Fruit Recognition from Fruits-360 dataset using CNN

Jyoti Yadav; School of Computer Science and Engineering, Lovely Professional University, Punjab, India; jyoti.11709941@lpu.co.in

Abstract: During this report, I actually have introduced a top quality dataset of pictures containing numerous sorts of fruits and vegetables. Once longing the data set, I'm about to build a model to classify the varied classes of fruits and vegetables from same family, aimed to adapt CNN-Convolutional Neural Network technique for image classification. Except beginner friendly technique, the main reason behind selecting this method is to achieve a most attainable accuracy in image classification.

Keywords: Deep Learning, Image Classification, Fruits Dataset, Tensorflow, CNN

1. Introduction

The Aim of this report is to propose a dataset comprising of various common fruits. The dataset was named Fruits-360 and therefore the digit '360' has been used here to show 360 degree aka full (maximum possible) coverage of common fruits and vegetables. Presently the dataset contains approximately 12,791 images of 20 distinct class of fruits. Training data size: 9586 and Testing data size: 3205.

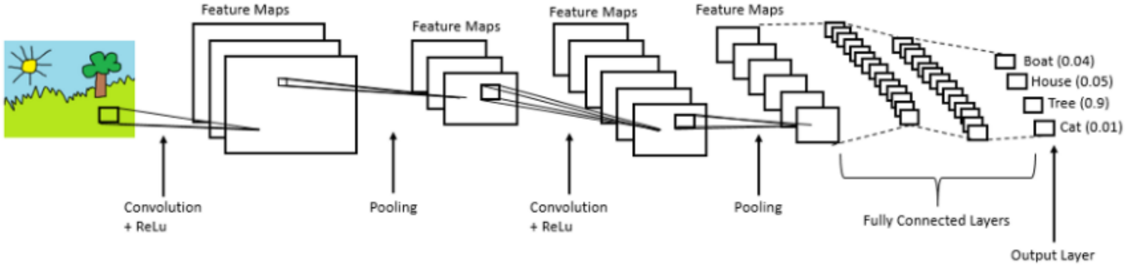
Having a high-quality dataset is important for obtaining good classifier. Most of the prevailing datasets with images contain both the thing and noisy background. This might probably cause to cases where changing the background will lead to incorrect classification of dataset.

The main reason for choosing this dataset to perform the task of identifying fruits for several reasons. On one side, fruits have certain categories that are hard to differentiate, like the citrus genus, which goes both for oranges and grapefruits. Thus the main objective behind this project is to check that how accurately the CNN model successfully accomplished the task of image classification.

2. Technology Used

2.1 Convolutional Neural Network

Convolutional neural networks (CNN) are part of the deep learning models. Such a network are often composed of convolutional layers, pooling layers, ReLU layers, fully connected layers and loss layers. In a typical CNN architecture, each convolutional layer is followed by a Rectified linear measure (ReLU) layer, then a Pooling layer then one or more convolutional layer and eventually one or more fully connected layer. A characteristic that sets apart the CNN from a daily neural network is taking under consideration the structure of the pictures while processing them. Regular neural network converts the input during a one dimensional array which makes the trained classifier less sensitive to positional changes.



Complete CNN Architecture

2.2 Convolution Layers

Convolutional layers are named after the convolution operation. In mathematics convolutional is an operation on two functions that produces a 3rd function that's the modified (convolutional) version of one of the first function. The resulting function gives in integral of the pointwise multiplication of the 2 functions as a function of the number that all in all the initial functions is translated.

2.3 Pooling Layers

Pooling layers are used on one hand to scale back the spatial dimensions of the representation and to scale back the quantity of computation wiped out the network. The other use of pooling layers is to regulate overfitting.

