AWS Machine Learning Engineer Nanodegree Program

Capstone Project Proposal

Inventory Monitoring at Distribution Centers

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. Occasionally, the robots may pick the wrong item or the wrong quantity. This causes customer dissatisfaction and it is difficult to track inventory. Due to the large volume of orders, it is time consuming for workers to double-check each order manually.

This is considered an image classification problem as the model should be able to detect different items in each bin. Below, I will provide a brief history of how modern computer vision, powered by Convolutional Neural Networks (CNNs), came to be.

Back in 1959, two neurophysiologists – David Hubel and Torsten Wiesel – proposed that there are simple and complex neurons in the primary visual cortex. Both types of cells are used in pattern recognition. Later in 1962, they proposed that simple detectors can be summed to create more complex detectors. This forms the basis of CNNs. In the 1980s, Dr. Kunihiko Fukushima proposed using simple (first layer) and complex (second layer) mathematical operations for visual pattern recognition.

Modern CNNs were invented in 1990s when Yann LeCun demonstrated using a CNN model for handwriting recognition. This model was trained using the MNIST dataset. More recently in 2012, a CNN called AlexNet achieved high accuracy labelling pictures in ImageNet. Thereafter, various CNNs achieved better accuracy with different datasets such as CIFAR-10, VisualGenome and medical images (such as chest x-rays).

Problem Statement

This is an image classification problem. The model should count every object instance in the bin. If there are two same objects in the bin, the model should count them as two.

Solution Statement

CNNs are commonly used for image classification tasks. I will fine tune a pre-trained CNN, such as ResNet, to count the number of items in each bin.

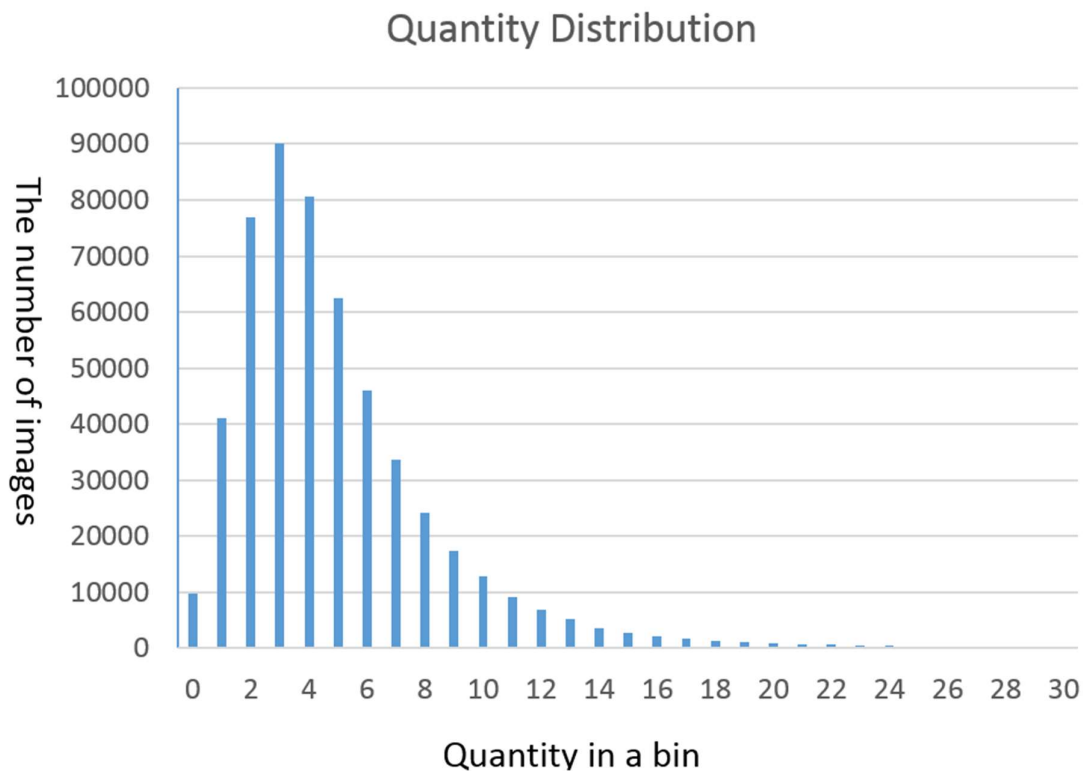
Datasets and Inputs

I will be using the Amazon Bin Image Dataset which contains 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 contains 500,000 images of bins containing one or more objects. For each image there is a metadata file containing information about the image like the number of objects, it's dimension and the type of object. For this task, I will try to classify the number of objects in each bin.

The histogram below shows the distribution of quantity in a bin. More than 90% of the bin images contain less than 10 items. In this project, I will restrict my dataset to images with 5 or less items.

To avoid exceeding the course budget, I will only use a subset of images for training and validation purposes.



Benchmark Model

I will fine tune a ResNet model for this project. ResNet models has been pre-trained on millions of images. This helps them learn general features that can come in useful for a variety of datasets.

Firstly, I will freeze all the convolutional layers and add my own fully connected layer. Next, I will train the model using my training dataset. This entails running the forward pass for the whole network and perform the backward pass only for the fully connected layer.

Since I am only training the fully connected layer, training can be done quickly and easily. However, I am using the whole network when performing prediction, so I can get the improved accuracy of using a CNN.

Evaluation Metrics

I will use accuracy to evaluate my model.

For each image, my model will output the probability of each class. I will take the largest probability as the predicted class. Accuracy = number of correct predictions / total number of predictions

If my model is working well, I should see that accuracy increases with more epochs.

Project Design

Before I begin working on the model, I need to create a new notebook instance in SageMaker as well as a new S3 bucket for this project.

Next, I will prepare the data by downloading the bin images to my workspace and uploading them to my S3 bucket. As the dataset is very large, I will only pick a subset for training and validation. I will

use the metadata to group the images according to their item quantity. The training and validation folder structure will be:

root/1/xxx.jpg

root /1/xxy.jpg

root /2/abc.jpg

root /2/def.jpg

After data preparation, I will create a training script which fine tunes a Resnet model using my training and validation dataset. This entails resizing and normalizing the images before passing them into the pre-trained Resnet model.

Once the training script is ready, I will prepare a Jupyter notebook which specifies the training environment (e.g. instance type, instance count). This notebook will setup the training estimator and submit the job. The final model is saved in SageMaker and can be used for deployment.

If time permits, I will attempt some of the standout suggestions such as hyperparameter tuning and model deployment.