Capstone Project Proposal for
Inventory Monitoring at Distribution Centers

Domain Background

Artificial intelligence in inventory management is yielding hefty and impressive improvements for the companies that are utilizing it likes Amazon and others. We are on the verge of a major upheaval in the way inventory is managed. This revolution is a result of the availability of the huge amounts of real-time data that are now routinely generated on the internet and through the interconnected world of enterprise software systems and smart products. In order to make effective use of this new data and to stay competitive, managers will need to redesign their supply-chain processes. Amazon, for example, implemented artificial intelligence throughout their inventory operations, at an unprecedented scale. In almost every aspect of their operations, A.I methodologies such as time series prediction and reinforcement learning systems are being deployed. User demand, supplier backorders, warehouse optimization, stock levels are all being guided by either machine learning or more complex artificial intelligence systems. [1]

Problem Statement

As we mentioned above about how inventory management is critical to Amazon's success. Thus, in distribution centers Amazon often uses robots to move objects as a part of their operations, In this project, we will have to build a model that can count the number of objects in **(Amazon Bin Images Dataset)** [2]. A system like this can be used to track inventory and make sure that delivery consignments have the correct number of items.

Dataset

To complete this project we will be using the Amazon Bin Image Dataset.
The dataset contains 500,000 images of bins containing one or more objects. For each image there is a metadata file containing information about the image like the number of objects, it's dimension and the type of object. For this task, we will try to classify the number of objects in each bin.

An example of an image and the corresponding metadata file is shown as below: (Source [3])

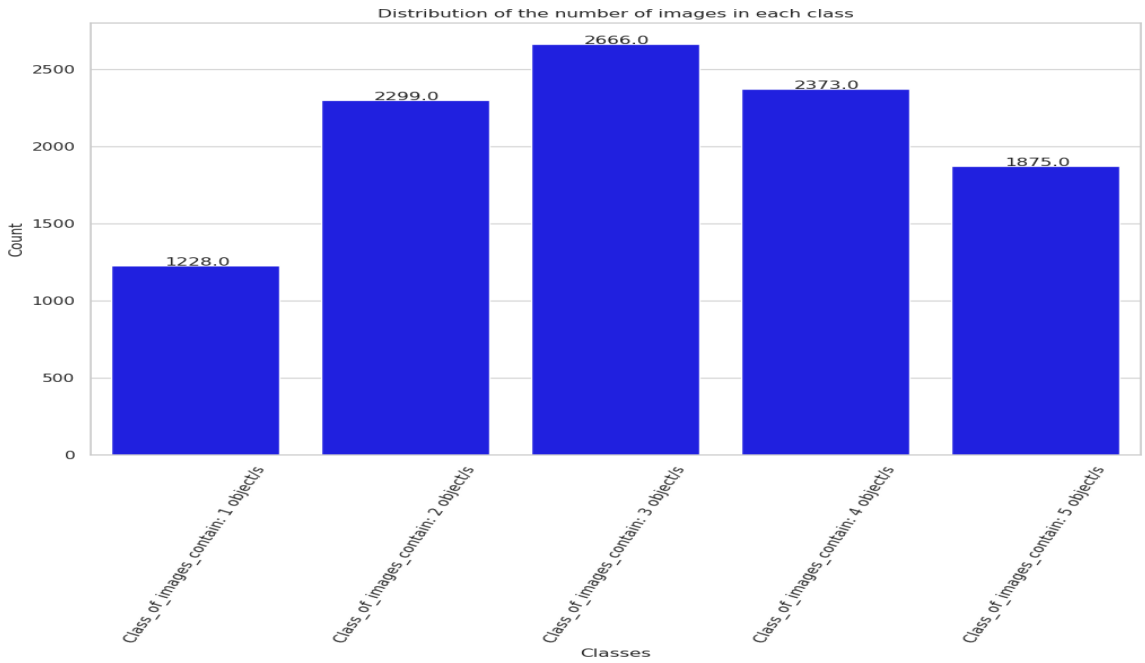

```
{
    "BIN_FCSKU_DATA": {
        "B000A8C5QE": {
            "asin": "B000A8C5QE",
            "height": {
                "unit": "IN",
                "value": 4.200000000000001
            },
            "length": {
                "unit": "IN",
                "value": 4.7
            },
            "name": "MSR PocketRocket Stove",
            "quantity": 1,
            "weight": {
                "unit": "pounds",
                "value": 0.45
            },
            "width": {
                "unit": "IN",
                "value": 4.4
            }
        },
        "B0064LIWVS": {
            "asin": "B0064LIWVS",
            "height": {
                "unit": "IN",
                "value": 1.2
            },
            "length": {
                "unit": "IN",
                "value": 5.799999999999999
            },
            "name": "Applied Nutrition Liquid Collagen Skin Revitalization,
10 Count 3.35 Fl Ounce",
            "quantity": 1,
            "weight": {
                "unit": "pounds",
                "value": 0.3499999999999999
            },
            "width": {
                "unit": "IN",
                "value": 4.7
            }
        }
    },
    "EXPECTED_QUANTITY": 2,
    "image_fname": "523.jpg"
}
```

