The Astrophysical Journal, 869:91 (6pp), 2018 December 20 https://doi.org/10.3847/1538-4357/aaed40

Application of the Deep Convolutional Neural Network to the Forecast of Solar Flare Occurrence Using Full-disk Solar Magnetograms

Eunsu Park, Yong-Jae Moon, Seulki Shin, Kangwoo Yi, Daye Lim, Harim Lee, and Gyungin Shin
School of Space Research, Kyung Hee University, 1732, Deogyeongdae-ro, Giheung-gu, Yongin-si Gyunggi-do, 17104, Republic of Korea; moonyj@khu.ac.kr
Received 2018 April 25; revised 2018 October 30; accepted 2018 October 30; published 2018 December 14

Abstract

In this study, we present the application of the Convolutional Neural Network (CNN) to the forecast of solar flare occurrence. For this, we consider three CNN models (two pretrained models, AlexNet and GoogLeNet, and one newly proposed model). Our inputs are *SOHO*/Michelson Doppler Imager (from 1996 May to 2010 December) and *SDO*/Helioseismic and Magnetic Imager (from 2011 January to 2017 June) full-disk magnetograms at 00:00 UT. Model outputs are the "Yes or No" of daily flare occurrence (C, M, and X classes) and they are compared with *GOES* observations. We train the models using the input data and observations from 1996 to 2008, covering the entire solar cycle 23, and test them using the data sets from 2009 to 2017, covering solar cycle 24. Then we compare the results of the CNN models with those of three previous flare forecast models in view of statistical scores. The major results from this study are as follows. First, we successfully apply CNN to the full-disk solar magnetograms without any preprocessing or feature extraction. Second, the results of our CNN models are slightly better in Heidke skill score and true skill statistics, and considerably better in false alarm ratio (FAR) and critical success index than the previous solar flare forecasting models. Third, our proposed model has better values of all statistical scores except for FAR, than the other two pretrained models. Our results indicate a sufficient possibility that deep learning methods can improve the capability of the solar flare forecast as well as similar types of forecast problems.

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

1. Introduction

A solar flare is a phenomenon that releases a very broad spectrum of emissions and a large amount of high energy particles from the Sun into interplanetary space. The X-ray enhancement due to a strong solar flare can affect the radio communication system, the global positioning system, the safety of satellites and astronauts, especially can cause increased ionization of the upper atmosphere, which can cause short-wave radio fade-outs (Wheatland 2005; Huang et al. 2018), and these hazards can result in numerous economic and commercial losses. Thus, it is very important to develop solar flare prediction models for space weather forecast.

Over the past decades, many solar flare forecast models have been developed using two representative methods: statistical and machine-learning algorithms. Many authors have presented the solar flare forecast models based on statistics using several types of input data such as solar flare occurrence rate for each sunspot class (e.g., Lee et al. 2007, 2012; Mason & Hoeksema 2010; Falconer et al. 2011; Bloomfield et al. 2012; Falconer et al. 2014; Barnes et al. 2016; Anastasiadis et al. 2017; Park et al. 2017). In the past decade, several authors have applied machine learning to the solar flare forecast models with various types of algorithms such as Support Vector Machine, Random Forest, Multi-Layer Perceptron, and Artificial Neural Network (e.g., Colak & Qahwaji 2009; Yuan et al. 2010; Li et al. 2011; Ahmed et al. 2013; Huang et al. 2013; Li & Zhu 2013; Al-Ghraibah et al. 2015; Bobra & Couvidat 2015; Muranushi et al. 2015; Barnes et al. 2016; Shin et al. 2016; Liu et al. 2017; Nishizuka et al. 2017; Raboonik et al. 2017).

Recently, Deep Neural Networks (DNNs), called "Deep Learning," and one of the machine-learning algorithms, have been developed to learn how humans think and recognize an object using their deep hierarchical layer structure. They have become very popular in several fields due to the large amount of data, advanced hardware, and improvements in machine-learning algorithms they provide.

