For the detection of flare in the images I have used deep learning model [CNN] wherein I have created the CNN model with 3 Convolutional model and 2 fully connected model.

For This model we will need a machine with keras ,CV2, PIL installed.

A training directory and validation directory which contains subdirectory as per image class having all the .jpg images.

**Outline of Algorithm:**

For building a powerful image classifier using small data we are going to use CNN(Convolution Neural Networks) as it is good for extracting the features of the image and is widely used for image classification.

For training purpose there is 8o images with both class having equal number of images, For Validation purpose we use 15 images from each class to evaluate our models .

For training this deep learning model ,Data is fed by using batches with the help of **ImageDataGenerator.** This is usually used for data augmentation process (Rotating the image ,zooming ,etc.)This often is used to get more from the data when the dataset is small size.

In this Case ,we have defined our CNN with

* 3 Convolution layers
* 2 Fully connected layer

The Convolution layers extract relevant features from the images and the fully connected layers will focus on using these features to classify well our images .A convolution operator extracts features from the input image using sliding matrices to preserve the spatial relations between the pixels.

We apply ReLu to introduce nonlinearity in our CNN.

We end the model with a single unit and sigmoid activation which is perfect for binary classification. Also, we have used Binary\_Crossentropy loss to train our model

We have used Dropouts to reduce overfitting by not updating the weights of some nodes which is basically preventing the Neural network from relying on one node in the layer too much.

Here we have defined the CNN model from scratch.

We had kept 25 epoch which means the model will be trained 25 times and at each epoch keras will check if our model performed better than the model of the previous epoch.

Finally, we have saved the model to be used further for testing purpose.

Our best model manages to get the Validation accuracy of 93.33 %

On executing the detector.py on the test folder having all the flare image we got an accuracy of 70%

While for the good image we got 100% accuracy.

On executing the model on randomly selected 30 images from flare and good we got 12 image classified as flare and 18 image classified as good.

All the files has been uploaded in the github link as below:

**https://github.com/PRANEET09/Image-Classification**