Electronic Letters on Computer Vision and Image Analysis 20(1):28-41, 2021

Investigation of Solar Flare Classification to Identify Optimal Performance

Aditya Kakde*, Durgansh Sharma+, Bhavana Kaushik*, Nitin Arora@

* *Department of Systemics, University of Petroleum and Energy Studies, Dehradun, India*
+ *Department of Cybernetics, University of Petroleum and Energy Studies, Dehradun, India*
@ *Electronics and Computer Discipline, Indian Institute of Technology, Roorkee, India*

Received: 29th July 2020; Accepted: 12th January 2021

Abstract

When an intense brightness for a small amount of time is seen in the sun, then we can say that a solar flare has emerged. As solar flares are made up of high energy photons and particles, thus causing the production of high electric fields and currents and therefore results in the disruption in space-borne or ground-based technological systems. It also becomes a challenging task to extract its important features for prediction. Convolutional Neural Networks have gained a significant amount of popularity in the classification and localization tasks. This paper has given stress on the classification of the solar flares that emerged in different years by stacking different convolutional layers followed by max-pooling layers. From the reference of Alexnet, the pooling layer employed in this paper is the overlapping pooling. Also, two different activation functions that are ELU and CReLU have been used to investigate how many numbers of convolutional layers with a particular activation function provides the best results on this dataset as the size of the dataset in this domain is always small. The proposed investigation can be further used in novel solar prediction systems.

Key Words: Solar Flares, Convolutional Neural Network, ELU, CReLU.

1 Introduction

The term solar flare is related to space weather as it refers to the adverse condition on the sun that may affect the space-borne or ground-based technological system and can endanger human health or life [1]. Most of the harmful radiations are protected by the earth's atmosphere, but some charged particles can enter the atmosphere. These space weathers are further classified into a ground-based system, communication system, and space-based systems. The ground-based systems are related to the gas pipelines, railway signaling, telecommunication, etc. where high electric fields and currents can cause disruption. The second one is the communication-based systems, which represents the direct impact of solar flares of the communication system, thus causing frequency jamming and radio bursts. The third one is the space-based system, this can cause health issues to the pilots and astronauts. This is not because they go deeper in the atmosphere, but the presence of harmful radiations and charged particles on the pole. Solar Flares yet closer to the sun emits an

Correspondence to: adityakakde100@gmail.com

Recommended for acceptance by Angel D. Sappa
https://doi.org/10.5565/rev/elcvia.1274
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

entire electromagnetic spectrum from the radio waves at the long-wavelength end, through optical emission of X-rays and gamma rays at the short-wavelength end www.qrg.northwestern.edu/projects/vss/docs/space-environment/3-what-is-solar-flare.html. As each year, there is an increased effect on the day-to-day life of living organisms, thus the prediction of solar flares becomes a necessary task. Many systems in recent years are being prepared for the prediction of solar flares as many infrastructures could be affected by that [2]. Some of the systems which are made for the prediction is, THEOPHRASTUS which was built by Space Environment Service Center [3] and was the first solar flare prediction system and ASSA (Automatic Solar Synoptic Analyzer), MAG4(Magnetic Forecast System) [4] and ASAP(Automated Solar Activity Prediction) [5] which are developed in recent years. In recent years, deep learning has advanced a lot. Deep learning is a subset of Machine Learning but with one major difference which is, it is an ability to discover and learn good representations using feature learning as specified by Yoshua Bengio [6]. Nowadays, the data and its complexity is increasing day by day. Thus it becomes very important to implement these networks based on feature learning. In this paper, we are dealing with solar flares which are made up of photons and high energy particles. Thus mapping its features becomes a challenging task. Deep Learning is a better option for solving this problem as it consists of the ability to map complex features. In this paper, we clarify that how many numbers of convolutional layers with a particular activation function should best suit the prediction of the solar flares at a particular year so that it can be used in any further solar prediction systems.

