

Inventory Monitoring at Distribution Centers

Lesson Downloads Cloud Resources

Project Overview: Inventory Monitoring at Distribution Centers

Distribution centers often use robots to move objects as a part of their operations. Objects are carried in bins which can contain multiple objects. In this project, you will have to build a model that can count the number of objects in each bin. A system like this can be used to track inventory and make sure that delivery consignments have the correct number of items. To build this project you will use AWS SageMaker and good machine learning engineering practices to fetch data from a database, preprocess it, and then train a machine learning model. This project will serve as a demonstration of end-to-end machine learning engineering skills that you have learned as a part of this nanodegree.

How it Works

To complete this project we will be using the [Amazon Bin Image Dataset](#). 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, its dimension and the type of object. For this task, we will try to classify the number of objects in each bin. To perform the classification you can use a model type and architecture of your choice. For instance you could use a pre-trained convolutional neural network, or you could create your own neural network architecture. However, you will need to train your model using SageMaker.

Once you have trained your model you can attempt some of the Standout Suggestion to get the extra practice and to turn your project into a portfolio piece.



Example of an image from the dataset

Pipeline

To finish this project, you will have to perform the following tasks:

1. Upload Training Data: First you will have to upload the training data to an S3 bucket.
2. Model Training Script: Once you have done that, you will have to write a script to train a model on that dataset.

3. Train in SageMaker: Finally, you will have to use SageMaker to run that training script and train your model

Here are the tasks you have to do in more detail:

	Setup AWS To build this project, you will have to use AWS through your classroom. Below are your main steps: <ul style="list-style-type: none"> ▪ Open AWS through the classroom on the left panel (Open AWS Gateway) ▪ Open SageMaker Studio and create a folder for your project
	Download the Starter Files We have provided a project template and some helpful starter files for this project. You can clone the Github Repo . <ul style="list-style-type: none"> ▪ Clone or download starter files from GitHub ▪ Upload starter files to your workspace
	Preparing Data To build this project you will have to use the Amazon Bin Images Dataset <ul style="list-style-type: none"> ▪ Download the dataset: Since this is a large dataset, you have been provided with some code to download a small subset of that data. You are encouraged to use this subset to prevent any excess SageMaker credit usage. ▪ Preprocess and clean the files (if needed) ▪ Upload them to an S3 bucket so that SageMaker can use them for training ▪ OPTIONAL: Verify that the data has been uploaded correctly to the right bucket using the AWS S3 CLI or the S3 UI
	Starter Code Familiarize yourself with the following starter code <ul style="list-style-type: none"> ▪ <code>sagemaker.ipynb</code> ▪ <code>train.py</code>
	Create a Training Script Complete the TODO's in the <code>train.py</code> script <ul style="list-style-type: none"> ▪ Read and Preprocess data: Before training your model, you will need to read, load and preprocess your training, testing and validation data ▪ Train your Model: You can choose any model type or architecture for this project
	Train using SageMaker Complete the TODO's in the <code>sagemaker.ipynb</code> notebook <ul style="list-style-type: none"> ▪ Install necessary dependencies ▪ Setup the training estimator ▪ Submit the job
	Final Steps An important part of your project is creating a <code>README</code> file that describes the project, explains how to set up and run the code, and describes your results. We've included a template in the starter files (that you downloaded earlier), with <code>TODOs</code> for each of the things you should include. <ul style="list-style-type: none"> ▪ Complete the <code>README</code> file

Standout Suggestions

Standout suggestions are some recommendations to help you take your project further and turn it into a nice portfolio piece. If you have been having a good time working on this project and want some additional practice, then we recommend that you try them. However, do not that these suggestions are all optional and you can skip any (or all) of them and submit the project in the next page.

Here are some of suggestions to improve your project:

- **Model Deployment:** Once you have trained your model, can you deploy your model to a SageMaker endpoint and then query it with an image to get a prediction?
- **Hyperparameter Tuning:** To improve the performance of your model, can you use SageMaker's Hyperparameter Tuning to search through a hyperparameter space and get

the value of the best hyperparameters?

- **Reduce Costs:** To reduce the cost of your machine learning engineering pipeline, can you do a cost analysis and use spot instances to train your model?
- **Multi-Instance Training:** Can you train the same model, but this time distribute your training workload across multiple instances?

Once you have completed the standout suggestions, make sure that you explain what you did and how you did it in the README. This way the reviewers will look out for it and can give you helpful tips and suggestions!

