Solar Flare Prediction Based on CNN

Jiayi Li
A53270752
j6li@eng.ucsd.edu

Mulin Yang
A53265076
ymulin@eng.ucsd.edu

Zachary Deneris
A12295511
zdeneris@ucsd.edu

Nansu Xu
A53270689
naxu@eng.ucsd.edu

***Abstract*—The motivation for this topic is the accurate prediction of an unpredictable celestial event: solar flares. Solar flares are studied using 10 sensors on the SDO NASA satellite mission, and the data consists of images from that mission. For our project, we built CNNs using both a regression model and a binary classification model as well as a vgg16-based network and a pre-trained autoencoder network. The results showed that with auto-encoded data and a regression classification model, the network is relatively successful at predicting the presence of a solar flare with an appropriately chosen threshold.**

Our GitHub repository:
https://github.com/NansuXu/ece228-Solar-flare.git

I. INTRODUCTION

A. Background

Solar flares are an example of celestial event that causes direct effects on human activities on Earth and in Space. Solar flares are sudden increases of brightness that are often accompanied by energetic outbursts of solar radiation. The frequency and magnitude of these events has unpredictability comparable with weather on Earth. While they can sometimes be predicted, the precision remains low. [1]

Solar flares have several effects on human activities both on Earth and in Space. Solar flares eject high energy particles that can damage satellite equipment. These ejections can also damage power grids by inducing extra current in sensitive grid hardware. [2] For these reasons, it is important to develop strategies for predicting these flares as accurately as possible. Predicted flares' risks might be mitigated by terrestrial organizations and space agencies.

B. Related works

Other researchers have also looked at machine learning as a way to predict solar flares. Most of the variety of research depends on different approaches to the learning rules, data sets, or data processing. Qahwaji et. al. has focused on creating a ruleset for solar flare prediction based on expert opinions and predictions of previous events [3]. Their system used a dataset of a large flare storm, and managed to accurately predict all flares during that events.

Zhang et. al. instead used image processing techniques similar to those used in this paper to create learning rules instead of relying on expert skills. Their research involved testing several models, the best of which achieved accurate prediction within 48 hours [4]. This research was similar to the work conducted here as they focused on feature extraction to better show the relevant aspects of the data. This research goes further than our work as it directly categorizes the different features such as outward and inward flux and the size of the sun spot. This research represents a logical avenue for the continuation of the work here.

Yu et. al. focused in image management with their data set [5]. They combine large image management techniques with hypersphere codeword strategies for their LM-sphere model [5]. The use of hyperspheres for this particular dataset is clever because it more clearly categorizes what is otherwise a relatively unorganized dataset. A given image of the sun likely does not look similar to any other to a computer, but an image depicting higher flux will likely look closer to an average picture of a high flux image than a low flux image. This technique leverages coding theory to better delineate the ruleset of the algorithm.

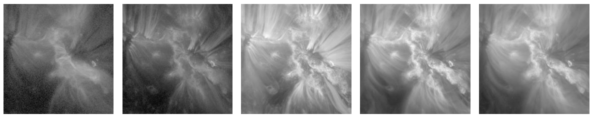
Huang et. al. focused on the magnetogram sensor measurements of the Sun's surface for their dataset. This paper also focuses on work using these magnetogram images [2]. Their research however attempted to employ deep learning techniques to the image along with a binary decision result. While a binary decision result was tested in the work presented here, it was quickly discovered that a class regression model yielded better performance as the varied flux values represent a wide range of data that is difficult to define with a binary classification.

Winter et. al. looked at predicting flux values of solar flares based on image processing [6]. While this research is not exactly the same as the work presented here, it represents a similar effort to make a connection between the image constructed from sensor data and a resulting flare. Their research involved looking at likelihood statistics of flares based on the comparison between local area flux and background flux. This novel interpretation of the sensor data allowed clear class delineations to be set before the machine learning step. the learning step uses these classes with a learning set to effectively match unseen data to a class. This research represents a clever and involved analysis of the actual data.

