****

**Flower Detection and Recognition using Convolution Neural Network**

**Project**

****

**Submitted To: Ma’am Mehreen Sirshar**

**Class: BSE-6A**

**Submitted By:**

**Aatiqa Ejaz (2017-BSE-001)**

**Hamna Baqai (2017-BSE-009)**

**Introduction:**

Identifying and recognizing the type of flower is the basic requirement for researches done in the field of Botany as well as other areas. Flowers identification is done by color, shape, texture or any patterns it contain.

**Working with CNN:**  
An image of flower is input, that we’re going to train out model using CNN. Convolutional Neural Network (CNN), also known as convents was proposed by Yann LeCun in 1988, it is used to identify any object from any image. Convolutional neural network has different architecture than other neural networks. Input given to the CNN algorithm is basically image to be classified and output generated by algorithm is set of features. Features of image means that the characteristics of image for instance flower’s image has features like sepals, petals, stem and leaves. CNN is classified into four components named as Convolution, Non Linearity (ReLU), Sub-Sampling or Pooling and Fully Connected Layer.

**Convolution (Convolutional Layer)** is responsible for identifying and recognizing the low level features of objects such as edges, corners etc. from an image. It means the whole network will learn a specific pattern and will identify objects according to the pattern in any image. Convolution is basically the input image using a filter (usually a 3x3 matrix) or a kernel so for convolution the image is scanned from top left corner to the right moving covering the whole width of the image and then repeating the same process until we are done with the whole image. The filter multiplies it’s all values with corresponding values of input image and add all of them making only value to output image unit till the entire image is traversed. The element-wise multiplication generates the output commonly known as feature map. In case of image having multiple channels matrix multiplication is performed in between all the images with three different filters (red, green and blue) and all the results are then summed up to give final output pixel value. Each neuron in output value has overlapping receipted field. After passing through Convolutional Layer the size of image is reduced. Inorder to regain the size same as input image padding is done. Furthermore the Convolutional Layer is organized by three parameters Depth (defines the total number of filters applied during the process of Convolution), Stride (defines the number of pixels to be jumped i.e. 1 pixel, 2 pixels or more) and Zero-Padding (process of adding rows and columns on each side of the feature map).

After passing the input image through filter the resulting image is passed through a function called activation function, usually activation function known as **ReLU (Rectified Linear Unit)** is used which transforms all the negative pixels value to 0 and other positive pixels values remains the similar.

After passing the input image from the convolutional layer we get the feature map then **Pooling or Sub-Sampling Layer** works on image. Similarly like convolutional layer the responsibility of pooling layer is to reducing the spatial size of the features. It is used to extract dominant features which are rotational and positional variants. Commonly Pooling is of two types one of it is max pooling and the other one is average pooling. **Max pooling** returns the max value from the slice of the image convolved by the filter, it also work as noise sub percent it discards the noise from an image. **Average pooling** returns the average value from the slice of the image convolved by the filter.

**The Fully Connected Layer** is a traditional Multi-Layer Perceptron. And basically is an activation function known as classifier. In Fully connected Layer each neuron present in the previous layer is connected to every neuron present in the next layer, this layer uses the output generated from previous layer to identify image.

**Project Flow Chart:**



**Dataset:**

The dataset had 18540 images which we used for training, 2009 images for testing, similarly separate excel files for training as well as testing. We will be classifying images in 102 categories based on their groups they fall in. For instance image number 0, 10, 16, 45 and many more and fall in category 77. Data was divided 60 40 for training and testing but then we also tried with 80 20 and 70 30. Our CNN has 4 layers and has a training accuracy of 92%.



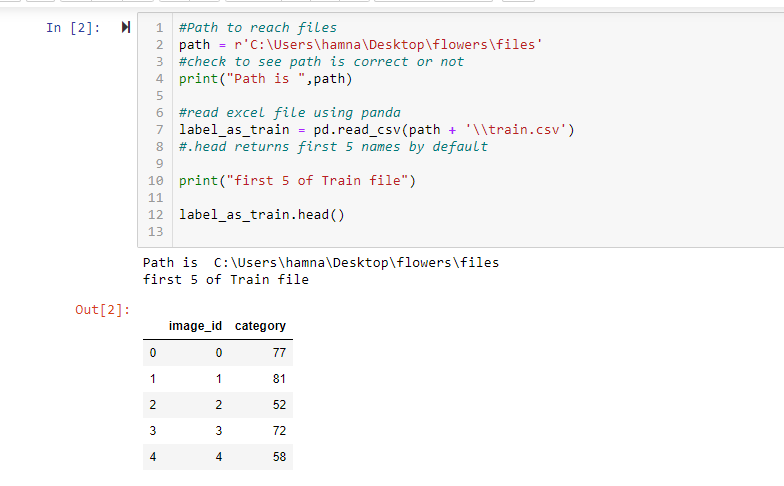
Coding:

Following are the libraries used:

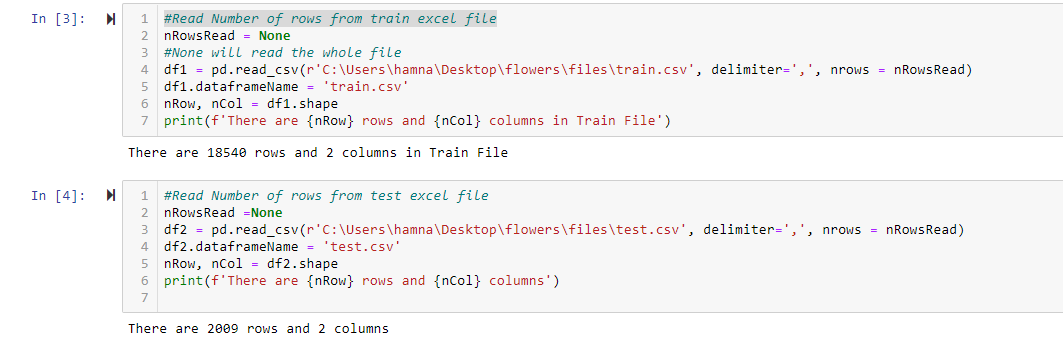


Initially to store path of file, simply copy the destination of folder which contains the dataset and paste it in a variable.

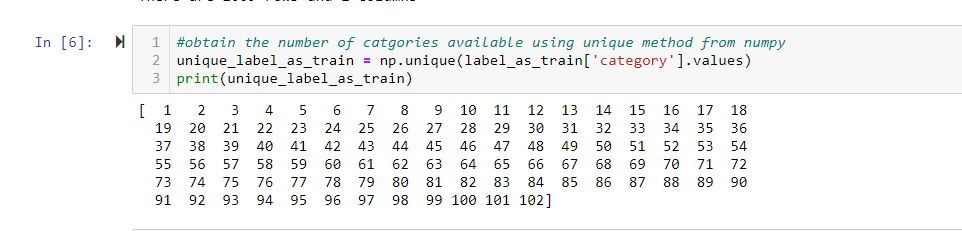
Reading images from train file and storing them in a variable. Displayed only few of them



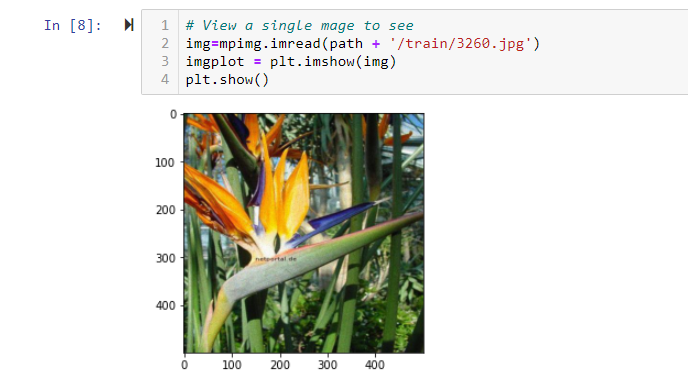
Counting the number of images in train as well as test files



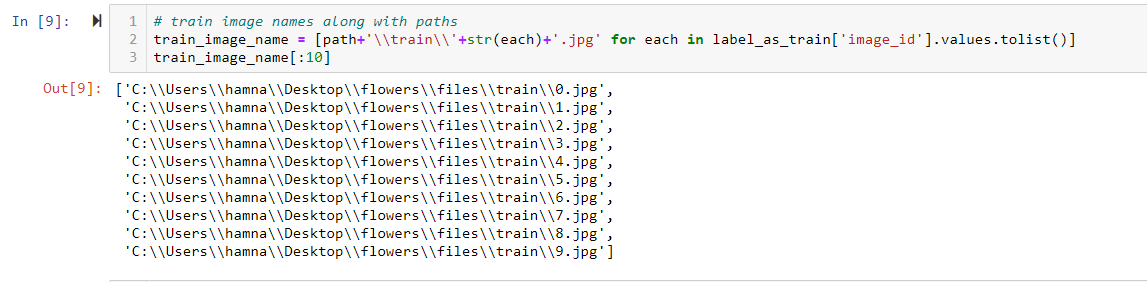
We have 102 category images. Now to uniquely find out the image categories,