2 Related Work

2.1 Convolutional Neural Networks

Alex Krizhevsky et al.[7] presented a paper on ImageNet Classification with Deep Convolutional Neural Networks in which they proposed a novel architecture called Alexnet where the first convolutional layer consists of 96 kernels of size 11x11x3 with a stride of 4 pixels. The second convolutional layer takes a response-normalized and pooled output from the first convolutional layer which consists of 256 kernels of size 5x5x48. The third, fourth, and fifth convolutional layers are stacked one after another where the third layer is connected to the normalized and pooled output from the second layer. The third layer consists of 384 kernels of size 3x3x256, the fourth layer consists of 384 kernels of size 3x3x192 and the fifth layer has 256 kernels of size 3x3x192. Further, fully connected layers are also used consisting of 4096 neurons. The network was tested on the ILSVRC-2010 dataset where it achieved a Top-1 error rate of 37.5% and a Top-5 error rate of 17.0%. With more number of CNN's tested on ILSVRC-2012 validation and test set, 7 CNNs achieved the least top-5 error rate of 15.3%. Further on the ImageNet dataset, top-1and top-5 error rates are 67.4% and 40.9%. Jiuxiang Gu et al.[8] presented a paper on Recent Advances in Convolutional Neural Network where basic components are specified. A review of improvements in CNNs was also mentioned in CNN's are also mentioned that are Tiled Convolution, Transposed Convolution, Dilated Convolution, Network in Network, and Inception Module. Types of pooling layers were also described that Lp pooling, Mixed pooling, Stochastic pooling, Multi-scale orderless pooling. Different types of activation functions were also described that are ReLU, Leaky ReLU, Parametric ReLU, Randomized ReLU, ELU, Maxout, and Pro bout. Loss functions were described that are Hinge loss, Softmax loss, Contrastive loss, Triplet loss, Kullback-Leibler Divergence. Regularization techniques were specified that are lp-norm regularization, Dropout, and DropConnect. Optimization techniques were specified that are data augmentation, weight initialization, stochastic gradient descent, batch normalization, and shortcut connections. Further, fast processing of CNNs was described that are structured transforms, low precision, weight compression, sparse convolution and last was the applications of CNNs which are image classification, object detection, object tracking, pose estimation, text detection and recognition, visual saliency detection, action recognition, scene labeling, speech processing, and natural language processing.

2.2 Regularization and Optimization

Nitish Srivastava et al.[9] presented a paper on Dropout: A Simple Way to Prevent Neural Network from overfitting in which they have proposed a novel method to reduce the problem of overfitting in the neural network. It was seen that the generalization process was improved when tested on different datasets that are MNIST, TIMIT, CIFAR-10 and CIFAR-100, Street View House Numbers, ImageNet, Reuters-RCU1, and Splicing dataset. On the MNIST dataset, DBM with Dropout fine-tuning achieved the least error rate of 0.79%. On-street view house numbers dataset, ConvNet followed by max-pooling and dropout in all layers achieved the least error of 2.47%. On CIFAR-10 and CIFAR-100 datasets, convert followed by max-pooling and dropout achieved the least error rate of 37.20% but was not able to achieve the least error on the CIFAR-10 dataset. Also, results of top-1 and top-5 error when tested on ILSVRC-2010 and ILSVRC-2012 shows that convert with dropout achieved the least error rate of 37.5% and 17.0% on ILSVRC-2010 and 38.1% and 16.4% on ILSVRC-2012. On the TIMIT dataset, DBN having 4 to 8 layers with dropout achieved the least error of 19.7%. When tested on Reuters-RCV1 data, achieved the test loss of 29.62%. When tested on the splicing dataset, Bayesian Neural Network[10] achieved the highest code quality of 623 bits. A further comparison was done with different regularization techniques on the MNIST dataset where Dropout with Max-norm achieved the least test error of 1.05%. Also, a comparison between Bernoulli dropout and Gaussian dropout on MNIST and CIFAR-10 was performed. Sergey Ioffe and Christian Szegedy[11] presented a paper on Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift where a new methodology was proposed named batch normalization. The execution of the algorithm consists of four processes that are mini-batch mean, mini-batch variance, normalize, scale, and shift. It was also seen that it also acts as a regularizer thus eliminating the need for dropout and local response normalization when tested on the LSVRC2012 dataset, it was seen that BN-x30 achieved the maximum accuracy of 74.8%. BN-x30 means the initial learning rate was increased by a factor of 5 to 0.0075. Also, it was seen that by only Batch Normalization, the accuracy of Inception was less than half the number of training steps. A further test was conducted on the ImageNet dataset where the Batch-Normalized Inception ensemble achieved the least top-5 error rate of 4.9%.

2.3 Activation Functions

Djork-Arne Clevert et al.[12] presented a paper on Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) in which a new activation function was proposed. It was first tested on the MNIST dataset where the lines in the graphs are average over five runs with different random initializations. The ELU was also applied to autoencoders using the MNIST dataset where a comparison with other activation functions and learning rates showed medians over several runs with different random initializations. The comparison was done with ReLUs, LReLUs, and SReLUs. The ELU was tested on the CIFAR-10 dataset with Fact. Max-Pooling achieved a 4.50 % error rate and when tested on CIFAR-100 achieved a 24.28 % error rate. Further, it was tested on the ImageNet dataset and compared with ReLU. It was seen that both of them showed convergence but error started reducing earlier and reached 20 % top-5 error after 160K iterations while ReLU took 200K iteration to reach the same rate. Aditya Kakde et al.[13] presented a paper on the Novel Approach towards an optimal classification of Multilayer Perceptron where a cost-effective method was proposed to classify binary digits. The architecture describes how many numbers of layers with which activation function should be used. The comparison was done with sigmoid, ReLU, ELU, and SELU where it was seen that ELU performed better than the other three when the average of the results was compared. Further analysis showed that the ELU achieved the least error rate at minimum iteration. Wenling Shang et al.[14] presented a paper on Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units where a new activation named CReLU was proposed. It was seen that replacing ReLU with CReLU at lower CNN layers increased the performance. When tested on CIFAR-10 dataset, CReLU+half means CReLU with a model that halve the number of filters, achieved the second least error rate if 8.37% at single(one of the multiple ways of error rate) whereas in average and vote, CReLU achieved the least error rate of 9.39+-0.11% and 7.09%. On the CIFAR-100 dataset, CReLU achieved the least test error in a single of about 31.48%, and on average and vote, it

