Capstone Project Proposal

Mustafa Mahmoud abd Elkawy

Machine Learning Nano degree

14th of August 2018

I. Domain Background

With the recent huge advances in technology, the importance of visual recognition and image processing has significantly increased. As Deep Learning and Computer Vision help us better understand images and extract useful information out of them. In the field of Self-driving cars, healthcare and pretty much every field that deals with images, visual recognition is proven to be of high importance.

For this project I chose the task of identifying fruits. This task can be achieved by a deep neural network that is capable of identifying fruits from images. This is part of a more complex project that has the target of obtaining a classifier that can identify a much wider array of objects from images.

Related academic research: this research paper.

II. Problem Statement

Fruits have certain categories that are hard to differentiate, like the citrus genus that contains oranges and grapefruits. Thus I want to see how well an artificial intelligence can complete the task of classifying them.

In this project we will be using this data set on kaggle , data set on GitHub to train a model to classify the image with the Fruit into its respective category using CNN architecture implemented with keras.

The goal of the project is to make classifier capable of taking an image of Fruit and predict its respective category.

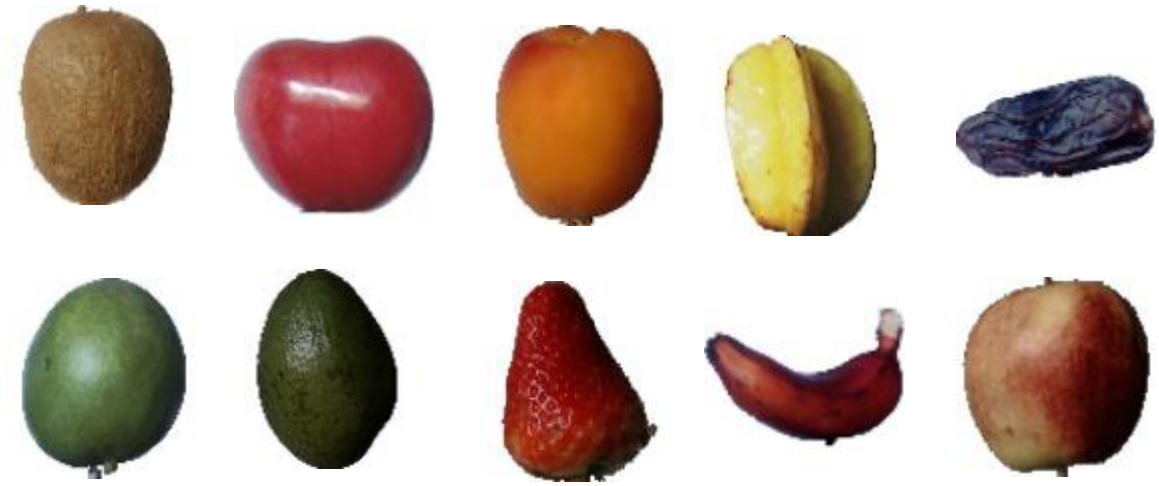
III. Datasets and Inputs

In this project, I'll be using this data set on kaggle , data set on GitHub it contains 55244. images of Fruits. It'll be the main motive for the project. It was collected by Mihai Oltean.

Here’s the content of the dataset. It

- Total number of images: 55244.
- Training set size: 41322 images (one fruit per image).
- Test set size: 13877 images (one fruit per image).
- Multi-fruits set size: 45 images (more than one fruit (or fruit class) per image)
- Number of classes: 81 (fruits).
- File format: standard RGB images
- Image size: 100x100 pixels.

Preview for the content of the dataset



Input:

When using Tensor Flow as backend, Keras CNNs require a 4D array (which we'll also refer to as a 4D tensor) as input, with shape

(nb_samples, rows, columns, channels)

Where `nb_samples` corresponds to the total number of images (or samples), and `rows`, `columns`, and `channels` correspond to the number of rows, columns, and channels for each image, respectively.

For instance (subset of the data) the target variable distributed as:

Label	Number of training images	Number of test images
Avocado	427	143
Avocado ripe	491	166
Banana	490	166
Banana Red	490	166
Cactus fruit	490	166
Cantaloupe 1	492	164
Cantaloupe 2	492	164
Carambula	490	166
Cherry 1	492	164
Cherry 2	738	246
Cherry Rainier	738	246
Cherry Wax Black	492	164
Cherry Wax Red	492	164
Cherry Wax Yellow	492	164
Clementine	490	166
Cocos	490	166
Dates	490	166
Granadilla	490	166
Grape Pink	492	164
Grape White	490	166
Grape White 2	490	166
Grapefruit Pink	490	166

There is no imbalance in the data, each class almost has the same number of training images and test images.

IV. Solution Statement

The solution of this problem is to build and train a model that can classify the new unseen image into one of the 81 mentioned classes accurately.

I intend to make classification using Convolutional Neural Network (using Transfer Learning), to get an acceptable accuracy. I intend to use CNN using keas.

V. Benchmark Model

I'll benchmark my model against the results presented in this research paper.

The structure of the neural network used in the paper:

Layer type	Dimensions	Output
Convolutional	5 x 5 x 4	16
Max pooling	2 x 2 — Stride: 2	-
Convolutional	5 x 5 x 16	32
Max pooling	2 x 2 — Stride: 2	-
Convolutional	5 x 5 x 32	64
Max pooling	2 x 2 — Stride: 2	-
Convolutional	5 x 5 x 64	128
Max pooling	2 x 2 — Stride: 2	-
Fully connected	5 x 5 x 128	1024
Fully connected	1024	256
Soft max	256	60

The calculated accuracy of the neural network used in the paper was 96.19%.

I am looking to achieve this accuracy or to approach it to some extent.

VI. Evaluation Metrics

I'll be evaluating my model using accuracy score test on the test set to check the accuracy of my model.

VII. Project Design

In this project, I will break the notebook into separate steps.

- Step 0 : Import Datasets
- Step 1: preprocessing steps supply images to a pre-trained network in Keras
- Step 2: Extract Bottleneck Features for Train set, valid set, Test Set.
- Step 3: rescale the images by dividing every pixel in every image by 255.
- Step 4: Obtain Bottleneck Features
- Step 5: create Model Architecture
- Step 6: Train the Model
- Step 7: Test the Model
- Step 8: Test the Model on Sample Images!