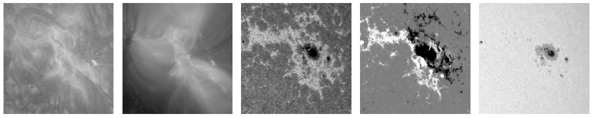
C. Dataset and features

Our dataset was from the Institute for Data Science, FHNW, Switzerland and downloaded from kaggle, which was derived from SDO (Solar Dynamics Observatory), it was launched for NASA's Living With a Star (LWS) Program, a program designed to understand the causes of solar variability and its impacts on Earth.

1) Dataset introduction: Solar flares are studied using 10 sensors on the SDO NASA satellite mission and the data consists of images from that mission. Each sample in this dataset have up to 40 images, but the total number of samples is only 8,000. All the images were in the resolution of 256×256 in *JPEG* format. They were taken in the time range from 2012 to 2018, so they are correlated in time domain. All the images are labeled with *peak flux* which represents the probability of a sun flare event occurring. the value ranges from 10^{-3} to 10^{-9}, divided into 157 classes.



(a) 94 (b) 131 (c) 171 (d) 193 (e) 211



(f) 304 (g) 335 (h) 1700 (i) magnet. (j) contin.

Fig. 1: ten different wave images,coming from 2 detectors onboard the SDO satellite: a to h are from AIA from wave length from 93 to 1700, i and j are from HMI, corresponding to magnetogram and continuum

2) Preprocessing and feature extraction: We split the whole data set into training set and testing set with a ratio of 9:1. And re-classified the training data into 5 classes of A, B, C, M, X according to their peak value[7].

Classification	Label	peak flux range
A	(4)	$< 10^{-7}$
B	(3)	10^{-7} - 10^{-6}
C	(2)	10^{-6} - 10^{-5}
M	(1)	10^{-5} - 10^{-4}
X	(0)	$> 10^{-4}$

TABLE I: Classification table

In the training dataset, we found an interesting trend of the image quality degradation. The average brightness has been steadily declining over the course of most of the mission's duration, which means the images has degraded over time. This phenomenon is obvious in AIA images. We take the average brightness of all training dataset images and draw the brightness curve. A trend does exist in our dataset as shown in curve plot (a) in Fig. 2. An offset is needed to correct the trend. We implemented a simple method to scale the brightness up.

The mean μ of all the images in first two years was calculated and used as a standard. I If the average brightness of the image in following year was smaller that μ, the following correction was performed.

$$I_{new}(x, y) = I_{old}(x, y) + p \tag{1}$$

Where p is the adaptive parameter. The curve plot after scaling shows that the method successfully offset this trend. 3 kinds of images with a relative flat brightness curve were selected and combined together in 3 channels for use as an input to our network. The AIA wavelength 1700, magnetogram and continuum sensors were chosen as shown in Fig. 3.

The Encoder-Decoder was used to extract the features to train our model, which will be explained more in the Methodology section.

II. METHODOLOGY

A. Four-Layer CNN Network

Since the goal of our project was to make predictions based on images, it is intuitive to think about using Convolutional Neural Networks (CNN). The key property of a CNN is that it uses convolution layers to extract information in a increasingly explicit hierarchy. We can exploit this important characteristic to train the network itself to learn the feature rather than manually extract them.

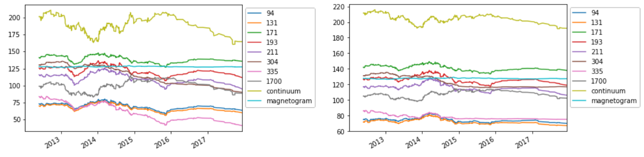
In this project, we implemented a simple four-layer CNN. The network contains 4 convolution layers with relu as the activation function. We first try to use this network to resolve a regression problem, and then, by adding sigmoid function at the final layer, we can convert it to resolve a binary classification problem.

