A Semisupervised Deep Learning Framework For Tropical Cyclone Intensity Estimation

Guangchen Chen[1, 2], Zhao Chen[1*], Feng Zhou[1], Xingxing Yu[1], He Zhang[1], Lingyun Zhu[2]
1. School of Computer Science and Technology, Donghua University, Shanghai, China
2. College of Information Science and Technology, Donghua University, Shanghai, China
chenzhao@dhu.edu.cn

***Abstract*—Tropical cyclone intensity estimation is important to catastrophic weather forecast. In this paper, it is treated as a classification task, with the intensity categories as class labels. Normally, traditional supervised methods require a large amount of prior knowledge for training. However, in reality, only a small amount of labeled samples can be available. Therefore, this paper proposes a novel semisupervised deep learning framework based on convolutional neural networks (CNNs) for FY-4 multispectral images (MSI). The new model only needs a small set of samples labeled *a priori* to accurately classify the images and estimate cyclone intensity. Moreover, the model involves an iterative training set update process with a hybrid similarity measurement especially designed for the task. The experiments show that the classification performance of the network is improved during the iterations. Evaluation on the estimated intensity categories indicate that the proposed method is significantly better than several existing methods, including the state-of-the-art cyclone intensity estimation model based on CNN, while small training sets are used.**

Keywords—semisupervised, classification, convolutional neural network (CNN), tropical cyclone, intensity estimation

I. Introduction

Tropical cyclones of high intensity can cause serious damages to coastal areas. Therefore, it is crucial to accurately analyze remote sensing images of cyclones and estimate cyclone intensity in advance. Cyclone intensity estimation can be addressed as a classification problem and solved by learning machines. For example, Multiple Logistic Regression (MLR), Support Vector Machine (SVM) and Back-Propagation Neural Network (BPNN) have been employed and achieved desirable performance on multispectral images (MSI) of cyclones [1]. As a powerful classifier, convolutional neural network (CNN) [2] has been intensively studied in the past years. Many variations of CNN (e.g., LeNet [3], GoogLeNet [4] and ResNet [5]) were proposed, making remarkable achievements in a wide range of vision tasks. Recently, CNN has been successfully applied to cyclone intensity estimation for infrared (IR) images [6]. However, CNN as well as most supervised methods in existence have a problem in common that they require large training sets. Unfortunately, correctly labeled samples are hard to come by, especially in the case of cyclone intensity estimation based on MSIs acquired by China's No. 4 meteorological satellite (FY-4). As this satellite was launched in December 2016, the FY-4 MSIs have not yet been intensively studied. Due to the complexity of cyclone image features in MSIs, it is difficult for unsupervised methods to achieve satisfactory classification accuracy since these methods are purely data-driven. Therefore, neither supervised nor unsupervised methods are very suitable for cyclone intensity estimation. Although labeled data are scarce, the amount of unlabeled images is enormous with rich information, which allows for semisupervised classification to deal with the problems.

Therefore, we propose a semisupervised deep learning framework for tropical cyclone classification, embedded with a specially designed sample-wise similarity measurement. It mainly consists of three parts, feature extraction, semisupervised CNN, and training set update. For efficiency, spectral feature extraction is applied to the original FY-4 MSI. Initially, the labeled samples are used for training a vanilla CNN, denoted as CNN0. Then each unlabeled sample has a class label predicted by CNN0. Given the classification result, similarity between every sample with a prior label and every sample with a predicted label of the same class are measured and sorted in descending order. The samples with the predicted labels of large similarity values are considered reliable and added to the present training set, which is hence updated. After that, another CNN, transferred from CNN0 and referred to as CNN1, is constructed and fine-tuned by the updated training set. Then the optimized CNN1 is used for classification of the present unlabeled set, followed by the training set update operations defined previously. The aforementioned procedure is repeated until all the samples are classified, i.e., each set of the original cyclone MSI assigned to an intensity category. It should be noted that the proposed framework is semisupervised, as both labeled samples and unlabeled samples are engaged in training.

The contributions of our works are summarized as follows. 1) A novel semisupervised deep learning framework based on CNN is proposed. It only requires a small amount of training samples for accurate cyclone intensity estimation. 2) A hybrid similarity measurement is designed and employed within the framework for training set update. It can reflect different aspects of the features. The efficacy of the proposed framework is demonstrated by adequate experiments. The results show that the proposed framework significantly outperforms several methods for cyclone intensity estimation, including the state-of-the-art model based on CNN [6], while only a few samples are labeled *a priori*.

