Deep Learning for Typhoon Intensity Classification Using Satellite Cloud Images

ZONGSHENG ZHENG,1,a CHENYU HU,1,a ZHAORONG LIU,1,a JIANBO HAO,1,a QIAN HOU,1,a AND XIAOYI JIANG2

a College of Information and Science, Shanghai Ocean University, Shanghai, China
b International Centre for Marine Research, Shanghai Ocean University, Shanghai, China
c National Marine Information Center, Tianjin, China

(Received 21 December 2019, in final form 5 May 2021)

ABSTRACT: A tropical cyclone, also known as a typhoon, is one of the most destructive weather phenomena. Its intense cyclonic eddy circulations often cause serious damage to coastal areas. Accurate classification or prediction of typhoon intensity is crucial to disaster warning and mitigation management. But typhoon intensity-related feature extraction is a challenging task as it requires significant preprocessing and human intervention for analysis, and its recognition rate is poor due to various physical factors such as tropical disturbance. In this study, we built a Typhoon-CNNs framework, an automatic classifier for typhoon intensity based on a convolutional neural network (CNN). The Typhoon-CNNs framework utilized a cyclical convolution strategy supplemented with dropout zero-set, which extracted sensitive features of existing spiral cloud bands (SCBs) more effectively and reduces the overfitting phenomenon. To further optimize the performance of Typhoon-CNNs, we also proposed the improved activation function (T-ReLU) and the loss function (CE-FMCE). The improved Typhoon-CNNs was trained and validated using more than 10,000 multiple sensor satellite cloud images from the National Institute of Informatics. The classification accuracy reached to 88.74%. Compared with other deep learning methods, the accuracy of our improved Typhoon-CNNs was 7.43% higher than ResNet50, 10.27% higher than InceptionV3, and 14.71% higher than VGG16. Finally, by visualizing hierarchic feature maps derived from Typhoon-CNNs, we can easily identify the sensitive characteristics such as typhoon eyes, dense-shadowing cloud areas, and SCBs, which facilitate classifying and forecasting typhoon intensity.

KEYWORDS: Classification; Deep learning; Neural networks

1. Introduction

A mature tropical cyclone (TC), also known as a typhoon or hurricane, is an intense mesoscale vortex characterized by a calm eye and a ring of violent winds in the eyewall (Frank 1977). A TC is one of the most severe weather systems on Earth that usually brings destructive winds and extreme rainfall, causes strong surges and floods, and often results in loss of life and damage to property. According to the statistics, more than 90% of marine catastrophes are caused by such meteorological disasters. Currently, typhoon detection, monitoring, and forecasting still remain a challenging task. Over the past a few decades, meteorological information about TCs has been extensively conducted with the utilization of various instruments such as aircrafts, dropsondes, and Doppler radar wind profilers (Powell 1980; Weatherford and Gray 1988; Aberson et al. 2006; Giammanco et al. 2012; Xie et al. 2013). The predicted weather condition of a TC was calculated using mathematical modeling software and meteorological data. Such a process-based numerical prediction model is usually driven by typhoon-characteristics information such as typhoon locations, central atmospheric pressures, and maximum wind speeds near typhoon centers. Even with modern high performance computing (HPC) facilities, it is still computationally expensive to run those models, and the accuracy of the prediction is heavily affected by the initial and boundary conditions (Bender et al. 2007; Jiménez et al. 2007; Hsiao et al. 2009).

Remote sensing technologies have great potential for providing spatial and temporal observations of environmental variables on large scales (Tralli et al. 2005; Handcock et al. 2006). Optical remote sensing data, in particular, provide better repeated spatial coverages than in situ surface meteorological station data, offering efficient monitoring of TCs (Hasler et al. 1983; Wu et al. 2003; Bell and Montgomery 2008; Liu et al. 2010; Zhang et al. 2014). In the intertropical convergence zone (ITCZ), the typhoon intensity and the cloud turbulence degree have a very strong correlation, which is demonstrated as the spiral cloud bands (SCBs) in satellite remote sensing images (Montgomery and Kallenbach 2010). Extracting intensity-associated features of SCBs provides an alternative approach to classifying and predicting TC. The most mature Dvorak analysis for assessing tropical cyclone intensity utilizes empirical regulations and constraint conditions on features of cloud system structures, by using specific parameters in satellite cloud images such as data T-number (DT), model expected T-number (MET), and pattern T-number (PT) (Dvorak 1975; Velden et al. 2006). But this method demands abundant domain knowledge for feature extraction and hardly captures structural information about typhoon inner cores. Currently, the implicit intensity-associated features of SCBs are extracted manually; therefore, there is an urgent need to develop methods to automatically classify intensity-based typhoons.

Researchers have already applied machine learning (ML) algorithms in the field of marine meteorology, such as support
vector machine (SVM), extreme learning machine (ELM), and back propagation (BP) to analyze meteorological cloud images (L. Li et al. 2015; P. Li et al. 2015; Xia et al. 2015). Traditional ML algorithms usually achieve unsatisfactory classification results from sophisticated intensity-associated typhoon information about satellite cloud images. A subfield of machine learning—deep learning (DL), as an extremely generalized approach—has been applied to various domains, such as pattern recognition, computer vision, artificial intelligence, and so on (Xiao et al. 2010; Hinton et al. 2012; LeCun et al. 2015). As one of the most popular deep learning models, convolutional neural networks (CNNs) have been demonstrated as a promising tool for image classification due to its locally connected layers and feasibility to integrate with local features; it outperforms the most traditional techniques mentioned above with a great margin of image classification and scene recognition (Taigman et al. 2014; Ren et al. 2015). In the hierarchical structure of CNNs, lower layers that involve series of multiscale convolutional filters learn low-level features such as edges, textures, and colors from raw data automatically and form more abstract concepts hierarchically along feed-forward layers, synthesizing semantic representation of input data at the highest layer eventually (Zeiler and Fergus 2013). Several studies have employed a conventional CNN model or multidimensional CNN (2D/3D) to estimate tropical cyclone intensity, which obtains a small root-mean-square error (RMSE; Rodés-Guirao 2019; Chen et al. 2019; Wimmers et al. 2019; Lee et al. 2020). These CNN-based approaches are applied to intensity-based typhoon imagery classification and focused on the application and improvement in network structure. However, there is a lack of in-depth exploration on the internal mathematical operation mechanism of CNNs, which could theoretically improve the model accuracy to some extent.

