Kai Tai Tam

Domain Background

Ever since the beginning of human existence, fruit have always been considered as healthy food in human diet. There are thousands of diets menu available online; regardless of the purpose of the menus, almost all of them suggest including at least one type of fruits in the diet. The nutritional values of fruit are irreplaceable for human. In fact, the United States spent $6.6 billion dollars on imported fruit in 2015, while they only spent $1.8 billion dollars in 1995 [1]. It shows that American have started to pay more attention to their health, as well as the nutritional value of fruits.

As there are many fruits available in the market today, consumers may not know every fruit and their nutritional value. Nowadays, to learn more about anything you imagined, using a search engine would be a smart choice. However, you would need to know the fruit's name to complete this task. It would be more convenient for people, say consumers in the market, to simply take a picture with their phone and it would know what fruit is in the picture.

Muresan and Oltean [2] conducted a study similar to this problem. They trained a convolutional neural network model on a "dataset of images containing fruits" in order to develop a model that was able to identify fruits from images. They collected the data by taken picture themselves using a camera. They took them in a close angle and removed the background using software.

While they have an impressive result: achieving more than 90% accuracy in the test set, their data would only be valid if the images were taken in a certain angle and distance. An image of a grape, for example, was taken much closer than an image of an orange. Therefore, it is difficult for general people to utilize this model, because not many people would take out one particular grape, take a picture in a very close distance, and remove their background. Therefore, the dataset used by Muresan and Oltean[2] was not applicable for my purpose. I will be constructing a model that would recognize fruits based on pictures that a normal individual would take on their cell phone.

Problem Statement

The goal for this paper is to develop an algorithm that will predict a fruit based on a fruit image provided by users. To achieve such a goal, utilizing image recognition would be a great choice. As image recognition become more popular, it is possible to train a model to recognize fruits.

Datasets & input

In order to train a model to recognize fruit images, it is essential to obtain fruit image data, and the fruit names. They are not many existed dataset that are suitable for my need. Although there is a relevant fruit images dataset in Kaggle, which will discuss more detail in benchmark model section, the images are not taken in a way that a normal individual would. Therefore, that dataset is not able to generalize to real world situation and is not suitable for our purpose of my study.

Instead, I am going to create my own dataset by searching images on Google. I am going to use an image extraction tool developed by Hardik Vasa [3]. This tool will download images showed on Google based on my keywords, and each fruit will have its own folder. I randomly chose 22 fruits from a list of fruits from Wikipedia [4] and search for their images on Google using the image extraction tool.

Even though I will try as precise as possible to obtain images of fruits, there will be images that does not match what I searched for, such as searching for orange when the image only showing orange color. To deal with this issue, I will clean up the data by going through all images and deleting the images that do not contain the fruit itself. I am expecting each categories will have approximately 100 images, a total of 2,200 images, in order to make sure each category is balanced and has enough data for validation set and test set.

Solution statement

Given the nature of the question is image based data, Convolutional neural network would be an appropriate choice, because CNN performs well in detecting features such as shapes and textures. These features will be the clues for our model to predict what fruit is in the image. Each fruit will have its own binary label. If an image belongs to a particular label, such as orange, it will display as 1, 0 otherwise. Therefore, we will have 22 label columns corresponding to the fruit names. Since each image can only have one label, my CNN will use “softmax” as the last layer. The highest probability of the label will be selected as the predicted fruit.

Benchmark model

Since the most important concern for my model is to be accurate, it is important to show that the model would perform better than random guessing. As there are 22 categories and each category is about 100 images, the probability of predicting a correct answer is 1 in 22, or 4.5%. The model should pass 4.5% to indicate that it has in fact learned through training. I would call this benchmark model as benchmark 1.

For benchmark 2, I am going to train a benchmark model using neural network. I will first flatten the images in order to pass them to a neural network model. Then I will use at most two hidden layers in the model, depends on which will perform better. The number of nodes in final layer will be the same as the number of fruit labels in the dataset and the activation will be “softmax.”

Evaluation matrix

When training the model, I will be using categorical cross-entropy loss and accuracy on training set and validation set to determine how well the model will perform. Categorical cross-entropy loss is best used when constructing a multi-class classifier. It checks whether the model has done well when classifying a label by comparing to the true label. It will return a lower value if the model has done well in predicting labels. Categorical cross-entropy loss is perfect in this case because each image in the dataset can only belong to one of the twenty two categories. If the model did a great job in predicting fruits, the score for categorical cross-entropy will be low.

In terms of accuracy, I will use it to keep track how well the model was learning by looking at the accuracy on training set, and how well the model is going by checking the accuracy on the validation set. It is calculated by the number of corrected labels divided by the total labels. The higher the accuracy in validation set, the better the result in general. As the number of epochs increases, the accuracy in training set and validation set will also increase. However, I should pay attention to the accuracy for validation set, as when the accuracy validation set decreases and increases in training set, it maybe a sign of overfitting.

Project design

To begin with, the image will be downloaded from google using Hardik [3] download tool on python. Each fruit will have its own folder and all images that belong to the fruit will be placed in the folder. There will be image-quality check in order to make sure all images in the folder will be appropriate, because it is possible that some images do not belong to the fruit. I then will find additional images for that particular fruit in order to replace the inappropriate image, if that particular label samples are below 90. All the images then will resize to 100x100 pixels then divide by 255 in order to standardize the images.

After all images are clean, I will randomly break the dataset down into three sets: 80% of training set, 15% of validation set, and 5% testing set. I am treating the testing set as real-life pictures that a normal individual would take, so I will not interfere the testing set until the very last part of the study.

Then I will construct a Convolutional Neural Network from scratch. The architecture will be uncertain, but will be constructed similar to below:

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 100, 100, 16)	208
max_pooling2d_4 (MaxPooling2	(None, 50, 50, 16)	0
conv2d_5 (Conv2D)	(None, 50, 50, 32)	2080
max_pooling2d_5 (MaxPooling2	(None, 25, 25, 32)	0
conv2d_6 (Conv2D)	(None, 25, 25, 64)	8256
max_pooling2d_6 (MaxPooling2	(None, 12, 12, 64)	0
dropout_1 (Dropout)	(None, 12, 12, 64)	0
global_average_pooling2d_2 (	(None, 64)	0
dense_2 (Dense)	(None, 22)	1430

Total params: 11,974
Trainable params: 11,974
Non-trainable params: 0

If the model failed to reach 50% accuracy in the validation set, I would try to implement a pre-trained model (such as VGG16) in keras using transfer learning in order to test whether a pre-trained model could do a better job in predicting fruits.

The model with the best performance will be chosen along with their loss scores and accuracy. I will also talk about what will be the potential improvement for the model as well as the design of this study in conclusion.

Reference

1. https://fas.org/sgp/crs/misc/RL34468.pdf
2. https://github.com/Horea94/Fruit-Images-Dataset
3. https://github.com/hardikvasa/google-images-download
4. https://simple.wikipedia.org/wiki/List_of_fruits