Article:
Liu, S., Xu, L., Zhao, Z. et al. (2022) Deep learning based solar flare forecasting model. II. influence of image resolution. The Astrophysical Journal, 941 (1). 20. ISSN: 0004-637X

https://doi.org/10.3847/1538-4357/ac99dc

The Astrophysical Journal, 941:20 (11pp), 2022 December 10 https://doi.org/10.3847/1538-4357/ac99dc

Deep Learning Based Solar Flare Forecasting Model. II. Influence of Image Resolution

Sixuan Liu[1,2,3], Long Xu[1], Zhongrui Zhao[1,2,3], R. Erdélyi[4,5], Marianna B. Korsós[5,6], and Xin Huang[1,7]
[1] State Key Laboratory of Space Weather, National Space Science Center, Chinese Academy of Sciences, NO.1 Nanertiao, Zhongguancun, Haidian District, Beijing 100190, People's Republic of China; sxliu@bao.ac.cn, huangxin@nssc.ac.cn
[2] National Astronomical Observatories, Chinese Academy of Sciences, A20 Datun Road, Chaoyang District, Beijing 100012, People's Republic of China
[3] University of Chinese Academy of Sciences, A19 Yuquan Road, Shijingshan District, Beijing 100049, People's Republic of China
[4] Solar Physics & Space Plasma Research Center (SP2RC), School of Mathematics and Statistics, University of Sheffield, UK
[5] Department of Astronomy, Eö tvös Loránd University, Pázmány Péter sétány 1/A, H-1112 Budapest, Hungary
[6] Department of Physics, Aberystwyth University, Ceredigion, Cymru, SY23 3BZ, UK
Received 2022 March 3; revised 2022 October 5; accepted 2022 October 5; published 2022 December 8

Abstract

Due to the accumulation of solar observational data and the development of data-driven algorithms, deep learning methods are widely applied to build a solar flare forecasting model. Most of the works focus on how to design or select proper deep networks for the forecasting task. Nevertheless, the influence of image resolution on the learning based solar flare forecasting model has not been analyzed and discussed. In this Paper, we investigate the influence of the resolution of magnetograms on the accuracy of solar flare forecasting. We study the active regions by the Solar Dynamics Observatory/Helioseismic and Magnetic Imager (SDO/HMI) magnetograms from 2010 to 2019. Then, we downsample them to get a database containing active regions with several resolutions. Afterwards, three deep neural networks (i) AlexNet, (ii) ResNet-18, and (iii) SqueezeNet are implemented to evaluate the performance of solar flare forecasting compared to different resolutions of magnetogram. In experiments, we first did comparative experiments on our own simulated HMI database with different resolutions. Then we conducted experiments on two selected actual overlapping databases, Hinode–HMI and Michelson Doppler Imager–HMI, to reconfirm our conclusions. The experiment results show that all the selected deep learning networks are insensitive to the resolution to a certain extent. We visualized the regions of interest of the network from an interpretable perspective and found that the deep learning network pays more attention to the global features extracted from active regions that are not sensitive to local information in magnetograms.

Unified Astronomy Thesaurus concepts: Solar physics (1476); Solar activity (1475); Convolutional neural networks (1938)

Supporting material: animation

1. Introduction

A solar flare is an intense burst of radiation that occurs in the lower solar atmosphere. In a case of an intense flare event, the space environment is disturbed, and its fastest impact takes around 20 minutes on the Earth. Therefore, a solar flare forecast is an important topic in a solar activity forecast.

There are two types of approaches to forecasting solar flares, the first one is the event-based approach, and the second one is the source-region-based approach.

In the event-based approach, the forecasting model is based on the law of historical flare events that obey Poisson statistics in time and power-law statistics in size (Wheatland 2004; Hazra et al. 2020). If these statistical laws are followed, then an event-based forecasting model could be created (Wheatland 2005). Not only the statistical method but also the machine-learning method has been applied to build the event-based flare forecasting model. Stanislavsky et al. (2020) built an event-based forecasting model by using the hidden Markov model with two hidden states. The one hidden state is a background dominated state, and the other hidden state is a flare dominated state.

Most forecasting models are based on studying the source regions of solar flares, namely the solar active regions (active regions, ARs). The active regions are well-defined areas with a strong magnetic field on the Sun. The properties of active regions are parameterized to distinguish the flaring from the quiet ones. The morphological (Lee et al. 2012; Kontogiannis et al. 2018), magnetic (Ahmed et al. 2013; Bobra & Couvidat 2015; Korsós et al. 2015; Domijan et al. 2019; Wang et al. 2020a), and coronal features (Pagano et al. 2019) are extracted from white-light images, magnetograms, and extreme-ultraviolet images, respectively (Schrijver 2009; Toriumi & Wang 2019). Bloomfield et al. (2012) calculate the flare rates for McIntosh classifications of active regions from solar cycles 21 and 22.

Cui et al. (2006, 2007) analyze the relationship between the solar flares and magnetic parameters, for example, the magnetic gradient, magnetic neutral line, and magnetic shear. Raboonik et al. (2016) calculate the Zernike moments of magnetograms that can be used to forecast large solar flares. Aschwanden (2020) study the scaling relationships between magnetic field parameters and solar flares. Kusano (2020) present the k scheme, a physics-based model to forecast large solar flares through a critical condition of magnetohydrodynamic instability triggered by magnetic reconnection. Cicogna et al. (2021) develop a new algorithm to compute

[7] Corresponding author.

the R value (Aschwanden 2020) of an active region and estimate the topological complex of an AR. The relationships between features of flow field and subsurface flow fields in ARs and solar flares are discussed in Welsch et al. (2009) and Gao et al. (2014), respectively. Deshmukh et al. (2020) compute the topology and geometry of the ARs on the magnetograms. Korsós et al. (2020) identify an optimal height range in a three-dimensional extrapolated magnetic field of an AR for an earlier solar flare forecast. Jonas et al. (2018) capture the precursors of solar flares from solar images of various wavelengths. Krista & Chih (2021) propose the detection and extreme-ultraviolet flare tracking algorithm to identify flare signatures and their precursors.

Based on features extracted from source regions, different modeling methods, for example, statistics (Barnes et al. 2007), expert system (Miller 1988), and traditional machine learning (Hazra et al. 2020; Benvenuto et al. 2020), are applied to build the solar flare forecasting model. Cinto et al. (2020a) propose a framework that involves a modeling algorithm, feature selection, hyperparameter optimization, data resampling, and performance evaluation, to build a solar flare forecasting model. Chen et al. (2021) apply K-Nearest Neighbors, Random Forest, and XGBoost to build a flare forecasting model. An ensemble learning technique could combine multiple base forecasting models (Guerra et al. 2015) or fuse short-, mid-, and long-term properties of ARs (Lim et al. 2019). In the data set of a solar flare forecast, the number of flaring samples is far less than the number of nonflaring samples. This is called the class imbalance problem. Generative adversarial networks are used to balance samples for different flare classes in Deng et al. (2021). The decision-making boundary is determined to treat the class imbalance problem in Park et al. (2017). Wan et al. (2021) compare three strategies (resampling the training samples, changing the decision-making boundary, and assigning weights to the training samples) in the class imbalance problem of a solar flare forecast.

Because deep learning has a stronger automatic feature extraction ability and nonlinear relationship learning ability, recently, it has been widely used to discover the precursors and build the forecasting model (Abduallah et al. 2021). Huang et al. (2018) constructed a convolutional neural network (CNN) to automatically extract the flare forecasting patterns from the magnetograms of active regions. A convolutional neural network and long short-term memory are applied to build flare forecasting models in Li et al. (2020) and Wang et al. (2020b), respectively. Tang et al. (2021) propose an ensemble learning model, which fuses a deep neural network, convolutional neural network, and bidirectional long short-term memory neural network for a solar flare forecast. Nishizuka et al. (2020) propose deep flare net-reliable (DeFN-R) to build a reliable probabilistic solar flare forecasting model. Usually, a deep learning model is considered a black box. Synthetic magnetograms (Bhattacharjee et al. 2020) or gradient-weighted class activation mapping (Yi et al. 2021) can be used to improve the interpretability of the flare forecasting model.

