

Machine Learning Engineer Nanodegree

Capstone Proposal

Augustas Volbekas

May 18, 2017

Proposal

Image recognition for self-service checkout stations in grocery stores.

Domain Background

In self-service checkout systems, a person has to scan everything by finding a bar code or selecting products from a list (if a product does not have a bar code).

The search of a specific product is often a hassle for an untrained person. It would be much more efficient if suggestions could be displayed for selection.

It could be done using machine learning.

Problem Statement

Image recognition of a range of common grocery items with a constant background. Supervised learning seems as an obvious choice.

As inputs (images) could be taken just after something has been placed on scales of the self-service checkout machine.

Labels would be available when the shopper chooses the right product on the screen.

A convolutional neural net could be a solution to this problem.

Data-sets and Inputs

As described above inputs and labels could be obtained in real usage of the machine, however as this project is more of a "proof of concept".

So data is pictures of groceries that I have taken in a white background, and camera placed above. The pictures that I would be using are common groceries in 6 categories each 500 pictures, total of 3000.

The categories that I have is pumpkin, potatoes, cucumbers, lemon, pumpkin_seeds, onions.

The photos were taken by constantly changing the position of the products, varying the number of items (for example photographing few potatoes in a bag and also photographing one potato in a bag) and photographing in a plastic bag or without a bag.

Some categories have a quite similar colors for example potatoes, onions especially when in a plastic bag.

Data could be seen with a link from Dropbox (in README.md), also a few samples are in a "data_samples" directory.

Solution Statement

The solution is a multilayer convolutional neural net done in Python using Keras library.

Benchmark Model

A simple benchmark model would be random guessing: $\text{accuracy} = (1 / \text{number_of_categories})$.

Evaluation Metrics

Evaluation would be a simple accuracy: $\text{accuracy} = \text{number_of_accurate_classifications} / \text{number_of_classifications}$

Project Design

Split data into training and testing.

Make a data pipeline(read, rescale, normalize) for images to be fed to a model for training, mainly using Scikit-image, numpy and pandas.

Make a convolutional neural net using Keras.

Make a testing script to test the accuracy.

Make a classifier script, and visualizations.