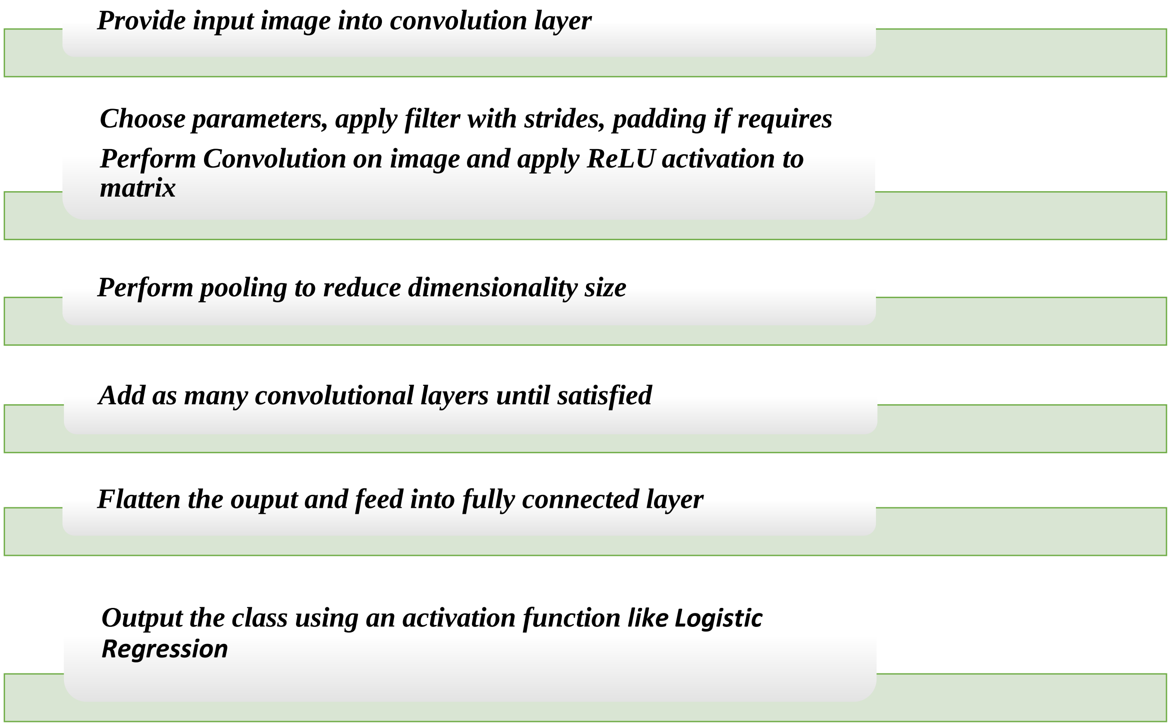
2.4 Fully Connected Layers

Fully connected layers are layers from a daily neural network. Each neuron from a completely connected layer is linked to every output of the previous layer. The operations behind a convolutional layer are a similar as during a completely connected layer.

2.5 Loss Layers

Loss layers are wont to penalize the network for deviating from the expected output. This is usually the last layer of the network. Various loss 8 function exist: softmax is employed for predicting a category from multiple disjunct classes, sigmoid cross-entropy is employed for predicting multiple independent probabilities.

2.6 Breakdown of process involved in CNN



Provide input image into convolution layer

Choose parameters, apply filter with strides, padding if requires

Perform Convolution on image and apply ReLU activation to matrix

Perform pooling to reduce dimensionality size

Add as many convolutional layers until satisfied

Flatten the ouput and feed into fully connected layer

Output the class using an activation function like Logistic Regression

3. Tensorflow Library

For the purpose of implementing, training and testing the network described in this report we used the TensorFlow library. This can be open source framework for machine learning created by Google for numerical computation using data flow graphs. Nodes within graph represent mathematical operations and graph edges represent the N-dimensional data array called tensors. The main components during a TensorFlow system are the client, which uses the Session interface to talk with the master, and one or a lot of employee processes, with every employee method answerable for arbitrating access to minimum of 1 or a lot of procedure devices (such as CPU cores or GPU cards) and for executing graph nodes on those devices as educated by the master. TensorFlow offers some powerful features such as: it allows computation mapping to multiple machines, unlike most other similar frameworks; it's inbuilt support for automatic gradient computation; it can partially execute subgraphs of the entire graph and it can add constraints to devices, like placing nodes on devices of a particular type, make sure that two or more objects are placed in the same space etc.

4. Fruits-360 dataset

These images were obtained by filming the fruits while they are rotated by a motor and then extracting frames. Since, there exist some fruits for which there has been no scientific/popular name assigned to it, so on the basis of physical characteristics they are labeled along with digits. For instance, apple red 1, apple red 2, similarly with other categories also.