achieved an error rate of 33.76+-0.12% and 27.60%. Further CReLU when applied on different layers of VGG network where CReLU at conv1 and conv3 achieved the least error rate in all three parameters that are single, average, and vote of about 5.94%, 6.45+-0.02%, and 5.09%. On the CIFAR-100 dataset, conv1, conv3, and conv5 achieved the least error rate of 26.16%, 27.67+-0.07%, and 23.66%. VGG with CReLU is also compared with a different state of the art models where it achieved the least test error of 5.09% on CIFAR-10 and 23.66% on CIFAR-100 datasets. Further investigation was made by testing VGG with CReLU on ImageNet dataset where conv1-4 achieved a least top-1 and top-5 error of 39.82% and 18.28% but by taking its average conv1, 4, and 7 achieved least top-1 error of 35.70% and top-5 error of 15.32%.

2.4 Solar Flare Systems

Tarek AM Hamad Nagem et al.[2] presented a paper on Deep Learning Technology for Predicting Solar Flares from (Geostationary Operational Environmental Satellite) Data where they proposed a novel architecture for the prediction of solar flares. This system used three neural networks where the first neural network is used to convert (GOFS) x-ray flux 1-minute time series data. The second neural network is based on unsupervised learning which was used to extract MTF image features and the last one used a convolutional neural network to generate predictions. The network which is used for unsupervised learning is the autoencoder. The best Quadratic Score was achieved at 20 minutes. The prediction of the output was measured using standard forecast verification where HSS proved to be the best indicator for overall performance. The result from HSS shows that the generated predictions are not generated by chance. R. Qahwaji et al.[15] presented a paper on Automatic Short-Term Solar Flare prediction using Machine Learning and Sunspot Associations where it was predicted whether the sunspot at a particular time is likely to produce a flare and that flare is further divided into two classes that are X or M flare. The machine learning algorithms were implemented to identify which one is better for this task that is Cascade-Correlation Neural Network, Support Vector Machine, and Radial Bias Function. For CFP, SVM gave the best performance of 93.07% and for CFTP, CCNN gave the best performance of 88.02%. T. Colak et al.[16] presented a paper on Automated Prediction of Solar Flares Using Neural Networks and Sunspots Associations where a neural network is used for two classification tasks. The first task is to identify if it is a flare or not and the other classification task is to identify which type of flares it is (X or M type). In [17], flares types that are C or M type has been classified and predicted using a novel deep learning architecture which used five convolutional layers with batch normalization and dropout followed by ReLU activation function and achieved the accuracy of 0.889 ± 0.029 in C class and 0.817 ± 0.084 in M class. As in [2], [15] and [16], neural networks were used in which some used three layers consisting of unsupervised learning and convolutional neural network, some used SVM, radial bias and cascade correlation neural network and neural network with nine neurons in a hidden layer but there was no optimization or regularization during the training of the data. Thus it suffered from vanishing gradient and overfitting problems. Therefore this paper tends to solve these problems to solve it. Further in [17], batch normalization and dropout have been used to solve the above-mentioned problems but it used the ReLU activation function which results in making the activation 0 when it identifies any negative input thus can cause a problem in the overall performance of the neural network architecture. Therefore, this paper used ELU and CReLU activation functions which can accept negative inputs and further compared these two to identify the optimal one.

3 Proposed Work

In recent years, convolutional neural networks have gain popularity, especially after the published work by [7]. The convolutional neural network was coined by [18] when it was first tested on the MNIST dataset. The word convolutional means that the dot product of the values in the filters with the values of the input image. This can be seen in figure 1 When dealing with a 3-Dimensional image, the depth of the filter should always be equal to the depth of the input volume. Through this, the convolutional layer scans the complete image and extract the most important features of it. Before getting into the pooling layer, first, it passes through the activation

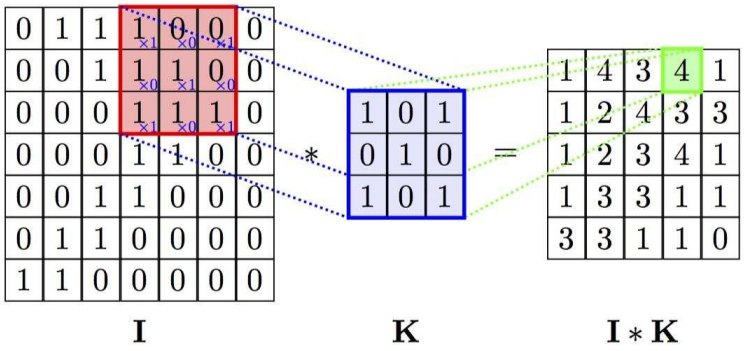

Figure 1: The above figure show the process of dot product between the data of input image and the filter https://github.com/PetarV-/TikZ/tree/master/2D%20Convolution.

function which checks the information coming out from the convolutional layer is sensible or not. After that, it passes through the pooling layer which downsamples the input so that only those features can be used which are most important. The output from the convolutional layer is computed by the number of filters, size of filters, and padding. Padding means having the same dimension for both input and output. It is denoted as valid or the same. The valid means that there is no padding and the same means we are applying some padding.

