Inventory Monitoring at Distribution Centers

(Amazon Bin Image counting)

Mohsen Mahmoodzadeh
NLP Engineer at Part AI Research Center
1mohsen.smm@gmail.com

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. Sometimes, objects are misplaced while being handled, resulting in a mismatch between the recorded bin inventory and its actual content.

A system that can facilitate inventory tracking and ensure complete delivery of consignments by counting the number of items in each bin will be so valuable.

In this project, we 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.

Problem Statement

Specifically and with more scientific and precise literature, our mentioned application is an object counting task.
The goal of the object counting task is to count the number of object instances in a single image or video sequence. It has many real-world applications such as traffic flow monitoring, crowdedness estimation, and product counting.

We count individual instances separately, which means if there are two same objects in the bin, we count them as two.

In the figure below, we can see an example of the input and output of this task:

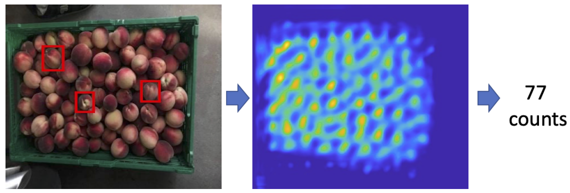

Figure 1: An example object counting task

Solution Statement

In this project, I focused on Improving the previous solutions and having an affordable model at the same time.

The base model I used for the problem is one of the EfficientNet family called EfficientNet_B4 architecture. [Here]() you can find a table which shows the accuracy of the models(on ImageNet dataset) and also the number of parameters each has.

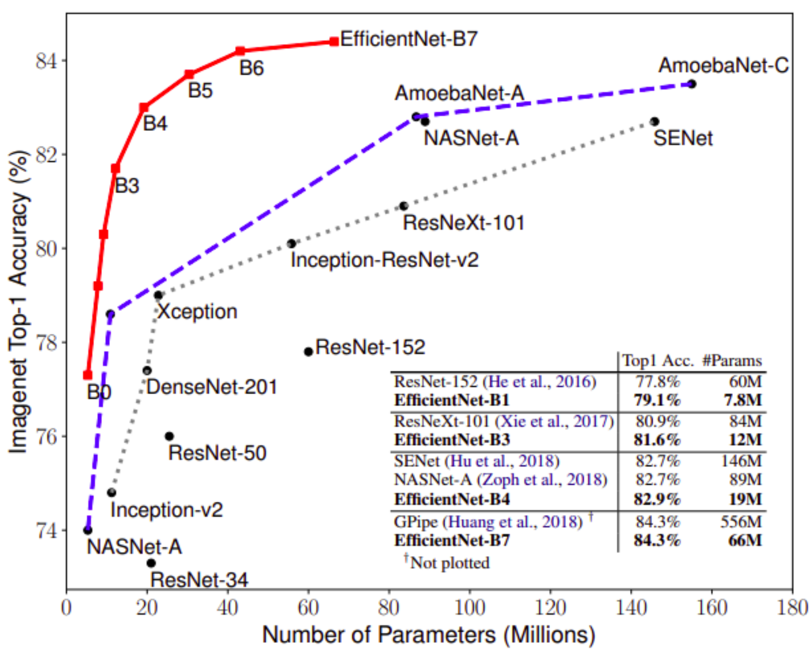

Figure 2: The outperforming of EfficientNet over other models on model size

I selected EfficientNet_B4 due to its compromise between resource computation and accuracy power.

