Capstone Project Proposal for Inventory Monitoring at Distribution Centres

1. The Domain and Background of the Project

Distribution centers often use robots to move objects as a part of their operations. Objects are carried in bins which can contain multiple objects. In this project, you will have to build a model that can count the number of objects in each bin. A system like this can be used to track inventory and make sure that delivery consignments have the correct number of items.This is where robots come in to help in Inventory Monitoring. They can be trained with Machine Learning Models, to perform tasks like Object Detection, Outlier & Anomaly Detection and much more. Once trained, these models are scalable, and can be deployed at a low cost for usage in actual warehouses and distribution centres on industry level robots.

To build this project i will use AWS SageMaker and good machine learning engineering practices to fetch data from a database, preprocess it, and then train a machine learning model. This project will serve as a demonstration of end-to-end machine learning engineering skills that you have learned as a part of this nanodegree.

2. The Problem Statement

Distribution centres often have robots which carry objects. These objects are present in bins, each bin contain 1-5 objects.

The problem in this project is to count the number of items in the bin. We will create a model, which can take in a picture of a bin, and accurately return the number of objects present in that, we could solve & thus fully automate one crucial step in the Inventory Management process!

3. Dataset and Inputs

The dataset used in this problem is the Amazon Bin Image Dataset. This dataset have 500,000 images of bins and the data have one or more objects present in it. One of each image contains information about the image, like the number of objects it has, the dimensions and type of objects. For our problem statement, we only need the total count of objects in the image

```
n [11]:
#TODO: Perform any data cleaning or data preprocessing
from os import listdir
import os.path

count = 0
dataset_folder = 'train_data'

for file in os.listdir(dataset_folder):
    path = os.path.join(dataset_folder, file)
    number = len(os.listdir(path))
    print(f'Number of images with the label of "{file}": {number}')
    count += number
print(f'\nNumber of images in total: {count}')
total_images = count
```

```
Number of images with the label of "4": 2373
Number of images with the label of "2": 2299
Number of images with the label of "1": 1228
Number of images with the label of "5": 1875
Number of images with the label of "3": 2666
```

We will be further using a split from this for our final testing. If required, we can also get separate data from the AWS Bin Image Dataset, which isn't in our subset, for testing, leaving us with more data for training

4. **Solution Statement**

To solve our problem, we will will create model machine leanring to detect object,. Essentially, we would be using Multi-Class Image Classification, with Number of Objects from 1-5 asan individual class. To do this, we can make use of Resnet18, We will use Pretrained models with Transfer Learning to solve the problem

5. **Evaluation Metrics**

To evaluate our model, we need some good metrics, which align with the problem statement. The metrics must be mathematically sound and we should be able to optimise our model for them.
We have a Classification problem, we can use Accuracy, Recall, Precision and F1 scores as our metrics. These can be for the overall data, and also class-wise, to identify if a model is doing better on a particular class, or has a high bias for

one of them.
We have Multi-Class classification, the definition of Precision & Recall must be clearly understood. So we assume that we have 3 classes: A, B & C. With Multi-Class classification, we have precision and recall for each individual class. For example, metrics for Class A will be defined as follows:
Precision:
#Correctly Predicted Class A Instances / #All Instances predicted as Class A
Recall:
#Correctly Predicted Class A Instances / #Total Class A Instances
F1 Score: 2*Precision*Recall / (Precision + Recall)
So to evaluate our model, we will see per class metrics of Precision, Recall and F1 Score, and see the overall accuracy of the model, which includes all classes, and is defined as:
Accuracy: #Total Correctly Predicted Instances / #Total Instances

6. **Project Design**

The aim of this project is to use SageMaker and good Machine Learning Engineering practices to solve our problem. We define some outline steps, to help accomplish this. **Note, this is not a strict outline and is there to guide the project. The goals could be flexible depending on the outcomes and scope with the resources available.**
- Load the Corresponding Images from the JSON File and Upload to S3 Bucket:
In this step we will aim to define the training, validation and test splits for the subset of data we have. We will upload data in a S3 Bucket
- Data Visualisation and Data Exploration:
We can verify that the ratio of train-validation-test split makes sense. We can be visuallize some image to show information about it too in this part
- Choose the Transfer learning model to classification image for our Dataset:
We need to decide which Transfer Learning model – Resnet18 - is good for us
Beside that we will decide which Loss Function and Optimiser we need to use
- Train the Model with Sage Maker
We need to prepare our code to work with GPUs for faster training. We need tofinalise the training and testing code. We can decide if we want to try for Hyper Parameter Tuning or try parameters from intuition, for refinement

- Deploy model as a Endpoint service for Predict

We will deploy model we trained in last step to one instance like a predictor service for classification image

- Model Evaluation

After deploy model inference, we can you that service to predict one image and see the result