The Astrophysical Journal, 885:73 (14pp), 2019 November 1 https://doi.org/10.3847/1538-4357/ab46bd

Solar Flare Prediction with the Hybrid Deep Convolutional Neural Network

Yanfang Zheng[1], Xuebao Li[1], and Xinshuo Wang
College of Electrical and Information Engineering, Jiangsu University of Science and Technology, Zhangjiagang, People's Republic of China; zyf062856@163.com
Received 2019 June 2; revised 2019 September 16; accepted 2019 September 19; published 2019 November 1

Abstract

We propose a hybrid Convolutional Neural Network (CNN) model (Model 2) and modify a popular CNN model (Model 1) to predict multiclass solar flare occurrence within 24 hr. We collect samples of solar active regions provided by the Space-weather Helioseismic and Magnetic Imager Active Region Patches data from 2010 May to 2018 September. These samples are categorized into four classes (No-flare, C, M, and X), containing 10 separate data sets. Then after training, validating, and testing our models, we compare the results with previous studies in forecast verification metrics with an emphasis on the true skill statistic (TSS). The main results are summarized as follows. (1) This is the first time that the CNN models are used to make multiclass predictions of solar flare. (2) Model 2 has better values of all statistical scores than Model 1 in every class. (3) Model 2 achieves relatively high average scores of TSS = 0.768 for No-flare class, 0.538 for C class, 0.534 for M class, and 0.552 for X class, which are the best results from the existing literatures. (4) Model 2 also can be used to make binary class flare predictions for ⩾M-class major flares, and the performance yields a TSS with 0.749 ± 0.079. (5) Model 2 obtains fairly good scores in other metrics for both multiclass flare predictions and ⩾M-class major flare predictions. We surmise that some crucial features extracted automatically by our models may have not been excavated before and could provide important clues for studying the mechanism of flare.

Key words: magnetic fields – Sun: activity – Sun: flares – techniques: image processing

1. Introduction

Solar flare refers to the sudden and large-scale energy release process occurring in a local area of the solar surface. The high energy particles due to strong solar flare can significantly affect the near space environment, such as the safety of satellites and astronauts, the radio communication system, the global positioning system, etc. Therefore, it is very significant to develop a high-accuracy and real-time predictive model for solar flare.

Up to now, there has still been a lack of definitive physical theory to explain the flaring mechanism of an active region (AR). Therefore, many methods of solar flare prediction based on statistical and machine learning algorithms have been developed and studied. Statistical methods have been applied in many solar flare prediction studies (Song et al. 2009; Mason & Hoeksema 2010; Bloomfield et al. 2012; Barnes et al. 2016). On the other hand, classic machine learning algorithms have become an increasingly popular approach for solar flare forecasts, such as the support vector machine (Yuan et al. 2010; Bobra & Couvidat 2015; Nishizuka et al. 2017; Sadykov & Kosovichev 2017), the artificial neural network (Qahwaji & Colak 2007; Ahmed et al. 2013; Li & Zhu 2013; Nishizuka et al. 2018), the Bayesian network approach (Yu et al. 2010), the random forest algorithm (Liu et al. 2017; Florios et al. 2018), and the ensemble learning (Colak & Qahwaji 2009; Huang et al. 2010; Guerra et al. 2015). All of the above studies have done binary class prediction, and Liu et al. (2017), Bloomfield et al. (2012), and Colak & Qahwaji (2009) have done multiclass prediction. Classic machine learning algorithms learn from engineered features extracted from the observational data. The flare prediction efficiency of the learning algorithms mainly depends on the quality of the features used.

In recent years, deep learning neural networks, as a branch of machine leaning, have emerged as a highly dependable technique to drive large-scale learning problems in astronomy and other branches of sciences. Deep learning is advantageous in situations where engineered features do not completely capture the physics of the raw data and the machine learning algorithm cannot learn with minimal loss (Arel et al. 2010; LeCun et al. 2015). Convolutional neural networks (CNNs; LeCun et al. 2015), a very popular deep learning method falling within the realm of image processing and computer vision, are able to carry out automatic feature extraction. These suffer less information loss compared to the classic machine learning algorithms, and are more suited to high-dimensional data sets (LeCun et al. 2015). The CNNs are based on neural networks that include several hidden layers. Each layer extracts complex features in the data before executing a classification or regression task. The raw data can be fed into the network, thus minimal to no feature engineering is required, and the network learns to extract the features through training. Park et al. (2018) presented a CNN model, which is a combination of GoogLeNet (Szegedy et al. 2014) and DenseNet (Huang et al. 2018a) to predict solar flare occurrence. Their model used full-disk solar line of sight (LOS) magnetograms to make binary class predictions within 24 hr. Huang et al. (2018b) presented a model based on CNN for flare prediction via binary classification. Their model used many patches of solar ARs from LOS magnetograms located within ± 30° of the solar disk center to avoid projection effects.

In this work, we attempt to propose a hybrid CNN model and modify a popular CNN model to predict solar flare occurrence with the outputs of four classes (i.e., No-flare, C, M, and X) within 24 hr. Input data for the models in this paper is from LOS magnetograms of ARs provided by the Helioseismic and

[1] These authors contributed equally to this work.

Magnetic Imager (HMI; Schou et al. 2012) on board the *Solar Dynamics Observatory* (*SDO*; Pesnell et al. 2012). The outputs for the models are compared with *Geostationary Operational Environment Satellite* (*GOES*) observations of the daily flare occurrence. To the best of our knowledge, this is the first time that CNN models are used to make multiclass solar flare predictions.

This paper is organized as follows. The data is described in Section 2, and the method in Section 3. Results are given in Section 4, and finally, conclusions and discussions are provided by Section 5.

2. Data

We use *SDO*/HMI LOS magnetograms of ARs as the input data of the proposed hybrid CNN model and the modified CNN model. *SDO*/HMI began its routine observation on 2010 April 30, and provided a continuous and high-quality photospheric magnetic field observation. Near the end of 2012, the *SDO*/HMI team started to release a new data product named Space-weather HMI Active Region Patches (SHARP; Bobra et al. 2014), which is convenient for AR event forecasting. These data are publicly available at the Joint Science Operations Center, and the LOS magnetograms of ARs can be obtained from SHARP (hmi.sharp_cea_720s). The LOS magnetograms as the input data of our models from SHARP, from 2010 May 1 to 2018 September 13, covering the main peak of solar cycle 24 are included, and these data are taken continuously and averaged to a cadence of 12 minutes. In order to minimize or avoid the influence of projection effects, only LOS magnetograms located within $\pm 45^{\circ}$ of the central meridian are considered for this work (Ahmed et al. 2013; Bobra et al. 2014).

In order to train and test our CNN models, we need to build a catalog of data sets. The solar flare is classified as B, C, M, or X class according to the peak magnitude of the soft X-ray flux observed by the *GOES*. This classification of solar flare is also applied in our models. The solar flare information is obtained from https://www.ngdc.noaa.gov/stp/space-weather/solar-data/solar-features/solar-flares/x-rays/goes/xrs/, which is constructed by the *GOES* X-ray flare listings. It is worth noting that many records of the flare events miss a location and National Oceanic and Atmospheric Administration (NOAA) AR number in the solar flare data, then we add the missing location information and AR number associated with the flare events, which are obtained from Solar Geophysical Data solar event reports (http://www.solarmonitor.org/index.php).

We then prepare a catalog of data sets in the following way: (1) for each magnetogram sample of AR, the No-flare (weaker than C 1.0) label is attached to the magnetogram if the AR does not flare within 24 hr after this magnetogram sample. (2) For each magnetogram sample of AR, the corresponding flare label (i.e., C, M, or X) is attached to the magnetogram if a C/M/X-class flare occurs within 24 hr after this magnetogram sample. For recurring flares of AR within 24 hr with different classes, the magnetograms within 24 hr before the first flare are labeled as the class of the first flare. When more than one flare occurs within the 24 hr time window, we set its label to the maximum-class flare that occurred. (3) Based on the above ways, we further categorize ARs into four classes according to the most powerful flare produced. It demands that an AR is classified into a specific class producing at least one flare with such *GOES* class but no flares with higher *GOES* class (Song et al. 2009; Yuan et al. 2010; Liu et al. 2017). "Label = No-flare" means that the AR only produces microflares (weaker than C1.0 flares); "Label = C" means that the AR produces at least one C-class flare but no M/X-class flares; "Label = M" means that the AR produces at least one M-class flare but no X-class flares; "Label = X" means that the AR produces at least one X-class flare. In total, we collect 870 ARs and 136134 magnetogram samples, including 443 X-class, 6534 M-class, 72412 C-class, and 56745 No-flare magnetogram samples.

Eventually, we take interest in developing a real-time prediction model to predict future flare activity given current solar data. In order to train and test our models on yet-unseen data, we simulate this process by segregating the whole data set into an 80% training data set and a 20% testing data set. The training data set contains the data that our algorithm will learn from, while the testing data set is used to evaluate our models. Note that it is difficult to correctly partition the whole data set into training and testing data set. In our study, we perform this segregation by NOAA AR number. In other words, all of the magnetogram samples of a given AR are placed in either the training or testing data set. It guarantees that our models are evaluated on ARs that are not used in the training data set.