This paper also used convolutional neural networks where the dimensions of the input images are 70x70x3 and the number of filters starts with 32. As the number of convolutional layers increases, the number of filters becomes the twice previous one means as the first layer uses 32 number of filters and thus the second layer uses 64 number of filters and so on. The value of padding is 0, the strides are equal to 1 and the size of filters is 3 for all convolutional layers.

3.1 Overlapping Pooling

Pooling is a process that is applied after the convolutional layer. It is used to downsample the images to get the most important features so that further fully connected layers can deal with it when it comes to the classification task. Overlapping pooling comes under the domain of the pooling layer where the filter size of the pooling layer should be less than the size of the convolutional filter. The advantage of using overlapping pooling is that it reduces the problem of overfitting during the extraction process as mentioned by [7]. Thus, this paper also used overlapping pooling where the filter size of the pooling layer was given 2 pixels and the filter size of the convolutional layer was given 3 pixels and this properly matched with the property of overlapping pooling as 3 greater than 2.

3.2 Augmentation and Preprocessing of the network

Data Augmentation is applied when we are dealing with less number of training data means to make a powerful image classifier with less number of training data, we use data augmentation. When the data is in the form of an image, then we can say that we are applying Image Augmentation. It is the artificial creation of the training data or a combination of multiple processing so that the training images can be processed in different ways. The subdomains which lie under the Image Augmentation are random rotation, shifts, shear, flips, etc. As this paper deals with only 1197 training images, thus Image Augmentation is applied. The image of our training data has been augmented in three different ways such that the first is the random 90-degree rotation of 4 times. After that, the random flip up down and random flip left-right.

3.3 Regularization and Optimization of the network

One of the major aspects to avoid the loss of training is to reduce the problem of overfitting. As the noisy data increases, the learning of the model becomes flexible at the risk of overfitting. It means that during the training time there are chances that some noisy data can be learned with the actual data. It shrinks the coefficient estimates to zero means it discourages learning a more complex or flexible model to avoid the risk of overfitting `https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a`. Whereas optimization means to be a more cost-effective model and to achieve the highest possible performance. This paper tends to use Dropout and Batch Normalization to reduce the problem of overfitting and vanishing gradient in the models.

3.3.1 Dropout

The dropout method is coined by [9]. It is the method of dropping a neuron from the hidden layers at an alternate iteration. Suppose that in a neural network, the last hidden layer has 2 neurons, and after that comes the output layer with a single neuron. When the data is passed through both the neurons from the previous layers, then suppose the first neuron gives the output of 80% and the second neuron gives a random output. The best thing the output neuron can do is

$$1.[ON1] + 0.[ON2] = [ON1] \tag{1}$$

where ON means the Output of Neuron. This shows that the output layer will give the result of 80 %. This means that the gradient is only passing from neuron 1 to the output neuron. So out of three neurons, only two neurons are doing their work. To prevent the loss of learning, the dropout function is used. Due to this function, one out of two neurons in the last hidden layer will be dropped at an alternate iteration, and because of this, a non-zero gradient will be generated for neuron 2 and thus it has to provide with some results which will result in more than 80% from the output layer. This is how it reduces the problem of overfitting. Thus two processes come under the dropout which are dropping units while training and the second is scaling output to be matched between training and testing. This paper is divided into two parts where the first part used the dropout function with fully connected layers as a part of a convolutional neural network for the classification task. The keep probability which means the probability at which each element is kept used for the dropout in this paper is 0.8.

3.3.2 Batch Normalization

It means the normalization of the output from the hidden layer. It solves the problem of vanishing gradient as it makes the learning of the initial layer capable as it is in further layers. It comprises two learn-able parameters that are β and γ. γ is used to shift the value so that mean becomes zero and β is used to scale so that it becomes unit variance. Being mean=0 and variance=1 can speed up the learning by normalizing all the features to take on a similar range of values as features that differ from 0-10 and 0-100. Each mini-batch is scaled by the mean and variance which makes it to add some noise to each hidden layer activation thus has a slight regularization effect. The second part of this paper also used batch normalization with fully connected layers as a part of a convolutional neural network. The formula can be represented as:

$$\mu_\beta \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{// Mini batch mean} \tag{2}$$

$$\sigma_\beta^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2 \quad \text{// Mini batch variance} \tag{3}$$

$$x_i \leftarrow \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}} \quad \text{// Normalize} \tag{4}$$

$$y_i \leftarrow \gamma x_i + \beta \equiv BN_{\gamma,\beta}(x_i) \quad \text{// Scale and Shift} \tag{5}$$