II. Related Works

CNN is a powerful tool for pattern recognition, which has developed in recent years, attracting widespread attention. The CNN architecture mainly includes convolution layers, pooling layers, fully connected (FC) layers, and softmax layer, as indicated in the graphical illustration of the architecture of vanilla CNN in Fig. 1.

The convolutional layers first extract features from the input images. The pooling layers help reduce computation and control overfitting. The role of the fully connected layers

This research is financially supported by grants from the National Key R&D Program of China (No. 2017YFB0309800), the National Natural Science Foundation of China (No. 61702094), the Dawn Project of Shanghai Municipal Education Commission (No. 18CG38), the Young Scientists' Sailing Project of Science and Technology Commission of Shanghai Municipal (No. 17YF1427400) and the Fundamental Research Funds for the Central Universities (No. 17D111206). *The corresponding author is Zhao Chen.

is to pull the feature maps extracted from the convolutional layer or the pooling layer into a vector, and obtain the category label through the softmax layer. As indicated by Fig. 1, the vanilla CNN can be directly applied to cyclone intensity estimation, with cyclone images as the input and intensity categories as the output.

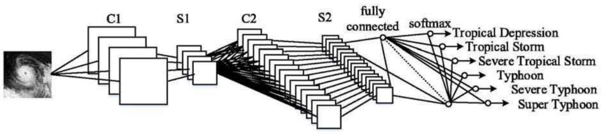


Fig. 1. Vanilla CNN for cyclone intensity estimation, where C stands for convolutional layers and S represents pooling layers.

III. THE PRORPSED FRAMEWORK

In this section, we introduce the proposed semisupervised deep learning framework for tropical cyclone classification. The proposed framework shown in Fig. 2 mainly consists of three modules: feature extraction, semisupervised CNN and training set update.

A. Feature Extraction

To reduce the size of network and boost efficiency, feature extraction is applied to the labeled samples $\mathbf{X}_{IL} \in \mathbb{R}^{S_1 \times S_2 \times B \times N_L}$ and the unlabeled samples $\mathbf{X}_{IU} \in \mathbb{R}^{S_1 \times S_2 \times B \times N_U}$, where $S_1 \times S_2$ is the spatial size, B is the number of bands, N_L is the number of the sets of MSIs labeled *a priori,* and N_U is the number of unlabeled sets of MSIs. Here, principal component analysis (PCA) is chosen as the method for the feature extraction. As results, unlabeled and labeled feature maps i.e. $\mathbf{X}_L \in \mathbb{R}^{S_1 \times S_2 \times N_L}$ and $\mathbf{X}_U \in \mathbb{R}^{S_1 \times S_2 \times N_U}$ are obtained through PCA.

B. Semisupervised CNN

The features extracted previously are fed to the proposed semisupervised CNN, which is designed to exploit limited amount of prior knowledge and abundant information in unlabeled samples. The network comprises two parts, CNN0 and CNN1. CNN0 is for initialization of the predicted labels of $\mathbf{X}_U$ and the parameters of CNN1.

CNN0 consists of two convolutional layers, two pooling layers, one fully connected layer and one softmax layer. It is trained with $\{\mathbf{X}_L, \mathbf{Y}_L\}$, where $\mathbf{Y}_L \in \mathbb{R}^{N_L}$ are the labels of $\mathbf{X}_L$. The unlabeled features $\mathbf{X}_U$ are then fed to the trained CNN0, where they are assigned to initial predicted labels. Due to the limited number of training samples, many initially predicted labels of $\mathbf{X}_U$ may be wrong. Therefore, some feature maps with reliable labels which is denoted as $\mathbf{X}_R \in \mathbb{R}^{S_1 \times S_2 \times N_R}$ (N_R is the number of selected feature maps) are selected from the $\mathbf{X}_U$ by the similarity selection mechanism which will be introduced in the next section. The reliable labels of the selected feature maps are denoted as $\mathbf{Y}_R \in \mathbb{R}^{N_R}$. The selected feature maps with their reliable labels which are denoted as $\{\mathbf{X}_R, \mathbf{Y}_R\}$ are added to $\{\mathbf{X}_L, \mathbf{Y}_L\}$.