We note that some magnetogram samples from SHARP contain multiple NOAA ARs (Bobra et al. 2014). In our study, such samples are excluded from the data set. It is obvious to see that the number of M/X-class magnetogram samples is far less than the number of No-flare/C-class magnetogram samples, so this is a strong class imbalance problem. It reflects the fact that most ARs do not produce major flares in any given 24 hr period. The class imbalance is a major issue for most machine-learning algorithms. In our work, we attempt to alleviate the class imbalance issue through undersampling and data augmentation techniques. The data set is undersampled by randomly selecting No-flare/C-class samples with about 2 samples per 10 samples. Meanwhile, we use data augmentation to artificially boost the number of M/X-class samples by rotating and reflecting images. In more detail, for M/X-class samples, we rotate each image by a multiple of 90°, and flip each image horizontally and vertically with a probability of 0.5 to exploit mirror symmetry. These data augmentation schemes make the resulting model more invariant to rotation and reflection in pixel values.

We utilize the cross-validation (CV) method to show the validity of our models, which is the standard approach in this field. There are several types of CV method such as shuffle and split CV, *K*-fold CV, leave-one-out CV, and so on (Nishizuka et al. 2017). As the number of ARs is strongly imbalanced in four classes, we select the shuffle and split CV method in our study. The data set is randomly shuffled by NOAA AR number, and partitioned into 10 separate training and testing data sets. We trained and evaluated our models on these 10 separate training and testing data sets, and the average results are shown. All information concerning 10 separate data sets is provided in Table 1.

3. Method

CNNs usually contain two main types of layers: convolutional and pooling layers. The input of a convolutional layer is an image, and the output of each convolutional layer is called a feature map. To produce output feature maps, we convolve the input image with a set of weights called kernels or filters. Mathematically, we consider the input of the nth convolutional layer to be a stack of M arrays $x_{m,n-1}(m = 1, \ldots, M)$, and the

Table 1
The Number of Solar Magnetogram Samples and ARs for 10 Separate Data Sets

Data Set	No-flare Class (Original/Preprocessed/AR Numbers)	C Class (Original/Preprocessed/AR Numbers)	M Class (Original/Preprocessed/AR Numbers)	X Class (Original/Preprocessed/AR Numbers)
No. 1: Training	46013/8796/359	55940/11091/237	5116/8862/60	238/2856/8
Test	10732/1331/57	16472/1756/39	1418/1146/8	205/1500/2
Total	56745/10127/416	72412/12847/276	6534/10008/68	443/4356/10
No. 2: Training	45587/8824/351	57040/11242/235	5417/9408/60	340/4080/8
Test	11158/1733/72	15372/1589/36	1117/876/7	103/60/1
Total	56745/10557/423	72412/12831/271	6534/10284/67	443/4140/9
No. 3: Training	46192/8996/362	58412/11549/235	4836/8562/59	399/4788/8
Test	10553/1382/58	14000/1379/35	1698/1950/11	44/312/2
Total	56745/10378/420	72412/12928/270	6534/10512/70	443/5100/10
No. 4: Training	44791/8807/347	56244/10864/239	5030/8814/60	307/3684/8
Test	11954/1565/66	16168/1635/37	1504/828/8	136/1416/2
Total	56745/10372/413	72412/12499/276	6534/9642/68	443/5100/10
No. 5: Training	45573/8951/352	58207/11611/234	5312/9270/59	313/3756/8
Test	11172/1664/68	14205/1686/38	1222/1218/9	130/600/2
Total	56745/10615/420	72412/13237/272	6534/10488/68	443/4356/10
No. 6: Training	46091/8836/359	56737/11189/238	5125/8580/62	304/3648/8
Test	10654/1500/62	15675/1641/39	1409/1332/5	139/1668/3
Total	56745/10336/421	72412/12830/277	6534/9912/67	443/5316/11
No. 7: Training	44940/8633/353	57521/11450/242	5015/8856/61	320/3840/8
Test	11805/1579/68	14891/1025/27	1519/1692/9	123/1476/3
Total	56745/10212/421	72412/12475/269	6534/10548/70	443/5316/11
No. 8: Training	46104/8908/357	59304/11689/243	5554/9468/58	386/4632/8
Test	10641/1459/64	13108/925/27	980/750/7	57/312/2
Total	56745/10367/421	72412/12614/270	6534/10218/65	443/4944/10
No. 9: Training	46139/8964/359	57209/11398/241	5023/9174/62	292/3504/8
Test	10606/1413/61	15203/1473/34	1511/1038/5	151/1440/2
Total	56745/10377/420	72412/12871/275	6534/10212/67	443/4944/10
No. 10: Training	45479/8632/357	58947/11473/236	5272/9216/60	324/3888/8
Test	11266/1517/65	13465/1266/30	1262/1392/10	119/252/1
Total	56745/10149/422	72412/12739/266	6534/10608/70	443/4140/9

Note. In every class of each data set, "Original" represents the number of magnetogram samples before image preprocessing, "Preprocessed" represents the number of magnetogram samples after image preprocessing used for our study, and "AR numbers" represents the number of ARs after image preprocessing used for our study. Image preprocessing indicates the sample images are processed through excluding the samples with multiple NOAA ARs, data augmentation, and undersampling.

output as L arrays $x_{l,n}(l = 1, \ldots, L)$. Thus, the output of nth layer is given by

$$x_{l,n} = f\left(\sum_{m=1}^{M} W_{m,l,n} \otimes x_{m,n-1} + b_{l,n}\right), \quad (1)$$

where $\otimes$ is the convolution operator, $b_{l,n}$ is the bias of nth layer, $W_{m,l,n}$ represents the kernels, and f is the activation function (or nonlinearity; Cabrera-Vives et al. 2017). The most popular nonlinearity at present is the rectified linear unit (ReLU; Nair & Hinton 2010), because it can achieve fast training and better performance. Pooling layers compute and choose an average (mean-pooling) or maximum (max-pooling) value within a sliding window of the input feature map to subsample each feature map. It reduces the dimensionality of the feature maps and makes the model invariant to small shifts and distortions (Boureau et al. 2010). In general, a common design for CNNs is composed of a mix of convolutional and pooling layers, followed by one or more dense fully connected layers. In order to reduce overfitting, a regularization technique called dropout is usually used in the fully connected layers (Hinton et al. 2012). Dropout consists of randomly turning off random neurons in the training phase with a probability p, usually selected as 0.5. Dropout is only activated in the training phase. However, all neurons become active when evaluating models in the testing phase. The output of the final fully connected layer is usually fed to a softmax activation function, which converts the score of each class to probabilities. The final output prediction is the highest probability class. In this paper, we consider two types of model to make multiclass flare predictions. One is a modified CNN model (Model 1), and the other is a proposed hybrid CNN model (Model 2). These models are trained and tested based on the Keras Deep Learning framework using the TensorFlow (Abadi et al. 2016) backend in Python programming language. The CNNs require a fixed size for all input images. Following a similar approach of Huang et al. (2018b), the input images from the data set are downsampled to 128 × 128 pixels.

3.1. Hybrid CNN Model and Modified CNN Model

The modified CNN model (Model 1) is a variant of VGGNet. VGGNet is introduced by Simonyan & Zisserman (2015), the winner of the Image Large Scale Visual Recognition Challenge (ImageNet Challenge) in 2014. VGGNet mainly