3.4 ELU and CReLU Activation Functions

The activation function can also be called a transfer function. It can be used with the hidden layer as well as with the output layer depending upon its property. It is used to generate a sensible functionally mapped output of the network layers. It introduces non-linear functional mapping and can range from 0 to 1 or -1 to 1. Two categories come under the activation functions which are inner activation function consisting of Linear, Tanh, Sigmoid, ReLU, Leaky ReLU, ELU, etc. Whereas the outer activation function property should match with the number of classes of the dataset used consisting of sigmoid and softmax. This paper used two different activation functions where the first part of this paper discussed the ELU (Exponential Linear Unit) activation function and the second part discussed the CReLU (Concatenated Rectified Exponential Linear Unit) activation function. The output layer used the softmax activation function as this paper deals with more than two classes.

3.4.1 Exponential Linear Unit

The Exponential Linear Unit can also be called ELU and was first proposed by [12] and is given by:

$$f(x) = \begin{cases} \alpha(e^x - 1), x < 0 \\ x, x \geq 0 \end{cases} \tag{6}$$

It's gradient is given by

$$\frac{df(x)}{dx} = \alpha(e^x - 1) + \alpha \tag{7}$$

For a positive value of x, it acts as ReLU but unlike ReLU, it also takes negative inputs. It saturates much faster than ReLU when compared in the positive domain thus makes it more efficient. In ReLU, the negative input converts automatically to zero and thus results in a dead neuron which overall makes performance suffer. This paper used the ELU activation function with both the parts that are with Dropout and with Batch Normalization.

3.4.2 Concatenated Rectified Exponential Linear Unit

Concatenated Rectified Exponential Linear Unit can also be termed as CReLU which can be given as:

$$f(x) = (max(0,x); max(0,1-x)) \tag{8}$$

CReLU is also an improvement to ReLU and was first proposed by [14]. Unlike ReLU, it also takes negative input but differs somewhat from ELU. As ReLU results in dead neurons due to the transformation of a gradient to zero in a row, CReLU at the negative part generates the output as [0, x] where x is the input and at the positive part generates the output as [x, 0]. After this, they both get concatenated to form concatenated ReLU. Due to this, it doubles the output dimension of the network. It also has an improved effect on the regularization, invariance, and reconstruction properties of the network. This paper also used CReLU in convolutional layers as well as in fully connected layers and is present in both the parts of the paper, one with the Dropout part and the other with the Batch Normalization part.

3.5 Experimental Setup

The input image is of size 70 x 70 x 1 where 1 signifies black and white color channel and the first layer uses 32 number of kernels of size 3 x 3 x 1 and the next layer used 64 number of kernels also of size 3 x 3 x 1. It can be seen in figure 2.

The next layers have twice the number of kernels that were used in the previous layer. As was explained previously that this investigation also used overlapping pooling, thus the size of the pooling layer is 2x2. For the classification task, two fully connected layers are used consisting of 1024 number of neurons. For the optimization task, Adam optimizer is used with a learning rate of 0.001 and loss of mean square. In the above

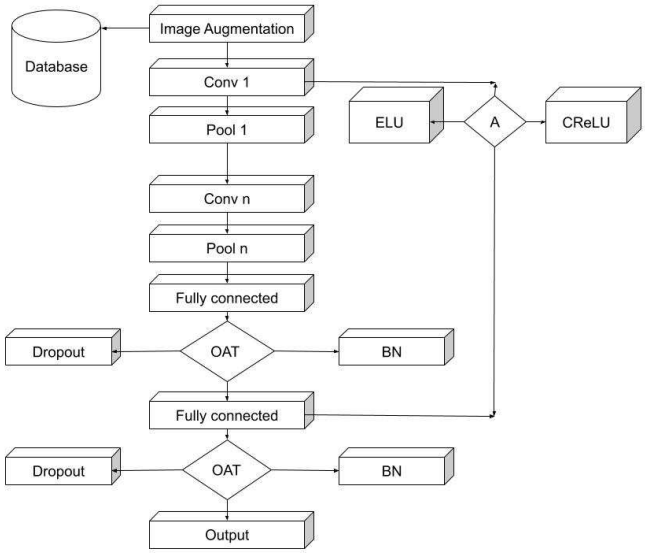

Figure 2: Architecture used for the proposed investigation.