After label initialization by CNN0, CNN1 is activated. The CNN1 is adapted from CNN0 with three new fully connected layers added before the softmax layer. The parameters of CNN1 are initialized by those of CNN0. The three new FC layers of CNN1 are fine-tuned by $\{\mathbf{X}_L, \mathbf{Y}_L\}$ produced by CNN0. After that, the remaining unlabeled samples are fed to CNN1 and have their labels predicted. Then new $\{\mathbf{X}_R, \mathbf{Y}_R\}$ are selected from the $\mathbf{X}_U$ by the same similarity selection mechanism, which are added to $\{\mathbf{X}_L, \mathbf{Y}_L\}$. Once the training set is updated, CNN1 is optimized. Likewise, CNN1 undergoes an iteration of fine-tuning every time $\{\mathbf{X}_L, \mathbf{Y}_L\}$ is updated, until every originally unlabeled cyclone sample is assigned to an intensity category.

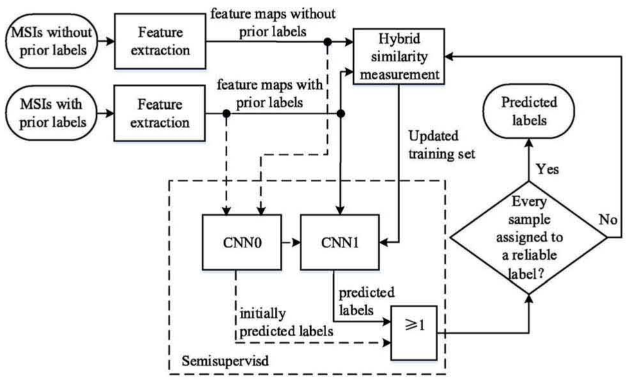


Fig. 2. Architecture of the proposed framework

It can be gathered that selection of new $\{\mathbf{X}_R, \mathbf{Y}_R\}$ using the prediction of the fine-tuned CNN1 and optimization of the CNN1 by the updated $\{\mathbf{X}_L, \mathbf{Y}_L\}$ are repeated alternatively until $\mathbf{X}_U$ is an empty set. To prevent the network from overfitting and increase its generalization ability, the dropout technique is applied to the extra fully connected layers [7]. Stochastic gradient descent (SGD) method is used for the network training.

C. Training Set Update

The above analysis reveals that $\{\mathbf{X}_L, \mathbf{Y}_L\}$ is updated iteratively based on the similarity between $\mathbf{X}_L(:,:,n_L)$ ($n_L = 1, 2, \ldots, N_L$) and $\mathbf{X}_U(:,:,n_U)$ ($n_U = 1, 2, \ldots, N_U$). According to the characteristics of the cyclone remote sensing images, histogram distance and Euclidean distance are combined and used as a hybrid similarity measurement.

In the proposed framework, the training sample selection begins with sorting $\mathbf{X}_U(:,:,n_U)$ with their predicted labels in descending order of the hybrid similarity, whereas the predicted labels are yielded by the trained CNN1. The number of the feature maps to be selected is fixed to N_R *a priori*. Only top N_R of all the sorted $\mathbf{X}_U(:,:,n_U)$ are selected as $\mathbf{X}_R$ while their corresponding labels $\mathbf{Y}_R$ predicted by CNN1 are considered reliable for the next round of training and optimization of CNN1. Hence, $\{\mathbf{X}_R, \mathbf{Y}_R\}$ is added to the present training set $\{\mathbf{X}_L, \mathbf{Y}_L\}$, which completes the training set update in the current iteration. In the meantime, $\mathbf{X}_R$ are removed from of $\mathbf{X}_U$ and will not be classified ever again.

As summary, the algorithm of the proposed semisupervised framework for tropical cyclone classification is given by TABLE I. It should be noted that $\{\mathbf{Y}_U\}_t \in \mathbb{R}^{N_U}$ are the final predicted labels of $\mathbf{X}_U$.