The Convolutional Neural Network (CNN; LeCun et al. 1998), which is a well-known deep learning method, is designed to imitate an animal's visual cortex and has become very popular in the area of image processing and computer vision. The CNN model consists of several convolutional filters (or layers) and the filters extract (or calculate) several features (or parameters) from input data sets. The CNN model trains the filters to select and extract features automatically without human knowledge, while a traditional machine-learning model requires feature extractors that manually select and extract the features.

Recently, there have been a few attempts to apply deep learning to flare forecast. Huang et al. (2018) presented a solar flare forecast model based on CNN. Their model used many patches of solar active regions (ARs) obtained from the solar magnetograms located within $\pm 30^{\circ}$ of the central meridian to avoid projection effects. Nishizuka et al. (2018) also presented a DNN model, which is not CNN but rather a multi-layer perceptron model, for solar flare forecast based on a deep residual network (He et al. 2015). This study considered 79 manually selected solar features, which were extracted from solar magnetograms, EUV images, and *GOES* X-ray flux data. It requires a feature extraction process before training, testing, and applying deep learning models.

In this study, we apply CNN to the forecast of solar flare occurrence using only solar full-disk magnetograms without any data preprocessing, while the previous studies perform data preprocessing to use the data sets as the input of their models. For this we consider two pretrained CNN models and our proposed CNN model. Input data for the models are original full-disk longitudinal magnetograms taken by the *Solar and Heliospheric Observatory* (*SOHO*; Domingo et al. 1995)/Michelson Doppler Imager (MDI; Scherrer et al. 1995) and the

Solar Dynamic Observatory (*SDO*; Pesnell et al. 2012)/ Helioseismic and Magnetic Imager (HMI; Schou et al. 2012). No type of image processing and feature extraction of input data has ever been made. Model outputs are 1 (Yes) or 0 (No) of daily flare occurrence (C, M, and X classes) and they are compared with *Geostationary Operational Environment Satellite* (*GOES*) observations of the daily flare occurrence.

This study is organized as follows. The data will be described in Section 2, and the method in Section 3. Results are given in Section 4, and a brief summary is presented in Section 5.

2. Data

We use solar *SOHO*/MDI and *SDO*/HMI full-disk line-of-sight magnetograms at every 00:00 UT as the input data of the CNN models. *SOHO*, which was launched on 1995 December 2, was a space mission that was part of the Solar-Terrestrial Science Program, developed in a collaborative effort by the European Space Agency and the National Aeronautic and Space Administration (Domingo et al. 1995). MDI, which is one of the onboard instruments on *SOHO*, was designed to provide velocity, continuum intensity, and line-of-sight magnetic field data (Scherrer et al. 1995). MDI has been superseded by *SDO*/HMI. *SDO*, which was launched on 2010 February 11, is a spacecraft mission that investigated how the Sun's magnetic field is generated and structured, how this stored magnetic energy is released into the heliosphere and geospace as the solar wind, energetic particles, and variations in the solar irradiance with three instruments (Pesnell et al. 2012). HMI is an instrument of the *SDO*, which is designed to measure the Doppler shift, intensity, and vector magnetic field at the solar photosphere (Schou et al. 2012). We select full-disk line-of-sight *SOHO*/MDI magnetograms from 1996 May to 2010 December for every 00:00 UT and those of *SDO*/HMI from 2011 January to 2017 June. We have calculated pixel-to-pixel correlations of the flux per pixels between MDI images and rebinned HMI images within 30 minute time difference during 2011 January. The average correlation coefficient is 0.96, which is significantly high enough to use these two data sets as the input data. Output data are the daily *GOES* observations of X-ray flare occurrence for "No-flare" (weaker than C 1.0) and "Flare" (equal to or stronger than C 1.0) from 00:00 UT to 24:00 UT and they are compared with model outputs. As a result, we select 4981 *SOHO*/MDI magnetograms and 2360 *SDO*/HMI magnetograms (total 7341 magnetograms), which are fully labeled for flare occurrence using *GOES* observations. We separate our data sets into the training data sets and the test data sets in chronological order, which is known to be more suitable for the model evaluation. Nishizuka et al. (2017) noted that the performance of forecasting models could depend on how the data sets are separated into training and testing. For example, randomly selected data sets have some similarity between the training and the test data sets because of closeness in time. In addition, evaluating the models through randomly separated data sets should carefully be considered because it can evaluate past events using future events. Thus, the comparison with forecasting models using the same data sets could be meaningful (Lera & Barnes 2017). For this reason, we use data sets in chronological order to train and test our model, and to compare our method with other methods that used chronological data sets (training using older and test using the remaining). For the training, we use the data sets (4298 images) from 1996 May to 2008 December, covering the entirety of solar cycle 23. For the test, we use the data sets (3043 images) from 2009 January to 2017 June, covering solar cycle 24. All information regarding the data sets is given in Table 1.