B. VGG16 Network

After the result is observed at the four-layer CNN network, The accuracy can be improved by adding more layers. The next network we used is VGG16. VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford [8]. It has been widely used for classification and detection. The architecture of VGG16 is in Figure 4. By adding a fully connected layer at the final layer of the VGG16 network, it can be used to classify for several classes. We first evaluate the performance of this VGG16 network on five-class classification, and then change it give binary classification.

C. Auto-Encoder Architecture

Given the input image, it's relatively hard for us to manually extract the features of the images due to the little knowledge to the satellite images, so we decided train a auto-encoder network to help us do the feature extraction. the general idea of the auto-encoder is that the auto-encoder tries to learn a function that make $X' \approx X$. In other words, it is trying to learn an approximation to the identity function so as to output



(a) origin curve plot (b) after correction

Fig. 2: curve plot for average brightness of images

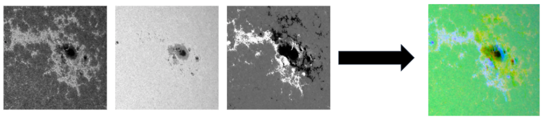


Fig. 3: combination of 3 different images

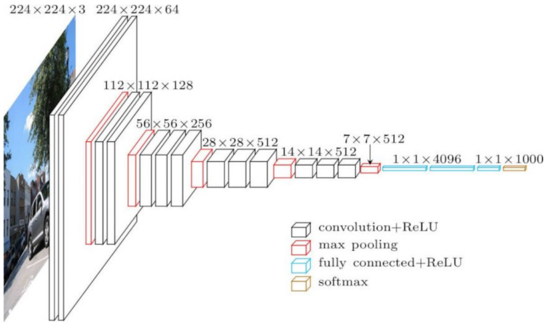


Fig. 4: The architecture of VGG16

x that is similar to x. And the middle Z is used as the feature extraction result. as shown in Fig.4.

A deep fully convolutional neural network architecture was also used for semantic pixel-wise segmentation termed SegNet[10]. The architecture of the encoder network is topologically identical to the 13 convolutional layers in the VGG16 network[11]. The structure of this network is shown in Fig.5. We can therefore initialize the training process from weights trained for classification on our datasets. The network's ability to prefer higher resolution feature maps at the encoder output over the fully connected layers also reduces the number of parameters in the SegNet encoder network significantly (from 134 to 14.7 M) as compared to other architectures [12]. In our dataset, we wanted generate a input image instead of doing segmentation, so The final decoder output is fed to a Relu function instead of multi-class soft-max classifier to produce class probabilities for each pixel independently.

III. EXPERIMENT SETTING

A. Dataset

The data set we use is created by Roman Bolzern and Michael Aerni from the Institute for Data Science, FHNW, Switzerland. The file is over 3 GB, but most of them cannot being used for training. We only use the magnetogram for training and testing. The size of the dataset we used for training is 32662, and 3476 for testing.

For classification, we use the result from [13] to classify the activation of the solar based on the flux value. We divide the activation into five class: flux value larger than $1e^{-}4$ as class 'X', flux value larger than $1e^{-}5$ as class 'M', flux value larger than $1e^{-}6$ as class 'C', flux value larger than $1e^{-}7$ as class 'B', and other values as class 'quiet'. We can see the distribution of different classes in our data set in Figure 7.

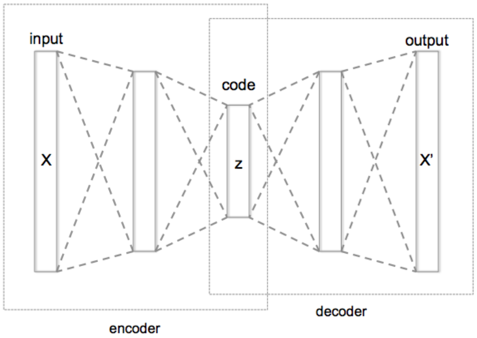


Fig. 5: Autoencoder structure [9]