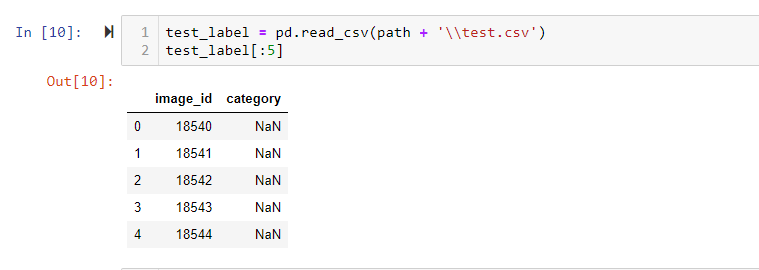


To view that images are correctly loading

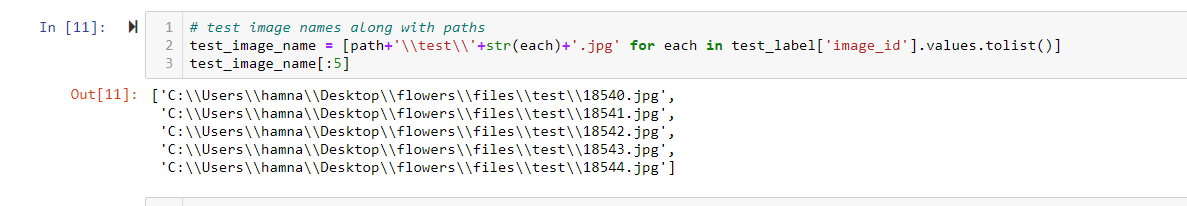


This can also be done for more images. To store path of all images in a variable and checking first ten stored using slicing method

