Capstone project for AWS ML Engineer Nanodegree

Montserrat Maceda Noda
November 19, 2024

Inventory Monitoring at Distribution Centers
(Amazon Bin Image counting)

Domain Background:

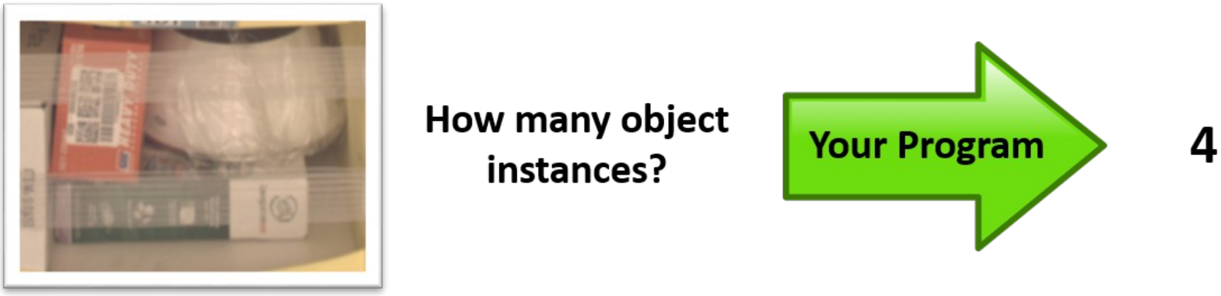
Distribution centers often use robots to move objects as a part of their operations. Objects are carried in bins which can contain multiple objects. In this project, I will have to build a model that can count the number of objects in each bin. A system like this can be used to track inventory and make sure that delivery consignments have the correct number of items.

To build this project I 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 I have learned as a part of this nanodegree.

Problem Statement:

This is a image classification problem like other we've seen throught this program, but in this case the result of the image classification must be the total number of different items in a bin image.

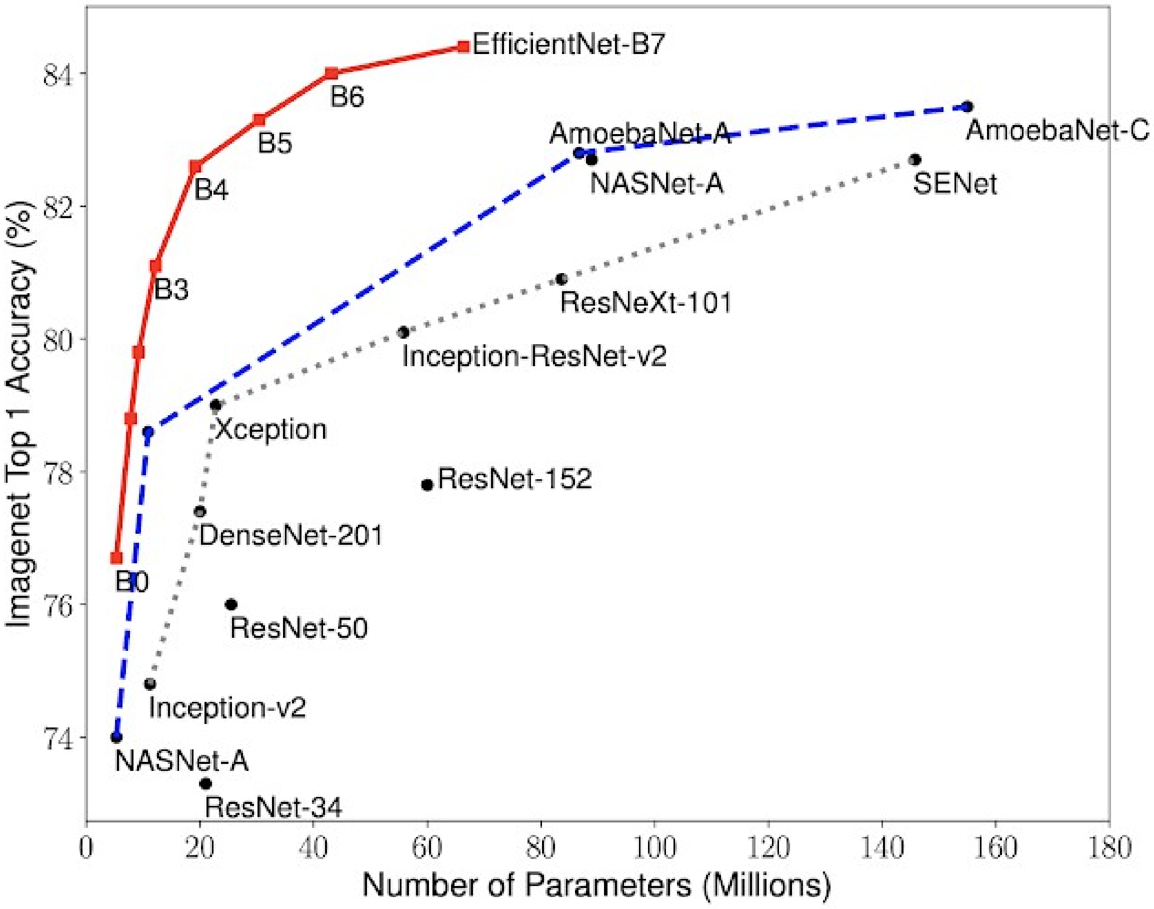
Example:



Solution Statement:

For resolving this image classification problem we can choose a pre trained convolutional neural network model and perform transfer learning with our image dataset, to create a new model adjusted to this problem,as we've already seen thought this program.

We can use models like ResNet like in previous projects, but investigating a little more of different model alternatives with better accuracy, I encounter several articles describing **EffiicientNet** as a better model for this problem, due to a better general accuracy even for not so big datasets as we can see in the graphic below:



References:
https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/#h-efficientnet
https://arxiv.org/abs/1905.11946

In this project we will use the Pytorch implementation of EfficientNet B4, for being a good compromise between cost and accuracy.

Documentation:
https://pytorch.org/vision/main/models/generated/torchvision.models.efficientnet_b4.html#torchvision.models.efficientnet_b4

Datasets and Inputs:

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 in json format containing information about the image like the number of objects, it's dimension and the type of object. For this task, we will try to classify the number of objects in each bin.

(opens in a new tab)

This dataset can be downloaded from a public s3 bucket, and for this project we will be uploading it to a s3 bucket in our aws account, in order to be able to perform the training through sagemaker studio.

For this project we will use a subset of this image dataset to avoid costs. This image subset list is present in the file *file_list.json.*

References:

https://github.com/awslabs/open-data-docs/tree/main/docs/aft-vbi-pds

Benchmark Model:

As we can see in https://registry.opendata.aws/amazon-bin-imagery/, there's two publications about examples usage of this dataset linked:

1. Amazon Bin Image Dataset(ABID) Challenge:

   In this work, thay have used a ResNet34 model as a base for training. The dataset has been organized on two tasks: moderate task, and hard task. In the hard task, all of the images will be considered for object counting, meanwhile in moderate one, only images having up to five objects are used. With this method, the deep CNN will classify the image as one of 6 categories (0-5 items, for moderate difficulty).

2. Amazon Inventory Reconciliation using AI:

   In this work, the authors have experimented with some linear and non-linear models. The project report explain they used logistic regression, classification trees, SVMs and CNNs for modeling. Among all this models, the ResNet50 (one of CNN architectures) with Adam optimization has given the best result.

For this project we will use as stated before the EfficientNet b4 model as a base for the training with Adam optimization, and try to improve or at least achieve aproximately the same level of accuracy that in the last project mentioned best scenarios (between 55-62% training accuracy)

Evaluation Metrics:

For this project we will use two metrics to evaluate the model:

Accuracy and **RMSE**(Root Mean Square Error), to evaluate the distance from the real value of the number of items in an image to the model predicted one.

Project Design:

- Download and perform EDA over the dataset
- Upload the resulting dataset to s3
- Perform training with hyperparameter tuning

- Perform model training with best hyperparameters
- Model evaluation