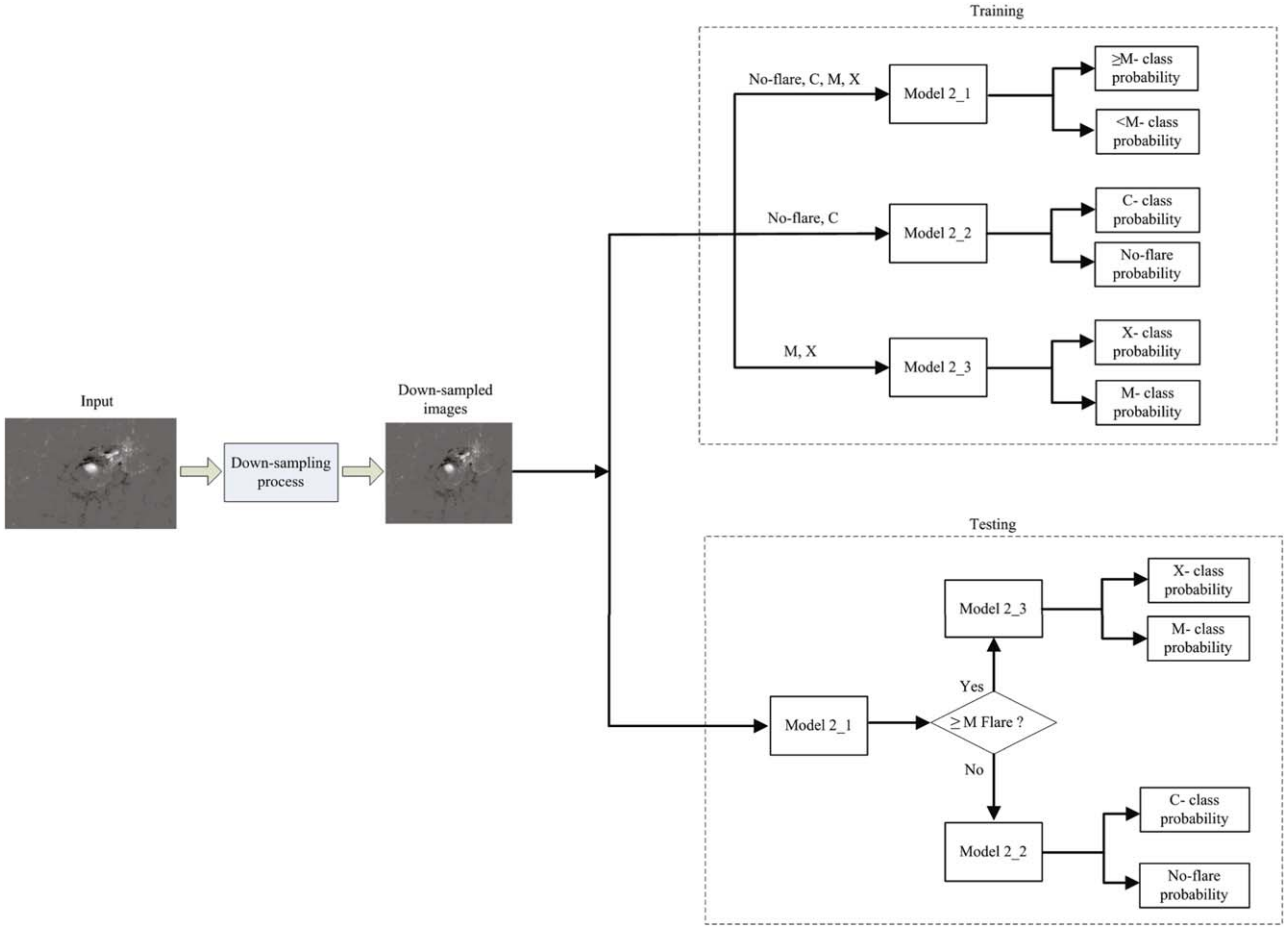

Figure 1. Proposed hybrid CNN model for four-class solar flare predictions.

contains 16–19 convolutional layers, 5 max-pooling layers, and 3 fully connected layers. The main contribution of VGGNet shows that the depth of network plays an import role in good performance. The original VGGNet cannot be applied to predict multiclass solar flares directly. In this work, we modify it as follows: (1) we make corrections to handle the number of classes, and select 16 convolutional layers, all of which apply a kernel size of 3×3, and all max-pooling layers apply a pooling size of 2×2. (2) We revise the model to work on single channel images, because the original network is designed to work on three-channel color images. (3) We exclude the three fully connected layers at the top of the network to avoid the size limitation of input image in the model on Keras. (4) We add two fully connected layers and one Batch Normalization (BN) layer (Ioffe & Szegedy 2015). The BN layer is usually added between activation function and convolutional layer or fully connected layer, which is used to standardize the input parameters at each layer during training and stabilizes the process. (5) We use the pretrained convolutional and max-pooling layers, and retrain the remaining BN and fully connected layers. In our study, there are two phases, one is the training phase, and the other is the testing phase. The model is fine-tuned in the training phase, and the testing result of evaluating the model is presented in Section 4.

We designed the hybrid CNN model (Model 2), which is proposed in this study and inspired by VGGNet, Colak & Qahwaji (2009), and Qahwaji & Colak (2007). Figure 1 shows the proposed hybrid CNN model for four-class solar flare predictions. The proposed model is composed of three CNN models (Model 2_1, Model 2_2, and Model 2_3), which are binary classifiers and work together as illustrated in Figure 1. In the training phase, the Model 2_1 adopts the whole training data set of input data, which contains all four-class magnetogram samples. It generates the probability that the AR will produce a $\geqslant$M-class solar flare. The Model 2_2 uses the subset of input data, which is extracted from the training data set and contains only No-flare and C-class magnetogram samples. Thus, it generates the probability that the AR will produce a C-class solar flare or No-flare (weaker than C1.0) event. In addition, the Model 2_3 also uses the subset of input data, which is extracted from the training data set and contains only M- and X-class magnetogram samples. Therefore, it generates the probability that the AR will produce an M/X-class solar flare. In the testing phase, each sample from the testing data set is first inputted to the Model 2_1. According to the output of the Model 2_1, this model predicts whether a $\geqslant$M-class flare will occur within 24 hr. In the case that the output of the Model 2_1 is "Yes," the Model 2_3 is activated and the sample is inputted to the Model 2_3 to further predict whether an M/X-class flare will occur within 24 hr. Otherwise, the Model 2_2 is activated and the sample is inputted to the Model 2_2 to further predict whether a C-class flare or No-flare event will occur within 24 hr. Eventually, according to the outputs of the above three CNN models, the proposed model (Model 2) makes four-class (i.e., No-flare, C, M, and X) flare predictions. In our work, flare prediction is a multiclass classification problem. The

one-versus-all method is canonical to combine the binary classification method for multiclass classification. If this method is selected in this study, we need to train four different binary classification models. The training samples of one category will be much less than the sum of the training samples of other categories, and this imbalance between training samples will affect the classification accuracy. Furthermore, because the samples are categorized into four classes according to the *GOES* magnitude, we surmise that the one-versus-all method could have difficulty in classifying C-versus-all (i.e., No-flare, M, X) and M-versus-all (i.e., No-flare, C, X). Our hybrid CNN model first divides four categories into two sub-classes, then sub-classes are further divided into two categories, and finally a separate category is obtained. Therefore, we need to train three different binary classification models in this study. This method gradually decomposes the complex multiclass classification problem into a series of binary classification problems. However, this method could propagate the error from top to bottom. Therefore, we make many efforts to enhance the separability between the two sub-classes (e.g., $\geqslant$M class versus $<$M class) to reduce the error accumulation.

The architecture of the proposed hybrid CNN model (Model 2) is illustrated in Figures 2–4. The three CNN models of Model 2 all adopt a stack of multiple 3×3 convolution filters, and increase the depth of network, which are learned from the architecture of VGGNet. In the Model 2_1, the model architecture is composed of two convolutional modules, two dense blocks, and a fully connected layer followed by a softmax activation function. The first convolutional module consists of two convolution blocks and each block has a convolutional layer with kernel of size 11×11, a BN layer, a ReLU function, and a 2×2 max-pooling layer. The second convolutional module consists of three convolutional blocks and each block has a convolutional layer with kernel of size 3×3, a BN layer, a ReLU function, and a 2×2 max-pooling layer. In two dense blocks, each block has a fully connected layer, a BN layer and a Dropout function. The corresponding architecture of the Model 2_1 is shown in Figure 2. Here, in the first few convolutional layers of the Model 2_1, we choose the kernel of size 11×11 to learn basic features, and 11×11 is the optimal size that we get after many attempts, which is also learned from the Alexnet network (Krizhevsky et al. 2012). However, in the last few convolutional layers of the Model 2_1, we choose the kernel of size 3×3 instead of 11×11 to learn complex high-level features. The kernel of large size will increase the number of model parameters to be calculated, which is not conducive to the increase of the model depth. In addition, the detailed descriptions for the architecture of the Model 2_2 and Model 2_3 are also given in Figures 3 and 4, respectively. The number of ARs in the training data set used in the Model 2_2 and Model 2_3 is small, and the kernel of large size (e.g., 11×11) will result in the increase of model parameters, which probably makes the model prone to overfitting. Therefore we choose a kernel of small size (i.e., 3×3) in all the convolutional layers of the Model 2_2 and Model 2_3.

3.2. The Optimization Method of Gradient Descent

In order to maximize the prediction accuracy, our models are trained to minimize a loss function, which is calculated from the cross entropy loss (Hinton & Salakhutdinov 2006). However, as the four-class flare occurrence ratio is imbalanced, we adopt the summation of the weighted cross entropy as the loss function,

$$L = \sum_{n=1}^{N}\sum_{k=1}^{K} \omega_k y_{nk} \log(p(y_{nk})), \tag{2}$$

$$\omega_k = \mathrm{len}(\mathrm{AR}_k)\mathrm{len}(\mathrm{sample}_k)\beta_k, \tag{3}$$

where N is the number of training samples, K is the number of classes, and $p(y_{nk})$ and y_{nk} are the predicted output and the expected output of each class during a forward propagation, respectively. Here, y_{nk} is the 4-valued label equal to X class, M class, C class, and No-flare for the Model 1, and the 2-valued label for three models of the Model 2, respectively. ω_k is the class weight of each class, $\mathrm{len}(\mathrm{AR}_k)$ is the number of AR of each class, $\mathrm{len}(\mathrm{sample}_k)$ is the number of sample of each class, and β_k represents the optimized parameter of each class.

The parameters of the model are iteratively learned by using the stochastic gradient descent (SGD; LeCun et al. 1998) method. In this method, the parameters θ of the model are updated on a small part of the training data called mini batch (Goodfellow et al. 2016).