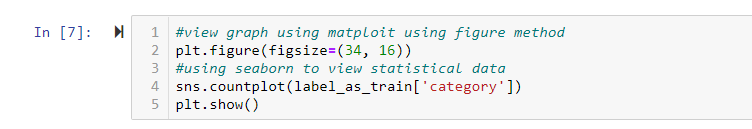




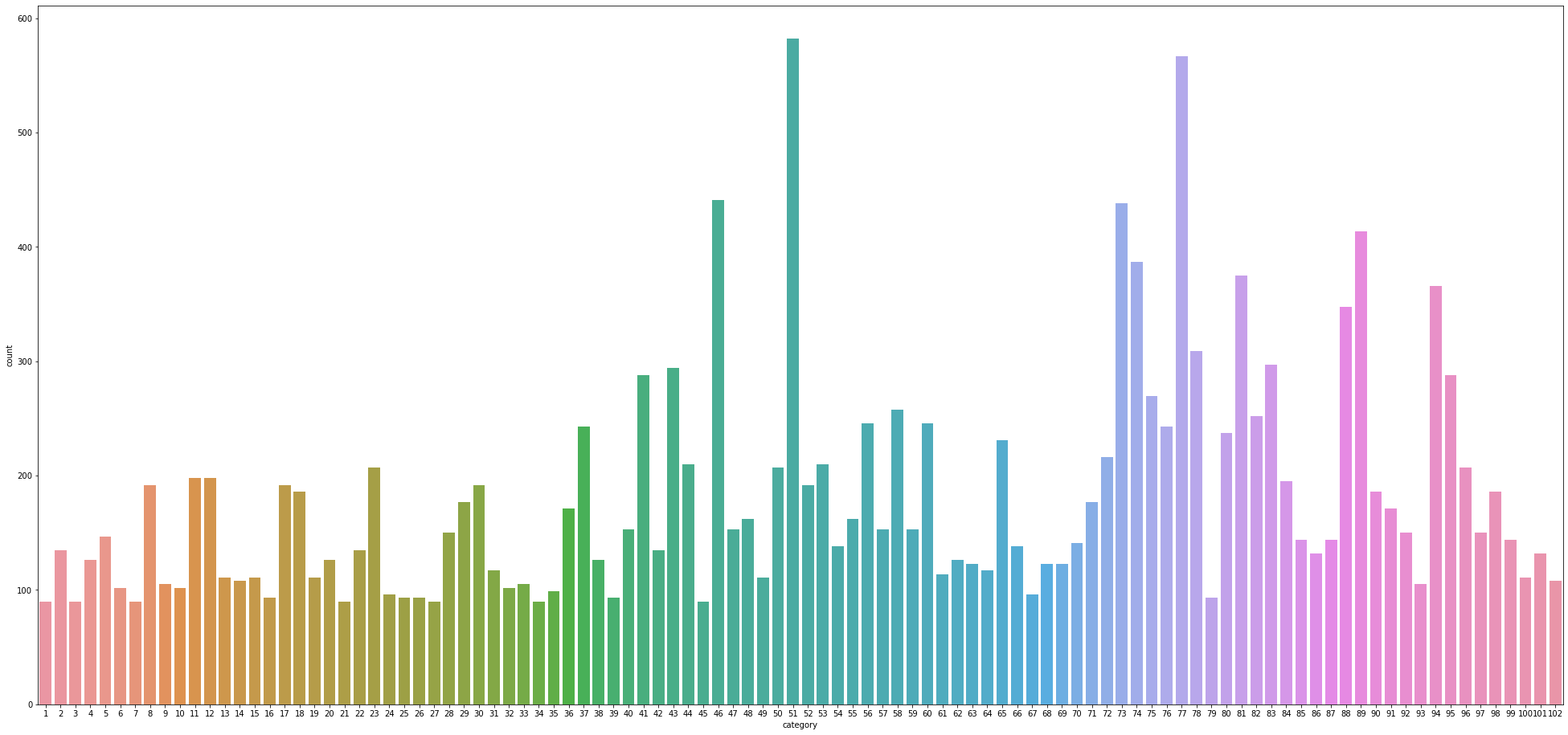
Similarly for testing images



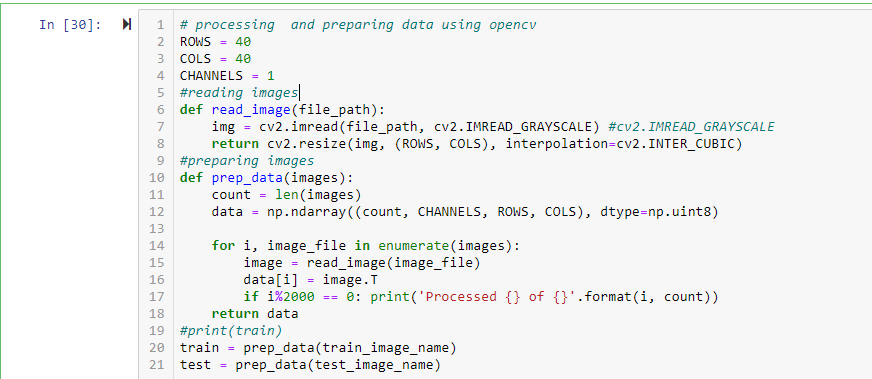
To create Histogram as mentioned in the flow chart

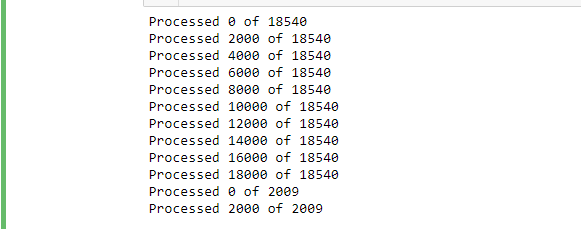


Following graph shows images according to the categories. For instance an image appeared 10 times in a particular category, so the number of times is shown on y axes and x axes show categories.



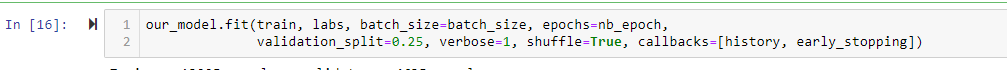
Processing data i.e images.