Deep learning models are data-driven techniques. The resolution of magnetograms is different for different monitoring instruments, and the input of the deep learning model is usually resized to a specified fixed size. Therefore, it is a crucial problem how the forecasting model responds to the change in magnetogram resolution. For the first time, we analyze the influence of magnetogram resolution on the forecasting model built by using deep learning algorithms. We published our code on https://github.com/whirgrunt/the-influence-of-resolution-on-flare-forecast. Our databases are available at http://QuickConnect.cn/DeepSolar/sharing/VS1LKoxEp.

This Paper is organized as follows: Section 2 describes the database. Section 3 introduces three popular CNNs. The influences of the magnetogram resolution on simulated and actual databases are evaluated in Section 4. Conclusions are provided in Section 5.

2. Database

The input of the flare forecasting model is the magnetogram of an active region, and its output is whether a flare will happen or not in this region. We obtain the magnetograms of active regions from Solar Dynamics Observatory/Helioseismic and Magnetic Imager (SDO/HMI). The routine observation of SDO/HMI began on 2010 April 30, and the magnetograms of active regions can be downloaded from the Joint Science Operations Center database.[8] The cadence of the downloaded magnetograms for ARs is 96 minutes, and its spatial resolution is $0\farcs5$ pixel^{-1}.

To study the impact of resolution changes on flare forecasts, we first reduce the resolution to 1/2, 1/4, 1/8, 1/32, 1/64, 1/128, 1/256, 1/384, 1/512 of the original resolution. Then, three CNNs are trained for solar flare forecasting at different resolutions separately. ARs as the resolution changes are shown in Figure 1.

The solar flare data are obtained from the website of NOAA National Centers for Environmental Information (NCEI).[9] When at least one flare of a given or greater flare occurs within a given forecasting time window from the time the magnetogram was observed, we define this corresponding magnetogram as a flaring sample. Otherwise, it is considered to be a nonflaring sample.

Here, the forecasting time window is 48 hr. The data of active regions and solar flares spans from 2010 May 4 to 2019 January 26. The database contains 2988 positive images and 70822 negative images. The data distribution for each year is shown in Table 1.

3. Methods

With the massive accumulation of solar data and the development of big data algorithms, deep learning techniques are widely used in solar flare forecasting. Considering the amount of active region data and the generality of the model. We select three widely used CNNs: AlexNet (Krizhevsky et al. 2012), ResNet-18 (He et al. 2016), and SqueezeNet (Iandola et al. 2016) to forecast flaring. The parameter of networks is shown in Table 2.

3.1. AlexNet

The architecture of AlexNet is shown in Figure 2. AlexNet consists of five convolutional layers and three FC layers. Each convolutional layer is followed by a rectified linear unit (ReLU; Glorot et al. 2011), which is proposed for improving training efficiency and first used in CNN models. Local response normalization is imposed on the activity of neurons to aid

[8] http://jsoc.stanford.edu/ajax/lookdata.html

[9] https://www.ngdc.noaa.gov/stp/space-weather/solar-data/solar-features/solar-flares/x-rays/goes/xrs/

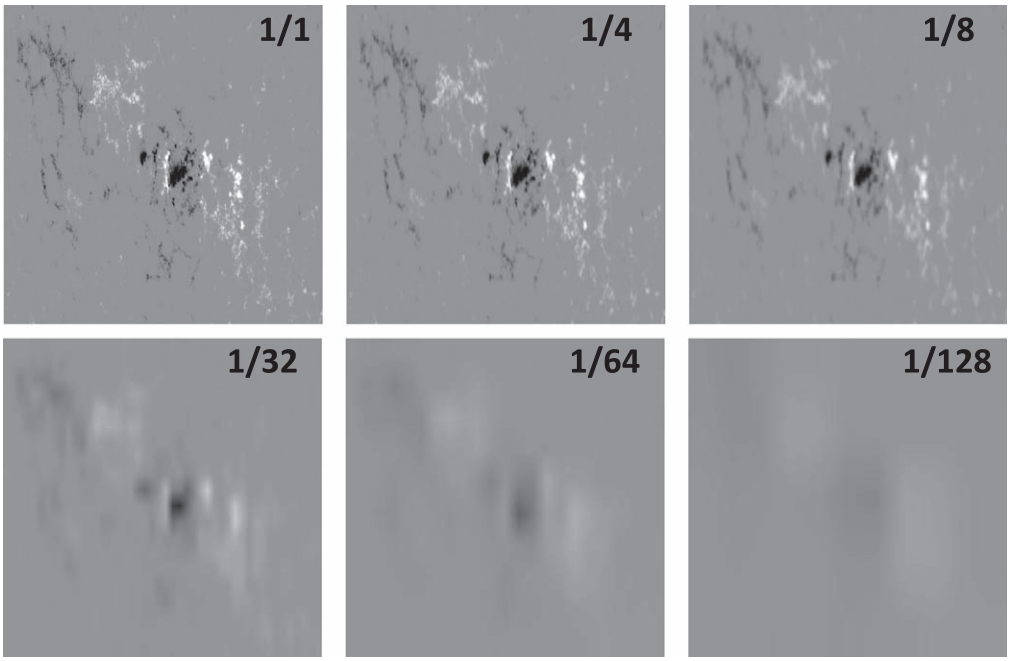

Figure 1. Examples for the magnetograms of ARs with different resolutions. This AR is NOAA 12497 on 2016 February 12. We artificially reduce the resolution of active regions to construct several databases. The resolution scale is marked in the upper right corner of the image. "1/1" represents the original magnetogram, "1/n" means downsampling the image to one nth of the original magnetogram.

Table 1
Data Distribution from 2010 to 2019

Year	Positive Numbers	Negative Numbers
2010	51	3637
2011	377	9209
2012	475	9762
2013	615	12684
2014	791	10862
2015	528	10715
2016	67	6769
2017	84	5133
2018	0	1964
2019	0	87

Table 2
Parameter of Three Networks

Network	Layer Number	FC Number	FLOPS	Parameter Number
AlexNet	8	3	1.82G	11.69M
ResNet-18	18	1	714.69M	61.10M
SqueezeNet	10	0	351.91M	1.24M

Note. "Layer number" represents the number of network layers. "FC number" represents the number of fully connected (FC) layers the network has. "FLOPS" means the amount of floating point operations. "Parameter number" represents the total number of parameters of the network.

generalization. The neighborhoods summarized by adjacent pooling units do not overlap in traditional pooling operations. In order to better integrate the characteristics of neighbors, overlapping max pooling is applied, which is proven to reduce the error rates in many databases. The details of the network are introduced as follows.

Convolutional Layer. Convolutional layers are the central network module used in CNNs. Stacking convolutional layers in deep models allows layers close to the input to learn low-level features (e.g., lines), and layers deeper in the model to learn more abstract features, like shapes or specific objects. One convolution operation uses a filter (kernel) sliding on the input, resulting in a feature map. One filter indicates a kind of attribute since we apply many filters to extract the different specific features. These filters can be learned automatically during the network training. Many optimization algorithms are proposed (e.g., gradient descent optimization (Ruder 2016) to adjust the parameters of the filter and to obtain the local optimal network. Then we can get good classification or forecast results using the trained network. Assuming the input image size is $W \times W$, the formula for calculating the size of the feature map after convolution is as follows:

$$N = \frac{W - F + 2P}{S} + 1, \quad (1)$$

where $F \times F$ is kernel size, P is the padding size, and S is the stride of sliding windows.