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta(L), \tag{4}$$

where t is the training step, ∇ represents the gradient operator, L is the loss function, and η is the learning rate. The learning rate determines how quickly the gradient updates following the gradient direction. The momentum update method used is Nesterov in our work. It evaluates the gradient at the future rather than current position, resulting in better informed momentum updates, and hence improved performance (Sutskever 2013). The training parameter choice of learning rate, momentum, and class weight is also crucial for achieving high predictive performance and speeding up the learning process. These training parameter settings of the Model 1 and the three CNN models of Model 2 are summarized in Table 2. As shown in Table 2, our models are trained over 100 epochs using SGD as an optimizer with a mini batch size of 16 or 32 (see Table 2). One epoch means that an entire training data set is passed through the model for training. After setting these training parameters, the model can be trained as follows. First, in the forward propagation, the samples with mini batch size are taken as input going through the forward propagation, and the output results of the model are obtained, and then the loss function is estimated. Second, in the backward propagation, gradients of the loss function about all weights are back propagated through the model. The weights in the convolutional layer and the fully connected layer and the parameters in the BN layer are updated by the gradient descent to minimize the loss function. Third, the forward propagation and backward propagation are repeated to obtain the best well-trained model.

4. Results

To evaluate the models, we characterize the prediction results by a confusion matrix, also called a contingency table for each AR class k. The class k ARs correctly predicted as class k are called true positives (TP), and the class k ARs wrongly predicted as other class are false negatives (FN). The ARs not in class k, correctly predicted not as class k are true negatives (TN), and in the case of predictions of class k, they are

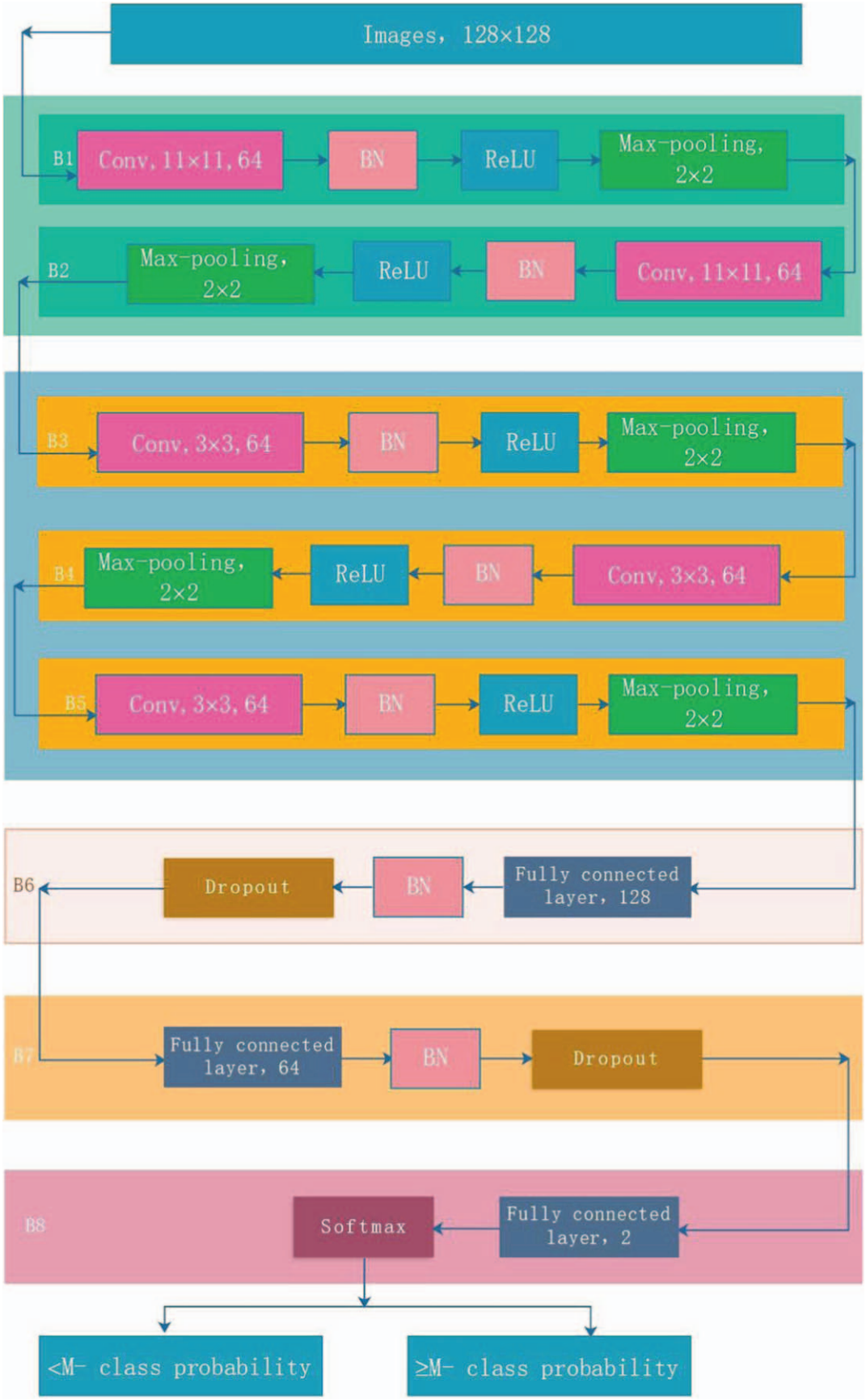

Figure 2. Architecture of the Model 2_1. The model consists of several blocks, where B1, B2, etc. represent block number, respectively. The model takes the downsampled images of size 128 × 128, and outputs the probability of ⩾M class and <M class.

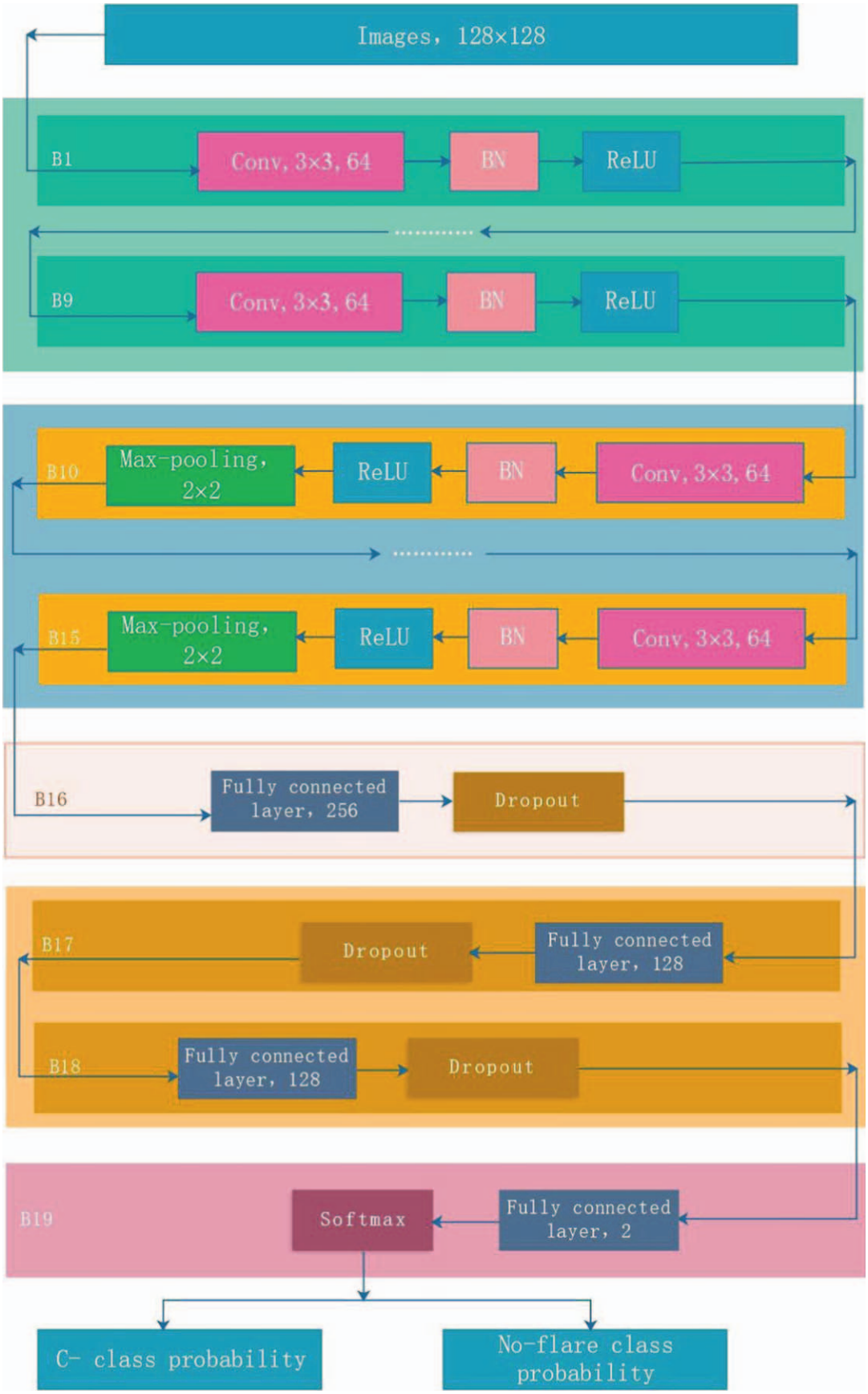

Figure 3. Architecture of the Model 2_2. The model consists of several blocks, where B1, B2, etc. represent block number, respectively. The model takes the downsampled images of size 128 × 128, and outputs the probability of C class and No-flare class.

