Inventory Monitoring at Distribution Centers

1 Domain Background

The Amazon Fulfilment Centers are not built to store products only but also fulfil million orders from customers every day. In these distribution centers, robots are used to automatically track products while carrying them in bins.

2 Problem Statement

The process of collecting and packaging items by robots can be error prone. If an item is missing from a package, the customer will be dissatisfied. This will probably result in an additional package being sent with the missing item, causing additional costs. If more items are packed than ordered, profit will be lost also. Therefore, checking the number of items in a bin is a good idea which can be solved with a trained model for image classification.

3 Datasets and Inputs

As a part of their normal operation, the robots take pictures of the products that they carry inside their bins. These images belong to the "Amazon Bin Image Dataset" and are store as JPEG files in an S3 container. Each image has a JSON metafile that describes the content of the image, for example the number of elements in the image.[1] It is important to note that items can be misplaced so the contents of the data may not match the recorded inventory.[2] To reduce the cost of model training with AWS SageMaker, only a subset of the over 500,000 images is used.[3] Figure 1 shows some captures of different bins. As is can be seen, it can be difficult to determine the quantity of the items inside a bin since they can be covered by objects and tape.

Figure 1: Example images

[1] https://registry.opendata.aws/amazon-bin-imagery/
[2] https://github.com/awslabs/open-data-docs/tree/main/docs/aft-vbi-pds
[3] https://github.com/udacity/nd009t-capstone-starter/blob/master/starter/file_list.json

Figure 2 shows the distribution of the quantity of items per image. The number of images that contain two or more items is around 2.000 to 2.500. The number of images containing one item is 1250 only.

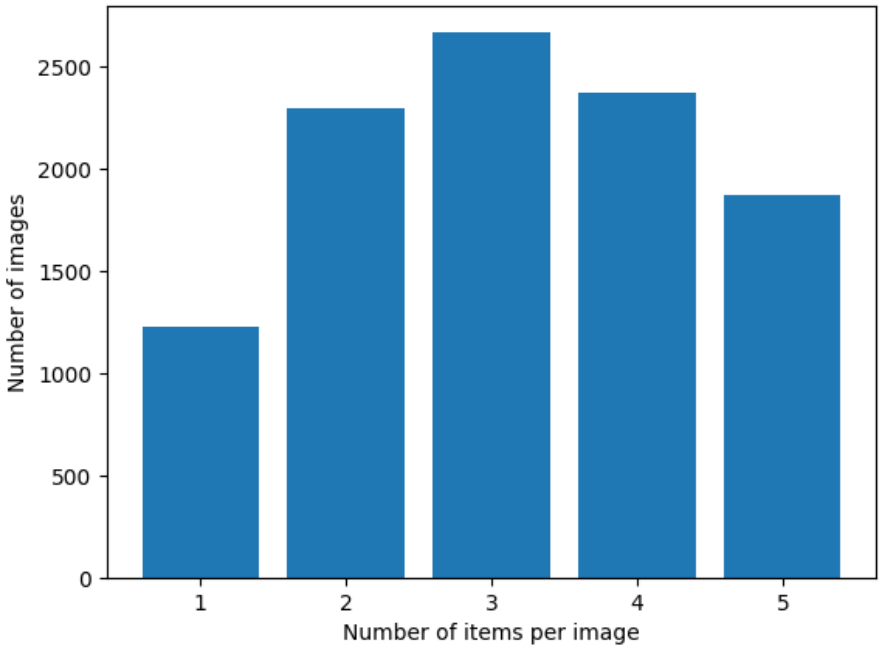

Figure 2: Distribution

4 Solution Statement

Since Convolutional Neural Network (CNN) are well established in the field of image classification problems, a pretrained model from the PyTorch framework will be used. Good candidates among others are:[4]

- ResNet50 (Microsoft, 2015)
- EfficientNet (Google, 2019)

Also, hyperparameter tuning will be done to find optimal hyperparameters for model training, such as learning rate, batch size or number of epochs. Additionally, an end point can be deployed to request predictions.

5 Benchmark Model

As a benchmark the model from the publication "Amazon Inventory Reconciliation Using AI" will be used which is also mentioned by the "Amazon Bin Image Dataset".[5] This study used the ResNet18 and ResNet34 Model. With the latter a train accuracy of 57.8 % and a test accuracy of 53.8 % were reached.

[4] https://towardsdatascience.com/4-pre-trained-cnn-models-to-use-for-computer-vision-with-transfer-learning-885cb1b2dfc

[5] https://github.com/pablo-tech/Image-Inventory-Reconciliation-with-SVM-and-CNN

6 Evaluation Metrics

For evaluation the accuracy on the test data will be calculated. The data will be split roughly into 70 % for training, 20 % for validation and 10 % for testing. This should help to get a better comparison between the trained model and the benchmark model.

7 Project Design

The design of the project can be split into different parts.

7.1 Prepare Dataset

First, the part of the dataset defined in the provided file “file_list.json” will be downloaded. These images will the be split into the three parts according to chapter 6. The images will then be split up further to identify the number of items from one up to five. After that, the images will be uploaded to a prepared S3 bucket.

7.2 Training a Model

As a next step, the data will be transformed to the correct size desired by the pretrained model. Then, the model is trained with a set of hyperparameters like batch size, learning rate or number of epochs. The optimal values can be obtained by hyperparameter tuning. The model. While training metrics for evaluation are captured and plotted afterwards.

7.3 Deployment and Prediction

In the end, the model will be deployed to an end point and example predictions will be made. This can be done with the SageMaker SDK or with a Lambda function which can be triggered by tests or HTTP requests.