Table 1
The Number of Solar Magnetogram Images Used for Our Research

	No-Flare (<C)	Flare Total			Total(Non-Flare +Flare)
		C	M	X	
MDI	2485	1675	719	102	4981
		2496			
HMI	784	1168	368	40	2360
		1576			
Training	1895	1596	705	102	4298
		2403			
Test	1374	1247	382	40	3043
		1669			
Total	3269	2843	1087	142	7341
		4072			

3. Deep CNNs

CNN, which is one of the deep learning methods, has recently become popular in the area of image processing and computer vision. It has been widely used in various fields such as image recognition, object detection, image segmentation, auto-driving car, and playing Go, known as the most difficult board game in the world. In general, CNN consists of several convolution layers, which are some of the most important components of CNN models. An input of a convolution layer is an image, and outputs are several convolved images, which are called feature maps. To make these output feature maps, the convolution layer convolves its input image with a set of filters (kernels). When we apply the convolution layer, we can select three hyper-parameters: (1) kernel size (or filter size), (2) filter depth, and (3) stride. The kernel size is the pixel dimension of convolution filters in a convolution layer. The filter depth refers to the number of convolution filters, which controls the number of output feature maps. The stride determines how many pixels the filters jump in every convolution in a convolution layer. After each convolution layer, we use an activation function that determines whether to forward the signal received from the previous layer to the next layer. Another important component for CNN models is pooling layers, which summarize the output feature maps to reduce a spatial size of output feature maps and amounts of operations by maximizing or averaging. A number of CNN models include max-pooling layers that adopt maximum value for each small patch of the output feature maps. To make a CNN model act like a classifier, the CNN model must include one or more fully connected layers. A fully connected layer, which looks like conventional neural networks, consists of a number of nodes, and each node is densely