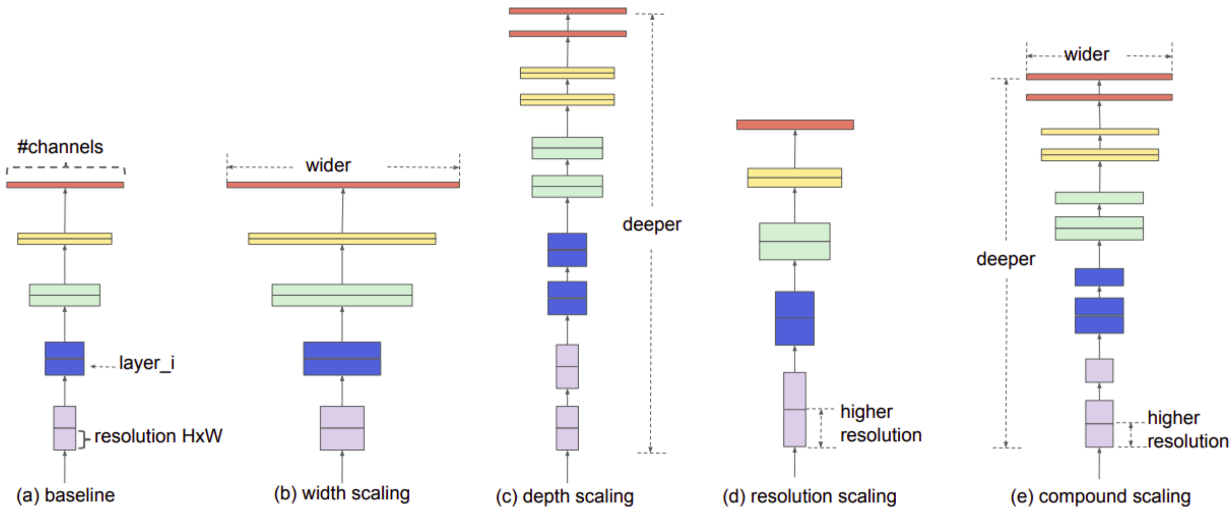

Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Figure 3: Model scaling methods w.r.t EfficientNet paper

The platform that will be used is AWS, more specifically the following services:

- S3: It's all about storage. Whether it is data storage or model storage.
- SageMaker Studio: it's all about logic of the pipeline: data preprocessing, modeling, training, evaluation and deployment.

Dataset and Inputs

For the mentioned above task, we use **Amazon Image Bin Dataset**. The dataset contains 536,435 bin JPEG 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.

Specifically, the images contain 459,558 different products skews of different shapes and sizes. Each image/metadata tuple corresponds to a bin with products. The metadata includes the actual count of objects in the bin, which is used as a label to train our model.

The input to our task is a raw color image of the products in a bin. And its output is the bin's predicted quantity, the number of products in the image.

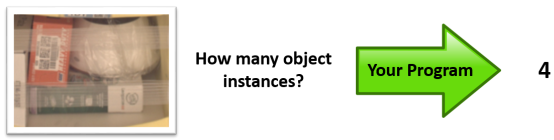

Figure 4: An example of input and output of the project for each sample

Benchmark Model

Most of the information used for this section was gotten from paperswithcode.com site. There are many papers that address various aspects of the object counting task with different approaches and architectures.

There are three main benchmarks for our work which will be described at the following:

1. Amazon Bin Image Dataset(ABID) Challenge:

   In this work, an interesting trick has been used and that is converting the continuous task to a classification problem after doing a good EDA. 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; while in moderate one, images having up to five objects are held and other ones are avoided. With this trick, the problem got converted to a classification one with six classes(from 0 to 5). The paper used ResNet34 architecture as a base for training.

2. Amazon Inventory Reconciliation Using AI:

   This work is an improvement on the previous benchmark. The paper has used the same trick for converting the task to a classification problem. For modeling, it has experimented with some linear and non-linear models. The paper used logistic regression, classification trees, SVMs and CNNs for modeling. Among these ones, ResNet50(one of CNN architectures) has given the best result.

3. Inventory Monitoring at Distribution Centers:

In this work, the focus is on cost reduction with using a simpler and yet accurate enough model. This work used some tricks such as weighted random sampling, data augmentation, model refining and multi-resolution training for performance improvement. The work has used EfficientNet B0 architecture for training.

Evaluation Metrics

As for model evaluation, the following metrics will be calculated:

- **Accuracy**

$$accuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{p_i}{g_i}$$

Where:

- N is the number of data
- p_i is the number of counted objects(at the ith image) by the algorithm
- g_i is the number of total objects

- **RMSE**

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - g_i)^2}$$

Where:

- N is the number of data
- p_i is the number of counted objects(at the ith image) by the algorithm
- g_i is the ground truth or the number of counted objects(at the ith image) by the algorithm()

- **F1 Score**

$$F1\,Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Where:

- $Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$

- $Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$

Since there is an existence of class imbalance in the dataset, accuracy can not be used as the main measure of model performance. Instead, I used F1 Score to reflect the side effects of class imbalance on model performance.

In general, accuracy is not a good metric for the cases in which there is an imbalance at the dataset. However the imbalance at some subsets of the dataset is trivial, we can ignore the side effects of the imbalance, and continue with accuracy metric.

Project Design

1. Hyperparameter tuning
2. Model training with best hyperparameters
3. Model evaluation

References

1. https://github.com/silverbottlep/abid_challenge
2. https://github.com/pablo-tech/Image-Inventory-Reconciliation-with-SVM-and-CNN
3. https://github.com/williamhyun/nd009t-capstone-starter

Authors

**Mohsen Mahmoodzadeh** is an ML Engineer at Part AI Research Center. His main task there is working on the G2P(Grapheme to Phoneme) system for Persian language.