MLND Capstone Proposal

Domain Background:

Computer vision is a science and engineering domain concerned with automating task involving the human visual system. These tasks include identifying objects within images and summarizing scenes in images. Studies conducted in the 1970s provided some of the foundations of computer vision algorithms including edge extraction and line labeling. The 1990s saw statistical learning techniques applied to the field that resulted in improved facial recognition (Eigenfaces). More recently, neural networks have been applied to computer vision tasks. Neural network techniques attempt to model the process by which humans and animals process images, namely, through a series of processing steps that recognize structural features in the early phases before combing those structures into more abstract objects in the later stages (see here for an example). One of the applications of computer vision is recognizing and reading information stored in bar codes. Using this capability, a person can take a picture of a bar code and be presented with information about the product. Fitbit has applied this technology to food tracking. With this technique, users can quickly record what they eat throughout the day. However, the current system does not allow for easily tracking food items without barcodes, such as whole fruits and vegetables. Some groups have used support vector machines to classify fruits (Zhang 2012, Rocha 2010). Rocha 2010 used specific features from the images and paired combinations of the these features with different classification algorithms to achieve 90% classification accuracy. Zhang 2012 used a similar approach in which the color histogram, texture, and shape features were used with principal component analysis and support vector machines to achieve a maximum accuracy of 88%. In recent years, convolutional neural networks have become the gold standard for image recognition and I would like to apply these updated techniques to the problem of fruit and vegetable image classification.

Problem Statement:

The problem is: given an input image of a fruit or vegetable, we classify the type of fruit or vegetable with greater than 90% accuracy? Fruits and vegetables can take on a variety of shapes, colors, and sizes, even for a particular type of fruit. There are also multiple types of specific kinds of fruits, such as fuji apples compared to granny smith apples. For the purpose of this project, I am concerned with classifying a greater number fruit types, without specifying specific subtypes.

Datasets and Inputs:

The datasets to be used in the project include the Supermarket Produce dataset used in the paper by Rocha 2010. The dataset consists of 15 different classes, plums, agata potatos, astexrix potatos, cashews, onions, oranges, taiti limes, kiwis, fuji apples, granny smith apples, watermelons, honeydew melons, nectarines, Williams pears, and diamond peaches. The images were captured on a clear background at a resolution of 1024 x 768 pixels and downsampled to 640 x 480 pixels. There are 2633 images and 75 to 264 images per classThe dataset can be downloaded at http://www.ic.unicamp.br/~rocha/pub/downloads/tropical-fruits-DB-1024x768.tar.gz. Convolutional neural networks are often trained 10s of thousands of images, so I will also include images from the Fruit Image Data set (Skrjanec 2013, http://www.vicos.si/Downloads/FIDS30). The Fruit Image Data set images are classified into 30 different classes, with approximately 32 images per class. The classes include aerolas, apples, apricots, avocados, bananas, blackberries, blueberries, cantaloupes, cherries, coconuts, figs, grapefruits, grapes, guava, kiwis, lemons, limes, mangos, olives, oranges, passionfruit,

peaches, pears, pineapples, plums, pomegranates, raspberries, strawberries, tomatoes, and watermelons.

Solution Statement:

To solve the problem, I will implement a convolutional neural network to classify fruit and vegetable images. Convolutional neural networks are the current best solution for image classification problems such as this one. Accuracy on training, validation, and test sets will be used to determine the acceptability of the solution.

Benchmark Model:

As there is not an even distribution of images for each class, the benchmark model will predict the most numerous class. A second benchmark model will utilize support vector machines which were used for this task in the papers above. In my implementation the SVMs will use the pixel values as features, instead of the features described in the above papers.

Evaluation Metrics:

The evaluation metrics to be used are the loss to quantify training and accuracy in the training and validation sets to prevent overfitting, and accuracy on the test set. The loss measures the progress of the gradient descent algorithm during training. Keeping track of the loss can guard against overfitting on the training data. The accuracy will be calculated on the training, validation, and test sets by comparing the predictions from the convolutional neural network or support vector machine to the ground truth label of the input. Accuracy = # correct predictions / total # predictions

Project Design:

The workflow for the project will be

1. Image preprocessing and transformation
    a. Images will be downsampled to 32x32 pixel images
    b. Images from each class will be transformed
        i. Flipped on horizontal axis
        ii. Flipped on vertical axix
        iii. Rotated 90 degrees left
        iv. Rotated 90 degrees right
    c. There are now 5x as many images for training and testing (2633*5+(30*32*5)=17965 images.
    d. Pixel value normalization
2. Randomly split the images into training, validation, and test sets
3. Split the training set into batches
4. Design initial convolutional neural network architecture
5. Train and validate
    a. Repeat and modify CNN architecture to attempting to get above 90% accuracy.
6. Test
7. Report final results