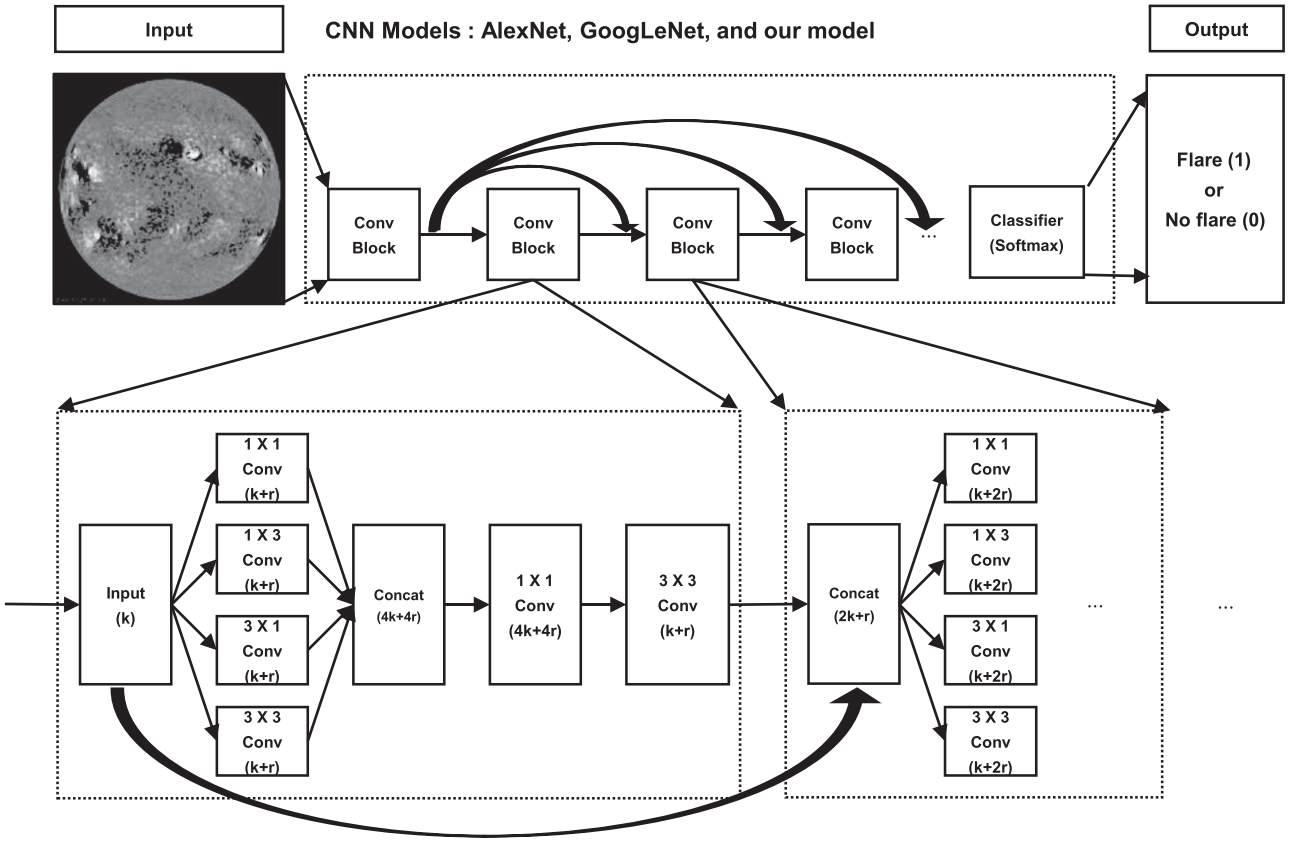

Figure 1. Flowchart for flare forecast and structures of our proposed model. Our proposed model consists of several convolution blocks and each block has stacks of convolution layers. k is the number of feature maps of the input data of a convolution block, and r is the growth rate.

connected with all nodes in another fully connected layer. To reduce an over-fitting problem, many of the CNN models include the dropout technique, which randomly excludes nodes for training to reduce overdependence on specific nodes and to train every node equally. Most of the CNN models include stacks of one or more convolution, pooling, and fully connected layers.

In this study, we consider three CNN models, two pretrained models and one proposed model. One of the pretrained models is AlexNet (Model 1), which is presented by Krizhevsky et al. (2012), the winner of the ImageNet Large Scale Visual Recognition Challenge (IMAGENET Challenge) in 2012. AlexNet consists of five convolutional layers, three max-pooling layers, and two fully connected layers with dropout. AlexNet applied several techniques to improve their scores such as the Rectified Linear Unit (ReLU) instead of "tanh" to add nonlinearity, dropout to deal with over-fitting, and overlap pooling to reduce the size of the network. The "tanh" is one of the activation functions forwarding a signal of -1 or 1 to the next layer. In the case of the ReLU layer, the signal from the previous layer can be forwarded when the signal is greater than 0; otherwise, the signal is not transmitted. AlexNet used overlap pooling, which applies a pooling size of 3×3 and strides of 2×2. Each pixel in the results of overlap pooling shares information from the previous layer. We implement this model on MatConvNet[1] based on MATLAB. The other pretrained model is GoogLeNet (Model 2), presented by Szegedy et al. (2014), the winner of the IMAGENET Challenge in 2014. GoogLeNet applied inception modules, including various scales of convolution filters, to make their model deeper with less computing power. Inception modules consist of multiple convolution layers, which are designed to have a small amount of computation but to show equivalent computational performance (Szegedy et al. 2014). In this study, GoogLeNet consists of 9 inception modules, 64 convolution layers, and 16 pooling layers. We implement GoogLeNet on DIGITS,[2] a deep learning GUI based on Caffe,[3] developed by NVIDIA. We expect that well-known pretrained models exhibit good performance for not only IMAGENET Challenge data but also space weather problems because they can efficiently extract features from their data sets. In addition, we design a CNN model (Model 3), which is a kind of combination of GoogLeNet (Szegedy et al. 2014) and DenseNet (Huang et al. 2016). Figure 1 shows a structure of our proposed model. Our proposed model consists of modules and each module has six convolutional blocks. Each convolutional block has six convolution layers with filter sizes of 1×1, 1×3, 3×1, and 3×3, and every block in a module has direct connectivities with other blocks. Our model first starts with 16 output feature maps, and the number of feature maps gradually increase by "Growth rate" (r) in every convolution block. In our experiment, we select 1024 as the input image size, 4 as the number of modules, 16 as the growth rate. After all convolution layers, we add Batch Normalization (BN) layers (Ioffe & Szegedy 2015) to stabilize the training and regularize input data of convolution layers, and the equation of the BN