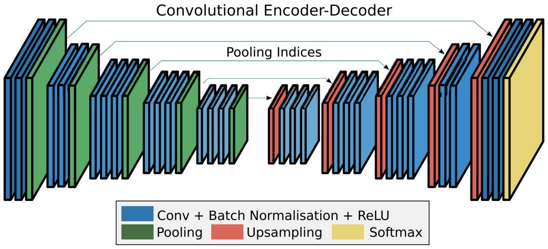


Fig. 6: Autoencoder structure, we replaced the softmax with Relu to fit our needs

IV. RESULT

A. Four-Layer CNN Network

1) Regression Result: We first try to use this network to give a regression prediction on the flux value. We trained the network for 60 epochs, and the mean-squared error on both training set and validation set is decreasing as the epoch increasing. However, after we plot the confusion matrix for the actual prediction (we consider the flux value larger than $1e^{-}5$ as flare, other values as non-flare), we can see in figure 8 this network only gives a trivial prediction by predicting all the images as non-flare.

2) Binary classification Result: We then use this network to give a binary classification prediction, with the flux value larger than $1e^{-}5$ as flare, other values as non-flare. We can see in figure 9, although the loss on neither training set or validation set is converging, the confusion matrix tells us the performance is actually better than regression prediction.

B. VGG16 Network

We then try to improve the accuracy of prediction by using a VGG16 network. We first evaluate the performance of five-class classification prediction. We can see in figure 10 that the network predicts most of the values for class 3 correctly, which is class 'C'. We then convert it to give binary classification predictions. In figure 10, we can see the network gives a trivial prediction. We then conclude that our data set is not suitable

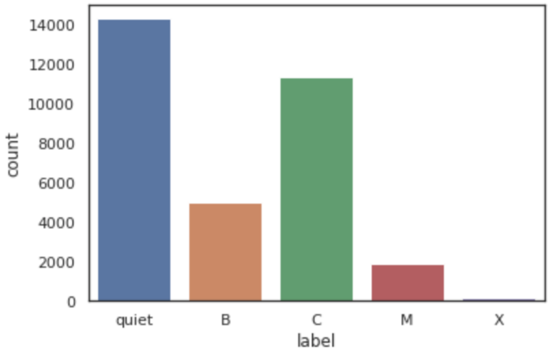


Fig. 7: The distribution of classes in training set

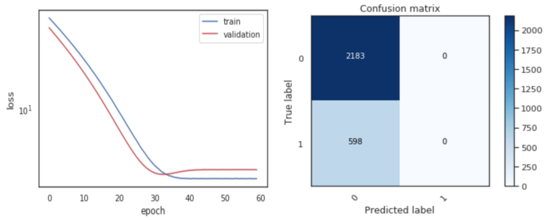


Fig. 8: The performance of the four-layer CNN network for regression prediction

for multiple layer deep learning network since the images are too simple and does not contain much information.

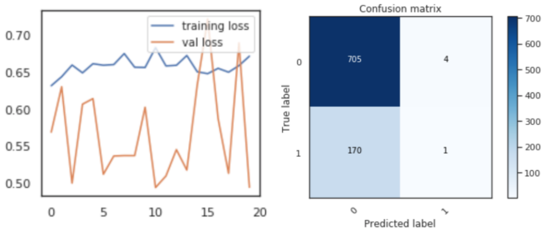


Fig. 10: The performance of the VGG16 network for binary classification

C. Auto-encoder Network

The output samples from the auto-encoder using our training set for 5 epochs is shown in figure **??**. The output from the first epoch has some distortion on the black area, after several iterations, the image is almost completely recovered from feature bank. The whole process can be described as figure 11.

In the training section, we take the L2 loss as our loss function, the training loss and validation loss versus epoch is shown in figure 12. The training loss contains the loss form each mini-batch iteration, so there is much more data than number of epochs. While the validation loss only calculated for once each epoch.

In figure 12, the training loss drops significantly before the second epoch and converges slower after that epoch. The evaluation loss shows the same trend as training loss which means that our model do not have the problem of overfitting.