In this paper, in order to explore inner physical mechanisms of CNNs in the typhoon intensity prediction, we build a Typhoon-CNNs architecture as a discriminating and automatic feature descriptor to classify and predict typhoon intensity using satellite cloud images. Based on the traditional structure of CNNs, the framework proposed the cyclic convolution strategy of a multiscale filter [section 2b(1)] to extract the representative hidden features of SCBs. Meanwhile, we focus on the activation function of network neurons and the cross-entropy loss functions based on the minimum classification error (MCE). To further enhance the performance of Typhoon-CNNs, we optimize the selection of network hyperparameters (such as the scale of filters, the rate of dropout zero-set, etc.) related to the design of typhoon CNNs. The improved Typhoon-CNNs is trained and validated using more than 10,000 multiple sensor satellite cloud images from the National Institute of Informatics (NII). Finally, the image sensitive areas that affect the intensity of the typhoon were visualized and the experimental results were also analyzed.

### 2. Materials and methods

#### a. Data acquisition and augmentation

Satellite remote sensing images for experiments are provided by NII, which provides real-time meteorological satellite data concerning typhoons in the northwest Pacific from 1995 to 2018 with intensity labels, captured by the satellites such as GMS-5, GOES-9, MTSAT-1R, MTSAT-2, and Himawari-8. Experimental samples are taken from the infrared band (IR3) images with multiresolutions, and the details about IR3 for these five satellites are shown in Table 1. The NII establishes a benchmark to rank typhoon intensity for five grades outlined in Table 2, such as tropical depression (TD), tropical storm (TS), severe tropical storm (STS), typhoon (TY), and severe typhoon (STY), on the basis of wind speeds near the typhoon center. We label the respective TC grades mentioned above from category 1 to 5 successively to perform supervised learning on the typhoon datasets.

Typhoon datasets contain 12,500 satellite cloud images, of which 2500 images in each category are divided into 2000 samples for training and 500 samples for testing. To compare with other models such as VGG16 and ResNet50 (Chatfield et al. 2014; He et al. 2015), the size of images in the typhoon datasets is scaled from $512 \times 512 \times 1$ to $224 \times 224 \times 1$ by using the nearest-neighbor interpolation, and

---

**Table 2. A benchmark for typhoon intensity established by NII.**

<table>
<thead>
<tr>
<th>Intensity-based typhoon classification</th>
<th>Maximum wind speed (10-min mean; m s$^{-1}$)</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tropical depression (TD)</td>
<td>&lt;17</td>
<td>0</td>
</tr>
<tr>
<td>Tropical storm (TS)</td>
<td>≥17–&lt;25</td>
<td>1</td>
</tr>
<tr>
<td>Severe tropical storm (STS)</td>
<td>≥25–&lt;33</td>
<td>2</td>
</tr>
<tr>
<td>Typhoon (TY)</td>
<td>≥33–&lt;42</td>
<td>3</td>
</tr>
<tr>
<td>Severe typhoon (STY)</td>
<td>≥42</td>
<td>4</td>
</tr>
</tbody>
</table>
normalization is performed to reduce information redundancy. Training a CNN architecture that owns numerous free parameters from scratch needs a large number of labeled images. To increase the diversity of images, data augmentation is adopted to expand a certain number of images on each training sample (Chatfield et al. 2014). Meanwhile, data augmentation can add appropriate noise to the input, making the model robust and adaptive. The data augmentation—which would not alter the label of the image—is rotating randomly. Notably, the amplitude of random rotation can be bounded to a maximum angle ranging from 0° to 180°. As strong correlations between adjacent pixels usually accompanied with redundancy among training images, ZCA whitening, performed as a preprocessing procedure, can decrease redundant information about input images and reduce the correlation between characteristics, eventually making the whitened image more approximate to the original input data (Krizhevsky et al. 2012). Figure 1 illustrates a portion of raw training images of each category, and the first column images are the raw data. The second are the pixelate images and the third are ZCA whitening images from the raw data.

b. Methods

1) ARCHITECTURE OF TYPHOOON-CNNs

CNNs, a milestone in the course of deep learning, are prone to be highly efficient in tackling computer vision issues in large-scale fields (LeCun et al. 2015). A typical CNN architecture is a topological structure, with a sequence of feed-forward layers such as convolutional and pooling layers arranged alternately. A convolution layer works as a generalized linear operation between input images and a filter bank. Each filter corresponds to a small receptive field from the input image. A pooling layer, normally placed behind each convolutional layer, operates a subsampling step that samples a specific number of pixels according to the stride to retain the maximum or average pixel in feature maps. Finally, a fully connected layer with a softmax function of nonlinear mapping provides a probability output for each category. Such a specific structure performs superiorly to a standard (fully connected) artificial neural network, by making use of sparse neural connectivity (limited by small receptive fields) and sharing weights of convolutional filters.

Different from ImageNet samples derived from natural scenes and Modified National Institute of Standards and Technology (MNIST) handwriting samples with recognizable features, the features in the spiral radius of cloud images datasets are indistinguishable due to intricate atmospheric factors during the formation of typhoons. When coping with inconspicuous features, part of the traditional topological structure with convolutional and pooling layers arranged alternately will lose part of crucial features at a certain extent and weaken model performance because the pooling operation lowers the resolution of feature maps in exchange for faster model convergence and lesser computational burden (Sabour et al. 2017). Hence, part of the traditional CNN architectures is possibly insensitive to extract implicit features from typhoon cloud images.

