally distanced nearest neighbor pairs are of the same class.

### 3) THE EFFECTIVENESS OF ENSEMBLE LEARNING

In this study, the effect of designed ensemble learning model voting is compared with four kinds of single classifiers, KNN, SVM, RF and AdaBoost. For the four classifiers, the SVM

algorithm is more robust when compared with other classifiers. Specially, with the voting method, the results can be further improved. This shows the designed ensemble learning method is effective to help the model to classify weather data.

### V. CONCLUSION

In this paper, an application of GAN is proposed to solve the problem of weather classification with imbalanced data. We turn the problem solution into two modules. The first

module is the design and implementation of GAN. The second module is the solution to the problem of weather classification. To alleviate the dependence on some unreliably generated images, we also introduce a data cleaning method. With these proposed methods, the performance of proposed imbalanced weather classification method gains significant improvement. In the future work, we will further develop more robust methods to ensure the high quality of the generated images for weather classification.

## REFERENCES

- [1] J. A. Nystuen and H. D. Selsor, "Weather classification using passive acoustic drifters," *J. Atmos. Ocean. Technol.*, vol. 14, no. 3, pp. 656–666, Jun. 1997.
- [2] R. Schnur and D. P. Lettenmaier, "A case study of statistical downscaling in australia using weather classification by recursive partitioning," *J. Hydrol.*, vols. 212–213, pp. 362–379, Dec. 1998.
- [3] C. Lu, D. Lin, J. Jia, and C.-K. Tang, "Two-class weather classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3718–3725.
- [4] M. Elhoseiny, S. Huang, and A. Elgammal, "Weather classification with deep convolutional neural networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3349–3353.
- [5] Z. Zhang and H. Ma, "Multi-class weather classification on single images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 4396–4400.
- [6] A. Lingua, D. Marenchino, and F. Nex, "Performance analysis of the SIFT operator for automatic feature extraction and matching in photogrammetric applications," *Sensors*, vol. 9, no. 5, pp. 3745–3766, 2009.
- [7] P. Yildirim, "Chronic kidney disease prediction on imbalanced data by multilayer perceptron: Chronic kidney disease prediction," in *Proc. IEEE 41st Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2017, pp. 193–198.
- [8] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, Boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
- [9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, X. Bing, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.
- [10] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: <http://arxiv.org/abs/1411.1784>
- [11] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.
- [12] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972.
- [13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
- [14] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.
- [15] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Springer, 2005, pp. 878–887.
- [16] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Inf. Sci.*, vol. 291, pp. 184–203, Jan. 2015.
- [17] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5375–5384.
- [18] C. Drummond and R. C. Holte, "C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *Proc. ICMLW*, 2013, pp. 1–8.
- [19] C.-M. Vong, W.-F. Ip, P.-K. Wong, and C.-C. Chiu, "Predicting minority class for suspended particulate matters level by extreme learning machine," *Neurocomputing*, vol. 128, pp. 136–144, Mar. 2014.
- [20] Z. Chen and C. Jiang, "Building occupancy modeling using generative adversarial network," *Energy Buildings*, vol. 174, pp. 372–379, Sep. 2018.
- [21] J. Yang, A. Kannan, D. Batra, and D. Parikh, "LR-GAN: Layered recursive generative adversarial networks for image generation," 2017, *arXiv:1703.01560*. [Online]. Available: <https://arxiv.org/abs/1703.01560>
- [22] Y. Cao, Z. Gan, B. Jia, R. A. Evans, and M. Gu, "High-photosensitive resin for super-resolution direct-laser-writing based on photoinhibited polymerization," *Opt. Express*, vol. 19, no. 20, p. 19486, Sep. 2011.
- [23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.
- [24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.
- [25] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for Image-to-Image translation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2849–2857.
- [26] H. Dong, P. Neekhara, C. Wu, and Y. Guo, "Unsupervised Image-to-Image translation with generative adversarial networks," 2017, *arXiv:1701.02676*. [Online]. Available: <http://arxiv.org/abs/1701.02676>
- [27] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain Image-to-Image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.
- [28] Y. Wang and H. Li, "A novel intelligent modeling framework integrating convolutional neural network with an adaptive time-series window and its application to industrial process operational optimization," *Chemometric Intell. Lab. Syst.*, vol. 179, pp. 64–72, Aug. 2018.
- [29] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.
- [30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: <http://arxiv.org/abs/1511.06434>
- [31] R. Palaniappan, K. Sundaraj, and S. Sundaraj, "A comparative study of the svm and k-nn machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals," *BMC Bioinf.*, vol. 15, no. 1, p. 223, Dec. 2014.
- [32] A. J. Wyner, M. Olson, J. Bleich, D. Mease, "Explaining the success of adaboost and random forests as interpolating classifiers," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1558–1590, 2017.
- [33] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.
- [34] G. Madzarov, D. Gjorgjevikj, and I. Chorbev, "A multi-class SVM classifier utilizing binary decision tree," *Int. J. Comput. Inform.*, vol. 33, no. 2, pp. 225–233, Jan. 2009.
- [35] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, Jan. 2005.
- [36] K.-Y. Feng, Y.-D. Cai, and K.-C. Chou, "Boosting classifier for predicting protein domain structural class," *Biochem. Biophys. Res. Commun.*, vol. 334, no. 1, pp. 213–217, Aug. 2005.
- [37] Z. Zhang, H. Ma, H. Fu, and C. Zhang, "Scene-free multi-class weather classification on single images," *Neurocomputing*, vol. 207, pp. 365–373, Sep. 2016.
- [38] D. Lin, C. Lu, H. Huang, and J. Jia, "RSCM: Region selection and concurrency model for multi-class weather recognition," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4154–4167, Sep. 2017.



**YANGRU HUANG** received the B.S. degree from Shanxi Agricultural University, Shanxi, China, in 2017. She is currently pursuing the master's degree with the School of Computer Science and Information Technology, Beijing Jiaotong University. Her research interests include computer vision and machine learning.



include computer vision, pattern recognition, image processing, and machine learning.

**YI JIN** (Member, IEEE) received the Ph.D. degree in signal and information processing from the Institute of Information Science, Beijing Jiaotong University, Beijing, China, in 2010. She was a Visiting Scholar with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2013 to 2014. She is currently an Associate Professor with the School of Computer Science and Information Technology, Beijing Jiaotong University. Her research interests



He is currently the Program Director with the Centre for Bio Devices and Signal Analysis, NTU. He served as the Chair of the IEEE CAS Singapore Chapter, from 2007 to 2008 and in 2019.

**ZHIPING LIN** (Senior Member, IEEE) received the B.Eng. degree in control engineering from the South China Institute of Technology, Canton, China, in 1982, and the Ph.D. degree in information engineering from the University of Cambridge, England, in 1987. He was with the University of Calgary, Canada, Shantou University, China, and DSO National Laboratories, Singapore, from 1987 to 1999. Since 1999, he has been with Nanyang Technological University



His researches focus on high-performance computing, privacy protection and information security, social network analysis, and parallel and distributed computing.

**YIDONG LI** received the Ph.D. degree in computer science from The University of Adelaide, in 2011. He is currently a Ph.D. Tutor and the Vice President of the School of Computer and Information Technology, Beijing Jiaotong University. His researches focus on high-performance computing, privacy protection and information security, social network analysis, and parallel and distributed computing.

...