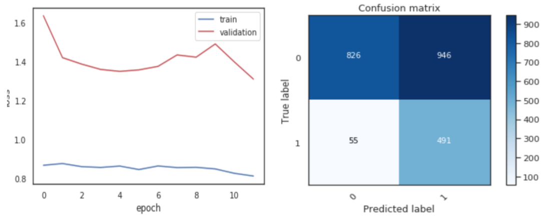


Fig. 9: The performance of the four-layer CNN network for binary prediction

Fig. 11: Encode and decode process

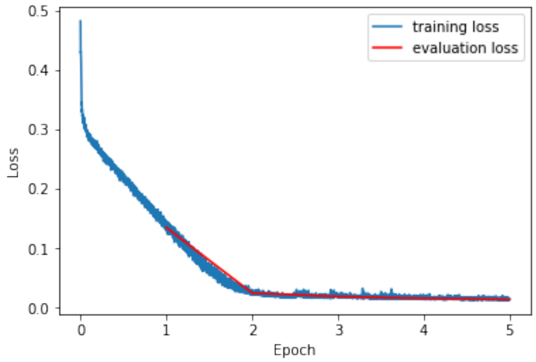


Fig. 12: Loss VS epochs

D. Regression Network with Pre-trained Auto-encoder

To estimate the solar flux, a regression network is implemented using features extracted from auto-encoder. By putting the scalar output into table I we can divide the data set into 5 classes. The confusion matrix of our network is shown in figure 13.

The confusion matrix shows that a large amount of the test dataset is classified as nearby classes and the validation dataset is unbalanced with too much data with smaller flux. This phenomenon can be explained by the property of the original dataset, which has few examples of large solar flux samples because it is a rare occurrence on the surface of sun. The network was originally designed to minimize the output flux with the true flux, so it made sense to put the sample in the nearby class.

We can do a little relaxation about our classification problem by treating a sample as correctly classified if a it is put into the true class or the two nearby classes to it. Then we can recalculated the classification accuracy which is about 74.3%.

V. DISCUSSION

It is important to note that with the chosen dataset, a balanced selection of samples to train would have contained

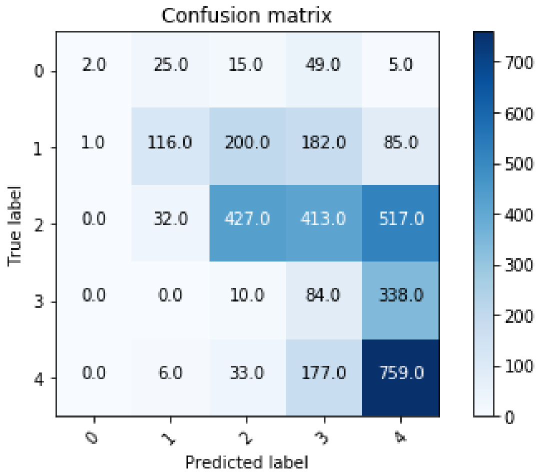


Fig. 13: Confusion matrix of 5 classes

so many ”no flare” events that the network would not have enough examples of flare events to properly map the features to all the classes. For this reason, an unbalanced dataset favoring the mid to high flux values was chosen.

The strange features present in the images made it difficult to find a model that would properly identify the features that mattered when classifying data. For this reason, the autoencoder step proved important as it reduced the complicated images to simpler samples that better represented the basic important features for classifying.

Another difficulty was the limited number of samples. For this reason, more time was spent on what data was available because the features in the data had to be easily learned because it was not an option to expand the training set.

In practice, the learning time for 1 epoch for the vgg16 network using cuda is over 250 seconds, and we can see that the loss on the validation set is not converging. Overall, the performance on Vgg16 is poor, and it shows that our data (images) is not suitable for deep learning with many layers.

With our first simple CNN network, we can see the performance is better for binary classification than regression. This improvement is due to the choice of peak flux as our prediction value in the regression problem, and this value is so small (from 1e-3 to 1e-9) such that taking a logarithm still does not sufficiently separate the classes. Therefore, our dataset is more suitable for a classification problem.

