MACHINE LEARNING NANODEGREE PROGRAM

CAPSTONE PROPOSAL

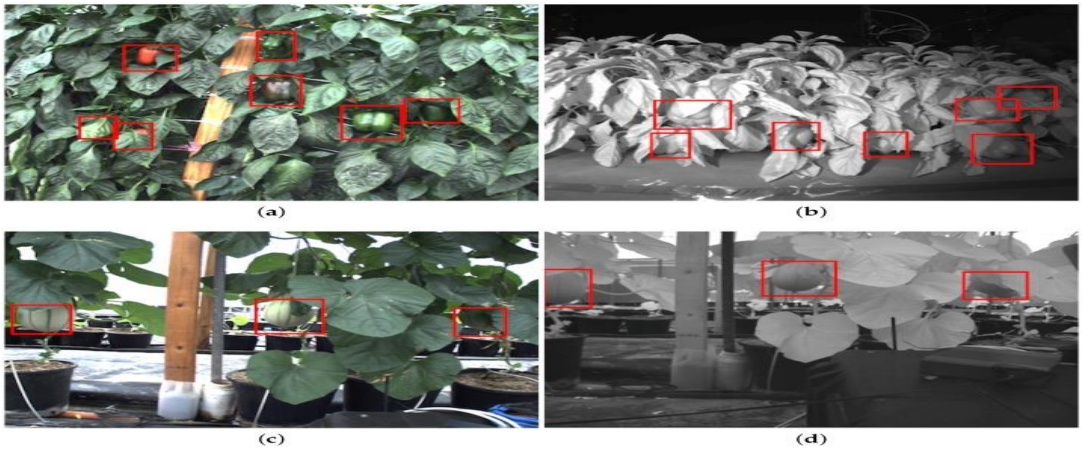
Satish Fulwani

02/12/2018

FRUITS CLASSIFIER

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.



Project Background

Although many researchers have tackled the problem of fruit detection, the problem of creating a fast and reliable fruit detection system persists, as found in the survey by "http://www.inderscience.com/offer.php?id=46419". This is due to high variation in the appearance of the fruits in field settings, including colour, shape, size, texture and reflectance properties. Furthermore, in the majority of these settings, the fruits are partially abstracted and subject to continually-changing illumination and shadow conditions. To solve this problem DCNN provides good performance in images with low quality, different shapes, different views.

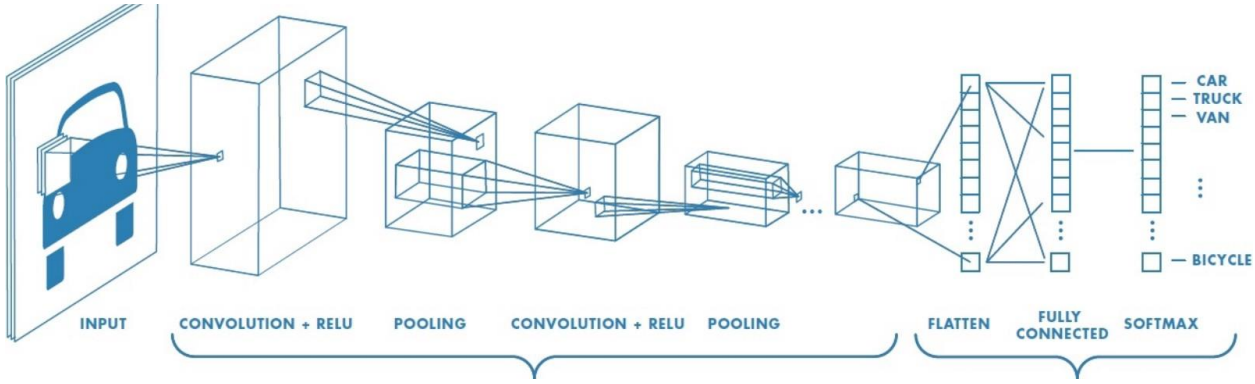
Problem Statement

In this project, we will provide an image to the model and it will predict the fruit present. The project will be using Conventional Neural Network. And it can be directly linked to an webpage and used to predict the fruits present. The project is referenced from Kaggle, it has 81 types of fruit images.

Domain Background

This project is mainly dependent on 2 domains:

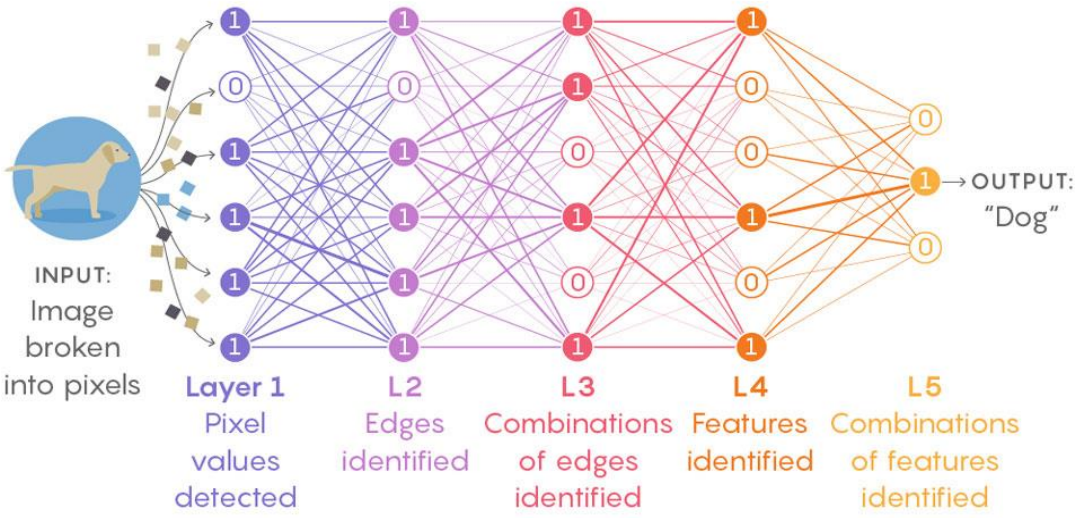
1) Convolutional Neural Network



2) Deep Learning

Learning From Experience

Deep neural networks learn by adjusting the strengths of their connections to better convey input signals through multiple layers to neurons associated with the right general concepts.



When data is fed into a network, each artificial neuron that fires (labeled "1") transmits signals to certain neurons in the next layer, which are likely to fire if multiple signals are received. The process filters out noise and retains only the most relevant features.

Dataset and Inputs

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.

1)

Label	Number of training images	Number of test images
Apple Braeburn	492	164
Apple Golden 1	492	164
Apple Golden 2	492	164
Apple Golden 3	481	161
Apple Granny Smith	492	164
Apple Red 1	492	164
Apple Red 2	492	164
Apple Red 3	429	144
Apple Red Delicious	490	166
Apple Red Yellow	492	164
Apricot	492	164
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

2)

Label	Number of training images	Number of test images
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
Grapefruit White	492	164
Guava	490	166
Huckleberry	490	166
Kaki	490	166
Kiwi	466	156
Kumquats	490	166
Lemon	492	164
Lemon Meyer	490	166
Limes	490	166
Lychee	490	166
Mandarine	490	166
Mango	490	166
Maracuja	490	166
Melon Piel de Sapo	738	246
Mulberry	492	164
Nectarine	492	164
Orange	479	160
Papaya	492	164
Passion Fruit	490	166
Peach	492	164
Peach Flat	492	164
Pear	492	164
Pear Abate	490	166

3)

Label	Number of training images	Number of test images
Pear Monster	490	166
Pear Williams	490	166
Pepino	490	166
Physalis	492	164
Physalis with Husk	492	164
Pineapple	490	166
Pineapple Mini	493	163
Pitahaya Red	490	166
Plum	447	151
Pomegranate	492	164
Quince	490	166
Rambutan	492	164
Raspberry	490	166
Salak	490	162
Strawberry	492	164
Strawberry Wedge	738	246
Tamarillo	490	166
Tangelo	490	166
Tomato 1	738	246
Tomato 2	672	225
Tomato 3	738	246
Tomato 4	479	160
Tomato Cherry Red	492	164
Tomato Maroon	367	127
Walnut	735	249

Solution Statement

- Deep Learning will be used for training the model.
- We will be designing Convolutional Neural Networks and connect it to deep neural network.
- Various filters and pooling layers for predicting the image more properly.
- Then we will use pre-existing model trained for large datasets and then check the accuracy.
- We will use image augmentation to increase the accuracy.
- CNN can detect edges, shapes and particular colors and backgrounds.

Evaluation Matrix

Existing models and study available in the domain to compare the accuracy.

- Previously available models to enhance the accuracy.
- Results available in the paper.

Table 3: Results of training the neural network on the fruits-360 dataset.

Scenario	Accuracy on training set	Accuracy on test set
Grayscale	99.53%	91.91%
RGB	99.51%	95.59%
HSV	99.32%	95.22%
HSV + Grayscale	98.72%	94.17%
HSV + Grayscale + hue/saturation change + flips	99.46%	96.41%

- Comparing other implementation on Kaggle.

Workflow:

1) Loading Dataset:

   To load dataset we populate a few variables through the use of the load_files function from the scikit-learn library:

   - train_files, valid_files, test_files - numpy arrays containing file paths to images
   - train_targets, valid_targets, test_targets - numpy arrays containing onehot-encoded classification labels
   - fruit_names - list of string-valued dog breed names for translating labels.

2) We will be using tensorflow as the backend, so we need to convert the image to 4D tensor which is supplied to keras CNN.

3) Third step is to design the architecture of the model using keras layers,to reduce overfitting we will be also using droupout layers from keras.

4) Now after preprocessing data, designing the architecture now we will train our model and find the accuracy for the same.

5) Now to reduce the training time we would also use transfer learning.

6) After the training we will check our model with various inputs and we will check for the losses and errors.

References

- Link to dataset Kaggle- https://www.kaggle.com/moltean/fruits
- Convolutional neural network- https://en.wikipedia.org/wiki/Convolutional_neural_network
- Fruit recognition from images using deep learning- https://www.researchgate.net/profile/Mihai_Oltean2/publication/321475443_Fruit_recognition_from_images_using_deep_learning/links/5bc8425e458515f7d9c4f97c/Fruit-recognition-from-images-using-deep-learning.pdf