TABLE I. TRAINING PROCEDURES OF THE SEMISUPERVISED DEEP LEARNING FRAMEWORK AFTER FEATURE EXTRACTION

Input: $\{\mathbf{X}_L, \mathbf{Y}_L\}$, $\mathbf{X}_U$, and t (the iteration of semisupervised CNN)

Step 1: Initialization

t =0, $\{\mathbf{X}_L, \mathbf{Y}_L\}_t \leftarrow \{\mathbf{X}_L, \mathbf{Y}_L\}$, $\{\mathbf{X}_U\}_t \leftarrow \{\mathbf{X}_U\}$.

Step 2: Similarity Computation

Compute histogram distance and Euclidean distance between $\{\mathbf{X}_L(:,:,n_L)\}_t$ and $\{\mathbf{X}_U(:,:,n_U)\}_t$.

Combine the histogram distance and Euclidean distance into the hybrid measurement of the similarity between $\{\mathbf{X}_L(:,:,n_L)\}_t$ and $\{\mathbf{X}_U(:,:,n_U)\}_t$.

Step 3: CNN0

Train CNN0 with $\{\mathbf{X}_L, \mathbf{Y}_L\}_t$.

Predict the labels of $\{\mathbf{X}_U\}_t$ by CNN0.

Sort $\{\mathbf{X}_U\}_t$ in descending order of the hybrid similarity measurements.

Select the top N_R feature maps of $\{\mathbf{X}_U\}_t$ and denote them as $\{\mathbf{X}_R\}_t$.

Denote the labels of $\{\mathbf{X}_R\}_t$ as $\{\mathbf{Y}_R\}_t \in \mathbb{R}^{N_R}$.

Let $\{\mathbf{X}_L, \mathbf{Y}_L\}_{t+1} \leftarrow \{\mathbf{X}_L, \mathbf{Y}_L\}_t \cup \{\mathbf{X}_R, \mathbf{Y}_R\}_t$, $\{\mathbf{X}_U\}_{t+1} \leftarrow \{\mathbf{X}_U\}_t - \{\mathbf{X}_R\}_t$, $\{\mathbf{Y}_U\}_{t+1} \leftarrow \{\mathbf{Y}_R\}_t$, $t \leftarrow t+1$.

Step 4: CNN1

While $\{\mathbf{X}_U\}_t \neq \varnothing$, **do.**

Initialize CNN1 by the parameters of CNN0.

Fine-tune the three new FC layers of CNN1 with $\{\mathbf{X}_L, \mathbf{Y}_L\}_t$.

Predict the labels of $\{\mathbf{X}_U\}_t$ by the fine-tuned CNN1.

Sort $\{\mathbf{X}_U\}_t$ in descending order of the hybrid similarity measurements.

Select the top N_R feature maps of $\{\mathbf{X}_U\}_t$ and denote them as $\{\mathbf{X}_R\}_t$.

Denote the labels of $\{\mathbf{X}_R\}_t$ as $\{\mathbf{Y}_R\}_t \in \mathbb{R}^{N_R}$.

Let $\{\mathbf{X}_L, \mathbf{Y}_L\}_{t+1} \leftarrow \{\mathbf{X}_L, \mathbf{Y}_L\}_t \cup \{\mathbf{X}_R, \mathbf{Y}_R\}_t$, $\{\mathbf{X}_U\}_{t+1} \leftarrow \{\mathbf{X}_U\}_t - \{\mathbf{X}_R\}_t$, $\{\mathbf{Y}_U\}_{t+1} \leftarrow \{\mathbf{Y}_R\}_t$, $t \leftarrow t+1$.

end

Output: $\{\mathbf{Y}_U\}_t$

IV. EXPERIMENTS

In this section, the dataset which is used in experiment and setup of experiment is introduced, the analysis of the results is also presented.

A. Dataset Description

The tropical cyclone images used in our experiments are originally collected by the FY-4 meteorological satellite over the past two years and provided by Shanghai Institute of Meteorological Science. After pre-processing such as cropping and data augmentation, we have obtained 5243 sets of MSIs, each depicting a cyclone with 14 bands and 240×240 pixels at a spatial resolution of 4000m. We have also gathered the sample labels, i.e., cyclone intensity categories, from the track data of the cyclones published online (http://en.weather.com.cn/).