Cyclic convolution strategy based on typical CNNs architecture is used, and the framework is called Typhoon-CNNs in our study. To implement the cyclic convolution process in Typhoon-CNNs, we proposed a set of two convolution layers and one pooling layer arranged successively in each group, which is called a cyclic convolution strategy in this paper. The framework used a max pooling layer by dividing the input into rectangular pooling regions and comparing the maximum of each region. The architecture of the Typhoon-CNNs is depicted in Fig. 2, and detailed classifier configurations for each layer are outlined in Table 3. The framework contains 10 feed-forward layers as follows: five convolutional layers, two max pooling layers, one fully connected layer, and an output layer. Input grayscale images are a matrix of 224 × 224, and the first convolutional layer convolves the input with learned filters of 11 × 11 typically. To preserve crucial features of the image, convolutional layers (con2 and con4) adopted a stride of 1, and con3 and con5 with stride 2 mainly was designed to reduce the dimension and the amount of calculation. The max pooling layer also used in the network structure considered the nonlinearity and noise suppression of the remote sensing images.

2) IMPROVED ACTIVATION FUNCTION

After the process of convolution operation, each filter produces a feature map through activation function existing in neurons, realizing automatic extraction for the implicit features. The nonlinear activation function should satisfy the following requirements: nonlinearity, saturability, continuity, and monotonicity, so as to highlight the back-propagation signal. The traditional activation functions are generally chosen to be logistic sigmoid, Tanh, ReLU (rectified linear unit), or ELU (exponential linear unit) function. These functions induce sparsity in hidden units and sparse feature representations for input images obtained by model (Rehn and Sommer 2007; Nair and Hinton 2010). Selecting an appropriate activation function adaptively on the typhoon dataset will improve the generalized classification performance of Typhoon-CNNs.

Currently, ReLU is the more mature activation function, which solves gradient vanishing with its merits of unilateral inhibition and sparse activation, conforming with activity of biological nerve (Glorot et al. 2012). When receiving a negative signal, neurons are in an inhibitory state, called “dead neurons.” Such a phenomenon makes the weights of inactive neurons never renewed while an optimization algorithm of gradient descent is applied during the feedback process in CNNs.

As a result, the forced sparsity of the ReLU function is likely to lose effective information, especially for the typhoon dataset that has hardly recognizable features. Thus, an improved function based on the ReLU, called the “T-ReLU,” function is employed in Typhoon-CNNs. Based on the advantages of ReLU and Tanh function, the T-ReLU function combines linearity for half of the input domain with saturation for
Fig. 1. Training images of clouds in the typhoon dataset. The number in front of "-" represents the typhoon intensity, and the letters after "-" represent different images: (a) raw images, (b) pixelate images, and (c) ZCA-whitening images.
the other half. The improved T-ReLU function is defined as the following formula:

$$T-\text{ReLU}(x) = \begin{cases} x, & x \geq 0 \\ k \left( \frac{2}{1 + e^{-x}} - 1 \right), & x < 0 \end{cases}$$

where $x$ represents the input of the nonlinear T-ReLU function, $k$ is a constant coefficient to control the extent of soft saturation, and its value varies from 0.1 to 0.5. Figure 3 shows the curves of original and derivative T-ReLU function, where T-ReLU retains saturation and nonlinearity on the negative axis and linearity on the positive axis.

From the perspective of residuals (the difference between actual and desired output), which are propagated back to the forward layers hierarchically to adjust network weights, the update process of CNNs weights during back-propagation algorithm can be formulated as follows:

$$\delta_n = \frac{\partial L(W, b)}{\partial z_n} = \nabla_a L \cdot \sigma'(z_n) = (W_{n,n-1} \delta_{n-1}) \sigma'(z_n),$$

where the residual of the $n$th layer $\delta_n$ is a derivative of the loss function $L(W, b)$ with respect to the input of the $n$th layer $z_n$. The $\sigma'(z_n)$ represents a derivative of the activation function, $W_{n,n-1}$ is the connected weight matrix between the $(n-1)$th and $n$th layers, and $a_n$ is the actual output.

In the formula (2), when $z_n > 0$, T-ReLU$'(x) = 1$, the gradient is invariance and neurons are in an active state. The residual $\delta_n$ is only determined by $W_{n,n-1}$ and $\delta_{n-1}$, which prevents the residual from being weakened by derivative of the activation function. When $z_n < 0$, T-ReLU$'(x) \neq 1$, the T-ReLU function ensures that inactive regions have a nonzero continuous gradient and some negative information. This information as small noise participates in updating weights to improve the learning ability and robustness of Typhoon-CNNs.

3) IMPROVED CROSS-ENTROPY LOSS FUNCTION

The loss function is a crucial component of CNN algorithms, and its value largely represents the robustness of the model. In this paper, we propose a loss function called “CE-FMCE,” which combines cross-entropy (CE) with the improved minimum classification error (MCE). Various loss functions such as mean square error (MSE), CE, and MCE have been extensively used to offer the gradient of weights in the back-propagation algorithm. The curve of cross-entropy loss function has fewer flat regions than MSE, which can jump out of local optimum easily; thus, cross-entropy loss function is more applicable to multiple classification tasks (Krizhevsky et al. 2012; Szegedy et al. 2016). While cross-entropy loss function is mainly expected to raise the output accuracy of the correct categories and ignore probabilities of other incorrect categories without distinction, which makes classifier difficult to distinguish between true and false categories usually accompanied with the decline in model accuracy.

The loss function based on MCE takes the most confused false categories into consideration and reduces misclassification probability of traditional shallow networks (McDermott et al. 2006). However, the constructor of MCE is a sigmoid function that can easily be saturated in the back-propagation algorithm, causing gradient vanishing. In particular, the direction of gradient descent in MCE is contrary to that in cross-

TABLE 3. The Typhoon-CNNs classifier configurations.