[1] https://github.com/vlfeat/matconvnet

[2] https://github.com/NVIDIA/DIGITS

[3] https://github.com/BVLC/caffe

layer is given by

$$y = \frac{\gamma x}{\sqrt{\mathrm{Var}[x] + \epsilon}} + \left(\beta - \frac{\gamma \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}}\right), \quad (1)$$

where x is an input batch, y is an output batch, $\mathrm{E}[x]$ and $\mathrm{Var}[x]$ are the average and the dispersion of the input batch, ϵ is a small value for avoiding zero-dividing error, and β and γ are selected parameters. We add ReLU layers behind all BN layers to add nonlinearity. We build, train, and test this proposed model on TensorFlow developed by Google.

A softmax layer is located at the end of the model to classify the flare occurrence by normalizing the output values and its equation is given by

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j^N e^{x_j}}, \quad (2)$$

where x_i is a value in the output of the CNN models, and N is the number of categories. For example, when percentage values of "Flare" in the output of the softmax layer are larger than those of "No-flare," we predict the occurrence of flares. In general, deep learning models train themselves to minimize their own loss functions assigned by users. To train our CNN models, we use the cross entropy given by

$$L = -\sum y_i \log(\hat{y}_i), \quad (3)$$

where y_i denotes the observations, and $\hat{y}_i$ denotes the model outputs. To minimize the loss function, we use the ADAM optimizer, which is adaptive momentum estimation (Kingma & Ba 2014), with a learning rate of 2×10^{-4}.

In our study, there are two steps, one is the training step and the other is the test step. In the step of the training, we use magnetograms that are labeled by the flare occurrence (Yes or No) according to *GOES* X-ray peak flux. The CNN model predicts an answer using the input data and modifies itself to increase the fraction of correct answers by minimizing a loss function, which is defined as the difference between the model output and the observation. In the testing step, we use nonlabeled magnetograms and the model predicts an answer using input data just like the step of the training. Then, we compare the answer with actual daily flare occurrence to assess the accuracy of the model.

4. Results and Discussion

To evaluate the models, we consider several statistical scores using the two-by-two contingency table shown in Table 2. The "Hit" is the number of cases where the event was observed and successfully predicted. The "Miss" is where the event was observed, but not predicted. The "False alarm" is where the event was predicted, but not observed. The "Null" is the number of cases where the event was not predicted and not observed.