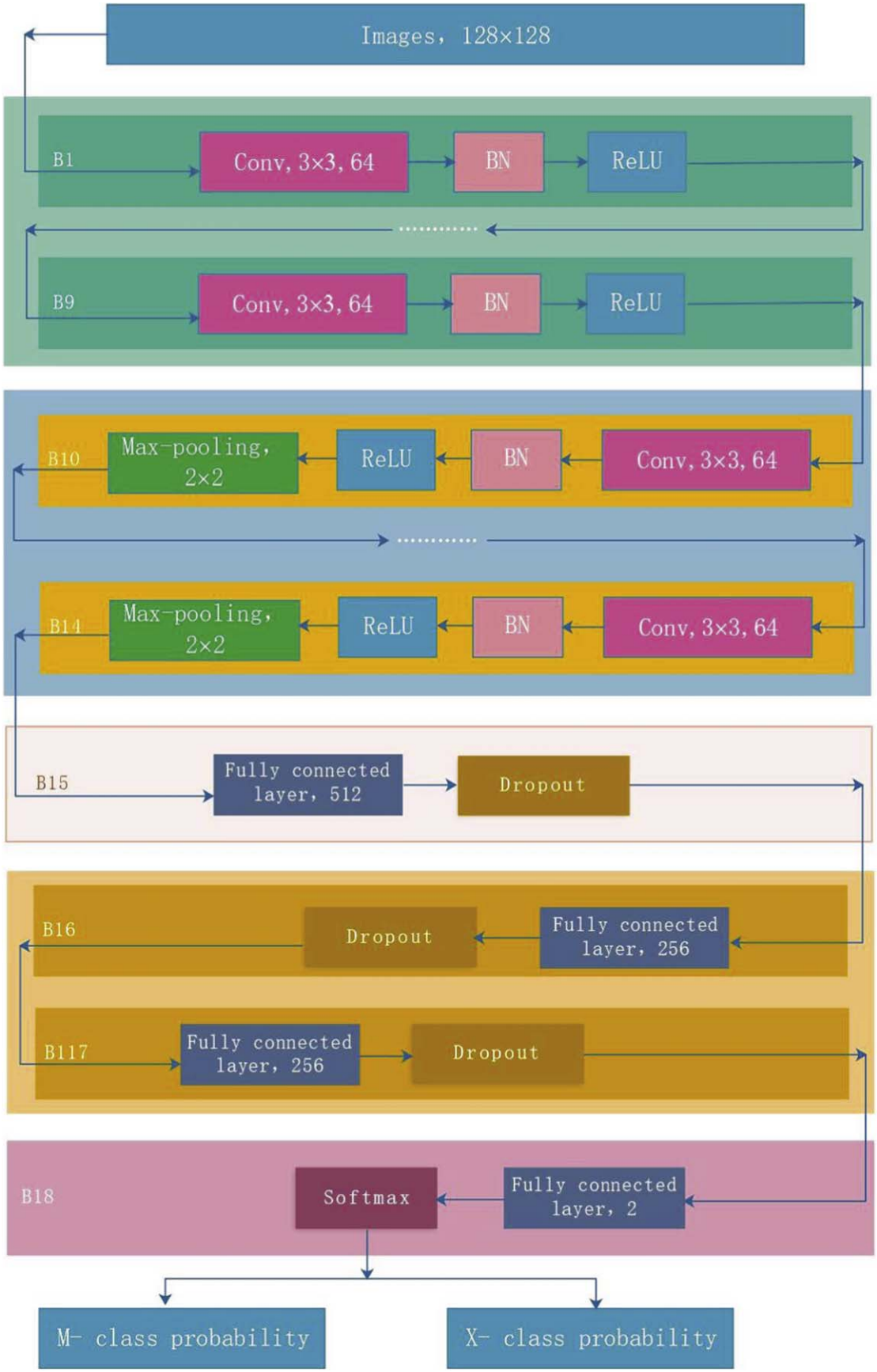

Figure 4. Architecture of the Model 2_3. The model consists of several blocks, where B1, B2, etc. represent block number, respectively. The model takes the downsampled images of size 128 × 128, and outputs the probability of M class and X class.

Table 2
The Training Parameter Settings of the Model 1 and the Proposed Hybrid CNN Model

	Optimizer	Learning Rate	Momentum	β_k	Batch Size	Epoch
Model 2_1	SGD	1.0×10^{-3}	0.5	(1, 30)	16	100
Model 2_2	SGD	5.0×10^{-4}	0.6	(1, 1.5)	16	100
Model 2_3	SGD	1.0×10^{-3}	0.7	(1, 140)	32	100
Model 1	SGD	1.0×10^{-4}	0.6	(1, 1, 25, 1000)	16	100

false positives (FP). Various well-known performance metrics are calculated from these four quantities, including recall = TP/[TP+FN], precision = TP/[TP+FP], accuracy = [TP+TN]/[TP+FP+TN+FN], false alarm ratio (FAR) = FP/[TP+FP], Heidke skill score (HSS; Heidke 1926), and the true skill statistics (TSS; Hanssen & Kuipers 1965). HSS and TSS are defined as follows:

$$\mathrm{HSS} = \frac{2[(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FN} \times \mathrm{FP})]}{(\mathrm{TP} + \mathrm{FN})(\mathrm{FN} + \mathrm{TN}) + (\mathrm{TP} + \mathrm{FN})(\mathrm{FP} + \mathrm{TN})}, \tag{5}$$

$$\mathrm{TSS} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} - \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{6}$$

In our study, we use these six metrics to evaluate our prediction results of flares. The recall, precision, and accuracy have a score range of 0–1, with 1 representing perfect prediction. FAR also has a score range of 0–1, with 0 representing perfect prediction. The score range of HSS is $-\infty$ to 1, with 1 representing perfect prediction and less than 0 representing no skill. HSS is most frequently used in flare forecasting (e.g., Barnes & Leka 2008; Yu et al. 2010), because it employs the whole contingency table to quantify the accuracy of achieving correct predictions. However, Bloomfield et al. (2012) reported that HSS changes when confronted with varying class ratio, highlighting the problem with using HSS to compare between different methods. The TSS ranges from -1 to 1, with 1 representing perfect prediction, 0 representing no skill, and -1 representing the worst score. Among these six metrics, all metrics except for the TSS are sensitive to the class-imbalance ratio. As the TSS score is unbiased over class-imbalance ratio (Woodcock 1976), it is recommended to be primary metric to evaluate the flare prediction model by recent studies (e.g., Bloomfield et al. 2012; Nishizuka et al. 2018), and we also follow the suggestion of Bloomfield et al. (2012) to adopt the TSS score as primary metric and others as secondary ones.

The validating of our models is performed during every epoch to keep track of the learning performance. The validating data sets used in the Model 2_1 are extracted from the testing data set including all four-class samples; that used in the Model 2_2 is the subset of the testing data set including only No-flare and C-class samples, and that used in the Model 2_3 is the subset of the testing data set including only M- and X-class samples. Our models are trained, validated, and tested by 10 CV data sets. We take the training, validating, and testing of the Model 2 on the first data set as an example. In the training phase, the Model 2_1 adopts a callback function named ModelCheckpoint (https://keras.io/callbacks/) in the Keras framework. The Model 2_1 is trained by the training data set over 100 epochs, and the ModelCheckpoint yields the trained model after every epoch, and then this model is validated by the validating data set after every epoch. Thus, the values of the training and validating loss are recorded at the end of every epoch. By monitoring the value of the validating loss, the ModelCheckpoint only saves the best trained model for the Model 2_1 when this value is minimum. This approach is similar to that of Park et al. (2018), which selects the best trained model among many models that are produced at every epoch. It also resembles the method of Huang et al. (2018b), which uses the early stopping strategy to stop training once performance on the testing data set stops increasing. Subsequently, the Model 2_2 and Model 2_3 perform the same operation with the Model 2_1, and then we obtain the best three trained models in total. In the testing phase, the three best trained models are combined into the hybrid model (Model 2), and the Model 2 is tested or evaluated by the testing data set in the first data set, and then it obtains the prediction result for the first data set. The detailed testing process of the Model 2 is given in Section 3. On the next nine data sets, the three models of the Model 2 repeat the above procedure, respectively. Finally, the Model 2 obtains 10 prediction results for the 10 separate data sets, and the three models obtain the results of training and validating loss per epoch, which are shown in Figures 5(a)–(f). The validating data set used in the Model 1 is also extracted from the testing data set containing all four-class samples. The training, validating, and testing of the Model 1 are similar to those of the Model 2. The Model 1 also obtains 10 prediction results for the 10 separate data sets, and the result of training and validating loss per epoch for the Model 1 is shown in Figures 5(g) and (h).