<table>
<thead>
<tr>
<th>Layers</th>
<th>Input sizes</th>
<th>Kernel numbers</th>
<th>Kernel sizes</th>
<th>Strides</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1</td>
<td>224 × 224</td>
<td>64</td>
<td>11 × 11</td>
<td>4</td>
</tr>
<tr>
<td>pool1</td>
<td>54 × 54</td>
<td>1</td>
<td>2 × 2</td>
<td>2</td>
</tr>
<tr>
<td>conv2</td>
<td>27 × 27</td>
<td>128</td>
<td>3 × 3</td>
<td>1</td>
</tr>
<tr>
<td>conv3</td>
<td>25 × 25</td>
<td>128</td>
<td>3 × 3</td>
<td>2</td>
</tr>
<tr>
<td>pool2</td>
<td>12 × 12</td>
<td>1</td>
<td>2 × 2</td>
<td>2</td>
</tr>
<tr>
<td>conv4</td>
<td>6 × 6</td>
<td>256</td>
<td>3 × 3</td>
<td>1</td>
</tr>
<tr>
<td>conv5</td>
<td>4 × 4</td>
<td>256</td>
<td>2 × 2</td>
<td>2</td>
</tr>
<tr>
<td>fc6</td>
<td>2 × 2</td>
<td>512</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>out7</td>
<td>512</td>
<td>5</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

FIG. 2. The overall architecture of Typhoon-CNNs.

FIG. 3. The original and derivative of the T-ReLU activation function.
entropy under misclassification situations, and finally results in training incompletely and converging slowly on CNNs.

An MCE criterion is proposed to define a loss function, called fixed minimum classification error (FMCE), which adds a rectified term to the traditional MCE, mainly tackling problems such as gradient vanishing and different directions of gradient descent between MCE and cross-entropy (Zheng et al. 2019). To illuminate for convenience, the following formulas are defined only for one sample. First, the softmax function is used as a constructor \( g_k(z) \), calculated as

\[
g_k(z) = \frac{\exp(z_k)}{\sum_{i=1}^{L} \exp(z_i)},
\]

where \( z_k \) is the input of the \( k \)th neuron from output layer and \( L \) represents the total number of sample categories. Then an MCE function \( d_k(z) \) is defined as follows to measure the difference between output probability of true labeled and the most confused unlabeled categories (Juang and Katagiri 1992):

\[
d_k(z) = -p_k + p_l = -\frac{\exp(z_k)}{\sum_{i=1}^{L} \exp(z_i)} + \frac{\exp(p_l)}{\sum_{i=1}^{L} \exp(z_i)}.
\]

In the misclassification, \( k \) is the true label and \( l \) is the most confused label. When \( 0 < d_k(z) < 1 \), the model performs the wrong classification. The traditional MCE function will not only fail to correct the error, but also cause a negative influence in parameter adjustment due to the opposite parameter gradient. Thus, we improved MCE by adding a rectified term \( \sum_{i=1}^{L} \exp(z_i) \) to correct the direction of gradient. The loss function FMCE for \( N \) samples is defined as follows:

\[
L_M = \begin{cases} 
\frac{1}{N} \sum_{n=1}^{N} \left[ d_k + \ln \sum_{i=1}^{L} \exp(z_i) \right], & 0 < d_k < 1 \\
\frac{1}{N} \sum_{n=1}^{N} d_k, & -1 < d_k < 0 
\end{cases}
\]

Finally, the improved cross-entropy loss function is proposed, called CE-FMCE, by adding FMCE and cross-entropy functions together. The final formula is

\[
L_C = -\frac{1}{N} \sum_{n=1}^{N} y_i^T \log(p_n),
\]

where \( L_c \) is a cross-entropy function, the second term \( L_M \) works as a regularization term, and the coefficient \( \lambda \) controls the degree of regularization. The initial value of \( \lambda \) is empirically set to \( 1/T \), where \( T \) represents the number of samples in a batch. Once the learning rate is optimized using the initial value of \( \lambda \), \( \lambda \) is also optimized continuously according to the validation accuracy.

According to the above formula definitions of the improved loss function, CE-FMCE cannot only raise the output accuracy but also reduce the probability of misclassification, by combing the cross-entropy and improved MCE function.

Proof: The residual of loss function is deduced theoretically, and the feasibility of FMCE function is proved. To facilitate analysis, we only consider a single sample and formula (5) can be simplified as follows:

\[
L_M = \begin{cases} 
d_k + \ln \sum_{i=1}^{L} \exp(z_i), & 0 < d_k < 1 \\
d_k, & -1 < d_k \leq 0 
\end{cases}
\]

In the gradient descent algorithm, residuals from output layers \( \delta_M \) can be formalized as

\[
\delta_M = \begin{cases} 
\delta_1 + \delta_2, & 0 < d_k < 1 \\
\delta_2, & -1 < d_k \leq 0 
\end{cases}
\]

where

\[
\delta_1 = \frac{\partial d_k(z)}{\partial z_j} = \begin{cases} 
p_j(-d_k-1), & j = k \\
-p_j, & j \neq k \text{ and } j \neq l \\
p_j(-d_k+1), & j = l 
\end{cases}
\]

Due to the relationship between residual and gradient value, we discuss separately as follows:

When \( 0 < d_k < 1 \):
### 3. Experimental results and discussion

All the proposed approaches are implemented by using the Keras framework based on Tensorflow, a library for Python, which extremely simplifies the usage and training for CNNs. The experiments are conducted on a CPU (intelXeonX5650 at 2.67 GHz) with 16 GB of memory, under the Windows operating system.

Test accuracy and training time of Typhoon-CNNs are taken as a comprehensive index to evaluate model performance. The model of Typhoon-CNNs is trained through over 10,000 grayscale samples, 20% of which as a validation set fine-tune the hyperparameters per epoch, and 2500 images are selected to test the Typhoon-CNNs. After 200 epochs, the performance of Typhoon-CNNs tends to be stable, the verification accuracy is 89.31%, and the loss value is 0.12. Figures 4a and 4b depict accuracy and loss curves, respectively, during the training procedure. Figures 5a and 5b depict test accuracy and loss under different epochs. The detailed settings of model hyperparameters during experiments are shown in Table 5.

The training accuracy curve in Fig. 4a tends to converge by reaching the accuracy of 0.93 ± 0.01 after 50 epochs while the validation accuracy converges slowly after approximately 140 epochs by achieving 0.88 ± 0.02 accuracy. Typhoon-CNNs controls the difference between training and verification accuracy within 4%, which alleviates the over fitting phenomenon to a certain extent. A total of 2500 image samples without training are chosen to test the classification performance of Typhoon-CNNs. During the testing process, batch size is set to 64 for the entire test set. Depicted in Fig. 5a, after iterating 150 epochs, the mean test accuracy on typhoon test set reaches 88.74% and the value of test loss reaches to 0.09 after 200 epochs. A total of 2218 images are correctly classified, and 282 images are wrongly predicted. In the following sections, the process of establishing Typhoon-CNNs will be illustrated in detail.

