Project Proposal

Domain Background

Distribution centres and warehouses require careful inventory management to avoid over- or understocking, which can negatively affect customer experience, hinder efficiency and increase costs. Amazon ships around 1.6 million packages per day, and it is therefore most desirable to have a modern, accurate system for tracking inventory. One possible component of such a system could be to monitor the number of items in each package being processed. Images of packages are captured as Amazon warehouse worker robots perform Fulfillment Centre operations and these would form the input data to be analysed for inventory management.

Problem statement

To use AWS services (S3, Sagemaker) to train an ML model on a subset of the Amazon Bin Image Dataset (see *Solution statement* for details on the subset). More specifically, the task is to train a model to count the number of objects in a given bin and obtain accuracy equal to or better than an established benchmark (see *Benchmark Model*).

Solution statement

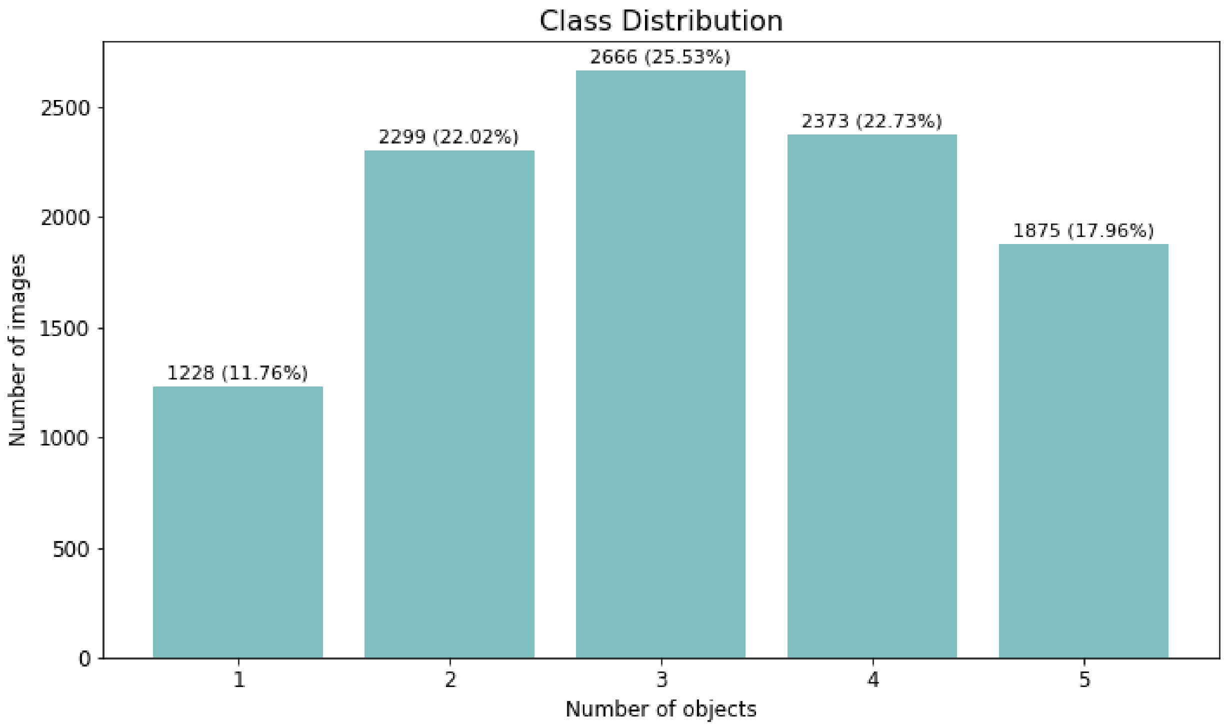
A full solution might work as follows. Once all items have been added to a package, an image is taken that is fed into a trained ML model for inference. The output is then fed back into the inventory database, which is integrated with the rest of the management system. This project is mainly concerned with the training and inference steps, though how the output of the inference may be integrated with external systems may also be discussed.

Our main solution will thus involve data extraction, any preprocessing necessary, ML training, including hyperparameter optimization, and inference. Although in reality we would want to optimise accuracy as much as possible in line with business priorities, in our case the focus will be on building an ML pipeline that utilises best practices and could in principle be deployed to production. That is, we assume that necessary initial work regarding thorough EDA, data quality and prototyping considerations have already been carried out and we are concerned with taking a model, training it on Sagemaker, and preparing and testing it for inference.

As this is a computer vision task, we will utilise a pre-trained CNN, such as ResNet50, and fine-tune on the bin images to obtain predictive model. We note here that we aim to count every object instance in the bin and count individual instances separately, i.e. if there are five objects of the same type in the image, we count this as five.

Datasets and inputs

The entire Amazon Bin Image Dataset contains over 500,000 images. A smaller subset of 10,441 has been provided for the purposes of this project, with class labels and file names given in the file `json_list.json`. A quick analysis of the latter yields the class distributions below.



The classes are imbalanced, with a relative majority of images (~26%) containing three images and a minority (~12%) containing just one image. That said, given that the most prevalent class is only a little over twice as common as the rarest, we assume that this will not affect learning significantly and therefore omit a preprocessing step wherein we balance the classes somehow, e.g. via over- or undersampling. Once the pipeline is built and assessed, we could take a look at how well the model is classifying each class and add this in later to optimise performance if necessary.

As part of the initial data preparation we will split the data into sets according to a 60:20:20 training:validation:testing split. This will leave about 700 images in the training set of the least-represented class, which should be enough to get okay results given that we are using transfer learning.

Benchmark Model

The Amazon Bin Image Dataset (ABID) Challenge was proposed in order to attract researchers in academia and industry to apply themselves to various computer vision tasks based on the dataset. One of the challenges outlined was to count the number of objects in each bin, i.e. the same task as here, and they provided an initial accuracy score (see *Evaluation Metrics* for a definition) of ~55% on a subset of the data using bins with 0–5 objects in them. Although our subset omits bins with no objects in them, i.e. we have five classes compared to their six, we will take 55% as a benchmark to compare model accuracy to. We note here that their accuracy varies somewhat from class to class, so it may be that a better score could be obtained if the class imbalance mentioned earlier was indeed mitigated ahead of training.

Evaluation Metrics

The specific ML task to be carried out here is multi-class classification, and there are various metrics available with which to judge model performance. The metric we will use is accuracy, simply defined as follows.

$$Accuracy = \frac{\#\ correct\ predictions}{\#\ total\ predictions}$$

This is straightforward to interpret: if we make 100 predictions and 80 of these are correct, our accuracy is 0.8. By using this metric we make two assumptions: (1) that class imbalance will not have any effect on learning and (2) that we value all misclassifications equally, i.e. a misclassification of class 1 as class 5 is just as unfavourable as one of class 1 as class 2. In reality, both of these should be checked, the first by checking the per class accuracies and the second by the business context, but we again assume for our purposes that this work has already been done previously and get on with preparing an ML pipeline using this as our evaluation metric.

Project Design

The initial design of the project is as follows.

1. Download the data and organise the images into training, validation and testing sets.
2. Fine-tune a CNN utilising on a pre-trained model.
3. Perform hyperparameter optimisation to obtain the best model.
4. Deploy the model and confirm that predictions can be carried out.