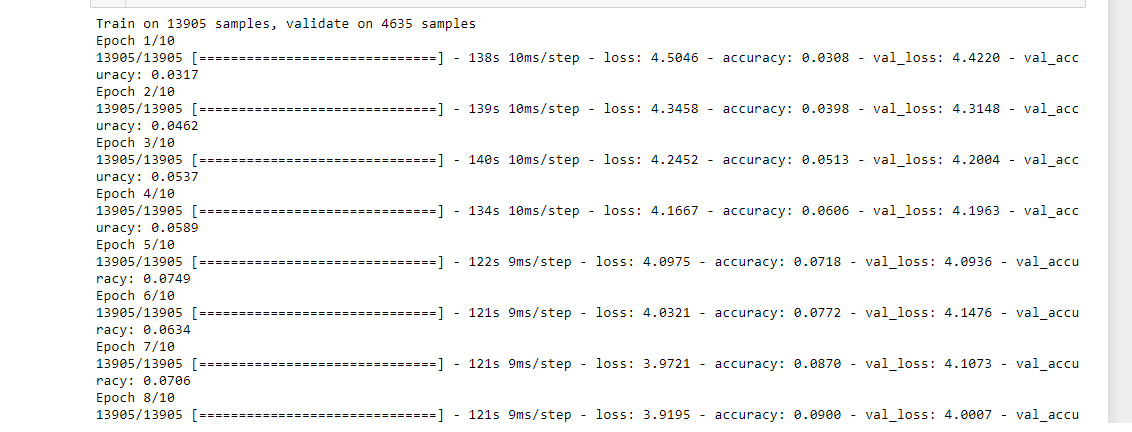


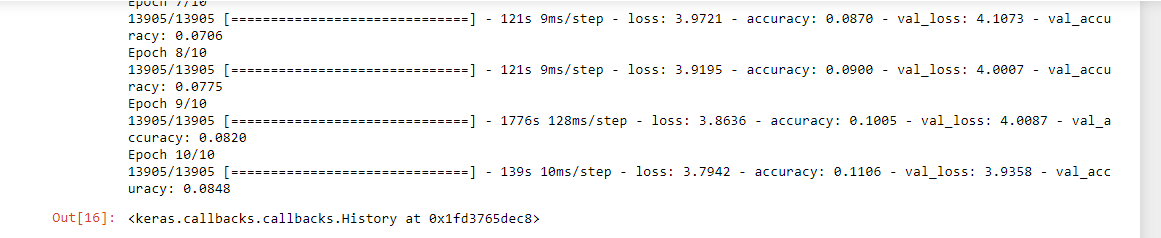


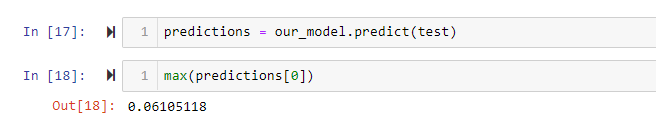
Creating CNN Model

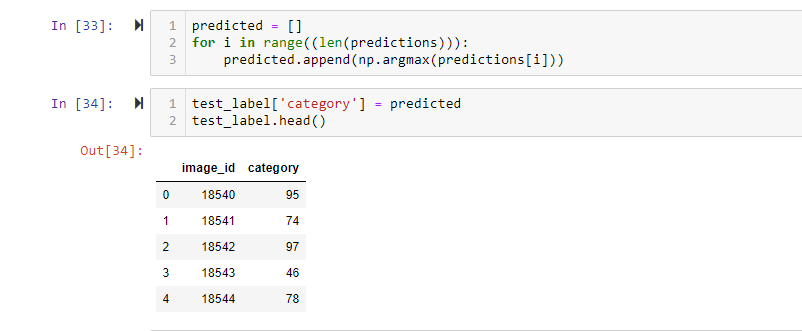




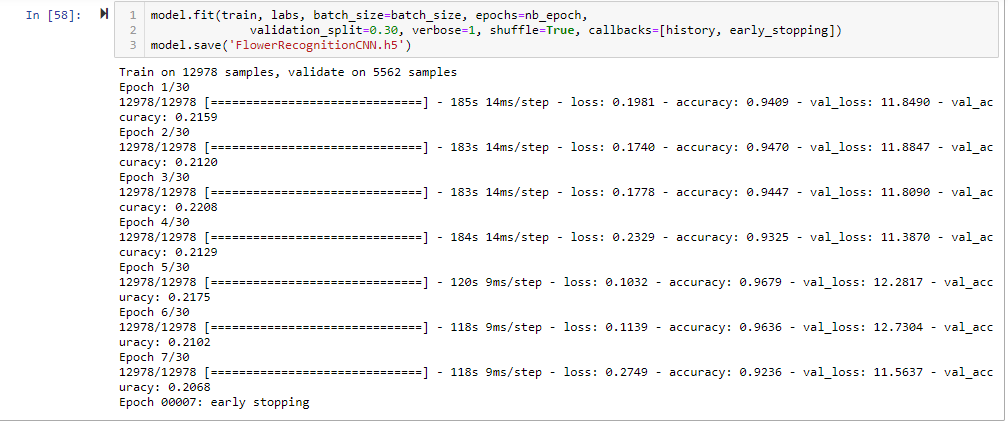




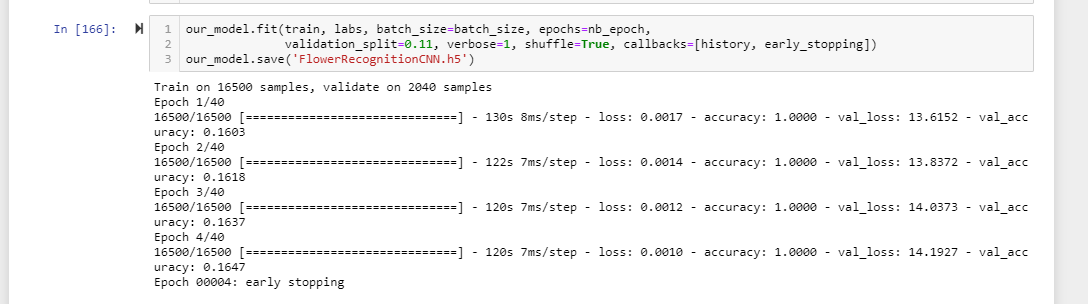




But when we repeatedly apply CNN to our model, increase epochs, change validation size, our accuracy of training increases but validation accuracy decreases.



When changing batch to 128 we got training accuracy up to 100%.



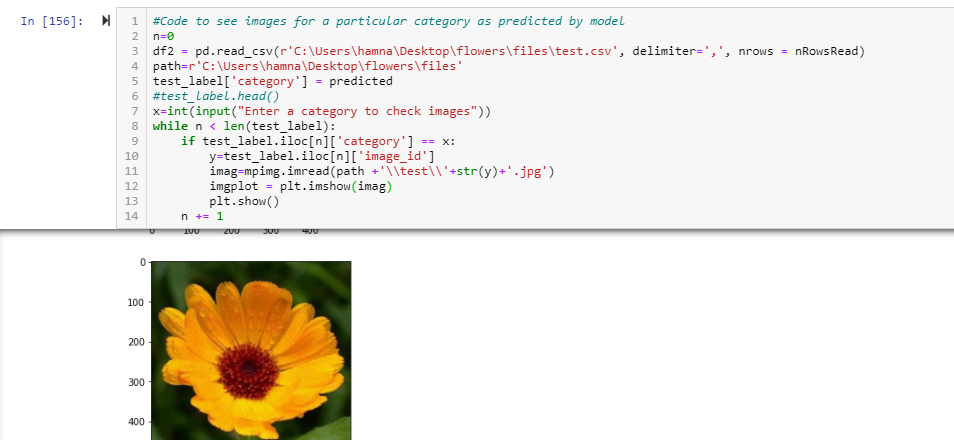
**Checking for Images: Incorrect vs Correct Images prediction**

Image 18541 is identified in category 75. 

Other images of this category in training set include, which do not match, so CNN is not predicting the exact flower from this category. This is because the validation accuracy is varying from 15 to 20 %.



It is working for same cases like the flower in above image is in category 5 as predicted in test and also in category 5 as in trained images.





Conclusion:

Accuracy of training increased to 100 percent but validation accuracy remained at 20%. Due to the large number of categories in dataset, it has become very difficult for the network to correctly identify images into those categories. The training accuracy is far larger than the validation or performance accuracy. It is only because categories are in a large number. But in spite of the fact, some images are predicted in correct category. To obtain more better validation accuracy, we can have more training images and more layers in CNN.