#### a. Selection for optimal sizes of convolution filters

The size of convolutional filters in Typhoon-CNNs affects the performance of feature extraction from associated pixels in typhoon cloud images. In detail, if the chosen size is too small, associated pixels can hardly be fully extracted, resulting in slow learning efficiency, while a too large convolutional filter size will increase redundancy and lose partial valuable information (Park and Bang 2010). Considering input images have a resolution of $224 \times 224$, the first convolutional layer is equipped with 64 filters of $11 \times 11$, and subsequent three convolutional layers with the same size of $1 \times 1$, $3 \times 3$, $5 \times 5$, or $7 \times 7$. For the sake of calculating, the last convolutional layer

#### Table 4. Analysis on gradient of loss function.

<table>
<thead>
<tr>
<th>$d_k$</th>
<th>$0 &lt; d_k &lt; 1$</th>
<th>$-1 &lt; d_k &lt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j = k$</td>
<td>$\delta_2$</td>
<td>$\delta_3$</td>
</tr>
<tr>
<td>$j \neq k, l$</td>
<td>$\delta_2$</td>
<td>$\delta_3$</td>
</tr>
</tbody>
</table>

The output of the softmax function represents posterior probability and $p_k, p_l \in [0, 1]$. When a sample changes from the wrong classification to the correct classification, $d_k$ transforms from 1 to $-1$. The relationship between the direction of $\delta_M$ and $d_k$ can be summarized in Table 4. According to formulas (12) and (13), no matter when a sample is wrongly or correctly classified, corresponding to $0 < d_k < 1$ or $-1 < d_k < 0$, the CE-FMCE function can distinguish a wrong classification from a correct classification in the loss function and ensure that the gradient direction of $\delta_M$ is consistent with that of residuals in the cross-entropy function.

### 3. Experimental results and discussion

All the proposed approaches are implemented by using the Keras framework based on Tensorflow, a library for Python, which extremely simplifies the usage and training for CNNs. The experiments are conducted on a CPU (intelXeonX5650 at 2.67 GHz) with 16 GB of memory, under the Windows operating system.

Test accuracy and training time of Typhoon-CNNs are taken as a comprehensive index to evaluate model performance. The model of Typhoon-CNNs is trained through over 10,000 grayscale samples, 20% of which as a validation set fine-tune the hyperparameters per epoch, and 2500 images are selected to test the Typhoon-CNNs. After 200 epochs, the performance of Typhoon-CNNs tends to be stable, the verification accuracy is 89.31%, and the loss value is 0.12. Figures 4a and 4b depict accuracy and loss curves, respectively, during the training procedure. Figures 5a and 5b depict test accuracy and loss under different epochs. The detailed settings of model hyperparameters during experiments are shown in Table 5.

The training accuracy curve in Fig. 4a tends to converge by reaching the accuracy of 0.93 ± 0.01 after 50 epochs while the validation accuracy converges slowly after approximately 140 epochs by achieving 0.88 ± 0.02 accuracy. Typhoon-CNNs controls the difference between training and verification accuracy within 4%, which alleviates the over fitting phenomenon to a certain extent. A total of 2500 image samples without training are chosen to test the classification performance of Typhoon-CNNs. During the testing process, batch size is set to 64 for the entire test set. Depicted in Fig. 5a, after iterating 150 epochs, the mean test accuracy on typhoon test set reaches 88.74% and the value of test loss reaches to 0.09 after 200 epochs. A total of 2218 images are correctly classified, and 282 images are wrongly predicted. In the following sections, the process of establishing Typhoon-CNNs will be illustrated in detail.

#### a. Selection for optimal sizes of convolution filters

The size of convolutional filters in Typhoon-CNNs affects the performance of feature extraction from associated pixels in typhoon cloud images. In detail, if the chosen size is too small, associated pixels can hardly be fully extracted, resulting in slow learning efficiency, while a too large convolutional filter size will increase redundancy and lose partial valuable information (Park and Bang 2010). Considering input images have a resolution of $224 \times 224$, the first convolutional layer is equipped with 64 filters of $11 \times 11$, and subsequent three convolutional layers with the same size of $1 \times 1$, $3 \times 3$, $5 \times 5$, or $7 \times 7$. For the sake of calculating, the last convolutional layer

### Table 4. Analysis on gradient of loss function.

<table>
<thead>
<tr>
<th>$\delta_M$</th>
<th>$0 &lt; d_k &lt; 1$</th>
<th>$-1 &lt; d_k &lt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j = k$</td>
<td>$\delta_2$</td>
<td>$\delta_3$</td>
</tr>
<tr>
<td>$j \neq k, l$</td>
<td>$\delta_2$</td>
<td>$\delta_3$</td>
</tr>
</tbody>
</table>

The output of the softmax function represents posterior probability and $p_k, p_l \in [0, 1]$. When a sample changes from the wrong classification to the correct classification, $d_k$ transforms from 1 to $-1$. The relationship between the direction of $\delta_M$ and $d_k$ can be summarized in Table 4. According to formulas (12) and (13), no matter when a sample is wrongly or correctly classified, corresponding to $0 < d_k < 1$ or $-1 < d_k < 0$, the CE-FMCE function can distinguish a wrong classification from a correct classification in the loss function and ensure that the gradient direction of $\delta_M$ is consistent with that of residuals in the cross-entropy function.

#### 3. Experimental results and discussion

All the proposed approaches are implemented by using the Keras framework based on Tensorflow, a library for Python, which extremely simplifies the usage and training for CNNs. The experiments are conducted on a CPU (intelXeonX5650 at 2.67 GHz) with 16 GB of memory, under the Windows operating system.