As shown in Figures 5(a)–(f), the result of training and validating loss tends to converge after 40 epochs for the Model 2_1, and after 60 epochs for the Model 2_2 and Model 2_3, respectively. However, the validating loss curves appear to fluctuate after 40 or 60 epochs, most probably because the number of ARs is small and unbalanced in the training and validating data set. The curves shown in Figure 5(f) fluctuate moderately, most probably because the number of X- and M-class ARs is very small and moderately unbalanced. The curve fluctuation in Figure 5(b) appears to be better than that in Figures 5(d) and (f). As shown in Figures 5(g) and (h), the result of training and validating loss tends to converge after 60 epochs for the Model 1. However, the validating loss curves seem to fluctuate moderately after 60 epochs, most probably because the number of four-class ARs is very small and moderately unbalanced in the training and validating data set, and there are so many categories (i.e., four classes) for the training and validating of the Model 1. Although there are fluctuations in the validating loss curves in Figure 5, the testing or prediction results of our trained models perform better in the testing phase below. In summary, Figure 5 shows that the Model 2 and Model 1 do not suffer from severe overfitting.

The four-class flare prediction results of our CNN models within 24 hr are given and compared with previous studies in recent years in Table 3. As the Model 2 and Model 1 obtain the 10 prediction results for the 10 separate data sets in the testing

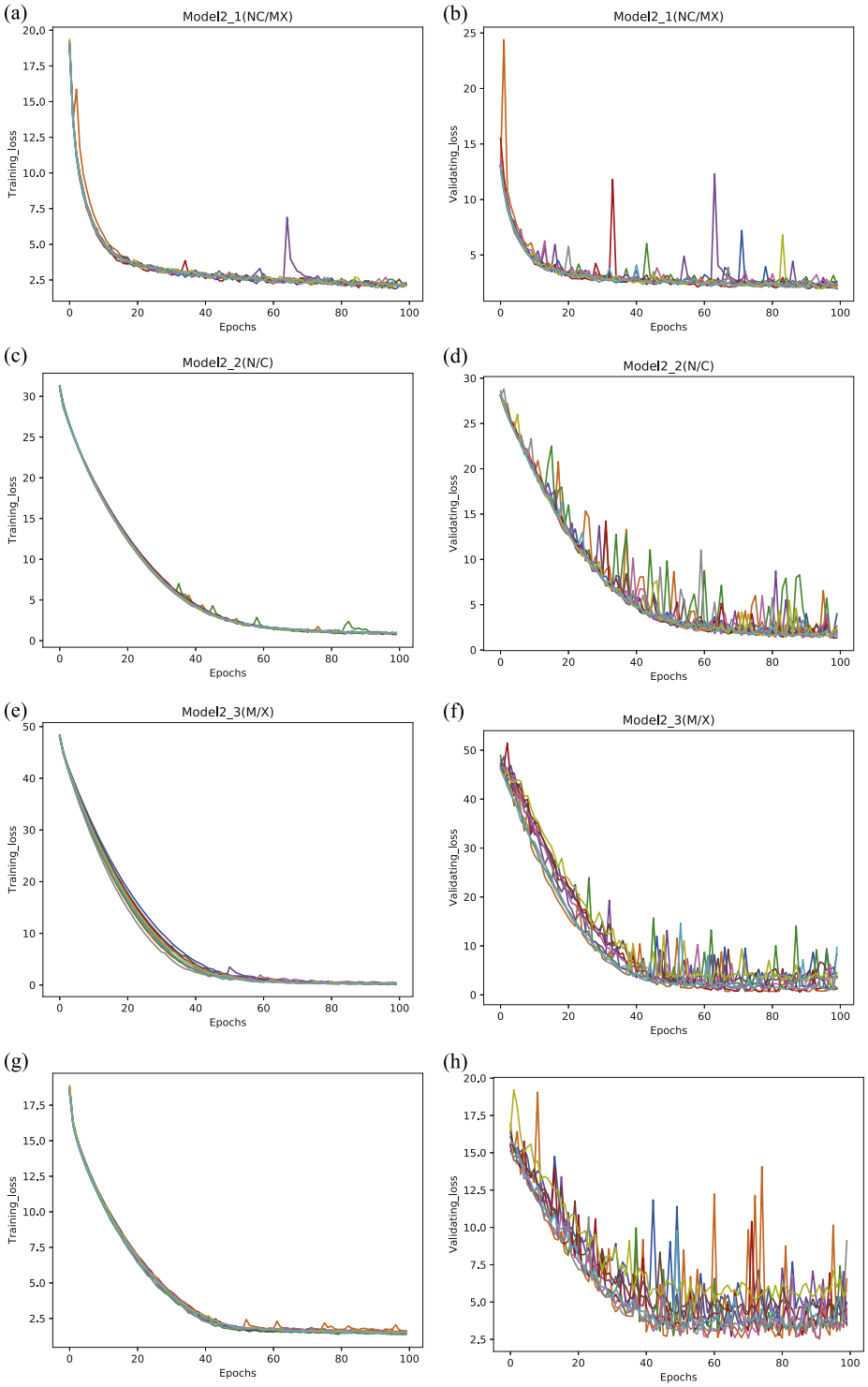

Figure 5. Learning curves showing the result of training and validating loss per epoch for the three CNN models of the proposed hybrid CNN model (Model 2), and the modified CNN model (Model 1). 10 different color curves show the changes of training and validating loss with epochs for the model trained and validated by 10 separate training and validating data sets. (a) and (b) show the result of training and validating loss per epoch for the Model 2_1, respectively. Similarly, (c) and (d) show the result of training and validating loss per epoch for the Model 2_2, respectively. (e) and (f) show the result of training and validating loss per epoch for the Model 2_3, respectively. (g) and (h) show the result of training and validating loss per epoch for the Model 1, respectively.

phase, the means and standard deviations of these prediction results are also provided in Table 3. As shown in Table 3, our proposed hybrid CNN model (Model 2) has better values of all six metrics than the modified CNN model (Model 1) in every AR class. Our average TSS scores of the Model 2 (0.768 for No-flare class, 0.538 for C class, 0.534 for M class, and 0.552 for X class) outperform Liu et al. (2017) in every AR class. Our TSS scores of the Model 2 also outperform Bloomfield et al. (2012) in every class except for X class in Table 3. It is noted that Bloomfield et al. (2012) used the best values of TSS in four classes. However, we provide the means of TSS with standard deviations in Table 3, and the best values of TSS of the Model

Table 3
The Multiclass Flare Prediction Results (within 24 hr) of our CNN Models and Comparison with Previous Studies

Metric	Model	No-flare (Weaker than C1.0) Class	C Class	M Class	X Class
Recall	This work (Model 1)	0.717 ± 0.131	0.432 ± 0.135	0.603 ± 0.127	0.143 ± 0.199
	This work (Model 2)	0.869 ± 0.034	0.671 ± 0.059	0.617 ± 0.148	0.594 ± 0.394
	Liu et al. (2017)	0.812 ± 0.039	0.526 ± 0.050	0.671 ± 0.037	0.297 ± 0.039
	Bloomfield et al. (2012)	...	0.737	0.693	0.859
	Colak & Qahwaji (2009)	...	0.772	0.865	0.917
Precision	This work (Model 1)	0.612 ± 0.058	0.495 ± 0.077	0.492 ± 0.118	0.269 ± 0.321
	This work (Model 2)	0.793 ± 0.054	0.670 ± 0.079	0.699 ± 0.087	0.562 ± 0.383
	Liu et al. (2017)	0.703 ± 0.037	0.563 ± 0.054	0.656 ± 0.036	0.745 ± 0.152
	Bloomfield et al. (2012)	...	0.330	0.136	0.029
	Colak & Qahwaji (2009)	...	...	...	...
Accuracy	This work (Model 1)	0.778 ± 0.029	0.703 ± 0.050	0.747 ± 0.073	0.840 ± 0.078
	This work (Model 2)	0.891 ± 0.018	0.812 ± 0.029	0.849 ± 0.034	0.933 ± 0.041
	Liu et al. (2017)	0.844 ± 0.017	0.712 ± 0.026	0.778 ± 0.019	0.957 ± 0.005
	Bloomfield et al. (2012)	...	0.711	0.829	0.881
	Colak & Qahwaji (2009)	...	0.811	0.944	0.981
FAR	This work (Model 1)	0.388 ± 0.058	0.505 ± 0.077	0.508 ± 0.118	0.731 ± 0.321
	This work (Model 2)	0.207 ± 0.054	0.330 ± 0.079	0.301 ± 0.087	0.138 ± 0.140
	Liu et al. (2017)	0.297 ± 0.023	0.437 ± 0.016	0.344 ± 0.020	0.255 ± 0.126
	Bloomfield et al. (2012)	...	0.670	0.864	0.971
	Colak & Qahwaji (2009)	...	0.319	0.688	0.967
HSS	This work (Model 1)	0.487 ± 0.068	0.245 ± 0.055	0.361 ± 0.131	0.117 ± 0.188
	This work (Model 2)	0.747 ± 0.037	0.535 ± 0.061	0.551 ± 0.120	0.539 ± 0.366
	Liu et al. (2017)	0.640 ± 0.032	0.334 ± 0.028	0.497 ± 0.031	0.406 ± 0.014
	Bloomfield et al. (2012)	...	0.296	0.177	0.049
	Colak & Qahwaji (2009)	...	0.493	0.470	0.169
TSS	This work (Model 1)	0.508 ± 0.080	0.241 ± 0.066	0.391 ± 0.134	0.104 ± 0.172
	This work (Model 2)	0.768 ± 0.028	0.538 ± 0.059	0.534 ± 0.137	0.552 ± 0.370
	Liu et al. (2017)	0.669 ± 0.039	0.328 ± 0.050	0.500 ± 0.037	0.291 ± 0.039
	Bloomfield et al. (2012)	...	0.443	0.526	0.740
	Colak & Qahwaji (2009)	...	...	...	...