Pooling Layer. One limitation of feature maps after the convolutional layer is that they record the precise location of the feature in the input. A slight movement of the feature position in the input image will result in a different feature map when recropping, rotating, shifting, and other minor changes are made to the input image. We expect the results to be more robust. The common method to solve this problem from signal processing is downsampling. The feature map after compression and integration is obtained through the sliding window. Three common pooling methods are listed below.

(1) Average pooling calculates the average value for each sliding window on the feature map. (2) Maximum pooling (or max pooling) calculates the maximum value for each feature map sliding window, which summarizes the most activated presence. (3) Global pooling; on a two-dimensional feature map, the first two pooling methods are typically applied in 2×2 patches of the feature map with a stride of (2, 2). Instead

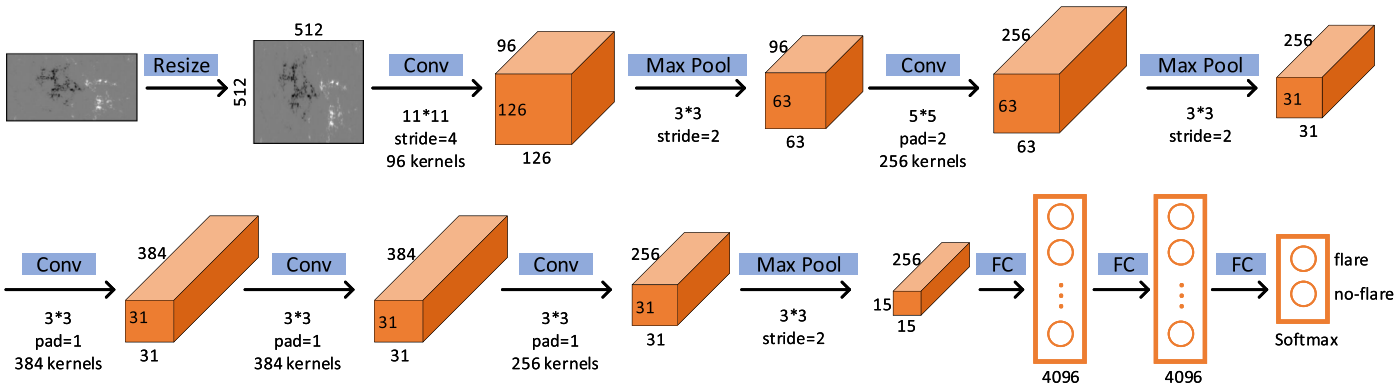

Figure 2. AlexNet structure. AlexNet contains eight layers. Blue rectangle denotes operations and layers. Among them, "Conv" stands for convolution layer, "Max pool" stands for overlapping max-pooling operation, "FC" stands for fully connected layer. Orange cube represents feature maps. Network output is a two-dimensional one-hot vector, which indicates the forecast result of input magnetogram image.

of downsampling the input feature map, global pooling downsamples the entire feature map to a single value, so the output vector dimension of this method is the channel numbers of the input feature map. The average pooling layer is always applied before FC, which is used to extract global features. This layer is parameterless and has a lesser amount of floating point operations (FLOPs).

Activation Function. Activation functions are inspired by the response of neurons to stimuli in biology. Our brains try to sort incoming information into "useful" and "not-so-useful" categories all the time. An activation function determines whether the neuron should be activated. This means that it will use simpler mathematical operations to determine whether the neuron's input to the network is essential in the current process. Many activation functions are proposed to increase the nonlinearity of the neural network. We apply a ReLU, which is defined in Equation (2):

$$ReLU(x) = max(0, x). \tag{2}$$

Fully Connected Layer. All input units from the last layer are connected to all units in the next layer in the fully connected layer. In the most popular deep learning models, the last few layers are FC layers, which compile the global information extracted from the previous layers to form the final output. It is the second most time-consuming layer after the convolutional layer, which can be written as Equation (3):

$$FC(x) = wx + b, \tag{3}$$

where w and b are learnable parameters.

3.2. ResNet-18

We can train deeper networks that express more complex and general models with the increasing amount of data. When networks go deeper, there emerge some problems with gradient transfer, which we call vanishing gradient and exploding gradient. In order to avoid these phenomena that hamper convergence in training, He et al. (2016) propose a residual module to train a deeper network (up to 152 layers) and get better classification results. Considering the limited amount of data, we choose ResNet-18, which has 18 layers to forecast flaring. The architecture is illustrated in Figure 3.

Residual Module. Instead of stack layers directly, the residual module adds mapping to maintain the original features. There are two residual operations designed: (1) The identity shortcuts in Equation (4) can be directly used when the input and output are of the exact dimensions. (2) The projection shortcut in Equation (5) is used to match dimensions that are done by 1×1 convolution:

$$y = F(x, W_i) + x \tag{4}$$

$$y = F(x, W_i) + W_s x. \tag{5}$$

Global Pooling. Global pooling is used to aggressively summarize a feature's presence in an image that can replace the FC layer to decrease the amount of calculation.

3.3. SqueezeNet

SqueezeNet is proposed by Iandola et al. (2016), which is less overhead and efficient while maintaining competitive accuracy. SqueezeNet achieves AlexNet-level accuracy on ImageNet with 50 times fewer parameters. Network architecture is illustrated in Figure 4. Solid and dashed arrows represent the two residual operations that are the same as ResNet-18.

To achieve the objective of efficiency and accuracy. There are three strategies when designing SqueezeNet.

Reduce Parameters. To reduce the convolution filter numbers. We first choose a 1×1 filter size with nine times fewer parameters than a 3×3 filter. Then, we replace the average pooling layer with the FC layer to reduce the FLOPs and filter parameters.

The Fire Module. We consider a traditional convolution layer in which filter sizes are all 3×3. The quantity of parameters is proportional to the number of input channels, the number of filters, and the filter size. The fire module is shown in Figure 5, which is comprised of a squeeze layer (which has only 1×1 filters) and an expand layer (which has a mix of 1×1 and 3×3 filters). The design of this module is carried out in both decreasing the number of 3×3 filters and the number of input channels to the 3×3 filters. There are three hyperparameters in a fire module: (i) S: the number of filters in the squeeze layer (all 1×1), (ii) E_1: the number of 1×1 filters in the expand layer, (iii) E_2: the number of 3×3 filters in the expand layer. When we use fire modules, we set S to be less than $(E_1 + E_2)$, so the squeeze layer helps to limit the number of input channels to the 3×3 filters.

Downsample Strategy. He & Sun (2014) make a point that delaying the downsampling layers will improve the accuracy. Since downsampling in early layers will lead to small activation maps that contain insufficient features. Meanwhile, downsampling can reduce the feature map size to control the number of calculations. SqueezeNet applies max pooling after

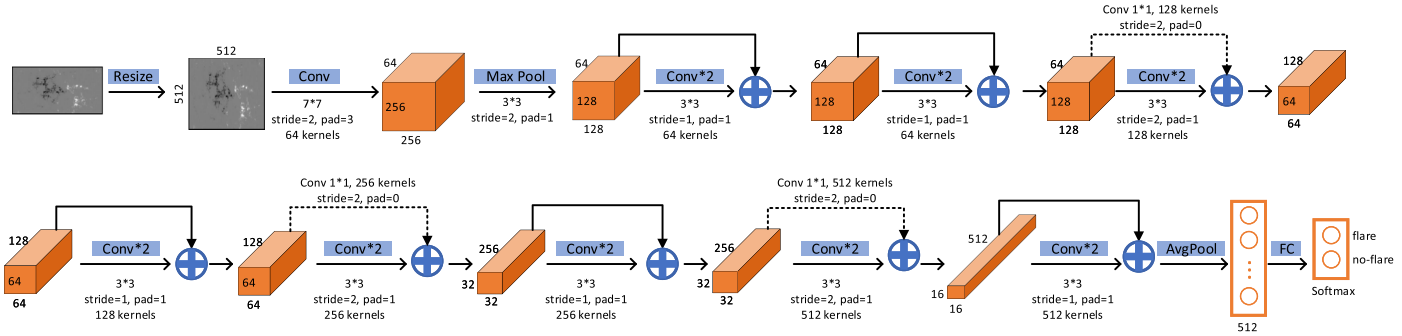

Figure 3. ResNet-18 structure. ResNet-18 contains 18 layers—a convolution layer, max-pooling layer, average pooling layer, and FC layer. Among the convolution layers, there are solid arrows and dashed arrows representing the two residual operations corresponding to Equation (4) and Equation (5).

