Capstone Project
Machine Learning Engineer Nanodegree

Carlos Alberto Villarreal Campos

May 28th, 2023

1. DEFINITION

1.1 Project Overview

Knowing exactly the count of a physical inventory has been a key issue in all industries that are involved in manufacturing and retail processes. Historically, this is a job that warehouse managers and supply chain professionals have performed through *cycle counting*, an audit process for managing inventory counts. Cycle counting always requires strategic planning to achieve high accuracy in stock recording without distracting staff from their essential tasks [1].

While this type of procedure might be valid for some particular industries, especially where the primary goal is to attain a balance between physical inventory counts and the inventory records (which are typically maintained on ERP platforms) there are other industries where time planning and staff time is prohibitive, especially in systems where a high level of automation has been implemented (i.e., distribution centers that use robots in their operations)

In these high-tech environments, a solution is required that can directly interact with robots, allowing for real-time counting of objects. Even such a solution could also be of great help in less technical environments with a proper interface that updates inventory records in ERP platforms. The benefits of an accurate inventory can be seen in [2]

1.2 Problem Statement

In this project, a solution for object counting is presented. The solution uses the Pytorch framework available from AWS SageMaker. The goal is to deploy a SageMaker endpoint that will query images to get predictions about the number of objects in the submitted image.

1.3 Metrics

The performance of the model obtained will be measured through the following metric:

$$\text{Accuracy: } \frac{1}{N}\sum_{i=1}^{N} 1[p_i == g_i]$$

Where, 1 is an indicator function (returns 1 when the event occurs and 0 otherwise), and *p* and *g* is prediction and ground truth respectively. [3]

2. ANALYSIS

2.1 Datasets

For this project, the Amazon Container Image dataset is used, which contains over 500,000 container images and metadata from a pod in a working Amazon fulfillment center. Container images in this dataset are captured when robot units transport pods as part of normal Amazon Fulfillment Center operations [4]. Since this is a large dataset, in order not to exceed the budget available for the project, a subset of the dataset will be used.

2.2 Benchmark

The results obtained in this project will be compared with the accuracy obtained in the Amazon Bin Image Dataset (ABID) Challenge [3] where a 34-layer Resnet architecture trained from scratch was used. In this project the model will be implemented from a pretrained Resnet50 model.

3. METHODOLOGY

3.1 Data Preprocessing

As mentioned earlier, this project will use a subset of the Amazon Bin image dataset. Some exploratory data analysis will be performed to understand the object categories, instances, and metadata available in the dataset.

The data will be uploaded to a S3 bucket, so that SageMaker can use it in the training process. In addition, training and split tests are required.

3.2 Implementation

Once data is proper processed and available in the S3 bucket, it is necessary to create a Training Script (a python file) and a Submission Script (a jupyter notebook).

In the Training Script, the following tasks will be executed:

- To read hyperparameters (defined in the Submission Script) and additional information required, that is passed as environment variables.

- To transform and load the data available in the S3 bucket in a format that can be understood by the train and test functions implemented.

- To create and configure the model that will be trained. In this case a ResNet50 model
- To define the optimizer used during the training process.
- To perform the training process according to the parameters defined in the Submission Script.

On the other hand, in the Submission Script the following tasks will be performed:

- To specify the hyperparameters for training and fine-tuning.
- To create and configure the estimator that uses the training script, defined earlier, as entry point.
- To define the infrastructure required for the training and endpoint deploying processes.
- To submit the training and deploying jobs.

REFERENCES

[1] Inventory Cycle Counting 101: Best Practices & Benefits. https://www.netsuite.com/portal/resource/articles/inventory-management/using-inventory-control-software-for-cycle-counting.shtml

[2] Physical Inventory. https://www.wallstreetmojo.com/physical-inventory-count

[3] https://github.com/silverbottlep/abid_challenge

[4] https://registry.opendata.aws/amazon-bin-imagery/