Note. Colak & Qahwaji (2009) did not provide the values of TSS and Precision. For Liu et al. (2017), we calculate the scores of HSS and FAR from the contingency table in Table 3 they provided.

2 calculated from Table 4 (i.e., the confusion matrix of No. 4) are 0.815, 0.493, 0.746, and 0.944 for No-flare, C, M, and X class, respectively, which are better than those of Bloomfield et al. (2012). Our average TSS scores of the Model 1 (0.508 for No-flare class, 0.241 for C class, 0.391 for M class, and 0.104 for X class) are lower than previous studies. The average HSS values of the Model 2 are 0.747, 0.535, 0.551, and 0.539 for No-flare, C, M, and X class, respectively, which are much better than previous studies, while the HSS values of the Model 1 are inferior to those of Liu et al. (2017) and Colak & Qahwaji (2009), but are comparable to or better than those of Bloomfield et al. (2012). Our FAR scores of the Model 2 also outperform previous studies, and the FAR score of X class is about two times lower than that of Liu et al. (2017) and is about eight times lower than that of Bloomfield et al. (2012) and Colak & Qahwaji (2009), while the FAR scores of the Model 1 are better than those of Bloomfield et al. (2012) and Colak & Qahwaji (2009), but are inferior to those of Liu et al. (2017). In addition, we obtain fairly good scores in recall, precision, and accuracy at the same time. In summary, experiment results indicate that for multiclass flare predictions, the performance of our proposed CNN model (Model 2) appears to be much better than previous studies, while the modified CNN model (Model 1) does not work better than previous models.

Table 4 is complementary to Table 3, because the metrics of Table 3 are calculated from Table 4. Table 4 shows the values in confusion matrix for the testing of the Model 2 on each of 10 data sets. The data of prediction result are more concentrated on the principal diagonal of the confusion matrix, implying the six metrics such as TSS etc. in Table 3 are better. The best prediction results of the Model 2 are presented in the confusion matrix of No. 4 in Table 4, which mainly distribute on the principal diagonal of the confusion matrix.

Our proposed model also can be used to make binary class flare predictions. The binary $\geqslant$M-class flare prediction results (within 24 hr) of our CNN model (Model 2_1) are shown and compared with previous studies in recent years in Table 5. As shown in Table 5, we also provide the means and standard deviations of prediction results. It is worth noting that the standard deviations of all metrics are significantly small, implying the Model 2_1 is considerably stable in predicting flares larger than the M1.0 class. Our model achieves a skill score of TSS $= 0.749 \pm 0.079$ for $\geqslant$M-class flare, which is much better than Huang et al. (2018b) also using the very popular CNN method among deep learning methods. Our TSS score is remarkably better than that of Liu et al. (2017) and Bloomfield et al. (2012), and roughly comparable to that of Bobra & Couvidat (2015). We note that Bobra & Couvidat (2015) did not consider C-class flare samples in their data set,

Table 4
Values in Confusion Matrix for the Testing of the Proposed Model (Model 2) on Each of 10 Data Sets

Observation ↓ Prediction →	No-flare Class	C Class	M Class	X Class
No. 1: No-flare Class	1125	142	64	0
C Class	437	1138	172	9
M Class	17	427	561	141
X Class	0	1	136	1363
No. 2: No-flare Class	1582	134	170	0
C Class	288	1113	188	0
M Class	68	268	540	0
X Class	0	0	60	0
No. 3: No-flare Class	1158	188	36	0
C Class	389	942	48	0
M Class	59	420	1471	0
X Class	0	21	291	0
No. 4: No-flare Class	1459	101	5	0
C Class	449	911	239	36
M Class	4	140	672	12
X Class	0	3	60	1353
No. 5: No-flare Class	1464	186	14	0
C Class	358	1132	124	72
M Class	5	508	704	1
X Class	0	16	61	523
No. 6: No-flare Class	1264	216	11	9
C Class	287	1205	128	21
M Class	65	294	838	135
X Class	0	117	319	1232
No. 7: No-flare Class	1338	209	4	28
C Class	137	622	150	116
M Class	17	390	995	290
X Class	0	1	264	1211
No. 8: No-flare Class	1239	202	12	6
C Class	199	685	32	9
M Class	0	104	499	147
X Class	0	37	42	233
No. 9: No-flare Class	1186	195	3	29
C Class	165	1096	134	78
M Class	128	95	282	533
X Class	0	50	96	1294
No. 10: No-flare Class	1375	131	11	0
C Class	287	787	192	0
M Class	125	203	1064	0
X Class	0	0	252	0

and used randomly shuffled data sets with some similarity between the training and testing data sets, which probably helped distinguish positive and negative samples easily to enhance flare predictive performance. The HSS value of the Model 2_1 is 0.759 ± 0.071, which is much better than that of previous models. It can be also noted that the FAR score of the Model 2_1 is very low, about two times lower than that of Liu et al. (2017) and about eight times lower than that of Huang et al. (2018b) and Bloomfield et al. (2012). Furthermore, by comparing the other three metrics with previous studies, the skill scores are better than or comparable to those of previous models in terms of recall, precision, and accuracy. In general, experiment results reveal that the overall performance of our model is greatly improved compared with previous studies for $\geqslant$M-class major flare prediction.

Table 5
The Binary $\geqslant$M- Class Flare Prediction Results (within 24 hr) of Our CNN Model (Model 2_1) and Comparison with Previous Studies

Metric	Model	No-flare/C Class	M/X Class
Recall	This work (Model 2_1)	0.933 ± 0.026	0.817 ± 0.084
	Huang et al. (2018b)	⋯	0.850
	Liu et al. (2017)	0.785 ± 0.036	0.747 ± 0.030
	Bobra & Couvidat (2015)	⋯	0.832 ± 0.042
	Bloomfield et al. (2012)	⋯	0.704
Precision	This work (Model 2_1)	0.887 ± 0.037	0.889 ± 0.056
	Huang et al. (2018b)	⋯	0.101
	Liu et al. (2017)	0.756 ± 0.033	0.777 ± 0.033
	Bobra & Couvidat (2015)	⋯	0.417 ± 0.037
	Bloomfield et al. (2012)	⋯	0.146
Accuracy	This work (Model 2_1)	0.891 ± 0.024	0.891 ± 0.024
	Huang et al. (2018b)	⋯	0.813
	Liu et al. (2017)	0.766 ± 0.023	0.766 ± 0.021
	Bobra & Couvidat (2015)	⋯	0.924 ± 0.007
	Bloomfield et al. (2012)	⋯	0.830
FAR	This work (Model 2_1)	0.113 ± 0.037	0.111 ± 0.056
	Huang et al. (2018b)	⋯	0.899
	Liu et al. (2017)	0.244 ± 0.010	0.233 ± 0.015
	Bobra & Couvidat (2015)	⋯	⋯
	Bloomfield et al. (2012)	⋯	0.854
HSS	This work (Model 2_1)	0.759 ± 0.071	0.759 ± 0.071
	Huang et al. (2018b)	⋯	0.143
	Liu et al. (2017)	0.532 ± 0.025	0.532 ± 0.025
	Bobra & Couvidat (2015)	⋯	0.517 ± 0.035
	Bloomfield et al. (2012)	⋯	0.190
TSS	This work (Model 2_1)	0.749 ± 0.079	0.749 ± 0.079
	Huang et al. (2018b)	⋯	0.662
	Liu et al. (2017)	0.532 ± 0.036	0.532 ± 0.030
	Bobra & Couvidat (2015)	⋯	0.761 ± 0.039
	Bloomfield et al. (2012)	⋯	0.539

Note. Bobra & Couvidat (2015) did not provide the value of FAR. For Huang et al. (2018b), we calculate the scores of Precision, FAR, and Accuracy from the contingency Table 4 they provided. For Liu et al. (2017), we calculate the scores of HSS and FAR from the contingency Table 4 they provided.