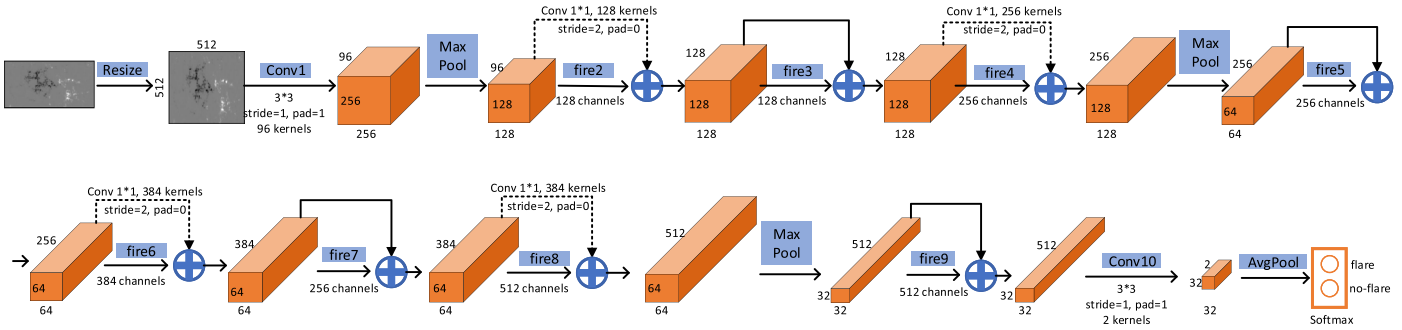

Figure 4. SqueezeNet structure. SqueezeNet contains 10 layers. The basic modules are almost the same as AlexNet and ResNet-18. Introduced is a new module called the "Fire module," which is explained in Figure 5.

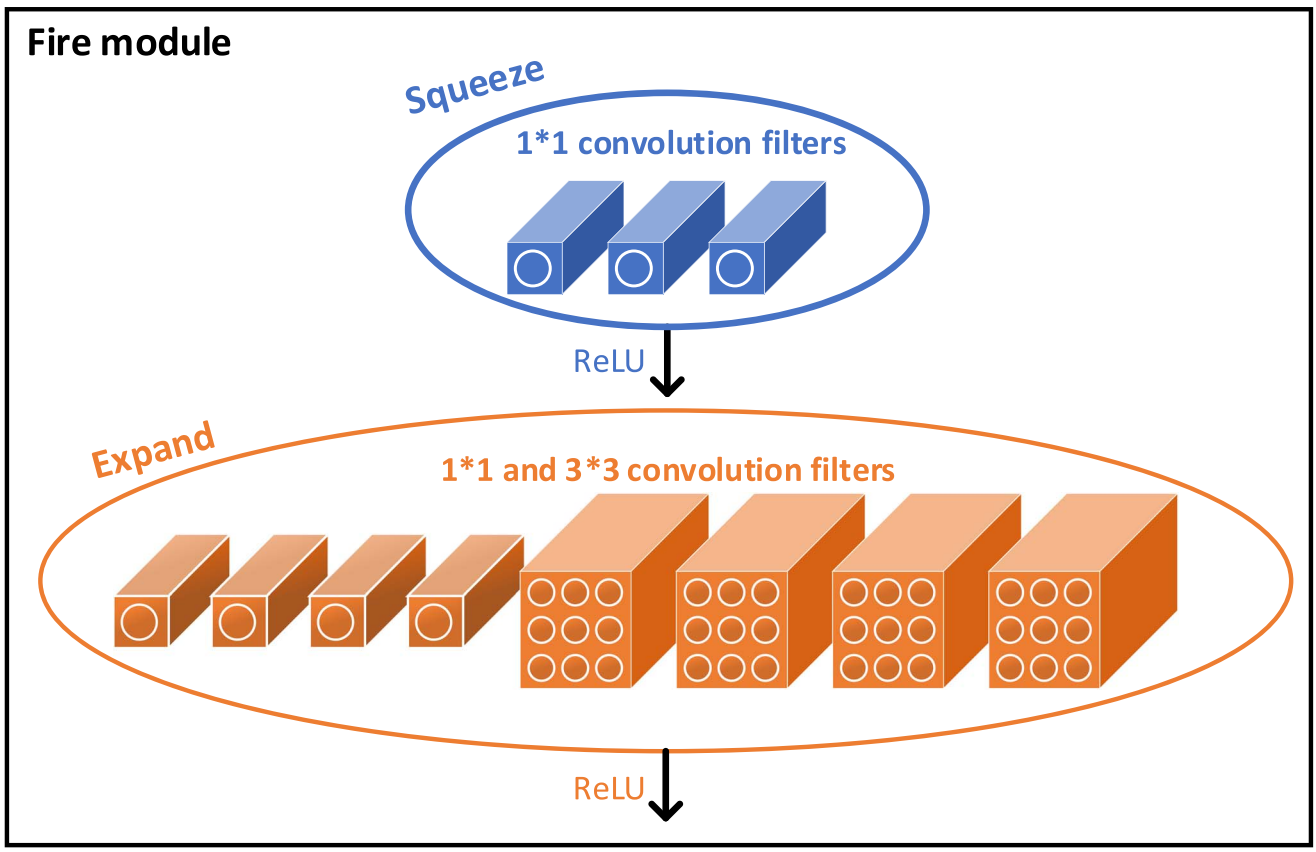

Figure 5. Fire module. This module consists of "Squeeze" and "Expand" two convolutional layers that both connect a ReLU activation function. In this example, the number of filters is set as $S = 3$, $E_1 = 4$, and $E_2 = 4$.