Test accuracy and training time of Typhoon-CNNs are taken as a comprehensive index to evaluate model performance. The model of Typhoon-CNNs is trained through over 10,000 grayscale samples, 20% of which as a validation set fine-tune the hyperparameters per epoch, and 2500 images are selected to test the Typhoon-CNNs. After 200 epochs, the performance of Typhoon-CNNs tends to be stable, the verification accuracy is 89.31%, and the loss value is 0.12. Figures 4a and 4b depict accuracy and loss curves, respectively, during the training procedure. Figures 5a and 5b depict test accuracy and loss under different epochs. The detailed settings of model hyperparameters during experiments are shown in Table 5.

The training accuracy curve in Fig. 4a tends to converge by reaching the accuracy of 0.93 ± 0.01 after 50 epochs while the validation accuracy converges slowly after approximately 140 epochs by achieving 0.88 ± 0.02 accuracy. Typhoon-CNNs controls the difference between training and verification accuracy within 4%, which alleviates the over fitting phenomenon to a certain extent. A total of 2500 image samples without training are chosen to test the classification performance of Typhoon-CNNs. During the testing process, batch size is set to 64 for the entire test set. Depicted in Fig. 5a, after iterating 150 epochs, the mean test accuracy on typhoon test set reaches 88.74% and the value of test loss reaches to 0.09 after 200 epochs. A total of 2218 images are correctly classified, and 282 images are wrongly predicted. In the following sections, the process of establishing Typhoon-CNNs will be illustrated in detail.

#### a. Selection for optimal sizes of convolution filters

The size of convolutional filters in Typhoon-CNNs affects the performance of feature extraction from associated pixels in typhoon cloud images. In detail, if the chosen size is too small, associated pixels can hardly be fully extracted, resulting in slow learning efficiency, while a too large convolutional filter size will increase redundancy and lose partial valuable information (Park and Bang 2010). Considering input images have a resolution of $224 \times 224$, the first convolutional layer is equipped with 64 filters of $11 \times 11$, and subsequent three convolutional layers with the same size of $1 \times 1$, $3 \times 3$, $5 \times 5$, or $7 \times 7$. For the sake of calculating, the last convolutional layer
adopts 256 filters of $2 \times 2$. According to filter size, five groups of comparative experiments are planned, and the respective losses curve of cross-entropy value are depicted in Fig. 6 after 300 epochs. Table 6 outlines the training and validation accuracy of Typhoon-CNNs against various size of filters on typhoon dataset.

As shown in Fig. 6, the loss curve of the filter size of $3 \times 3$ in typhoon CNNs converges the fastest after 120 epochs, reaching the minimum cross entropy loss of $0.06 \pm 0.03$, $0.10$ lower than $5 \times 5$ scale, $0.17$ smaller than $7 \times 7$, and $0.38$ smaller than $1 \times 1$. Simultaneously, the model achieves the optimal training accuracy of $91.35\%$ and validation accuracy of $82.90\%$ on typhoon dataset. The loss curve of $1 \times 1$ size converges the slowest and achieves the lowest accuracy outlined in Table 6. Proved by comparative experiments, convolutional filters with the size of $3 \times 3$ are more consistent with feature distribution in spiral cloud bands of the typhoon.

b. Optimization for rate of dropout in fully connected layer

The phenomenon of overfitting is common no matter which size of convolutional filters is adopted from Table 4 in section 3a. The mean difference between training and validation accuracy is over $8\%$. To alleviate overfitting caused by plenty of free trainable network parameters in Typhoon-CNNs, a dropout strategy is applied to the fully connected layers. We randomly freeze $10\%$–$90\%$ of neurons in the last fully connected layers (fc6 in Table 3) and compare with the classification performance of Typhoon-CNNs without freezing neurons in 10 experiments. Figure 7 depicts test accuracy of the network when freezing different proportions of neurons in the fully connected layer, and we find that the ratio of neuronal freezing is irregular with the generalization of typhoon CNNs. According to Fig. 7, the zero-set rate of $60\%$ is adopted to improve test accuracy to $86.72\%$, and the difference between training and test accuracy is less than $5\%$, which can alleviate the overfitting phenomenon significantly. Dropout strategy, which reduces sophisticated co-adaptations of neurons, makes the update of weights independent on the inherent relation of hidden nodes among fully connected layers (Hinton et al. 2012). In general, the overfitting phenomenon of typhoon CNN can be alleviated to some extent through these comparative experiments.

c. Comparison among different activation functions

Figure 8 compares several common activation functions such as Tanh, ReLU, and ELU with T-ReLU in our research by evaluating the performance of Typhoon-CNNs under various activation functions on the typhoon dataset. Figure 8 shows training and validation accuracy under four different activation functions, respectively, from which the improved activation function T-ReLU that we proposed presents the optimal classification accuracy on both training and validation dataset by achieving the accuracy of $92.09\%$

<table>
<thead>
<tr>
<th>Data</th>
<th>Batch size</th>
<th>LR</th>
<th>Decay</th>
<th>Momentum</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training dataset</td>
<td>128</td>
<td>0.01</td>
<td>1e-6</td>
<td>0.9</td>
<td>$0.93 \pm 0.01$</td>
</tr>
<tr>
<td>Test dataset</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td>$0.88 \pm 0.01$</td>
</tr>
</tbody>
</table>
and 87.71%, respectively, 2.8% higher than ReLU function. And the Tanh function performs worst, which tends to be overfitting. The ELU function adds a nonzero gradient on neurons in inhibitory state and makes the negative information to participate in training of network. While all of the negative signals are compressed between $-1$ and $0$ in the inhibitory region, the derivative of ELU function within such scope is close to 1, which means all of the negative values will affect the weight update of each layer in network (Zheng et al. 2018). Unlike the ELU function, the T-ReLU function compresses the negative signals between $-0.2$ and $0$. The derivative within such scope is far less than 1, which can ensure that negative values will participate in the update of weight in few network layers.

d. **Comparison among different loss functions**

Considering the limitations of cross-entropy function and MCE in improving model performance on classification, a new max-margin minimum classification error (M$^3$CE) training approach is proposed, which is inspired by traditional MCE. The M$^3$CE aims not only to increase posterior probability of the true category but also to decrease the output of the most confused category (Feng et al. 2016). In this section, loss functions such as MCE, MSE, M$^3$CE, and cross-entropy functions are compared with the CE-FMCE, to manifest the feasibility of CE-FMCE function used in Typhoon-CNNs. The batch size is set to 40, and the initial value of $\lambda$ is $0.025$ for the batch size. Finally, $\lambda$ in CE-FMCE is 0.03 after the optimization. Figure 9 shows curves of training accuracy on Typhoon-CNNs using five various loss functions.