In this study, we consider six statistical scores: Accuracy (ACC), Probability of Detection (POD), Critical Success Index (CSI), False Alarm Ratio (FAR), Heidke Skill Score (HSS), and True Skill Statistics (TSS). ACC measures the rate of correct prediction. POD represents the ratio of the number of events correctly forecast to the number of total observed events. CSI is a more balanced score because CSI includes false alarms. The range of ACC, POD, and CSI are 0 to 1, with a perfect prediction score 1. FAR measures a rate of incorrect forecast events by the ratio of false alarms to the total number of forecast events. The range of FAR is also 0 to 1, with a perfect prediction score 0. HSS indicates the improvement of the forecast over always forecasting that no flare will occur (Heidke 1926; Barnes & Leka 2008). This HSS has a range from $-\infty$ to 1, which indicates no skill when lower than 0 and perfect 1. TSS is a parameter that distinguishes well between "Yes" cases and "No" cases and allows one to examine whether forecasting an event leads to a significant increase in false alarms or not (Hanssen & Kuipers 1965). The range of TSS is -1 to 1, with 1 as the perfect score, 0 as no skill score, and -1 as the worst score. HSS is a good statistical score for measuring the forecast skill for rare events, but it is noted that it changes according to the ratio of the number of no-event and the number of event observations. Bloomfield et al. (2012) reported that HSS changes despite the prediction held constant, highlighting the problem with using HSS to compare between different methods. This study suggested TSS as an alternative parameter, which does not depend on the ratio of the number of no-flare and the number of flare observations, and several studies verified their models using TSS parameters. Therefore, we mainly consider TSS as a primary parameter and others as secondary ones. In order to obtain the results optimized for TSS, we select the best TSS model among many models that are produced at every epoch time, where one epoch is when an entire training data set is passed through our models for training.

Table 2
Contingency Table for Prediction and Observation, and Statistical Scores from the Contingency Table

	Observation	
Prediction	Yes	No
Yes	Hit(H)	False Alarm(F)
No	Miss(M)	Null(N)
ACC (Accuracy)	(H+N)/(H+F+M+N)	
POD (Probability of Detection)	H/(H+M)	
CSI (Critical Success Index)	H/(H+F+M)	
FAR (False Alarm Ratio)	F/(H+F)	
HSS (Heidke Skill Score)	2[(H x N)-(M x F)]/[(H+M)(M+N) +(H+F)(F+N)]	
TSS (True Skill Statistics)	H/(H+M)-F/(F+N)	

As shown in Table 3, we compare our CNN models with three previous models by Colak & Qahwaji (2009), Bloomfield et al. (2012), and Huang et al. (2018) because they used chronological selected data sets (training using older data and test using the remaining data) as we did. We do not include the other models (e.g., Nishizuka et al. 2017) that consider nonchronological data sets (randomly or alternately selected for training and testing). The model by Colak & Qahwaji (2009) is based on a machine-learning algorithm, that of Bloomfield et al. (2012) is based on statistics, and that of Huang et al. (2018) is based on deep learning. Our proposed

Table 3
The Results of Our CNN Models and the Comparison with Three Previous Flare Forecast Models

	This Work			Colak & Qahwaji (2009)	Bloomfield et al. (2012)	Huang et al. (2018)
	Model 1	Model 2	Model 3			
ACC	0.78	0.79	0.82	0.81	0.71	0.76
POD	0.72	0.84	0.85	0.81	0.75	0.73
CSI	0.64	0.68	0.73	⋯	⋯	0.31
FAR	0.14	0.21	0.17	0.30	0.65	0.65
HSS	0.57	0.57	0.63	0.51	0.32	0.34
TSS	0.57	0.56	0.63	⋯	0.46	0.49
Data sets	Training:1996–2008 Test:2009–2017			Training:1972–1998 Test:1999–2002	Training:1988–1996 Test:1996–2010	Training:1996–2010 Test:2010–2015

Note. Colak & Qahwaji (2009) did not provide the values of CSI and TSS, and Bloomfield et al. (2012) did not provide the values of CSI.

model has better values of all statistical scores, except the FAR, than the other two pretrained models. The CNN models have ACC values larger than 0.78, which are comparable with those of the previous models. The POD values of our CNN models are larger than 0.72, which are also similar to those of the previous models. The CSI values of our CNN models are greater than 0.64, which are much better than that of Huang et al. (2018). The FAR values of our CNN models are less than 0.21, which are a little better than that of Colak & Qahwaji (2009) and remarkably better than those of Bloomfield et al. (2012) and Huang et al. (2018). All CNN models have HSS values greater than 0.57, which are a little better than that of Colak & Qahwaji (2009) and much better that those of Bloomfield et al. (2012) and Huang et al. (2018). The TSS values of our CNN models are larger than 0.56, which are better than those of the previous models. In particular, the TSS value of our proposed model (Model 3) is noticeably better than those of the previous models. In summary, our models seem to be much better than the previous models in view of several statistical scores.