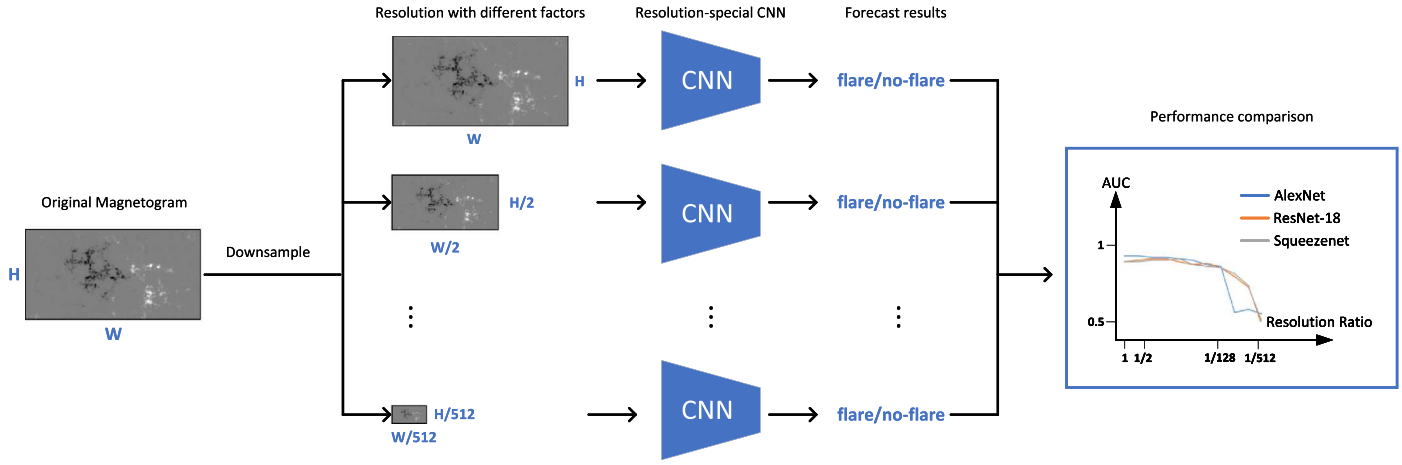

Figure 6. Flow chart of forecasting network. AlexNet, ResNet-18, and SqueezeNet: three convolutional neural networks (CNNs) are selected to forecast the flaring. The input of each CNN is a magnetogram of different resolution. The output of each CNN is the forecast result on whether a flare will burst in the next 48 hr. We evaluate the performance by synthesizing the forecast results of different networks at different resolutions.

the first convolutional layer and before the last fire module to balance the pros and cons of downsampling time.

4. Experiments

The flow chart of the forecasting network is shown in Figure 6. In order to evaluate the influence of different resolution magnetograms on solar flare forecasting, we first trained the models on different resolution databases, respectively. The model output is the forecasting results of whether a flare will occur in the active region within 48 hr. Then, we calculate the corresponding evaluation indexes and synthesize the forecasting results of different networks at different resolutions.

4.1. Evaluation Indexes

The output of the model is a two-dimensional vector representing the model's forecast of whether a flare will occur within the time window. Table 3 shows metrics in four forecast situations that are true positives (TPs), false positives (FPs), false negatives (FNs), and true negatives (TNs). TPs represent the number of positive instances that are classified as positive. FPs indicate the number of negative instances that are classified as positive. FNs indicate the number of positive instances that are classified as negative. TNs indicate the number of negative instances that are classified as negative. $P = TP + FN$ represents the total number of positive samples; $N = FP + TN$ represents the total number of negative samples.

4.1.1. True Positive Rate, False Negative Rate, True Negative Rate, and False Positive Rate

The true positive rate (TPR) is the percentage of positive instances correctly classified. On the contrary, the false negative rate represents the proportion of negative samples being classified incorrectly;

$$\mathrm{TPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

$$\mathrm{FNR} = 1 - \mathrm{TPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}. \tag{7}$$

The true negative rate (TNR) is the percentage of negative instances that are correctly classified. On the contrary, the false positive rate (FPR) represents the proportion of positive samples being classified incorrectly;

$$\mathrm{TNR} = \frac{\mathrm{TN}}{TN + FP} \tag{8}$$

$$\mathrm{FPR} = 1 - \mathrm{TNR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}. \tag{9}$$

4.1.2. True Skill Score

The true skill score (TSS) is used to estimate the whole model's performance considering both positive and negative classes;

$$\mathrm{TSS} = \mathrm{TPR} - \mathrm{FPR}. \tag{10}$$

In actual situations, flaring samples are far less than nonflaring samples. Consider the imbalance of positive and negative samples in the database. If a model judges all input active regions with flaring, it can also get good TPR performance, but the FPR index will be huge, and the TSS will get worse.

4.1.3. Receiver Operating Characteristic and Area Under the Curve

A receiver operating characteristic (ROC) curve is a graph showing the performance of a forecast model at all forecast thresholds. This curve has two parameters: TPR and FPR. We can get a ROC curve by increasing the threshold from zero to one. If the threshold is set to zero, all samples are forecasted to be positive. At this time, TPR and FPR are both one. If the threshold is equal to one, then all samples are forecasted to be negative; at this time TPR And FPR are both zero. The Area Under the Curve (AUC) score is the area below the ROC curve and is a general index to evaluate the two-class classification model. The larger the value of AUC, the better the model's performance. This score is between zero and one.

4.2. Experimental Setup

In an actual situation, flaring samples are much smaller than nonflaring samples. The number of positive samples of the data is small, and the positive and negative samples are unbalanced. To solve this two issue. We first apply three data augmentation methods, horizontal flip, random vertical flip, and random rotation, to augment the positive samples and enhance the

Table 3
Confusion Matrix for Binary Solar Flare Forecasting

		Forecasted Classes		Total Instances
		Positive	Negative	
Actual Classes	Positive	TP	FN	P
	Negative	FP	TN	N

robustness of the model. Data augmentation can help expand the data set. Image enhancement is a simple deformation of the image, which is used to deal with the deformation of the picture caused by different camera angles. In the face of the insufficient data volume of small databases on the solar deep learning task, through data enhancement, the data set becomes larger, the training data increase, and the robustness of the model is also enhanced. Second, we keep the quantity of positive and negative samples consistent during training to solve the problems caused by data imbalance.

We use the pretrained model provided by the PyTorch package to initialize the network parameters. We choose the cross-entropy loss function, a widely used optimization method in training networks. The stochastic gradient descent algorithm is applied to train the model. We train each network for about 30 epochs. Throughout training, we use an initial learning rate of 0.01, a batch size of 128, and a momentum of 0.9.

4.3. Experimental Results

In order to accurately verify our conclusions, we will introduce two groups of experiments: the simulated databases and the actual databases. The simulated databases are obtained by artificially downsampling the HMI magnetogram; the actual databases are the part of Hinode and Michelson Doppler Imager (MDI), which we selected, which overlap with the HMI.

4.3.1. Experiments on Simulated HMI Databases

After training the models with different resolutions, we calculate the above evaluation indicators. We first draw AUC and TSS graphs to analyze the influence of resolution. Then, we evaluate forecasting models by the tenfold cross-validation method. Meanwhile, train and test loss in different resolutions are retained. Finally, we visualize the regions of interest of the trained network.

AUC and TSS Results. Figure 7 shows the ROC curve of initial resolution. The animation shows that as the resolution decreases, the area under the ROC curve gradually becomes smaller. An AUC and TSS curve is shown in Figure 8. From Figure 8, the following three points can be concluded: (1) The capability of these three networks are comparable, with AlexNet slightly outperforming the other two networks. (2) At the beginning, as the resolution decreases, these two indicators remain at a high level. (3) Until the resolution is reduced to $1/64$, indicators begin to decrease significantly.

Cross-validation Results in 10 yr. The basic idea of cross validation is to group the whole data set into k parts. The cross-validation method uses one part as a training set and other $k-1$ parts as a test set. This evaluation method trains k times in turn and gets average indicator values to eliminate the instability caused by data set division and to increase the model's credibility. Inspired by this method, since our data have natural time-division characteristics, we split our database into 10 parts, each part contains active region data that spans a year. The results are shown in Table 4; we calculated AUC and TSS in the test set. We list TPR and TNR in both training and testing. Since the 2018 and 2019 yr have no positive samples, "\" indicates that this index cannot be calculated because the function's denominator is zero.

Train and Test Loss. We use ResNet-18 as an example to observe the trend of loss. Figure 9 shows the train and test loss curve. We focus on the loss at the later epochs because it is more stable than the former. Loss began to gradually increase when the resolution of input drops by more than $1/64$ and remains at the same level when the resolution is reduced to less than $1/64$ of the original image.