5. Conclusions and Discussions

In this paper, we propose a hybrid CNN model and modify a popular CNN model to predict solar flare occurrence with the outputs of four classes (i.e., No-flare, C, M, and X) within 24 hr. The *SDO*/HMI LOS magnetograms of ARs from the SHARP from 2010 May to 2018 September covering the main peak of solar cycle 24 are considered. We collect a total of 870 ARs and 136134 magnetogram samples, including 443 X-class, 6534 M-class, 72412 C-class (weaker than C1.0), and 56745 No-flare magnetogram samples, which are categorized into a specific class according to the maximum *GOES* magnitude of the most powerful AR flare produced. Based on these AR samples, we adopt the shuffle and split CV method to build 10 separate training and testing data sets, as the number of ARs is strongly imbalanced in four classes. In each of 10 data sets, we segregate the entire data set into the training and testing data set by NOAA AR number to simulate the real-time flare prediction, which is more suited to the model evaluation and

flare predictions. It is noted that it is difficult to carry out this segregation, ensuring not only the magnetogram samples in the training data set cannot overlap with those in the testing data set, but also the ARs in the testing data set have never been seen in the training data set. However, previous studies mainly segregated their data sets into the training and testing data sets in chronological order, or selected randomly shuffled data sets with some similarity between the training and testing data set because of closeness in time. Indeed, the difference in building the training and testing data sets may affect the performance comparisons, but it is difficult to build the common training and testing data sets. In addition, we exclude some samples with multiple NOAA ARs from the data set. To alleviate the class-imbalance issue, we also utilize undersampling and data augmentation techniques in each data set. Ultimately, the resulting data sets are used to train, validate, and evaluate our models, and our models do not suffer from excessive overfitting.

The main results from this study are summarized as follows. (1) To our knowledge, this is the first time that the CNN models are used to predict multiclass solar flares, without manually engineered features extracted from the observational data, and our models adopt the very popular CNN method among deep learning methods, while previous models utilize statistical or classic machine-learning methods. (2) For multi-class flare predictions, our proposed hybrid CNN model (Model 2) has better values of all statistical scores than the modified CNN model (Model 1) in every class, and the Model 2 is more suitable for predicting multiclass flare. (3) The performance of the Model 2 is greatly improved compared with previous models for multiclass flare predictions in terms of TSS. The Model 2 achieves higher mean scores of TSS $= 0.768$ for No-flare class, 0.538 for C class, 0.534 for M class, and 0.552 for X class, which are much better than that of Liu et al. (2017) in every class. The best TSS scores of the Model 2 are 0.815, 0.493, 0.746, and 0.944 for No-flare, C, M, and X class, respectively, which are better than that of Bloomfield et al. (2012) in every class. (4) The performance of our model (Model 2_1) is noticeably better than previous studies for $\geqslant$M-class major flare prediction in terms of TSS. For example, Huang et al. (2018b) achieved skill score of TSS $= 0.662$ for $\geqslant$M-class flare and TSS $= 0.49$ for $\geqslant$C-class flare, and Park et al. (2018) achieved skill score of TSS $= 0.63$ for $\geqslant$C-class flare, while our model achieves a higher skill score of TSS $= 0.749 \pm 0.079$ for $\geqslant$M-class flare, which is much better than Huang et al. (2018b) and Park et al. (2018) all using CNN methods. Moreover, our proposed model obtains quite good scores in the other five metrics for both multiclass flare predictions and $\geqslant$M-class major flare predictions. Experiment results show that our proposed model achieves an unprecedented performance in flare prediction. Therefore, we speculate that there may be some previously undiscovered features that could reveal the flare eruption mechanism, which are automatically extracted by the convolution filters of our model. However, the learned convolution filters automatically detecting features in image data act more like a black box, and these features involve high-level information that is not easily understood. Our current research has not been able to identify the specific features that would lead to such positive results. In near future work, we attempt to open this black box to study these high-level features associated with solar flare, to further improve predictive performance of our model.

It can be seen from Table 3 that the standard deviations of all metrics in No-flare and C class are very small, indicating our models are quite stable in predicting No-flare (weaker than C1.0) and C-class flare, while those of all metrics except for accuracy in class X and M are close to or larger than 0.1, indicating our models are not very stable in predicting M/X-class flare. The higher standard deviation is mainly due to the fact that the number of X-class ARs and samples is seriously insufficient for our models. Fortunately, some of X-class samples are mostly wrongly predicted as class M in Table 4 (e.g., the confusion matrix of No. 2, No. 3, and No. 10), in the case that they are not correctly predicted, which indicates the correct predictions of $\geqslant$M-class major flares are not missed. Based on all our experiments, our proposed hybrid CNN model is a valid method for flare forecasting with fairly reasonable prediction performance. In the near future, as the *SDO*/HMI continues to observe, we could obtain more LOS ARs and samples in X class to further optimize the Model 2_3 of the proposed CNN model, which can help improve the ability to predict M/X-class flare steadily. We will also consider *SDO*/HMI vector magnetograms and the extreme ultraviolet images of ARs in our data sets to further improve the performance of multiclass flare predictions of our model. In addition, as solar observational data from various instruments is growing rapidly, including those such as from *SDO* and the latest ground-based Hα images from the New Vacuum Solar Telescope (Li et al. 2015), the CNN methods may be helpful in solving other multiclass problem in solar physics.

We acknowledge the referee for his/her great effort to anonymously review our paper and for giving us very valuable and useful comments. The data used here is courtesy of NASA/*SDO* and the HMI science team, as well as the *GOES* team. This work is supported by the National Natural Science Foundation of China (grants No. 11703009, No. 11803010), and the Natural Science Foundation of Jiangsu Province, China (grant No. BK20170566).

ORCID iDs

Xuebao Li https://orcid.org/0000-0003-0397-4372

References

Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467
Ahmed, O. W., Qahwaji, R., Colak, T., et al. 2013, SoPh, 283, 157
Arel, I., Rose, D. C., & Karnowski, T. P. 2010, IEEE Computational Intelligence Magazine, 5, 13
Barnes, G., & Leka, K. D. 2008, ApJL, 688, L107
Barnes, G., Leka, K. D., Schrijver, C. J., et al. 2016, ApJ, 829, 89
Bloomfield, D. S., Higgins, P. A., James McAteer, R. T., & Gallagher, P. T. 2012, ApJL, 747, L41
Bobra, M. G., & Couvidat, S. 2015, ApJ, 798, 135
Bobra, M. G., Sun, X., Hoeksema, J. T., et al. 2014, SoPh, 289, 3549
Boureau, Y. L., Ponce, J., & LeCun, Y. 2010, in Proc. 27th Int. Conf. on Machine Learning (ICML-10) (Madison, WI: Omnipress), 111
Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, ApJ, 863, 97
Colak, T., & Qahwaji, R. 2009, SpWea, 7, S06001
Florios, K., Kontogiannis, I., Park, S. H., et al. 2018, SoPh, 293, 28
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (Cambridge, MA: MIT Press), http://www.deeplearningbook.org/
Guerra, J. A., Pulkkinen, A., & Uritsky, V. M. 2015, SpWea, 13, 626
Hanssen, A. W., & Kuipers, W. J. A. 1965, Meded. Verh., 81, 2
Heidke, P. 1926, Geogr. Ann., 8, 301
Hinton, G., Deng, L., Yu, D., et al. 2012, ISPM, 29, 82
Hinton, G. E., & Salakhutdinov, R. R. 2006, Sci, 313, 504
Huang, G., Liu, Z., Maaten, L., et al. 2018a, arXiv:1608.06993
Huang, X., Wang, H., Xu, L., et al. 2018b, ApJ, 856, 7

Huang, X., Yu, D. R., Hu, Q. H., Wang, H., & Cui, Y. 2010, SoPh, 263, 175
Ioffe, S., & Szegedy, C. 2015, arXiv:1502.03167
Krizhevsky, A., Sutskever, I., & Hinton, G. 2012, in Proc. Int. Conf. on Advances in Neural Information Processing Systems, ed. F. Pereira et al. (La Jolla, CA: Neural Information Processing Systems Foundation), 1097
LeCun, Y., Bengio, Y., & Hinton, G. 2015, Natur, 521, 436
LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. 1998, in Neural Networks: Tricks of the Trade, ed. G. Montavon, G. Orr, & K.-R. Müller (Berlin: Springer), 9
Li, R., & Zhu, J. 2013, RAA, 13, 1118
Li, X. B., Liu, Z., Wang, F., et al. 2015, PASJ, 67, 47
Liu, C., Deng, N., Wang, J. T. L., & Wang, H. M. 2017, ApJ, 843, 104
Mason, J. P., & Hoeksema, J. T. 2010, ApJ, 723, 634
Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. on Machine Learning (ICML-10) (Madison, WI: Omnipress), 807
Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., & Ishii, M. 2017, ApJ, 835, 156
Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2018, ApJ, 858, 113
Park, E., Moon, Y. J., Shin, S., et al. 2018, ApJ, 869, 91
Pesnell, W. D., Thompson, B. J., & Chamberlin, P. C. 2012, SoPh, 275, 3
Qahwaji, R., & Colak, T. 2007, SoPh, 241, 195
Sadykov, V. M., & Kosovichev, A. G. 2017, ApJ, 849, 148
Schou, J., Scherrer, P. H., Bush, R. I., et al. 2012, SoPh, 275, 229
Simonyan, K., & Zisserman, A. 2015, arXiv:1409.1556
Song, H., Tan, C., Jing, J., et al. 2009, SoPh, 254, 101
Sutskever, I. 2013, PhD thesis, Univ. Toronto
Szegedy, C., Liu, W., Jia, Y., et al. 2014, arXiv:1409.4842
Woodcock, F. 1976, MWRv, 104, 1209
Yu, D. R., Huang, X., Wang, H. N., et al. 2010, ApJ, 710, 869
Yuan, Y., Shih, F. Y., Jing, J., & Wang, H. M. 2010, RAA, 10, 785