To estimate the stability of our proposed model, we consider the 10-fold cross-validation. For this, we separate our data sets into 10 parts by random selection, then we use 9 parts as the training data sets, and the remaining one part as the test data set. Then we repeat the operation 10 times and calculate their mean performance. As shown in Table 4, 6 statistical scores are slightly better than those of our proposed model in Table 3 and the standard deviations are significantly small. These results imply that our proposed model is quite stable regardless of data sets.

Table 4
The Results of 10-fold Cross-validation from Our Proposed Model

	ACC	POD	CSI	FAR	HSS	TSS
Mean	0.84	0.83	0.75	0.11	0.69	0.69
StdDev	0.014	0.034	0.023	0.019	0.027	0.025

5. Conclusion and Summary

We have applied CNN to the forecast of solar flare occurrence using solar full-disk magnetograms from 1996 to 2017. We selected 7341 solar magnetograms at 00:00 UT including 4981 *SOHO*/MDI magnetograms and 2496 *SDO*/HMI magnetograms. We separated our data sets into the training data sets and the test data sets in chronological order, which is suitable for forecast. For training, we have used the data sets (4298 images) from 1996 May to 2008 December, covering the entirety of solar cycle 23. As a test, we have used the data sets (3043 images) from 2009 January to 2017 June, covering solar cycle 24. We have considered two well-known pretrained models and our proposed model.

The main results from this study are as follows. First, we successfully apply CNN to the full-disk solar magnetograms without any preprocessing or feature extraction. Second, the results by our CNN models are slightly better in HSS and TSS, and quite better in FAR and CSI than the previous solar flare forecasting models. Third, our proposed model has better values of all statistical scores except for FAR, than the other two pretrained models.

This study is an attempt to apply CNN to the forecast of solar flares. Our results are very impressive in view of several statistical scores and give us several implications in view of further developments of space weather forecast models such as the solar flare forecast as follows. First, the input data are full-disk solar magnetograms without any preprocessing of data, which are much simpler than those of the previous models. Second, our trained CNN models need only a few milliseconds to produce prediction results using the input data, which is enough to predict solar flares at a near-real time basis. Third, our results based on statistical scores show that CNN models are very effective for predicting solar flares because they are trained to extract some features from the location and size of ARs as well as the structure of ARs. Fourth, we successfully obtain small FAR values and large CSI values compared to the previous models by applying CNN to the flare forecasts, which are very important for space weather forecasters and operators of spacecraft because they can reduce their cost and loss. In addition, our results indicate a sufficient possibility that "Deep Learning" methods can improve the capability of solar flare forecasts. We have a plan to solve several space weather problems using "Deep Learning" methods such as Long-Short Term Memory, Generative Adversarial Network, and Reinforcement Learning.

We thank the numerous team members who have contributed to the success of the *SDO* mission, as well as the *SOHO* mission. We acknowledge the community effort devoted to the development of the following open-source packages that were used in this work: NumPy (numpy.org), Keras (keras.io), SunPy (sunpy.org), and TensorFlow (tensorflow.org). This work was supported by the BK21 plus program through the National Research Foundation (NRF) funded by

the Ministry of Education of Korea, the Basic Science Research Program through the NRF funded by the Ministry of Education (NRF-2016R1A2B4013131), NRF of Korea Grant funded by the Korean Government (NRF-2013M1A3A3A02042232), the Korea Astronomy and Space Science Institute under the R&D program supervised by the Ministry of Science, ICT and Future Planning, the Korea Astronomy and Space Science Institute under the R&D program "Development of a Solar Coronagraph on International Space Station (Project No. 2017-1-851-00)" supervised by the Ministry of Science, ICT and Future Planning, and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2018-0-01422, study on analysis and prediction technique of solar flares).