Visualization. Saliency methods can aid in understanding deep neural networks that link a deep neural network's prediction to the inputs that most influence that prediction. A saliency map indicates which input regions provide the most predictive power. To visualize the region of interest of networks, we use Grad-CAM ++ (Chattopadhyay et al. 2017), which draws the saliency map using gradients of the output over the input. The results are shown in Figure 10. When the resolution reduction is not so significant, the area that the network pays attention to does not change much. The network cannot extract the critical region when the image is too blurred. This result can also explain why the network performance does not decrease when the resolution is reduced very little. It is probably because the network forecasts a sample depending on an area. A slightly blurred magnetogram does not affect the network's feature extraction for this whole area.

We can conclude from these three experiments that before the resolution drops to $1/64$, the performance of CNNs can be maintained at a good level. As the resolution continues to drop, the input magnetogram loses too much information for the network to extract, so the network performance begins to decrease significantly. A possible reason for this is that CNNs are introduced to extract a global feature. It is difficult for CNNs to capture detailed features, which may affect the flaring. A suitably blurred image does not affect the network's comprehension of the whole magnetogram. When the resolution is reduced too much, the magnetogram loses too much information for the network to extract so that the performance will be degraded significantly.

4.3.2. Experiments on Hinode–HMI and MDI–HMI Databases

In order to further verify the impact of resolution on the deep learning prediction model, we chose actual data at different resolutions for experiments. By screening Hinode (high resolution), MDI (low resolution), and HMI (middle resolution) overlapping data, we prepare Hinode–HMI and HMI–MDI, which are two contrasted databases. We then study the resolution effect using deep learning models on two actual databases. The databases' overviews are shown in Table 5.

For the Hinode–HMI database, HMI observes the full disk with $0\farcs5$ resolution and Hinode has a better spatial resolution of $0\farcs3$ per pixel. The overlapping time of Hinode and HMI is from 2010 to 2015 and their magnetograms cannot correspond in time and space. Temporally, we choose the image closest to time as an alternative. Spatially, we use "label-me" to crop HMI to correspond to Hinode's magnetograms visually. Since the magnetograms are used to extract features for deep models, slight deviations in the visual have little effect on the model performance. The results are shown in Table 6. Even though Hinode's data resolution is higher, the performance of the

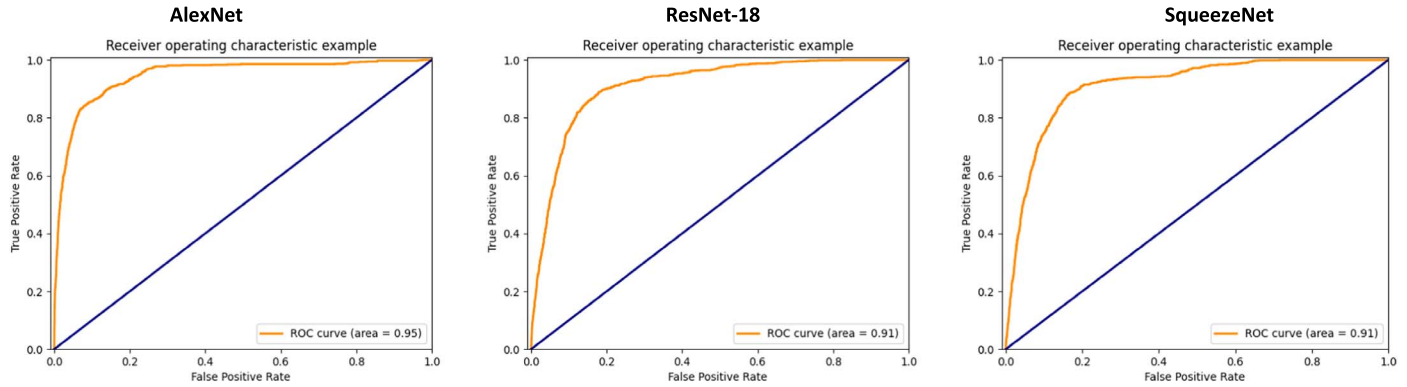

Figure 7. ROC curve of three networks with initial resolution. An animation with changing resolution is available. The animation shows the changing process of ROC from an initial resolution to low-resolution magnetogram. Each frame of animation corresponds to a different downsampling ratio. There are 11 frames in the animation, and their downsample ratios are 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/384, 1/512, respectively. The real-time duration of the animation is 8 s.

(An animation of this figure is available.)

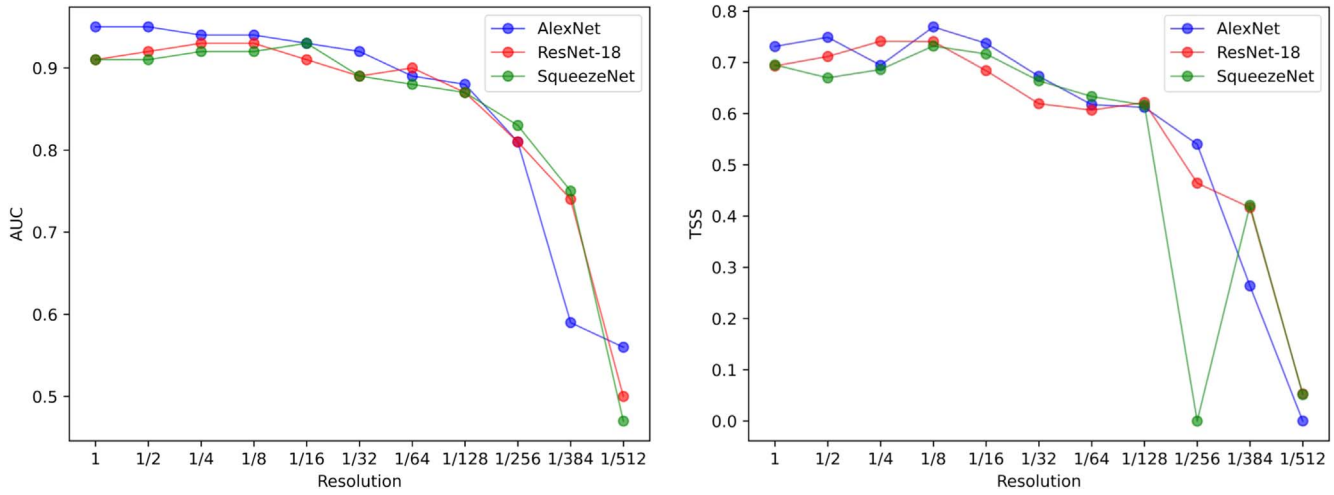

Figure 8. AUC and TSS curve with image resolution. The *x*-axis indicates the ratio of resolution reduction. Blue curve represents AlexNet, red curve represents ResNet-18, and green curve represents SqueezeNet.

Table 4
Cross-validation Results in AlexNet

Test Year	Train TPR	Train TNR	Test TPR	Test TNR	Test AUC	Test TSS
2010	0.8947	0.8635	0.7443	0.8627	0.80	0.61
2011	0.8829	0.7908	0.8548	0.8780	0.86	0.73
2012	0.9049	0.8736	0.7453	0.8156	0.86	0.56
2013	0.8959	0.8656	0.8846	0.8319	0.92	0.72
2014	0.8962	0.8612	0.9204	0.7737	0.93	0.69
2015	0.8809	0.8467	0.9432	0.8621	0.96	0.81
2016	0.8956	0.8490	1.0000	0.9167	0.99	0.92
2017	0.8998	0.8598	0.7381	0.9456	0.95	0.68
2018	0.8849	0.8497	...	0.9679	...	...
2019	0.8976	0.8497	...	0.7816	...	...

Note. We use 1 yr from 2010 to 2019 as the test set in turn, and use the remaining years' data as the training set. We compared AUC and TSS indicators in the test database.

models trained from the two resolution magnetograms is comparable.