The accuracy curve of using the CE-FMCE function converges the fastest after iterating 52 epochs and achieves the highest training accuracy of 93.37%. This is mainly due to the fact that descending direction of cross entropy was consistent with the MCE. Though M$^3$CE overcomes the shortcoming of gradient vanishing in MCE, it achieves the lowest accuracy of 82.51% and converges slower than the CE-FMCE function in Fig. 9. The main reason is that the descent direction of loss function gradient was different from that of cross entropy. Notably, MSE achieves training accuracy of 85.4%, 7.90% lower than CE-FMCE, which evaluates posterior probability in classification tasks and is more applicable to solve regression problems (Guo et al. 2011). Table 7 outlines training and test accuracy of Typhoon-CNNs in detail, which manifests the CE-FMCE loss function is applicable to back-propagation of residuals during the training phase of Typhoon-CNNs.

<table>
<thead>
<tr>
<th>Convolutional filter</th>
<th>Training accuracy (%)</th>
<th>Validation accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 × 1</td>
<td>84.31</td>
<td>77.62</td>
</tr>
<tr>
<td>3 × 3</td>
<td>91.35</td>
<td>82.90</td>
</tr>
<tr>
<td>5 × 5</td>
<td>88.30</td>
<td>82.37</td>
</tr>
<tr>
<td>7 × 7</td>
<td>87.05</td>
<td>80.44</td>
</tr>
</tbody>
</table>

**Table 6. Model performance with convolutional filter in various sizes.**
In conclusion, the CE-FMCE function makes sure to enhance the output probability of the correct category and consider the incorrect category meanwhile.

e. **Comparison with the existing works**

Many deep learning approaches, especially deep convolutional neural network (DCNN), have acquired excellent performance on a variety of computer vision issues. In particular, three representative works, VGGNet, GoogLeNet, and ResNet, have achieved prominent ability for feature learning on the ImageNet dataset in the Large-Scale Visual Recognition Challenge (ILSVRC) (Simonyan and Zisserman 2014; He et al. 2015; Szegedy et al. 2015). The Typhoon-CNNs proposed in our research is compared with these three representative DCNNs trained on the typhoon dataset in Table 8. Figure 10 describes the stabilization time required to train the four models to converge and stabilize. From Table 8 and Fig. 10, Typhoon-CNNs presents better performance than the other three representative models by consuming about 12400 s after iterating 100 epochs and achieving test accuracy of 88.74%, 7.43% higher than ResNet50, 10.27% higher than InceptionV3, and 14.71% higher than VGG16. The three DCNNs tend to be overfitting and acquire unsatisfactory model performance during testing phase, which results from the huge discrepancy between ImageNet dataset in natural scenes and typhoon dataset in the remote sensing situation mentioned in section 2b.

f. **Visualization of relevant patterns**

The process of visualization allows us to visualize patterns that are detected at each layer of the deep network (Zeiler and Fergus 2013). Such procedure contributes to make hierarchies of CNNs from the category of black-box models visible, highlighting the most relevant regions of input images for network output probabilities. In this paper, we visualize features extracted by Typhoon-CNNs hierarchically in a single typhoon cloud image. Figure 11 visualizes the feature maps of partial layers in Typhoon-CNNs. From Fig. 11a, we can find that the first convolutional layer (conv1) learns features which represent the absence or presence of edges at particular rotation direction and location of typhoon center in cloud images. As the number of convolutional layers increases, the features extracted by filters are becoming more abstract, contrasting with feature maps in the layers of conv2 in Fig. 11b. After the pooling operation (pool2), the outline around the typhoon eye is clearer, and texture structure of the cloud system provides more abundant, accurate, and reliable detail information in Fig. 11c. Figure 11 verifies the basic principle of typhoon CNN. First, the edge features are identified by shallow learning convolution layer, and then these features are combined to form a more abstract concept. The whole cognitive process of feature extraction for input typhoon cloud image is from shallow to deep, from low level to high level.

To further confirm that the filter size of $3 \times 3$ is the most optimal for Typhoon-CNNs to achieve better performance on the typhoon dataset in section 3b, we select four cloud images and visualize feature maps with the sizes of $1 \times 1$, $3 \times 3$, $5 \times 5$, and $7 \times 7$, respectively, in the conv4 layer, corresponding to Figs. 12b–e. The filters of $3 \times 3$ can capture comprehensive and detailed features (yellow regions in feature maps), which represent the regions of the typhoon eye, cloud wall, and spiral cloud bands in the original images. Typhoon-CNNs is more sensitive to these areas rather than the fibrous cloud at the edge of typhoons. Feature extraction capability of $3 \times 3$ kernel size is better than filters of $1 \times 1$ and $5 \times 5$, which results in the fastest convergence in Fig. 7. With the size of convolution kernel increases, redundant information is also increased to a certain extent. Therefore, instead of increasing the kernel size, the feature extraction ability of model is improved (Wimmers et al. 2019). The filters of $7 \times 7$ increase the similar redundancy of typhoon images and lose local feature information. Thus, focusing on capturing features in yellow regions is beneficial to the classification task for Typhoon-CNNs in different filter size. Such a conclusion is also consistent with some research on typhoon, such as locating the typhoon eye, segmenting dense-shadowing cloud areas, and extracting characteristics in spiral cloud bands (Kuo et al. 2008; Wei et al. 2011; Chen et al. 2013).

