Fruit Classification using Convolutional Neural Networks

Liam Eynan, liam.eynan@richmond.edu
Stephen Owen, stephen.owen1@richmond.edu

1 Abstract

Deep learning has become one of the dominant fields of computer science, and convolutional neural networks have become a favorite tool of those practicing deep learning. For this project, we created a model (that we have decided to name F.I.C.T.I.O.N. which stands for Fruit Image Classification That Is Ostensibly Nutritious)[1], that as the name would imply, able to take in pictures of various fruits (and vegetables!) and classify them using a convolutional neural network. Using the Fruit-360 data set as our training data and Keras as our back end, our model is able to achieve a 98.84% testing-accuracy. We have also implemented a system for uploading user images, modifying them to achieve a similar format to the testing data, and classify them on a one-off basis.

2 Introduction

You can rarely go anywhere in the computer science space without seeing something relating to machine learning, be it in the academic or, more recently, the industrial front (as evident by the astounding number of "machine learning start-ups", and the prevalence of buzz words like "deep learning," "neural network," "Artificial Intelligence," and thinks of the like). This is, however not without good reason. Machine learning has completely changed the way we approach certain computational tasks and has opened up countless advances in technology that were previously thought impossible.

At its core, machine learning is advanced applications of procedures from statistics, and many algorithms in the machine learning space are almost directly taken from statistics. Take regression, for example. Regression is a statistical method of parameter estimation in which you make an attempt to fit some function to a data set (the function used is depending on the generalized shape of the data, common ones are linear, logistic, and quadratic) and use this resulting function to make predictions about other potential data points that may lie outside of the observed data. As machine learning grew as a field, there was a widespread obsession over these new processes called "artificial neural networks."

Using these artificial neural networks involved using other algorithms to extract the "features" we look for in a data set, and then feeding the information extracted from that into a "network" of nodes that would apply various linear-algebra functions to the input and eventually give you an output. This output is compared to the output of the training data, and then the network would go back and correct itself if it preformed sub-optimally (more on this later). As people began to play with these more and recognize their computational power, it was realized that not only were neural networks better at the actual computations, but were also better at doing the feature extraction step as well. This process of using neural networks for both learning and feature extraction brought with it the widespread fascination and desire to use what would be named deep learning wherever possible.

As people studied deep learning more, people found certain network structures excelled at certain tasks. Convolutional neural networks became one of the go-to applications of deep learning because of it's ability to handle visual/image-based

[1] This name is a nod at old Cartoon Network show *Kids Next Door*, where the names of inventions made on the show always had over the top names that formed an acronym.

data in a highly intuitive and efficient way.

3 Related Work

4 Formulation

This project is formulated as a classification problem, so we use deep learning to accomplish this. We opt to use deep learning because deep neural networks automate the extraction of image features, which simplify design and accelerate model construction. More specifically, we train and use a multi-layered convolutional neural network because of their dominance and universal acceptance as the gold standard for handling visual data.
As mentioned, using a deep-learning based approach simplifies the design, because it allows the network to automatically extract features from the data set on its own without any direction from us.

Figure 1: Standardized Strawberry 1

Figure 2: Standardized Strawberry 2

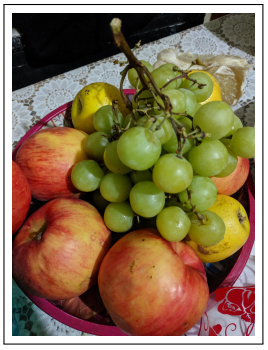

Figure 3: An example of a "non-standardized" photo

5 Approach

5.1 Data Set

We use the Fruit-360 data set for our model.[2] This data set provides 120 different types of labeled fruits (and vegetables) with a white background (which ended up being a hindrance, and will be discussed in the conclusions) in a standardized 100 × 100 pixel format. This data set contains a total of 82213 images showing off the fruits at various different angles. We used 60498 of these images as our training set and 20622 as our testing set (the rest of the images were a random assortment of images that were not in the standardized format, such as pictures of multiple types of fruit together).

The Above photos highlight the benefit of using a data set like this. For each fruit or vegetable, there are at least 150 images showing the fruit in multiple different angles, which aids in preventing model over-fitting, because it learns to recognize both of these images as "Strawberries" rather than only being able to categorize something that looks like, say, figure 2, for example.

[2]Available for free at https://www.kaggle.com/moltean/fruits courtesy of Mihai Oltean

5.2 Language & Tools

5.2.1 Python

We did all of our programming in Python 3.7 and interfaced with the language through the use of Jupyter notebooks. We chose this because Python is the standard programming language for doing machine learning of any variety. Jupyter notebooks were used because they allow programming in "blocks" where certain segments of code can be executed in any order, which makes prototyping, editing, and parameter-tweaking much more efficient than standard functional programming.

5.2.2 Numpy

Numpy is a python library that has become the gold standard for number manipulation and array based applications. They provide a wealth of sub-libraries and functions for almost any array-based application and Numpy has become the backbone of many commonly used modern python libraries and projects.

5.3 Keras

Keras is one of the most popular frameworks for doing machine learning tasks in python.

6 Experiments & Analysis

7 Conclusions

8 Appendix