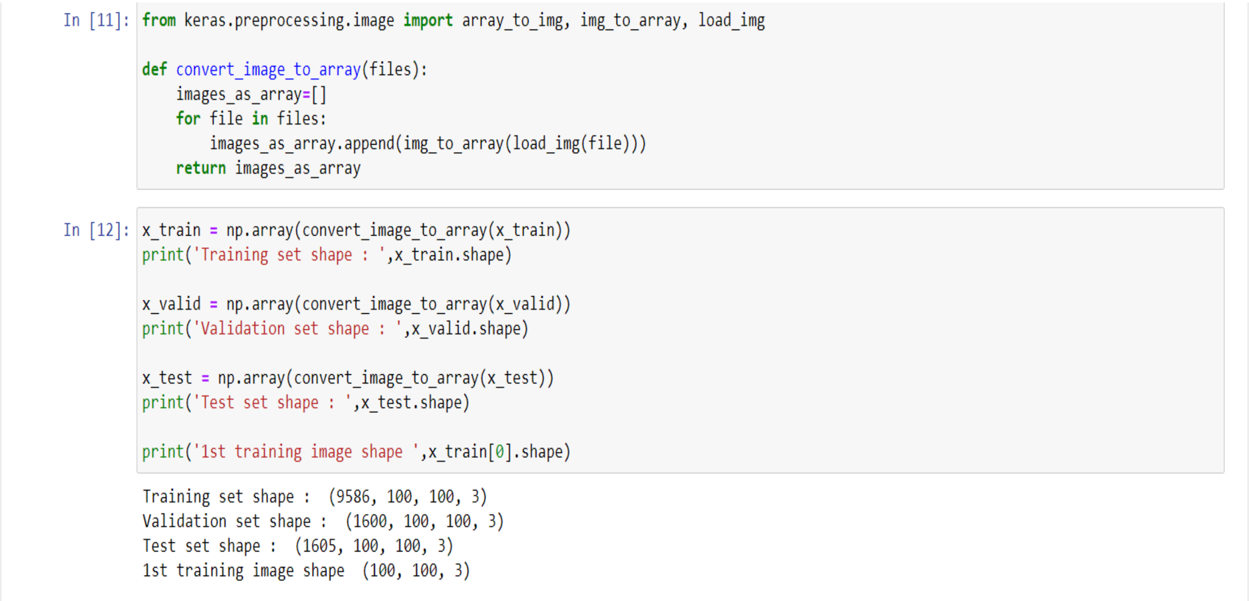
4.1 Few visual examples of fruits and vegetables used in dataset

 Apple	 Blueberry	 Cherry	 Kiwi

 Mango	 Papaya	 Pomegranate	 Lemon

5. Description of Architecture

Here I’m using Convolution Layer followed by maxpooling layers, dropout layer, flatten layer and some fully connected layer(Dense layer).



```
In [11]: from keras.preprocessing.image import array_to_img, img_to_array, load_img

         def convert_image_to_array(files):
             images_as_array=[]
             for file in files:
                 images_as_array.append(img_to_array(load_img(file)))
             return images_as_array
```

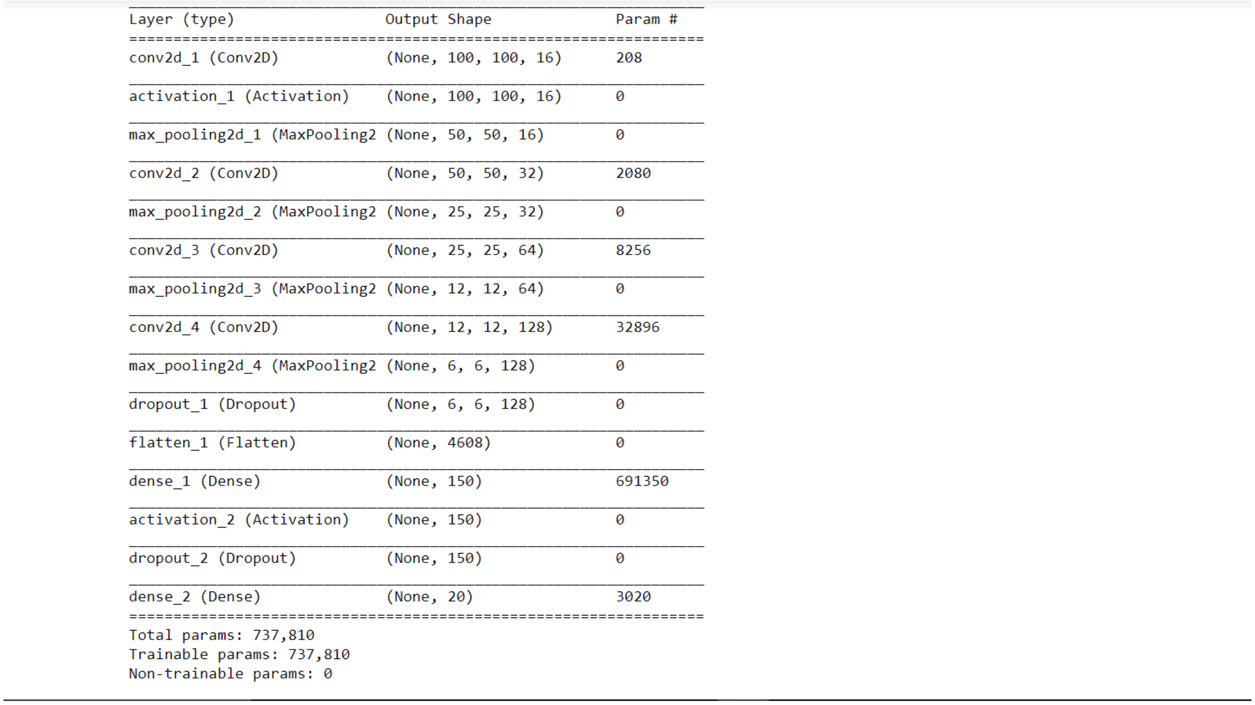
```
In [12]: x_train = np.array(convert_image_to_array(x_train))
         print('Training set shape : ',x_train.shape)

         x_valid = np.array(convert_image_to_array(x_valid))
         print('Validation set shape : ',x_valid.shape)

         x_test = np.array(convert_image_to_array(x_test))
         print('Test set shape : ',x_test.shape)

         print('1st training image shape ',x_train[0].shape)
```