TABLE II. WIND SPEED SCALES AND RELATED CATEGORIES OF TROPICAL CYCLONES

Tropical cyclone speed (m/s)	Category	Sample Number (Total: 5243)
10.8-17.1	Tropical Depression	155
17.2-24.4	Tropical Storm	2766
24.5-32.6	Severe Tropical Storm	692
32.7-41.4	Typhoon	588
41.5-50.9	Severe Typhoon	708
⩾51.0	Super Typhoon	334

According to reference [8], there are 6 categories of cyclone intensity. Speed scales, category names and sample numbers are shown in TABLE II. The 14th band images of some exemplary tropical cyclones are shown in Fig. 3.

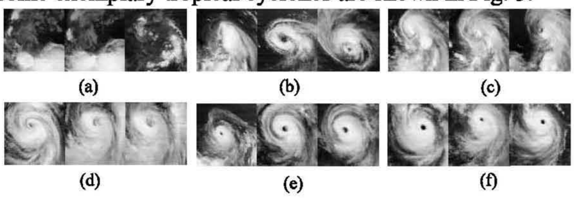


Fig. 3. The 14th band images (240×240) of exemplary tropical cyclones selected from each of the following categories, (a) Tropical Depression, (b) Tropical Storm, (c) Severe Tropical Storm, (d) Typhoon, (e) Severe Typhoon and (f) Super Typhoon.

B. Setup

For dimensionality reduction and efficiency boost, the tropical cyclone features are extracted by PCA and used as the initial inputs to the proposed network. The image features are divided into two parts: $\{\mathbf{X}_L, \mathbf{Y}_L\}$ and $\mathbf{X}_U$. Initial training samples $\mathbf{X}_L$ are randomly selected from the feature set with a proportion ranging from 5%, 10%, 15%, 20%, 25% to 30%. They are labeled *a priori*. In selection of high-similarity samples for training set update, N_R is set as 5% of the total number of samples (round-off).

TABLE III. CONFIGURATIONS OF THE PROPOSED NETWORKS

Layer	Input size	Output size	Kernel size
C1	[240,240,1]	[232,232,4]	[9,9]
S1	[232,232,4]	[58,58,4]	[4,4]
C2	[58,58,4]	[50,50,16]	[9,9]
S2	[50,50,16]	[10,10,16]	[5,5]
F1	[10,10,16]	[1600]	[10*10*16,1600]
F2	[1600]	[800]	[1600,800]
F3	[800]	[400]	[800,400]
F4	[400]	[200]	[400,200]
softmax	[200]	[6]	[200,6]

In the proposed model, CNN0 comprises two convolutional filers (C1 and C2) and two pooling filers (S1 and S2) and one fully connected layer (F1) followed by softmax. CNN1 is transferred from the architecture of CNN0 removed of its original softmax layer but concatenated by three new fully connected layers (F2, F3 and F4) followed by a softmax layer. The configurations of CNN0 and CNN1 is given by TABLE III. Throughout the training process, the probability of dropout is set as 0.5 and the Sigmoid function is adopted wherever nonlinear mapping is needed.

The proposed model is compared with several well-established classifiers, including CNN, SVM, BPNN, k-Nearest Neighbors (k-NN) and MLR. Here, CNN can be considered as a state-of-the-art model as it is been

TABLE IV. CLASSIFICATION RESULTS FOR DIFFERENT METHODS

Methods	5% labeled		10% labeled		15% labeled		20% labeled		25% labeled		30% labeled	
	OA (%)	*Kappa*	*OA (%)*	*Kappa*	*OA (%)*	*Kappa*	*OA (%)*	*Kappa*	*OA (%)*	*Kappa*	*OA (%)*	*Kappa*
SVM	67.13	0.504	71.48	0.566	76.89	0.646	80.11	0.692	84.46	0.760	86.61	0.792
BPNN	69.48	0.553	72.54	0.587	77.77	0.664	82.90	0.738	86.47	0.791	88.66	0.827
MLR	63.49	0.470	67.41	0.514	73.17	0.590	77.77	0.660	81.94	0.723	83.36	0.744
k-NN	54.08	0.211	62.79	0.379	70.36	0.503	74.71	0.590	77.66	0.645	80.46	0.690
CNN	64.62	0.471	78.53	0.680	82.90	0.739	86.29	0.789	88.76	0.826	90.16	0.849
The Proposed	**77.05**	**0.649**	**88.92**	**0.830**	**94.68**	**0.918**	**94.56**	**0.916**	**95.59**	**0.932**	**96.69**	**0.949**

successfully applied to cyclone intensity estimation with infrared images only recently [6]. In our experiment, the results of intensity categorization are evaluated by overall accuracy (OA) and Kappa coefficient (Kappa). All the algorithms are implemented by MATLAB 2016 running on a computer with Intel Xeon E5-2624 CPU and 256G RAM.

