Project Report: Analyzing Amazon Inventory Stock in Image Bins Using AWS Sagemaker

By: Derrick Moseti
Date: 16 January 2022

Project Overview

Background Information

According to city clean simple website: *the earliest form of inventory management* ***dates back over 50,000 years*** *in which people used “tally sticks” to count. ... Over time, inventory management developed into slightly more accurate systems of accounting and record keeping, particularly in ancient Greek and Egyptian societies.*

Humans have also evolved through many stages in terms of inventory management. Below are some of the stages according to https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.barcodedirect.com/2015/09/05/the-evolution-of-inventory-management/&ved=2ahUKEwip6cK4maL1AhVCEWMBHYwdAskQFnoECCYQAQ&usg=AOvVaw2bwwGsYWiRvToXLbGZRMTW.

- Industrial Revolution Era Tracking.
- 1880s – Enter the Machine Readable Punch Card.
- 1940s – The Barcode is Born!
- 1980s – Software Improves Tracking.
- 2000s – The Introduction of RFID in Barcode Technology.

Amazon is a giant ecommerce business. Its activities majorly involve selling and shipping items online with clients. The items sold often are put into cartons or parcels before they are shipped to designated clients’ locations. The items shipped are in thousands if not millions, hence counting and keeping inventory records for these huge chunks of products is greatly cumbersome and highly time consuming if only done by humans. This creates a huge problem that needs to be resolved.

Problem Statement

The huge inventory record management creates a huge challenge that is not easily solvable by ordinary humans. There is a need to create an automated tool that is able to accomplish this essentially required task with no error, no fatigue and with accelerated efficiency.
Such solutions required domain knowledge expertise in the field of Artificial Intelligence to help innovate streamlined tools for such.

This particular problem will need the application of deep learning, a subset of AI.This process will be used in **AWS Sagemaker** to train the model and also **AWS S3** for data storage.

According to the Stanford researchers' paper, ResNet consists of a series of convolutional filters followed by a fully connected layer. Deeper networks usually suffer with vanishing gradient issues and ResNet attempts to solve this problem by stacking identity mappings.

Pretrained ResNet18, ResNet34 and ResNet50 Convolutional Neural Network architectures have been used in the experiments in the research paper.However in this project we will use ResNet34 which generally shows better model performance in the data.

Solution Statement

I used a convolutional neural network to train on the bin images and count the number of objects on it. This was achieved using a pretrained ResNet model. The model uses a CNN network to train and test on data.

Training using a ***resnet34*** pretrained model was successfully , with a testing accuracy of **31.0%**

Data

The dataset used in this project is that of Amazon Image Data. According to the data source's official description*The Amazon Bin Image Dataset contains over 500,000 images and metadata from bins of a pod in an operating Amazon Fulfillment Center. The bin images in this dataset are captured as robot units carry pods as part of normal Amazon Fulfillment Center operations.*

The dataset contains bin images of items that are sold on the amazon ecommerce platform. The dataset has them grouped into folder with names denoted by numbers from 1 up to 5 that implies the number of items contained in each bin image.

The dataset can be found in [Amazon Imagery dataset.](https://registry.opendata.aws/amazon-bin-imagery/)

Link to dataset: https://registry.opendata.aws/amazon-bin-imagery/

Metrics

Evaluation metric for this project is to calculate the training loss of our preferred model.To measure this we will have to calculate the *accuracy of our model.* Our calculated model accuracy is **0.31.**

.

Data Preprocessing

The data used in the project is downloaded from Amazon Imagery bucket.
It is then split into train, test and validation chunks read for training, testing and validation tasks respectively.However this does not make the data totally read for model fitting. The data needs again to be modified into appropriate transformations using the t*orch* python module

transforms.RandomResizedCrop((224, 224)),

transforms.RandomHorizontalFlip(),

transforms.ToTensor()

The above activities are aimed to prepare the data into a suitable shape and size for model fitting.

Algorithms and Techniques

The model algorithm utilized in this project is the resnet34. It applies the use of a convolutional neural network to train and test out data. It also uses the cross entropy criterion to measure the model loss.
The choice of this model is due to its exemplary results on the same data in the benchmark model of a research project done by the students at Stanford University.
Link to the research; https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3311007

Implementation

After the data is in the right form, the next step is to implement and fit our model. This is achieved by creating an estimator for our model. The estimator is configured with functions that are responsible for data loading,training, testing and evaluation.

We call the ***.fit()*** method on the estimator with the training path name as a parameter to start model training.

Refinement

Inorder to improve the model performance, we change the parameters of our estimator in an aim to improve its accuracy. The parameters used in this project are *learning rate and batch size .* In our training we select the best learning rate and pass it to the next estimator training inorder to improve model performance.

Model Evaluation and Validation

Our project uses the resnet34 CNN network to train and test on data. After successfully running our model we get a testing accuracy of **31%**.

Benchmark Result

The benchmark results obtained by similar project *i.e* *Image-Inventory-Reconciliation-with-SVM-and-CNN* a project done by researchers at Stanford University show the performance to reach up to **60 %** testing accuracy. Our project got results 31% testing accuracy that are much lower than that. However this can be due to the fact that we used just a segment of the dataset.

Justification

When comparing the benchmark model and the training model. The benchmark model is the winner, it achieved an overall accuracy of over **56%** while this one achieved a **31%** percent model accuracy.
However in our model we just used a small part of the data. About 10,000 out of over 500000 images in the whole dataset.