Based on the traditional CNNs, our proposed Typhoon-CNNs is constructed by using the cyclic convolution strategy. The convolution layer and the convergence layer are arranged alternately. To compare the model performance of different

---

**Table 7. Model accuracy with various loss functions.**

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Training accuracy (%)</th>
<th>Test accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCE</td>
<td>82.51</td>
<td>79.48</td>
</tr>
<tr>
<td>MSE</td>
<td>85.46</td>
<td>81.05</td>
</tr>
<tr>
<td>Cross entropy</td>
<td>92.09</td>
<td>87.42</td>
</tr>
<tr>
<td>M²CCE</td>
<td>92.53</td>
<td>87.61</td>
</tr>
<tr>
<td>CE-FMCE</td>
<td>93.37</td>
<td>88.74</td>
</tr>
</tbody>
</table>

**Table 8. The accuracies on the typhoon cloud image datasets.**

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Training accuracy (%)</th>
<th>Validation accuracy (%)</th>
<th>Test accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>81.64</td>
<td>76.02</td>
<td>74.03</td>
</tr>
<tr>
<td>InceptionV3</td>
<td>85.03</td>
<td>80.94</td>
<td>78.47</td>
</tr>
<tr>
<td>ResNet50</td>
<td>86.95</td>
<td>83.55</td>
<td>81.31</td>
</tr>
<tr>
<td>Typhoon-CNNs</td>
<td>93.37</td>
<td>89.31</td>
<td>88.74</td>
</tr>
</tbody>
</table>
networks more intuitively, we visualize the hidden layers of traditional CNNs, VGG16, and Typhoon-CNNs, respectively, in Fig. 13. Figures 13a and 13b represent the learned abstract features in traditional CNNs. Figures 13c and 13d represent the learned abstract features in VGG16 and Typhoon-CNNs, respectively. Contrasting the area of yellow regions in Fig. 13, traditional CNNs lose a large proportion of local pivotal features, resulting in poor performance than Typhoon-CNNs and VGG16, because...
the pooling operation may drop a certain number of crucial features in yellow regions during the training phase.

g. Comparison with various dataset

The experimental datasets include IR3 satellite cloud images captured by the GMS-5, GOES-9, MTSAT-1R, MTSAT-2, and Himawari-8 satellites. Figure 14 depicts the performance of Typhoon-CNNs on five datasets, from which samples are captured respectively by these five satellites at infrared band. The test accuracy on the multisource experimental dataset is compared with the other single source datasets in Fig. 15.

Contrasting Fig. 14 with Fig. 15, the performance of Typhoon-CNNs on single-source datasets is superior to the experimental dataset we used. Because of the inconsistency in spectral resolution of the experimental samples resulting from various satellites, the test accuracy is 88.74%, lower than other datasets from only one satellite. If the data source of input is limited to one sensor, the accuracy will increase, but the adaptability and robustness of the model will decrease. The comparative experiments show that within the allowable range of accuracy reduction, the model we proposed can adapt to multisource merging datasets from different sensors.

4. Conclusions

Typhoons, as the most devastating disaster, cause extreme damage to human life and production. Precise classification for typhoon intensity is an urgent problem to be tackled. The technique of satellite remote sensing becomes the most convenient and effective of monitoring TCs with the usage of the high temporal–spatial resolution images captured by meteorological satellites. Satellite cloud images provide abundant weather information for deep learning approaches to extract meteorological characteristics automatically.

In this paper, a typhoon CNN based on convolutional neural networks (CNNs) is proposed. The model can be used to classify typhoon intensity from remote sensing cloud images. Its novelty and contribution lie the following aspects:

Typhoon-CNNs architecture. Based on the traditional CNN method, the typhoon classifier adopts a cyclical convolution strategy, which can extract sensitive features of spiral cloud bands. We optimize the selection for filter parameters to enhance the representation ability of Typhoon-CNNs. Dropout zero-set operation prevents the network from overfitting by freezing partial proportion of neurons randomly. At the same time, the optimized zero rate is selected, which improves the accuracy of the typhoon dataset from 82.90% to 86.72%.

Improved activation function, T-ReLU. On the basis of ReLU and Tanh, we improve the activation function called
T-ReLU in typhoon CNNs. The T-ReLU function activates neuron signals in a proper numerical range for negative values while retaining monotony in positive x axis, which can alleviate the dead neurons phenomenon, universally existing in ordinary activation functions. The Typhoon-CNNs using T-ReLU activation function achieves the training accuracy of 92.09% and test accuracy to 88.74%. These experiment results show the significant improvement on typhoon CNNs with T-ReLU than other activations.

Improved loss function, CE-FMCE. The cross-entropy loss function is widely used in CNNs during error back-propagation. The cross-entropy function neglects the influence of the most confused category in misclassification situations. We propose the CE-FMCE function based on the cross-entropy and MCE functions, through adding a rectified term to make the direction of gradient descent consistent with cross-entropy in a misclassification situation. The CE-FMCE function accelerates the speed of model convergence and finally increases training accuracy to 93.37% and test accuracy of 88.74%. Based on the above research, Typhoon CNNs has been established, and compared with other classical CNNs methods on typhoon datasets, it is found that the model accuracy and time consumption of Typhoon CNNs are better than other DCNNs.

We apply visualization techniques to observe that Typhoon-CNNs can detect specific features in the sensitive regions of typhoon cloud images automatically and generate abstract feature representation layer by layer, which verifies the feasibility of using Typhoon-CNNs to extract features automatically. The model of Typhoon-CNNs constructed on the multisource dataset is more robust and can adapt to multiple sensors.

Finally, based on visualization technology, we observe that typhoon CNNs can detect the specific features of the sensitive area of typhoon images, and generate abstract feature representation layer by layer, which confirms the feasibility of automatic feature extraction of typhoon CNNs. The typhoon CNNs model based on multisource datasets has stronger robustness and can adapt to multiple sensors.

Acknowledgments. This study was supported by grants from the NSF China (41671431), STCSM capacity-building projects in local universities (17050501900), scientific project from Shanghai Engineering Research Center of Estuarine and Oceanographic Mapping, and open fund project of Science Technology Key Laboratory in SOA Digital Ocean. We acknowledge NII for providing basic data on the typhoon events. The authors thank the reviewers in advance for their comments and suggestions.

REFERENCES


