**FruitTL: A Transfer Learning Approach towards Classification of Fruits**

Ankan Bhattacharyya, Shibaprasad Sen\*

*Abstract: A pretrained VGG16 model has been applied with transfer learning approach to the FruitNet dataset. The model has obtained an accuracy of 99%, which beats the state of the art models that has been tested on the FruitNet dataset. The code is available in* [*GitHub*](https://github.com/iamankan/fruits-ml/blob/conference1/FruitmlTL.ipynb)*.*

*Keywords: Transfer learning, VGG16, CNN*

Introduction

Dataset

The dataset [1] has been collected using high resolution mobile phone rear camera. Meshram et. al. [1] has not mentioned anything about the specifications of the mobile camera. However, the specification of the images taken has been mentioned to be .jpg images of dimension 3024x3024. The images has been resized down to 256x256. The dataset has three subfolders, namely, Good Quality, Bad Quality and Mixed Quality fruits. For each of these categories, there are mainly 6 fruit types, namely, Apple, Guava, Banana, Lime, Orange and Pomegranate. Meshram et. al. claims that the data were taken at different lighting conditions in different background.

This manuscript deals with the fruit classifications, so, the quality of the fruits are not in the scope of the current experiment. We rearranged the data into six classes of different fruit types. The data is uploaded to the [cloud](https://drive.google.com/drive/folders/11V65XEXsYM2pZV2lERbK0ESxSARWxCMS?usp=sharing). We are experimenting with a total of 18452 data samples.

Methodology

We have made a split of the data into training and testing samples. The ratio of the training to the testing samples is 8:2. Further the training data is split into 90% training and 10% validation set. All the splitting done for this experiment are random and shuffled, so that the distribution is uniform. So, after the splitting, the following are the data counts:

|  |  |
| --- | --- |
| Training | 11629 |
| Testing | 5534 |
| Validation | 1289 |

We have used VGG16 as the pre-trained deep learning model. Chakraborty et. al. [2] showed that using pre-trained weighted neural network is better that running a CNN from scratch, for a small scale of data. For our experiment, the scale of data is medium and cannot be considered as large.

A high level workflow diagram is shown in Fig. 1.

|  |
| --- |
| Diagram  Description automatically generated |
| Fig. 1. High level workflow diagram |

So, we approached the problem with transfer learning. A brief description of the VGG-16 is given in the following paragraph.

VGG-16 is a widely used CNN model. The network has 13 convolution layers and 3 dense fully connected layers. Hence it is called VGG-16. Fig. 2 illustrates the high-level block diagram for VGG-16. The convolution layers have 3 x 3 filter. Batch normalization has been used in this model. This is for achieving stability without overestimation or underestimation. The output layer has 1000 nodes. This network, when considered for transfer learning is assumed to have been pretrained with weights on imagenet dataset [3].

|  |
| --- |
| Diagram  Description automatically generated |
| Fig. 2. A high level block diagram for VGG-16 |

Transfer learning is the process where the pre-trained network is used with pretrained weights. As seen in Fig. 2, the last stack of dense layer produces a very high dimension of flattened features. We use this feature and pass it through 2 dense layer. The first layer contains 1024 neurons and the final layer contains 6 neurons, which represent 6 classes of the fruits in the dataset. Since the model is already pretrained, we don’t retrain the layers of VGG-16 and hence freeze those layers. The only thing that is trained is the two dense layers that we added to the last entries of the stack. A high level diagram is illustrated in Fig. 3. We term this model as FruitTL, which is an acronym for Transfer Learning on FruitNet dataset [1].

|  |
| --- |
| Diagram  Description automatically generated |
| Fig. 3. A high level block diagram for transfer learning using pretrained VGG-16 |

The training of the model is done over 20 epochs, in Google Colab. The dense layer of 1024 neurons uses the Rectified Linear Unit (ReLU) activation function [5]. The ReLU function is illustrated in Fig. 4.

|  |
| --- |
| A Practical Guide to ReLU. Start using and understanding ReLU… | by Danqing  Liu | Medium |
| Fig. 4. Rectified Linear Unit activation function |

For the last classification dense layer, having 6 neurons, Softmax activation function [5] has been used. The Softmax function is illustrated in Fig. 5.

|  |
| --- |
| SoftMax Activation Function-InsideAIML |
| Fig. 5. Softmax activation function |

The model saved the weights in the overall training process at its best performance. Quantitatively, the best performance is considered as the minimum loss value. The loss function used in the training process is categorical cross-entropy [6]. Categorical cross entropy is used because there are more than two data (hence not binary). Th optimizer used for the training is the Adam optimizer [7].

Results and discussions

Conclusion and future scope