Steps	Training Implementation
1	Inputting raw data with 256 x 256 pixels dimensions.
2	Altering the original dimensions to 70 x 70 pixels.
3	Augmenting the data.
4	Passing through convolutional layer by initializing F = 3, K = 32, S = 1 and P = 0.
5	Increasing the number of filters by 2x when stacking n number of convolutional layers.
6	Passing the convolve data through max-pooling initializing F = 2 and S = 1 making F(conv) $>$ F(max)
7	Repeating steps 4, 5 and 6 to n times.
8	Applying ELU activation in all convolutional layers.
9	Passing the pooled output as flattened through HL with random initialization of weights.
10	Applying ELU and CReLU activation in HL.
11	Applying batch normalization and dropout to get $y_i = \mathrm{BN}_{\gamma,\beta}(x_i)$ and $y_i = p(HL)^M$ where p = 0.5.
12	Calculating mean squared error of y_i then applying back propagation and adam optimizer to update weights for next iterations.

Table 1: Implementation about Network Training

figure, it can be seen that the input image is first drawn to the Image Augmentation process which is four times random 90-degree rotation, random flip up-down, and random flip left-right. Then it passes through the convolutional layers followed by the overlapping pooling stacked one after the other. The number 32 shows the kernel size of the first convolutional layer and 2x shows how that kernel size gets doubled as the input image passes through further convolutional layers. The first convolutional and pooling layer is shown by Conv1 and Pool1. Conv n and Pool n means nth convolutional layer and nth pooling layer. The symbol A means activation function which is used from first to last convolutional layers that are ELU and CReLU. Then comes two [19] fully connected layers that used only ELU activation function with RO means Regularization and Optimization methods that are Dropout and Batch Normalization. At last, is the output layer with 4 number of neurons as we have only 4 classes with a softmax activation function. In table 1, F represents filter size, K represents the size of the filter, S represents strides and P represents padding. In pooling, F(Conv) means the filter size of the convolutional layer, and F(max) means the filter size of the max-pooling layer. F(Conv) $>$ F(max) because to apply overlapping pooling. In the seventh point, n means the number of stacking that is done in the experiments. In the eleventh point, BN means batch normalization and p means regularization where its value is 0.5.

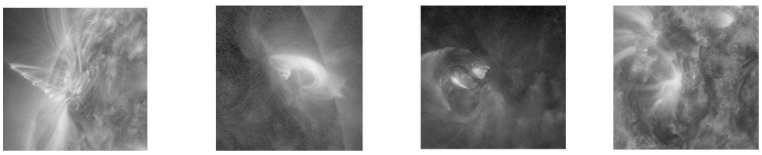

Figure 3: Figure shows how the data has been divided into four classes on the basis of years. The first image shows solar flare occurrence in year 2012, second image in the year 2014, third image in the year 2015 and fourth in the year 2017

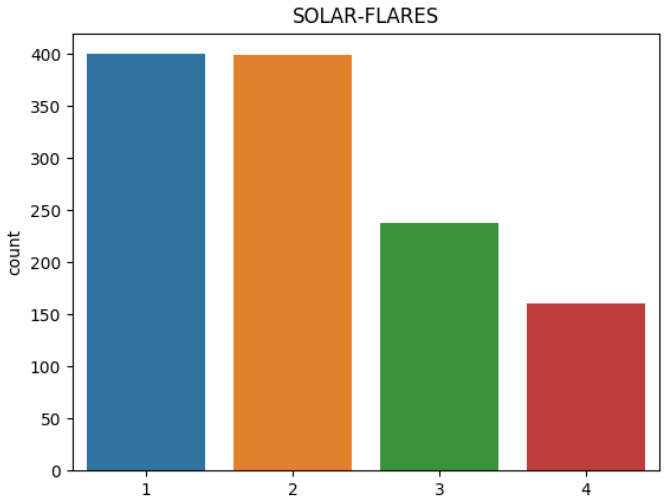

Figure 4: The above figure show the division of each classes.

4 Results and Discussions

4.1 Dataset

For this proposed investigation, the dataset is referred from `https://i4ds.github.io/SDOBenchmark/`. The dataset is divided into two parts that are SDOBenchmark_example and SDOBenchmark_full. This paper has considered the first part of the data for performing the task. This dataset also consists of some gaps means for the training, the years are given as 2012, 2013, 2014, 2015, and 2017. But in testing samples, it is divided into 2012, 2014, 2015, 2016, and 2017. The testing samples do not consist of the year 2013 and training samples do not consist of the year 2013. Thus only four different years that are present on both training and testing samples have been considered that are 2012, 2014, 2015, and 2017. Start and end dates are also provided with peak flux. But for our task, the dataset is divided into four classes just based on years as we are doing investigated predictions based on years by specifying an improved result. The original size of the images is 256 x 256 x 1 pixels which are then converted into 70 x 70 x 1 pixels for fast processing. For the training purpose, 1097 image samples and for testing purpose 100 image samples are considered. It can be seen in figure 3. The dataset has been altered as per the objective of the paper and therefore makes it the first attempt. Moreover, the architecture used in this paper has been compared with the architectures of the previous paper which is detailed in section 2.4. From the give figure 4, it can also be seen how each classes are divided. For the optimization of the gradient, Adam optimizer is used with a learning rate of 0.001 and loss of mean square. Below are the results achieved after training and testing according to the proposed investigation.