For HMI–MDI database, we select a total of 271 magnetograms overlapping with HMI from 2010 May to 2010 October. We align with the HMI magnetograms according to the processing method in Huang et al. (2018). The amount of data are too small to support the data-driven deep learning models, so we trained models on downsampled

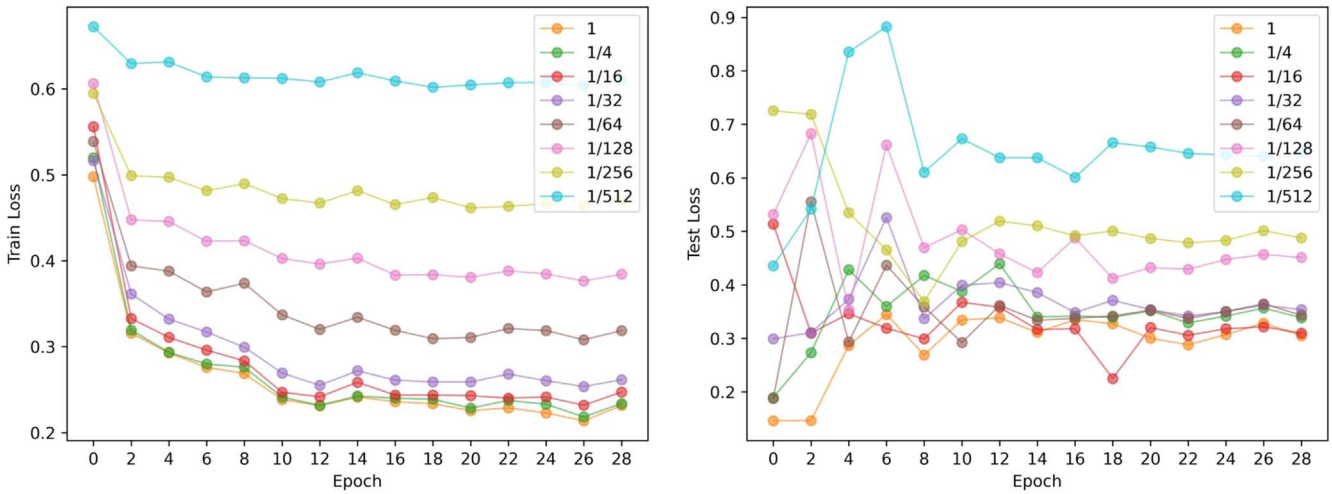

Figure 9. Loss curve with different epochs. Eight color curves represent different resolutions.

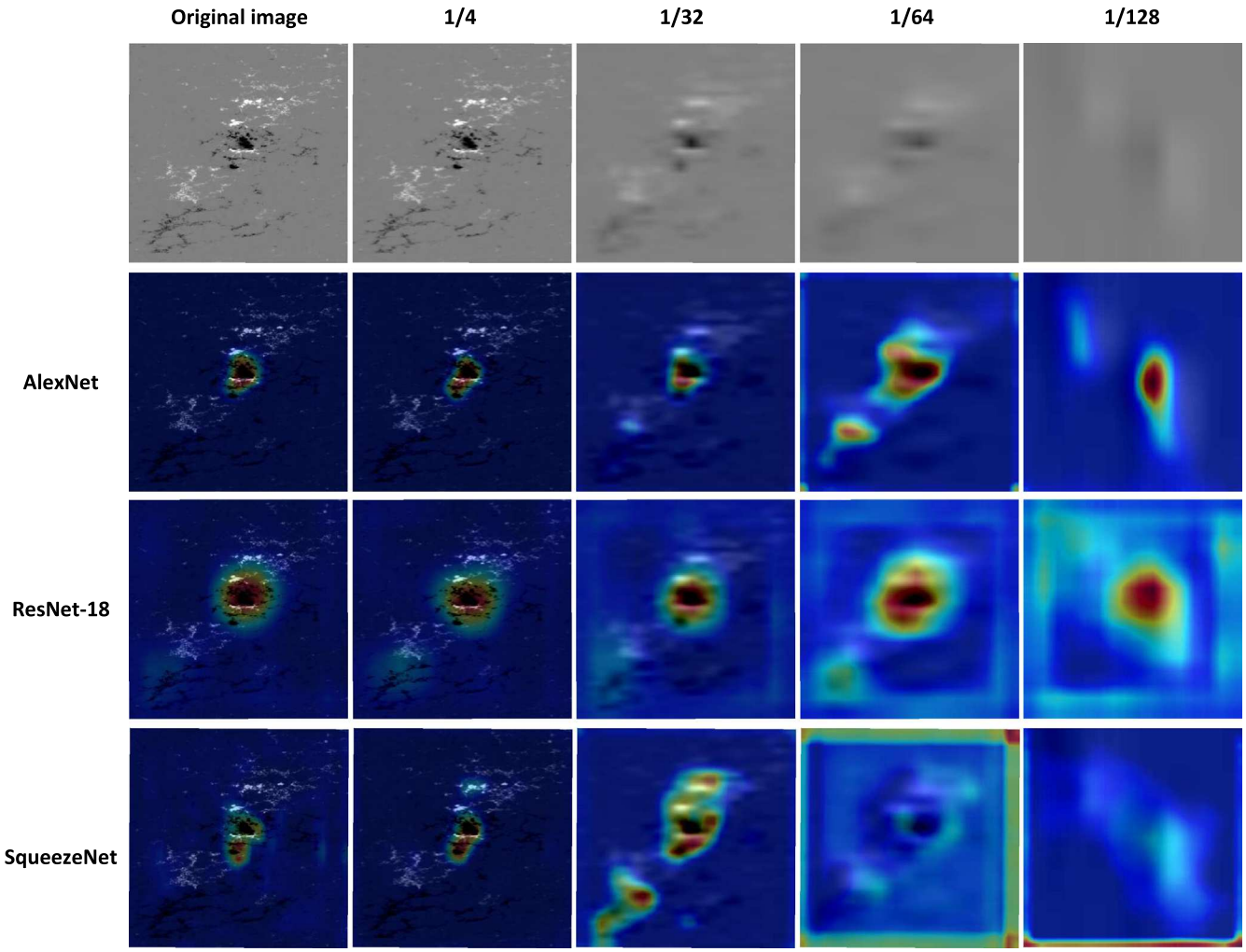

Figure 10. Visualization of network's interest area. The first row is the input image of different resolutions. The second, third, and fourth rows are the visualization results of different networks. The highlighted areas of the magnetogram, which were relevant for the classification. This active region sample is NOAA 12497 on 2016 February 12.

HMI magnetograms and tested them on the overlapping MDI magnetograms. The results are shown in Table 7. We can conclude that lower resolution MDI magnetograms can also have considerable prediction results on deep learning models.

The experiments on Hinode and MDI show that training on the Hinode magnetograms with higher resolution does not get a better model than HMI. Meanwhile, we can obtain a comparable result with HMI using MDI magnetograms with

Table 5
Data Split on Hinode and MDI

		Hinode	MDI
Train	Positive	65	...
	Negative	307	...
Test	Positive	49	42
	Negative	297	229
Sum	...	718	271

Table 6
Different Resolution Results on Hinode–HMI

		AlexNet	ResNet-18	SqueezeNet
Hinode	ACC	0.7717	0.7746	0.7023
	TPR	0.8367	0.8163	0.6531
	TNR	0.7609	0.7677	0.7104
	AUC	0.86	0.86	0.76
HMI	ACC	0.7832	0.789	0.7486
	TPR	0.8163	0.8163	0.7347
	TNR	0.7778	0.7845	0.7508
	AUC	0.86	0.85	0.8

Table 7
Different Resolution Results on HMI–MDI

		AlexNet	ResNet-18	SqueezeNet
HMI	ACC	0.8930	0.8819	0.9225
	TPR	0.4524	0.3810	0.6429
	TNR	0.9738	0.9738	0.9738
	AUC	0.95	0.93	0.91
MDI	ACC	0.8819	0.9446	0.8893
	TPR	0.1000	0.5333	0.0333
	TNR	0.9793	0.9959	0.9959
	AUC	0.97	0.99	0.95

lower resolution. The results in actual databases reconfirm our conclusions that the three selected deep learning networks are insensitive to changes in the resolution of Hinode, HMI, and MDI magnetograms.