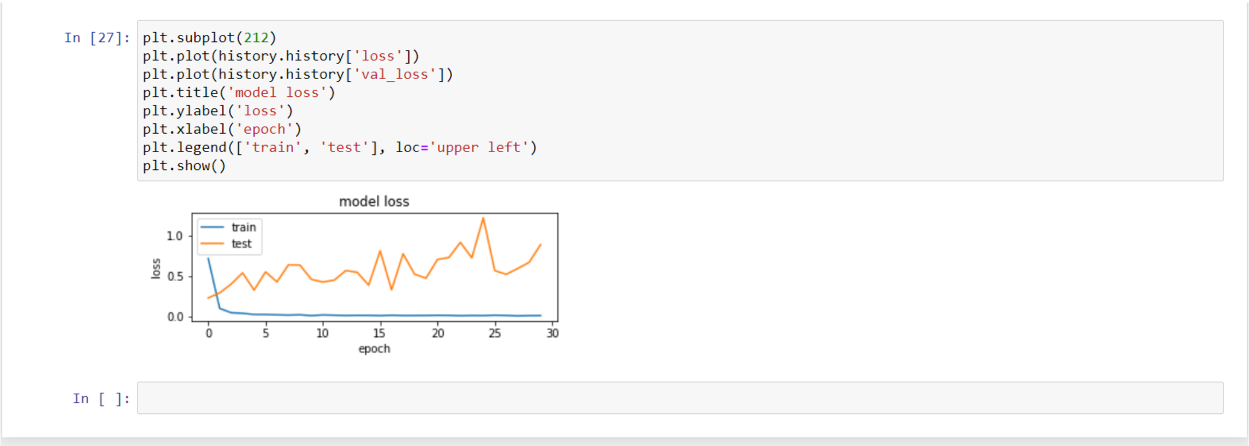
```
Training set shape :  (9586, 100, 100, 3)
Validation set shape :  (1600, 100, 100, 3)
Test set shape :  (1605, 100, 100, 3)
1st training image shape  (100, 100, 3)
```



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_1 (Conv2D)            (None, 100, 100, 16)      208       
_________________________________________________________________
activation_1 (Activation)    (None, 100, 100, 16)      0         
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 50, 50, 16)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 50, 50, 32)        2080      
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 25, 25, 32)        0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 25, 25, 64)        8256      
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 64)        0         
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 12, 12, 128)       32896     
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 6, 6, 128)         0         
_________________________________________________________________
dropout_1 (Dropout)          (None, 6, 6, 128)         0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 4608)              0         
_________________________________________________________________
dense_1 (Dense)              (None, 150)               691350    
_________________________________________________________________
activation_2 (Activation)    (None, 150)               0         
_________________________________________________________________
dropout_2 (Dropout)          (None, 150)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 20)                3020      
=================================================================
Total params: 737,810
Trainable params: 737,810
Non-trainable params: 0
```

6. Result & Discussion

After developing model for Fruits-360 dataset followed by data visualization and finding accuracy with respect to 30 epochs, I've summarized the history of loss through plotting:



```
In [27]: plt.subplot(212)
         plt.plot(history.history['loss'])
         plt.plot(history.history['val_loss'])
         plt.title('model loss')
         plt.ylabel('loss')
         plt.xlabel('epoch')
         plt.legend(['train', 'test'], loc='upper left')
         plt.show()
```

```
In [ ]:
```

7. Conclusion

I have taken a complex database of images with fruits. The task of classifying images of fruits and vegetable can be done by building a Convolutional Neural Network (CNN) through which the major objective is to attain maximum possible accuracy fruits and vegetables. Since, there are still more class of fruits and vegetables might be left out. So another objective is to expand the data set to include more fruits in order to maintain achieved accuracy with more wide variety of classes.