4.2 Convolutional Neural Network with Batch Normalization and ELU Activation Function

CNN Layers	Train Loss	Train Accuracy	Test Loss	Test Accuracy
Conv 1	0.0136	0.9701	0.0303	0.9300
Conv 2	0.0064	0.9838	0.1025	0.8000
Conv 3	0.0230	0.9462	0.0234	0.9400
Conv 4	0.0178	0.9584	0.0188	0.9500
Conv 5	0.0097	0.9775	0.0188	0.9400
Conv 6	0.0172	0.9628	0.0170	0.9600
Conv 7	0.0114	0.9709	0.0336	0.9000
Conv 8	0.0188	0.9560	0.0116	0.9700
Conv 9	0.0256	0.9457	0.0573	0.8800
Conv 10	0.0221	0.9475	0.0730	0.7900

Table 2: Results on the solar flare data using batch normalization and elu activation function on the basis of four parameters that are train loss, train accuracy, test loss and test accuracy and CNN layers means number of layers used in Convolutional Neural Networks.

From the table 2, it can be seen that, after applying 8 convolutional layers, the network achieved the best test accuracy of 97% with batch normalization of a CReLU. Further, it was seen that adding more layers decreased the performance of the network. So after 10 convolutional layers, we stopped conducting the results.

4.3 Convolutional Neural Network with Batch Normalization and CReLU Activation Function

CNN Layers	Train Loss	Train Accuracy	Test Loss	Test Accuracy
Conv 1	0.0096	0.9788	0.0181	0.9400
Conv 2	0.0203	0.9579	0.0218	0.9400
Conv 3	0.0169	0.9649	0.0203	0.9500
Conv 4	0.0081	0.9792	0.0143	0.9700
Conv 5	0.0152	0.9614	0.0165	0.9600
Conv 6	0.0275	0.9385	0.0370	0.8900
Conv 7	0.0178	0.9612	0.0723	0.8300

Table 3: This table shows different results on the solar flare data using batch normalization and CReLU activation on the basis of four parameters that are train loss, train accuracy, test loss and test accuracy.

From table 3, it was seen that when applying 4 convolutional layers, the network achieved the highest test accuracy of 97%. Further, it was seen that adding more layers decreased its performance. So after the 7th convolutional layer, we stopped conducting the test.

4.4 Convolutional Neural Network with Dropout and ELU Activation Function

From the above table 4, the network used with dropout did not perform that much well as batch normalization when used ELU activation function. The highest test accuracy is achieved is just 39% and after adding more layer the value of test accuracy starts degrading. We stopped the test after the 6th convolutional layer.

CNN Layers	Train Loss	Train Accuracy	Test Loss	Test Accuracy
Conv 1	0.3233	0.3534	0.3400	0.3200
Conv 2	0.3306	0.3388	0.3300	0.3400
Conv 3	0.3294	0.3411	0.3050	0.3900
Conv 4	0.3398	0.3203	0.3250	0.3500
Conv 5	0.3272	0.3455	0.3400	0.3200
Conv 6	0.3376	0.3248	0.3850	0.2300

Table 4: This table shows different results on the solar flare data using dropout and elu activation on the basis of four parameters that are train loss, train accuracy, test loss and test accuracy.

4.5 Convolutional Neural Network with Dropout and CReLU Activation Function

CNN Layers	Train Loss	Train Accuracy	Test Loss	Test Accuracy
Conv 1	0.3312	0.3374	0.3450	0.3100
Conv 2	0.3223	0.3543	0.3350	0.3300
Conv 3	0.3322	0.3355	0.3350	0.3300
Conv 4	0.3388	0.3223	0.3400	0.3200
Conv 5	0.3289	0.3412	0.3400	0.3200
Conv 6	0.3385	0.3230	0.3500	0.3000

Table 5: This table shows different results on the solar flare data using dropout and CReLU activation on the basis of four parameters that are train loss, train accuracy, test loss and test accuracy.

Also, while using CReLU with dropout function achieved almost the same results, see in table 5. Network with the 2nd and 3rd convolutional layer achieved the same test accuracy of 33%, but by viewing the other three parameters, it can be said that network with 2nd convolutional layer has given the best performance. After adding more convolutional layers, the results start degrading. Thus we stopped conducting the test after the 6th convolutional layer. From all the results, it can be seen that first of all batch normalization has given far better performance when compared with dropout function, and sometimes it acts as a regularizer said by [11]. Further, it was seen that when applying CReLU activation function with batch normalization has given the highest test accuracy of 97% when applied 4 convolutional layers whereas ELU activation function with batch normalization achieved the same accuracy that is of 97%but when applied 8 convolutional. Therefore, this classification method can be used further in novel solar prediction systems.