5. Conclusion

In order to study the influence of resolution to forecast flaring, we prepared simulated databases with different resolutions and reconfirmed our conclusions on Hinode–HMI and MDI–HMI databases. Three deep learning models, AlexNet, ResNet-18, and SqueezeNet, were selected, and we trained models on different resolution databases, respectively. The experiments show that before the resolution drops significantly, the performance of CNNs can be maintained at a good level. One possible explanation is that deep learning networks are insensitive to high-resolution features. A certain degree of blurring does not affect the network's feature extraction.

On the one hand, the resolution has little effect on deep learning models and has a high tolerance for low-resolution data. This is an exciting conclusion because high-resolution space-based equipment is expensive. We can choose ground-based equipment for the solar forecasting task because it is good enough for a deep learning model, which will significantly reduce the equipment construction cost. On the other hand, deep learning models are insensitive to relative high-resolution features on our simulated magnetograms. A probable explanation is that when the network goes deeper, it will focus on a broader area in the image and will ignore the fine features that may cause flaring. Deep learning models can get around 90% AUC performance. An attention module could be used to extract local features of active regions that may cause flaring if we want to achieve better performance.

Thanks to anonymous reviewers for their valuable comments, which made our paper more accurate and complete. We thank the SDO consortium for the data. This work was supported by the National Key R&D Program of China under grant 2021YFA1600504. X.H. acknowledges support from the National Natural Science Foundation of China (grant Nos. 11873060, 11790305, U1831121, 11973058, and 12103064) and the support from the Beijing Municipal Natural Science Foundation (grant No. 1202022).

ORCID iDs

Long Xu https://orcid.org/0000-0002-9286-2876
R. Erdélyi https://orcid.org/0000-0003-3439-4127
Xin Huang https://orcid.org/0000-0003-0569-5932

References

Abduallah, Y., L. Wang, J. T., Nie, Y., Liu, C., & Wang, H. 2021, RAA, 21, 160
Ahmed, O. W., Qahwaji, R. S., Colak, T., et al. 2013, SoPh, 283, 157
Aschwanden, M. J. 2020, ApJ, 897, 16
Barnes, G., Leka, K. D., Schumer, E. A., & Della-Rose, D. J. 2007, SpWea, 5, S09002
Benvenuto, F., Campi, C., Massone, A. M., & Piana, M. 2020, ApJL, 904, L7
Bhattacharjee, S., Alshehhi, R., Dhuri, D. B., & Hanasoge, S. M. 2020, ApJ, 898, 98
Bloomfield, D. S., Higgins, P. A., McAteer, R. T. J., & Gallagher, P. T. 2012, ApJ, 747, L41
Bobra, M. G., & Couvidat, S. 2015, ApJ, 798, 135
Chattopadhyay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. 2017, arXiv:1710.11063
Chen, A., Ye, Q., & Wang, J. 2021, SoPh, 296, 150
Cicogna, D., Berrilli, F., Calchetti, D., et al. 2021, ApJ, 915, 38
Cinto, T., Gradvohl, A. L. S., Coelho, G. P., & da Silva, A. E. A. 2020a, MNRAS, 495, 3332
Cui, Y., Li, R., Wang, H., & He, H. 2007, SoPh, 242, 1
Cui, Y., Li, R., Zhang, L., He, Y., & Wang, H. 2006, SoPh, 237, 45
Deng, Z., Wang, F., Deng, H., et al. 2021, ApJ, 922, 232
Deshmukh, V., Berger, T. E., Bradley, E., & Meiss, J. D. 2020, JSWSC, 10, 13
Domijan, K., Bloomfield, D. S., & Pitié, F. 2019, SoPh, 294, 6
Gao, Y., Zhao, J., & Zhang, H. 2014, SoPh, 289, 493
Glorot, X., Bordes, A., & Bengio, Y. 2011, PMLR, 15, 315, https://proceedings.mlr.press/v15/glorot11a.html
Guerra, J. A., Pulkkinen, A., & Uritsky, V. M. 2015, SpWea, 13, 626
Hazra, S., Sardar, G., & Chowdhury, P. 2020, A&A, 639, A44
He, K., & Sun, J. 2014, arXiv:1412.1710
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 770
Huang, X., Wang, H., Xu, L., et al. 2018, ApJ, 856, 7
Iandola, F. N., Moskewicz, M. W., Ashraf, K., et al. 2016, arXiv:1602.07360
Jonas, E., Bobra, M., Shankar, V., Todd Hoeksema, J., & Recht, B. 2018, SoPh, 293, 48

Kontogiannis, I., Georgoulis, M. K., Park, S.-H., & Guerra, J. A. 2018, SoPh, 293, 96
Korsós, M. B., Ludmány, A., Erdélyi, R., & Baranyi, T. 2015, ApJL, 802, L21
Korsós, M. B., Georgoulis, M. K., Gyenge, N., et al. 2020, ApJ, 896, 119
Krista, L. D., & Chih, M. 2021, ApJ, 922, 218
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems 25 (NIPS 2012), ed. F. Pereira et al. (Red Hook, NY: Curran Associates Inc.), 1097, https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
Kusano, K., Iju, T., & Bamba, Y. 2020, Sci, 369, 587
Lee, K., Moon, Y.-J., Lee, J.-Y., Lee, K.-S., & Na, H. 2012, SoPh, 281, 639
Li, X., Zheng, Y., Wang, X., & Wang, L. 2020, ApJ, 891, 10
Lim, D., Moon, Y.-J., Park, E., et al. 2019, ApJ, 885, 35
Miller, R. W. 1988, JRASC, 82, 191
Nishizuka, N., Kubo, Y., Sugiura, K., Den, M., & Ishii, M. 2020, ApJ, 899, 150
Pagano, P., Mackay, D. H., & Yardley, S. L. 2019, ApJ, 886, 81
Park, J., Moon, Y.-J., Choi, S., et al. 2017, SpWea, 15, 704
Raboonik, A., Safari, H., Alipour, N., & Wheatland, M. S. 2016, ApJ, 834, 11
Ruder, S. 2016, arXiv:1609.04747
Schrijver, C. J. 2009, AdSpR, 43, 739
Stanislavsky, A., Nitka, W., Małek, M., Burnecki, K., & Janczura, J. 2020, JASTP, 208, 105407
Tang, R., Liao, W., Chen, Z., et al. 2021, ApJS, 257, 50
Toriumi, S., & Wang, H. 2019, LRSP, 16, 3
Wan, J., Fu, J.-F., Liu, J.-F., et al. 2021, RAA, 21, 237
Wang, J., Zhang, Y., Hess Webber, S. A., et al. 2020a, ApJ, 892, 140
Wang, X., Chen, Y., Toth, G., et al. 2020b, ApJ, 895, 3
Welsch, B. T., Li, Y., Schuck, P. W., & Fisher, G. H. 2009, ApJ, 705, 821
Wheatland, M. S. 2004, ApJ, 609, 1134
Wheatland, M. S. 2005, SpWea, 3, S07003
Yi, K., Moon, Y.-J., Lim, D., Park, E., & Lee, H. 2021, ApJ, 910, 8