Implement logistic regression to classify fruits with dataset

Anand kumar

Varanasi, India

anandjoy2635@gmail.com

www.ai-techsystems.com

Introduction and project Overview:-

Skilled robots are being used for fruit harvesting in farms, this helps in reduction of man power and to increase the speed of work. The main part in robot harvesting is detection of the fruit , if robot cannot detect the correct fruit it cannot do the estimated work. So fruit detection is one part of robot harvesting and we will use Deep Convolutional Neural Networks (DCNN) to develop fruit classifier. So here we will develop Fruits classifier that will classify the fruits present in an image and will predict the percentage if there is no fruit in that image**.**

Analysis

Data Exploration

There in all 81 fruits with 55244 images. The following fruits are included: Apples (different varieties: Golden, Golden-Red, Granny Smith, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red), Cactus fruit, Cantaloupe (2 varieties), Carambula, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Clementine, Cocos, Dates, Granadilla, Grape (Pink, White, White2), Grapefruit (Pink, White), Guava, Huckleberry, Kiwi, Kaki, Kumsquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine, Orange, Papaya, Passion fruit, Peach, Pepino, Pear (different varieties, Abate, Monster, Williams), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum, Pomegranate, Quince, Rambutan, Raspberry, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red), Walnut.

Data Visualization

1. After loading data from dataset, using python’s matplotlib Iplotted the images according to their classes.
2. Here is a graph which shows the color representation in an training image using CV2, which shows that blue has some high value areas in the image.

Algorithm and techniques:-

1. The base of the model being made is Deep Learning.
2. The technique used is Convolutional Neural Networks, the model is being developed from scratch.
3. We will be designing Convolutional Neural Networks and connect it to deep neuralnetwork.
4. Various filters and pooling layers for predicting the image more properly.
5. The input to the model will be an image of size 224*224 , which is converted to an array of dimension 4.
6. The dataset being to large it is not possible to convert and train the model on local machine, so platform used here is the instance at google cloud.

Detailed Architecture:

- First I started with 16 filters and kept padding as 'same' as it may help me not losing the data.
- Then a MaxPooling layer with pool_size=2 to reduce our data width wise.
- Then to go deep inside and classify I went until the layer with 128 filters.
- A droupout layer to reduce chances of overfitting.
- To connect CNN layers to fully connected network flattering layer is used.
- Then a hidden layer with 500 nodes is connected having 'relu' activation function.

Benchmark

- The benchmark for the model being developed is calculated from the accuracy obtained from training set.

- There are various already created models available on Kaggle which will be used to compare the accuracy.

Methodology

Data Preprocessing:-

- LoadingDataset: To load dataset, we populate a few variables with the load_files function from the scikit-learn library: o train_files, valid_files, test_files - numpy arrays containing file paths to images o train_targets, valid_targets, test_targets - numpy arrays containing onehot-encoded classificationlabels. o fruit_names - list of string-valued dog breed names for translatinglabels.

- Now the data obtained is in image format, we must convert it in the form the CNN model takes input that is Numpy Array. To convert the steps followed are: o The input image is first converted to numpy array of dimension 3 by using img_to_array function. o Then the array from the previous array is converted to dimension 4.

Implementation

- Deep Learning is used for training the model.
- Convolutional Neural Networks was designed and connected to deep neural network.
- The model takes an input array of dimension 4, processes it further through the network.
- The filter size increases in following sequence 8,16,32,64.
- Model has dropout layers to avoid overfitting.
- Now model is ready for training.

Refinement

- After implementation of the model the changes made were few dropouts layer were added to the model as the dataset is so size 64000 image.
- The dropout values we used are 0.2 and 0.4, with increasing the dropout, there is some increase in validation accuracy and decrease in loss.

Results

Model Evaluation and Validation

- The accuracy obtained from the training set is 99.47% and from testing set is 97.68%.And when the model is tested on image from real world it failed to predict perfectly the type of fruit.

Justification

- According to our benchmark models our model proved to be very effective on testing data and our goal is achieved but still our model is not up to mark and still can improve more.

Conclusion

Free-Form Visualization

- Confusion Matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm.

Reflection

The overall process of development of the model from starch was little though but due to this project I learnt many new techniques and cleared my concept regarding machine learning. The most interesting aspect was to preprocess the data and to do the same it requires large memory. The model takes an input of 224*224, and predicts the output accordingly. This can be used as a part for many application and automation systems.

Improvement

The improvement in this project is training the model for more epochs and increasing the prediction accuracy for image on unseen data.