ORCID iDs

Eunsu Park https://orcid.org/0000-0003-0969-286X
Yong-Jae Moon https://orcid.org/0000-0001-6216-6944
Harim Lee https://orcid.org/0000-0002-9300-8073

References

Ahmed, O. W., Qahwaji, R., & Colak, T. 2013, SoPh, 283, 157
Al-Ghraibah, A., Boucheron, L. E., & McAteer, R. T. J. 2015, A&A, 579, A64
Anastasiadis, A., Papaioannou, A., Sandberg, I., et al. 2017, SoPh, 292, 134
Barnes, G., & Leka, K. D. 2008, ApJ, 688, L10
Barnes, G., Leka, K. D., Schrijver, C. J., et al. 2016, ApJ, 829, 89
Bloomfield, D. S., Higgins, P. A., James McAteer, R. T., et al. 2012, ApJL, 747, L41
Bobra, M. G., & Couvidat, S. 2015, ApJ, 798, 135
Colak, T., & Qahwaji, R. 2009, SpWea, 7, S06001
Domingo, V., Fleck, B., & Poland, A. I. 1995, SoPh, 162, 1
Falconer, D. A., Barghouty, A. F., Khazanov, I., et al. 2011, SpWea, 9, S04003
Falconer, D. A., Moore, R. L., Barghouty, A. F., et al. 2014, SpWea, 12, 306
Hanssen, A. W., & Kuipers, W. J. A. 1965, Meded. Verh., 81, 2
He, K., Zhang, X., Ren, S., et al. 2015, arXiv:1512.03385
Heidke, P. 1926, Geogr. Ann. Stockholm, 8, 301
Huang, G., Liu, Z., Maaten, L., et al. 2016, arXiv:1608.06993
Huang, X., Wang, H., Xu, L., et al. 2018, ApJ, 856, 7
Huang, X., Zhang, L., Wang, H., et al. 2013, A&A, 549, A127
Ioffe, S., & Szegedy, C. 2015, arXiv:1502.03167
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Proc. 25th Int. Conf. Neural Information Processing Systems, NIPS'12 (Newby: Curran Associates), 1097
LeCun, Y., Bottou, L., Bengio, Y., et al. 1998, IEEE, 86, 2278
Lee, J.-Y., Moon, Y.-J., Kim, K.-S., et al. 2007, JKAS, 40, 99
Lee, K., Moon, Y.-J., Lee, J.-Y., et al. 2012, SoPh, 281, 639
Lera, K. D., & Barnes, G. 2017, in Extreme Events in Geospace: Origins, Predictability, Consequences, ed. N. Buzulukova (Amsterdam: Elsevier), 65
Li, R., Wang, H. N., Cui, Y. M., et al. 2011, Science China Physics, Mechanics and Astronomy, 54, 1546
Li, R., & Zhu, J. 2013, RAA, 13, 1118
Liu, C., Deng, N., Wang, J. T. L., et al. 2017, ApJ, 843, 104
Mason, J. P., & Hoeksema, J. T. 2010, ApJ, 723, 634
Muranushi, T., Shibayama, T., Muranushi, Y. H., et al. 2015, SpWea, 13, 778
Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, ApJ, 835, 156
Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2018, ApJ, 858, 113
Park, J., Moon, Y.-J., Choi, S., et al. 2017, SpWea, 15, 704
Pesnell, W. D., Thompson, B. J., & Chamberlin, P. C. 2012, SoPh, 275, 3
Raboonik, A., Safari, H., Alipour, N., et al. 2017, ApJ, 834, 11
Scherrer, P. H., Bogart, R. S., Bush, R. I., et al. 1995, SoPh, 162, 129
Schou, J., Scherrer, P. H., Bush, R. I., et al. 2012, SoPh, 275, 229
Shin, S., Lee, J.-Y., Moon, Y.-J., et al. 2016, SoPh, 291, 897
Szegedy, C., Liu, W., Jia, Y., et al. 2014, arXiv:1409.4842
Wheatland, M. S. 2005, SpWea, 3, S07003
Yuan, Y., Shih, F. Y., Jing, J., et al. 2010, RAA, 10, 785