5 Conclusion

The proposed investigation was tested on solar flares data which consists of 4 classes and was tested up to 40 iterations which proved that the solar flare system which is based on convolutional neural networks will best perform only when it uses batch normalization with CReLU activation function as after applying only 4 convolutional layers, the test accuracy achieved was 97% even after having a small amount of data whereas batch normalization with ELU activation function achieved the same test accuracy but after applying 8 convolutional layers. Thus convolutional neural networks with batch normalization with CReLU activation function proved to be an optimized and accurate method when performed on 1097 train images and 100 test images, thus can be used in future solar flare prediction systems.

References

[1] Gombosi, T.I., Dezeeuw, D.L., Groth, C.P.T., Powell, K.G., Robert Clauer, C. and Song, P., "From Sun to Earth: Multiscale MHD Simulations of Space Weather", *In Space Weather (eds P. Song, H.J. Singer and G.L. Siscoe)*, 2013, doi: 10.1029/GM125p0169.

[2] Nagem, Tarek & Qahwaji, Rami & Ipson, Stan & Wang, Zhiguang & Al-Waisy, Alaa, "Deep Learning Technology for Predicting Solar Flares from (Geostationary Operational Environmental Satellite) Data", *International Journal of Advanced Computer Science and Applications*, 2018, doi: 10.14569/IJACSA.2018.090168.

[3] Colak, T., & Qahwaji, R., "Automated Solar Activity Prediction: A hybrid computer platform using machine learning and solar imaging for automated prediction of solar flares", *Space Weather*, 7, S06001, 2009, doi: 10.1029/2008SW000401.

[4] C. Chifor, D. Tripathi, H. E. Mason, B. R. Dennis, "X-ray precursors to flares and filament eruptions", *A&A* 472(3):967-979, 2007, doi: 10.1051/0004-6361:20077771.

[5] Ashamari, Omar & Qahwaji, Rami & Ipson, Stanley, "Enhanced Prediction of Solar Flares: Developing Automated Techniques for Active Region Feature Extraction to Enable the Efficient Prediction of Solar Flares", 2012.

[6] Bengio, Y., "Deep Learning of Representations for Unsupervised and Transfer Learning", *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, in PMLR 27:17-36, 2012.

[7] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey., "ImageNet Classification with Deep Convolutional Neural Networks", *Neural Information Processing Systems*, 25:10.1145/3065386, 2012, doi: 10.1145/3065386.

[8] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, Tsuhan Chen, "Recent advances in convolutional neural networks", *Pattern Recognition* 77:354-377, 2018, doi: 10.1016/j.patcog.2017.10.013.

[9] Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research* 15:1929-1958, 2014, doi: 10.5555/2627435.2670313.

[10] Xiong, Hui & Barash, Yoseph & Frey, Brendan., "Bayesian prediction of tissue-regulated splicing using RNA sequence and cellular context", *Bioinformatics (Oxford, England)* 27:2554-62, 2012, doi:10.1093/bioinformatics/btr444.

[11] Ioffe, Sergey & Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", 2015, doi: arXiv:1502.03167v3.

[12] Clevert, Djork-Arné, Thomas Unterthiner and Sepp Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)", *CoRR*, 2016, doi: abs/1511.07289.

[13] Kakde, Aditya & Arora, Nitin & Sharma, Durgansh, "Novel Approach towards Optimal Classification using Multilayer Perceptron", *International Journal of Research in Engineering, IT and Social Sciences* 810:29-38, 2018.

[14] Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee., "Understanding and improving convolutional neural networks via concatenated rectified linear units". *In Proceedings of the 33rd International Conference on International Conference on Machine Learning* 48:2217–2225, 2016, doi: 10.5555/3045390.3045624.

[15] Qahwaji, R., Colak, T., "Automatic Short-Term Solar Flare Prediction Using Machine Learning and Sunspot Associations", *Sol Phys* 241:195–211, 2007, doi: 10.1007/s11207-006-0272-5.

[16] Colak T., Qahwaji R., "Automated Prediction of Solar Flares Using Neural Networks and Sunspots Associations", *Soft Computing in Industrial Applications. Advances in Soft Computing* 39, 2007, doi: 10.1007/978-3-540-70706-6_29.

[17] Xuebao Li, Yanfang Zheng, Xinshuo Wang, and Lulu Wang, "Predicting Solar Flares Using a Novel Deep Convolutional Neural Network", *The Astrophysical Journal* 891:10, 2020, doi: 10.3847/1538-4357/ab6d04.

[18] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognitio", *Proceedings of the IEEE*, 86:2278-2324, 1998, doi: 10.1109/5.726791.

[19] Jaesoo Kim, Alistair Mowat, Philip Poole, Nikola Kasabov, "Linear and non-linear pattern recognition models for classification of fruit from visible–near infrared spectra", *Chemometrics and Intelligent Laboratory Systems*, 51(2):201-216, 2000, doi: 10.1016/S0169-7439(00)00070-8.