C. Result Analysis

From TABLE IV, it can be seen that the proposed semisupervised framework largely outperforms all the other methods in comparison, including the state-of-the-art CNN. Its OA is at least 6 percentage points higher than that of CNN under all the circumstances. It validates the efficacy of the proposed method, which can indeed stand the scarcity in prior knowledge.

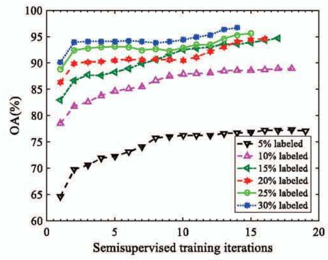
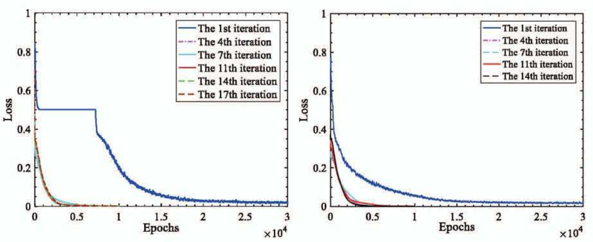


Fig. 4. Classification accuracy against semisupervised iterations as the proportion of the samples with prior labels goes up from 5% to 30%.



(a) (b)

Fig. 5. Loss against training epochs in different semisupervised iterations while the proportion of the samples with prior labels is equal to a) 15% or b) 30%.

Fig. 4 shows the training process of the proposed method. It can be observed that overall accuracy increases with the semisupervised training iterations, whereas CNN1 is fine-tuned and the training set is updated alternatively. It demonstrates that the design of training sample selection with the hybrid similarity measurement is successful.

Moreover, TABLE IV and Fig. 4 show that the final OA increases with the amount of the labeled samples. It suggests that the proposed model functions stably as the labeled data size goes up. TABLE IV and Fig. 4 also indicate that the increment of OA slows down when the proportion of the labeled samples passes 15%. It means that when the labeled set reaches a certain size, the information it can provide for classification tends to be saturated. Meanwhile, there may be a few cyclone images whose patterns cannot be effectively extracted or classified by the proposed model due to severe intra-class heterogeneity and/ or inter-class homogeneity.

Fig. 5 demonstrates the convergence of the proposed semisupervised network whereas the proportion of the samples with prior labels is 15% or 30%. The loss gradually converges as all the unknown samples have their labels predicted and updated iteratively. Therefore, the proposed model is not only accurate; it is also a reliable and functional model for cyclone intensity estimation.

V. CONCLUSION

In this paper, we propose a new semisupervised deep learning framework for accurate tropical cyclone classification, which only requires a small number of labeled samples. The proposed method updates training set by selecting the samples with reliably predicted labels based on a specially designed hybrid similarity measurement. It trains two CNNs in a semisupervised and iterative manor. Experiments show that the proposed method is significantly better than several popular classification methods. In the future, we will construct semisupervised regression network which can better characterize cyclone patterns and directly estimate the wind speed of tropical cyclones.

ACKNOWLEDGMENT

The authors thank the Shanghai Meteorological Service for providing the FY-4 data.

REFERENCES

[1] Z. Chen, X. Yu, G. Chen and J. Zhou, "Cyclone Intensity Estimation using Multispectral Imagery from FY-4 Satellite," *International conference on Audio, Language and Image Processing (ICALIP)*, Shanghai, China, 2018.

[2] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, December, 1989, pp. 541-551.

[3] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, November, 1998, pp. 2278-2324.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, June, 2015.

[5] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, June, 2016

[6] R. Pradhan, R. Aygun, M. Maskey, R. Ramachandran, and D. Cecil, "Tropical Cyclone Intensity Estimation Using a Deep Convolutional Neural Network," *IEEE Transactions on Image Processing*, vol. 27, no. 2, October, 2017, pp. 692-702.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, 2014, pp. 1929-1958.

[8] Grade of tropical cyclones, GB/T 19201, 2006.