Project Proposal
Name: Joan Gerard

Domain Background

Distribution centers often use robots to move objects as a part of their operations. Objects are carried in bins which can contain multiple objects. These objects are then shipped to the customers who order them for instance via Amazon.

Inventory Monitoring at Distribution Centers is an important task for some companies in charge of delivering packages. When you order something from Amazon they need to make sure that they are delivering all the product items you ordered within a single package, nothing more and nothing less. A simple way to verify this is making sure that the number of objects in the bin matches the number of objects the customer has ordered.

Problem Statement

The problem is avoiding distribution centers to deliver packages with missing items. An image of a bin containing the ordered items is provided and the number of objects in the bin needs to be determined.

Solution Statement

This project uses the Amazon Bin Image Dataset that contains images of orders together with the amount of products that are in every order. The dataset contains 500K images but only a subset of them are used here to train a model using ConvNets. The objective of this project is to build a classifier that receives an image of an order with the products in it and it tries to guess how many products there are. Also, this project illustrates mainly how to train/evaluate/deploy a model in Sagemaker to be ready to be used for making predictions via an endpoint.

Datasets and Inputs

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 was obtained from here and it contains images of customer's orders previously to be sent to them. Also contains labeled information about the images such as how many products it has, what the products are, their weights, etc.

The dataset is publicly available and it can be accessed either from an S3 bucket or HTTP request.

Benchmark Model

There are many architectures for ConvNets for image classification in general that can be used for this project such as ResNet18, ResNet50, ImageNet, etc. Nevertheless, a benchmark model that can be cited specifically for the Amazon Bin Image object count task was provided by Xu, and Salloum in their work title "*Deep Neural Networks for Object Enumeration*" where they used a ResNet for classification using data augmentation. The best model they found obtained 54.64% of test accuracy which is much better than the accuracy of 32.42% obtained by our best model.

Evaluation Metrics

The metric used for assess the classifier is the accuracy, defined as follows:

*Accuracy = Number of Correct Predictions/Total Number of Predictions * 100%*

Project Design

The project will consider the following steps:

- *Data downloading:* the dataset is downloaded locally and then upload to S3 so it can be used by other instances:
- *Data cleaning:* some of the images presented incorrect annotations of were very blurry so they were deleted from the dataset
- *Data transformation:* the images were resized to 224x224 pixels and rotated randomly before training the model
- *Model Selection:* hyperparameter tuning was performed with 4 jobs and then the best model was selected to be trained.
- *Model Training:* the best model was trained and evaluated
- *Debug and Profiling:* a complete profiler report was obtained with relevant information such as GPU/CPU utilization and debugger information such as training loss was reported.
- *Model Deployment:* it deploys the best model to a Sagemaker endpoint.
- *Model Inference:* once the model is deployed, it can be used to make predictions for different images never seen before.

Conclussion

In an effort to improve accuracy many things were tested out such as resizing the images to 224x224 instead of 112x112, centering them, croping them, normalizing them, trying data augmentation making a rotation of the images, modifying the architecture to include more or less layers, adding a dropout layer to avoid overfitting, adding early stopping to have the chance to train more hyperparam jobs, etc. Unfortunately, none of these techniques made a lot of improvement in the final model.

This project presented a lot of difficulties to obtain a final-ready-to-use model because the accuracy of the model presented is low and it may not be sufficient to move it to a production-ready environment. Further work needs to be done to have a better model such as the use of more images from the dataset, more experimentation with some other pretrained models, some more architecture, the use of more hyperparameters, etc. Nevertheless, the present project has accomplished its purpose which is show how to have a model togheter to the end-to-end process needed using AWS resources and tools.