Capstone Project Proposal

1. Domain Background

A warehouse management system is a set of policies and processes intended to organize the work of a warehouse or distribution center, and ensure that such a facility can operate efficiently and meet its objectives (Wikipedia 2024a). Today, warehouses are constantly growing in size and complexity and it's important to develop efficient processes that can accurately track the inventory. A major player in the field of warehouse management is Amazon. I'm interested in how a company, such as Amazon, has been able to build global warehouse management systems, and I want to look into how to streamline and improve processes in these systems.

In terms of previous work, a team from Stanford University has evaluated different machine learning models for inventory reconciliation in Amazon warehouses (Bertorello et al. 2019). In these warehouses, occasionally, items are misplaced while being handled, which results in a mismatch between the recorded bin inventory versus its actual content. The Stanford team describes methods to predict the number of items in a bin, thus detecting any inventory variance. The detected mechanism is based on the photo of the bin. The performance of convolutional neural networks has been evaluated vis-a-vis support vector machines (SVMs). In the research, convolutional networks outperformed SVMs by 75% .

2. Problem Statement

Industrial robots are used in warehouses to move items as part of operations. These items are carried in moveable shelf units containing a number bins. Each bin contains one or more items. The aim of this project is to build a machine learning model that can count the number of items in each bin. A system like this can be used for inventory tracking and reconciliation. As we are dealing with counting the number of items in a bin, the problem is quantifiable and measurable.

3. Solution Statement

The warehouse robots are equipped with cameras and take photos of the bins in the moveable shelf units. Amazon has made these images publicly available. The solution is to use a **supervised machine learning algorithm** to classify these images according to the number of items in each bin. In the model, each class corresponds to the number of items detected in the image. A pre-trained convolutional neural network can be used or a customized neural network architecture can be developed. A suitable place to train and evaluate this model is by means of a Jupyter Notebook in Amazon SageMaker.

4. Datasets and Inputs

The project will use the **Amazon Bin Image Dataset**. This dataset is publicly available and contains 500,000 images of bins containing one or more items. The bin images are captured while the robots move shelf units around the warehouse. This forms part of the normal operations in an Amazon Fulfillment Center. Information about the dataset can be found in following location:

https://registry.opendata.aws/amazon-bin-imagery/

Furthermore, each bin image has a corresponding metadata file, containing information about the image, such as, the number of items, its dimension and object type. Below are sample images from the dataset.

Figure 1: Sample images from the Amazon Bin Image Dataset

To reduce the cost of training, a subset of the full dataset will be used. This subset has 10,441 images, which represents about 2% of the total number of images. These images have been divided into 5 classes, where each class represents the number of items present in the image. The distribution of images is as follows: 1228 (1 item), 2299 (2 items), 2666 (3 items), 2373 (4 items) and 1875 (5 items). Below is bar chart of the subset of images used in the project:

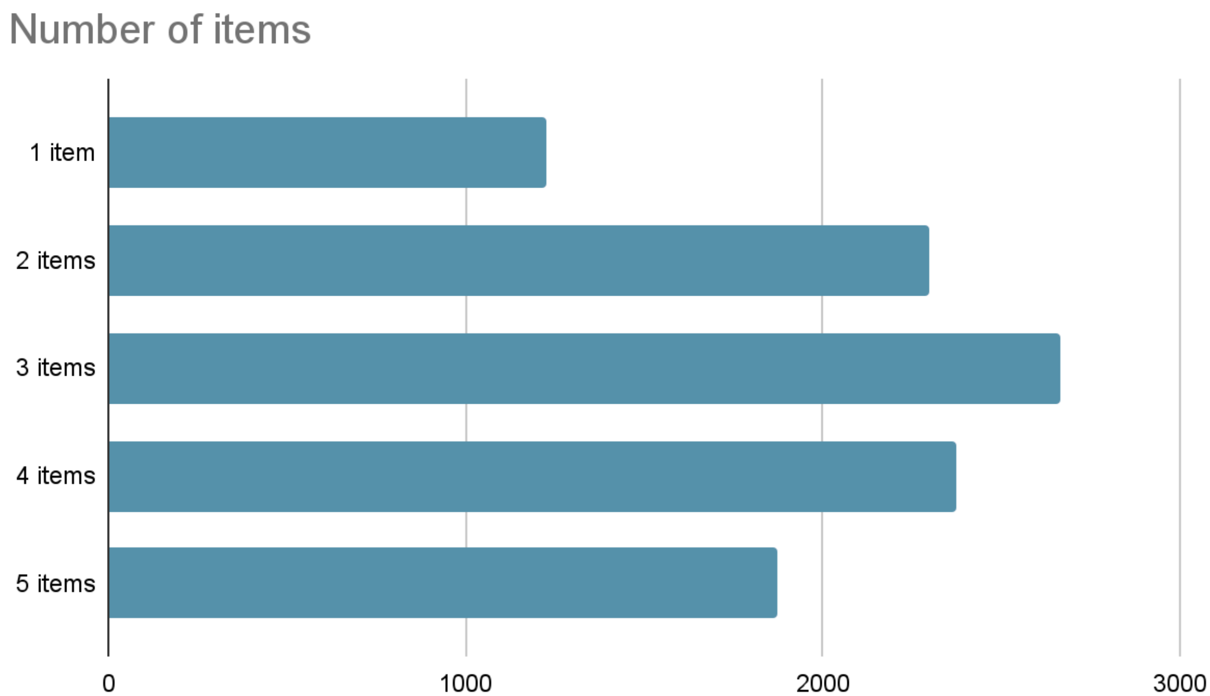

Figure 2: Bar chart showing distribution of images in the subset

The data has a slightly skewed distribution with a mean of about 3 items per image. According to the distribution, the model will be trained more heavily on bins with 3 items, followed by bins with 2 or 4 items and then 1 or 5 items. This can affect the resulting accuracy of the model. Also, the images have imperfections. In some of the images, ít can be difficult to detect the exact number of items with the naked eye, meaning the model might struggle as well.

This subset of data will be further divided into 3 categories: training, validation and testing. The training category contains 90% of images, the validation category contains 5% of images and the testing category contains 5% of images.

5. Benchmark Model

A suitable benchmark model is **ResNet** (Residual Network). It is a deep learning model which contains residual blocks, where weight layers learn residual functions with reference to the input layers. The model was developed in 2015 for image classification. It won that year's **ImageNet** Large Scale Visual Recognition Challenge (Wikipedia 2024d). ImageNet is a large visual database designed for use in visual object recognition software. It serves as a dataset benchmark and contains more than 14 million hand-annotated images and more than 20,000 classes (Wikipedia 2024e).

The model comes in 5 versions: ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152 (where the number indicates the depth in terms of layers). These models use residual blocks. In a traditional neural network, each layer feeds into the next layer. In a network with residual blocks, each layer feeds into the next layer and directly into the layers 2-3 hops away (He et al. 2016).

In Pytorch, model builders are available for all 5 versions. These model builders can be instantiated with or without pretrained weights (Pytorch 2024).

In the project, ResNet will be used for both benchmark and training/evaluation. For the training/evaluation, transfer learning will be used to reduce time and cost of the project. During transfer learning, the weights in a pretrained-version of ResNet will be frozen. Then, an output layer will be added with the correct number of classes (in our case 5). Only the weights in this output layer will be trained. Two of the models will be evaluated: ResNet50 and ResNet101.

6. Evaluation Metrics

A suitable evaluation metric in this project is the **test accuracy**. As the name suggests, the test accuracy is measured on images reserved for testing only.

For a binary classification problem, accuracy is measured by means of following formula (Wikipedia 2024b):

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

where,

TP are true positives
TN are true negatives
FP are false positives
FN are false negatives

This method can be expanded to measure the accuracy of a multiclass classification problem. In our case, there are 5 classes, where each class corresponds to the number of items in the bin. There are also other types of evaluation metrics that can be used in this project, for example, **test loss**. Test loss represents the ‘price paid’ for inaccuracy of predictions on the test dataset. In other words, it provides an estimate of how well the trained model can be generalized to new data (Wikipedia 2024c).

7. Project Design

In general, a machine learning workflow consists of following steps:

- Preprocessing
- Training
- Evaluation
- Deployment

During preprocessing, the data is collected and stored. If necessary, the data is then cleaned and refined. The dataset is then divided into training, validation and testing. Once the preprocessing has been completed, training is then carried out on the **training dataset**. In terms of algorithms, this project will use ResNet which as mentioned is a popular, high-performing algorithm for image classification. The training may involve several iterative steps, including tuning of the hyperparameters. During training the model is evaluated using the **evaluation dataset**. After the training has been completed, the model is evaluated on the **test dataset**, which is untouched i.e. has not been used during training. If the evaluation is satisfactory, the trained model is then deployed. As we are using SageMaker in this project, the deployment consists of creating a SageMaker endpoint that can be queried by the user.

References

Bertorello, P.R., Sripada, S. and Dendumrongsup, N (2019) “Amazon Inventory Reconciliation using AI”. Available online [pdf document]: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3311007

He, K., Zhang, X., Ren, S., Sun, J. (2016) “Deep Residual Learning for Image Recognition”. Available online [pdf document]: https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

PyTorch 2024, “ResNet: Model Builders”, Available online [html document]: https://pytorch.org/vision/main/models/resnet.html

Wikipedia 2024a, “Warehouse management systems”. Available online [html document]: https://en.wikipedia.org/wiki/Warehouse_management_system

Wikipedia 2024b, “Precision and recall”. Available online [html document]: https://en.wikipedia.org/wiki/Precision_and_recall

Wikipedia 2024c, “Loss functions for classification”. Available online [html document]: https://en.wikipedia.org/wiki/Loss_functions_for_classification

Wikipedia 2024d, “Residual Neural Network”. Available online [html document]: https://en.wikipedia.org/wiki/Residual_neural_network

Wikipedia 2024e, “ImageNet”. Available online [html document]: https://en.wikipedia.org/wiki/ImageNet