The most amazing outcome is from pre-trained autoencoder network. For the autoencoder part, it is capable of fully recovering the image from features, which means that the features extracted are a good match for this type of network. After connecting the autoencoder to two fully connected layers, the performance is the best among all the methods we used.

VI. Conclusion and Future Work

In this report, we detailed a neural network solution built to predict solar flares. A dataset was compiled from images gathered by sensors on the SDO NASA satellite mission. To predict the flares, we trained several networks based on different models: a four layer network with binary and regression models, a VGG16 network, and a pre-trained auto-encoder network with both a binary and regression model. The auto-encoder network coupled with a regression model proved to have the best results for our testing. The auto-encoder step improved feature extracting on difficult to analyze images and the regression model classes better fit the data in terms of useful classification. In the end, our network predicted flares with the testing set with an accuracy of 74.3%.

If work on this project continued, we would investigate other ways of combining image data to improve learning since our models use three channels when taking the sensors in total yields 10 sensors. We would also try to better understand the features in the solar flare data to better extract the important indicators of a solar flare for the network to investigate, similar to Winter et. al. [6].

VII. Contributions

Zachary Deneris focused on understanding the value and motivation for the work writing the introductions and parts of the analysis of the results, and helped organize the distribution of tasks. He also edited the formatting, spelling, and grammar of all written products from the group.

Jiayi Li implemented Auto-encoder to a pre-trained network for feature extracting and connect a regression layer to predict the solar flux. In addition, he analysed the result of the the auto-encoder based network.

Nansu Xu evaluated the accuracy of the prediction using four different method. Two of them are using simple four-layer CNN network, and the other two are using VGG16 network. He also finished part of the method and result part of this paper according to his implementations.

Mulin Yang did some research of the solar flare prediction and found the brightness problem and figured out a way to scale them up.He analyzed the original data and write down the pre-processing part of the code, he came up with the idea of combining three picture into 3 channel and write a code to implement it.

References

[1] G. Kopp, G. Lawrence, and G. Rottman, “The total irradiance monitor (tim): Science results,” *Solar Physics*, vol. 230, no. 1, pp. 129–139, Aug 2005.

[2] X. Huang, H. Wang, L. Xu, and W. Sun, “Learning solar flare forecasting model from magnetograms,” in *Proceedings of the IEEE Visual Communications and Image Processing*, St. Petersburg, FL, USA, November 2018.

[3] R. Qahwaji and T. Colak, “Automatic prediction of solar flares using machine learning: Practical study on the halloween storm,” in *2007 3rd International Conference on Recent Advances in Space Technologies*, June 2007, pp. 739–742.

[4] X. Zhang, J. Liu, and Q. Wang, “Image feature extraction for solar flare prediction,” in *2011 4th International Congress on Image and Signal Processing*, vol. 2, Oct 2011, pp. 910–914.

[5] Daren Yu, Xiaopeng Zhang, Jinfu Liu, and Qiang Wang, “A new solar flare prediction model based on large-scale image management,” in *2012 IEEE International Conference on Computer Science and Automation Engineering*, June 2012, pp. 351–354.

[6] L. M. Winter and K. Balasubramaniam, "Using the maximum x-ray flux ratio and x-ray background to predict solar flare class," *Space Weather*, vol. 13, no. 5, pp. 286–297. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2015SW001170

[7] W. Zhou, S. Gu, M. Ge, and J. Wang, "Analysis of the effect of the 06-09-2017 solar flare on gnss signal and positioning performance," in *China Satellite Navigation Conference*. Springer, 2018, pp. 555–569.

[8] A. Z. Karen Simonyan, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[9] Wikipedia, "Autoencoder — Wikipedia, the free encyclopedia," http://en.wikipedia.org/w/index.php?title=Autoencoderoldid=897670835, 2019, [Online; accessed 10-June-2019].

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[13] G. Barnes and K. L. NWRA, "A comparison of flare forecasting methods, i: Results from the all-clear workshop."