The “EXPECTED_QUANTITY” field tells us the total number of objects in image.
However, since this dataset is too large to use as a learning project, and due to cost limitations on the Udacity AWS Portal, we will be using a subset of the data provided to us by Udacity itself. ([2])

This subset of data has 5 classes, corresponding to number of objects prese

in each bin: 1, 2, 3, 4 & 5. The total number of images in this subset are: 10,441.

The graph below shows the distribution of classes in the subset which is imbalanced:



Solution Statement

The proposed solution involves applying deep learning techniques that have proved to be successful for computer vision. We can use convolutional neural network with a pre-trained model (ResNet Model), and leverage this pre-trained model with transfer learning to solve the problem.

As it is a multi-classification problem; we will adopt the cross-entropy loss function to maximize prediction accuracy.

Besides accuracy, we will evaluate the performance of the model with (Precision, Recall, F1 Score) to identify if a model is doing better on a particular class, or has a high bias for one of them.

Precision: **True positives (correctly predicted class) / Total predicted positives**

Recall: True predicted positives / Total actual positives

F1 Score (Harmonic Mean): 2* precision / (Precision + Recall)

F1 score: is a way to computing an average of sorts that pays more attention to whichever is lower. Also it is useful for unbalanced classes, as we have here with Amazon Bin Images.

Benchmark

The benchmark model will be a pre-trained convolutional neural network, as pre-trained models (usually trained on the Image Net dataset) are now publicly available for download and can be used to bootstrap powerful vision models out every little data. Also, we can't train a model from scratch because the training process will be intense, and due to the constraints of resources available from the Udacity AWS Gateway; we will apply a pre-trained CNN model on a subset of (Amazon Bin Images Dataset).

Evaluation Matrix

Since we have a Classification Task, abd we aim to identify if a model is doing better on a particular class**;** we can use Accuracy, Recall, Precision and F1 scores as our metrics.

Project Design

Step 1 Preparing The Data:

* Download a small subset of the data

* Preprocess and clean the file (if needed)

* Upload the data to an S3 Bucket so that SageMaker can use them for training

Step 2 Data Exploration:

* We can visualize some images class wise to see the quality for each class and if that makes sense.

Step 3 Create Training Script

***** Read and preprocess data before training: we will need to read, load and preprocess our training, testing, and validation data.

* Train the data with the chosen architecture.

Step 4 Deploying the model:

* Deploying endpoint for inference to test our model with unseen images.

References

[1] [Online]. Available: https://medium.com/@RemiStudios/artificial-intelligence-for-inventory-management-c8a9c0c2a694

[2] [Online]. Available: https://registry.opendata.aws/amazon-bin-imagery/

[3] [Online]. Available: https://github.com/awslabs/open-data-docs/tree/main/docs/aft-vbi-pds