Inventory Monitoring at Distribution Centers

CAPSTONE PROJECT REPORT

Sathyapriyan Kannan

Udacity's AWS Machine Learning Engineer Nanodegree Program

1. Domain Background:

Inventory Monitoring is a critical process in a Distribution Center. It is the process of ensuring the right number of products or items present in a particular bin. It is an integral part of the supply chain management, that ensures the flow of goods from manufacturing venue to the warehouse or a distribution center and to the sale location.

Long before the industrial age, keeping track of things consisted of manually counting and tallying items. The earliest form of inventory management dates back over 50,000 years in which people used “tally sticks” to count.

The empowerment of the distribution centers to automate its process through a digital workplace is the need of the hour to manage the demand and supply chain. During the time of the pandemic, there is a huge demand for the logistics services.

2. Problem Statement:

Manual Inventory Management requires a huge workforce, and it is also error prone. So, there is a need for an effective inventory management.

Effective Inventory Management enables the business to save time and increase productivity. The automation of this process would eliminate the manual errors, thus saving cost and time. For larger organization, the process is much complicated.

With the recent advancement in the technology, we could solve the above problem with machine automated tasks. The Computer Vision Process aids us in solving the defined problem.

With the huge rise in the demand for inventory management, scalability is a major concerned that affects the long term run of the business. With the automated workflow, scalability becomes much simpler and cost efficient.

The solution can be replicated to different objects with respective datasets for training and the outcomes are measurable with the proposed evaluation metrics[5].

The problem is identified as a classification problem and upon given an input image from the live camera, the count of the objects in the bin can be predicted using the machine learning model.

3. Solution Statement:

The identified problem can be solved using Computer Vision Techniques. By feeding appropriate data to train the machine learning model, a robust model can be produced.

Components:

1. Dataset / Image Source
2. Algorithm
3. Environment or the platform

Dataset or the Image Source:

The **Amazon Bin Image Dataset** is to be used to train the model.

It contains over 500,000 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.

Documentation about the open source dataset can be found here.

These are some typical images in the dataset. A bin contains multiple object categories and various number of instances. The corresponding metadata exist for each bin image and it includes the object category identification(Amazon Standard Identification Number, ASIN), quantity, size of objects, weights, and so on. The size of bins are various depending on the size of objects in it. The tapes in front of the bins are for preventing the items from falling out of the bins and sometimes it might make the objects unclear. Objects are sometimes heavily occluded by other objects or limited viewpoint of the images.

Algorithm :

1. The Algorithm will be built using the Convolution Neural Network (CNN) architecture.
2. Deep Learning Framework – PyTorch
3. A corresponding Sage Maker instance will be created and data will be fed from the S3 bucket.
4. The model will also be tuned to find out the best hyper-parameters.

Environment or the platform :

1. Amazon Web Services
2. Sage Maker Studio – to train, tune and deploy the model.

S3 – Storage Bucket

Sample Image and JSON Pair :

This is an example of image(jpg) and metadata(json) pair. This image contains 3 different object categories. For each category, there is one instance. So, "EXPECTED_QUANTITY" is 3, and for each object category "quantity" field was 1. Unique identifier("asin") is assigned to each object category, e.g. here "B00CFQWRPS", "B00T0BUKW8", and "B00C3WXJHY"


```
{"BIN_FCSKU_DATA": { "B00CFQWRPS": {
      "asin": "B00CFQWRPS",
      "height": {
        "unit": "IN",
        "value": 2.399999997552
      }, "length": {
        "unit": "IN",
        "value": 8.199999991636
      },
      "name": "Fleet Saline Enema, 7.8 Ounce (Pack of 3)",
      "normalizedName": "(Pack of 3) Fleet Saline Enema, 7.8 Ounce",
      "quantity": 1,
      "weight": {
        "unit": "pounds",
        "value": 1.8999999999999997
      },  "width": {
        "unit": "IN",
        "value": 7.199999992656
      }
    },
    "ZZXI0WUSIB": {
      "asin": "B00T0BUKW8",
      "height": {
        "unit": "IN",
        "value": 3.99999999592
      },
      "length": {
        "unit": "IN",
        "value": 7.899999991942001
      },
      "name": "Kirkland Signature Premium Chunk Chicken Breast Packed in Water, 12.5 Ounce, 6 Count",
      "normalizedName": "Kirkland Signature Premium Chunk Chicken Breast Packed in Water, 12.5 Ounce, 6 
Count",
      "quantity": 1,
      "weight": {
        "unit": "pounds",
        "value": 5.7
      },
      "width": {
        "unit": "IN",
        "value": 6.49999999337
      }
    },
    "ZZXVVS669V": {
      "asin": "B00C3WXJHY",
      "height": {
        "unit": "IN",
        "value": 4.330708657
      },
      "length": {
        "unit": "IN",
        "value": 11.1417322721
      },
      "name": "Play-Doh Sweet Shoppe Ice Cream Sundae Cart Playset",
      "normalizedName": "Play-Doh Sweet Shoppe Ice Cream Sundae Cart Playset",
      "quantity": 1,
      "weight": {
        "unit": "pounds",
        "value": 1.4109440759087915
      },    "width": {
        "unit": "IN",
        "value": 9.448818888
      }
    }
  },
  "EXPECTED_QUANTITY": 3
}
```

4. Benchmark Model:

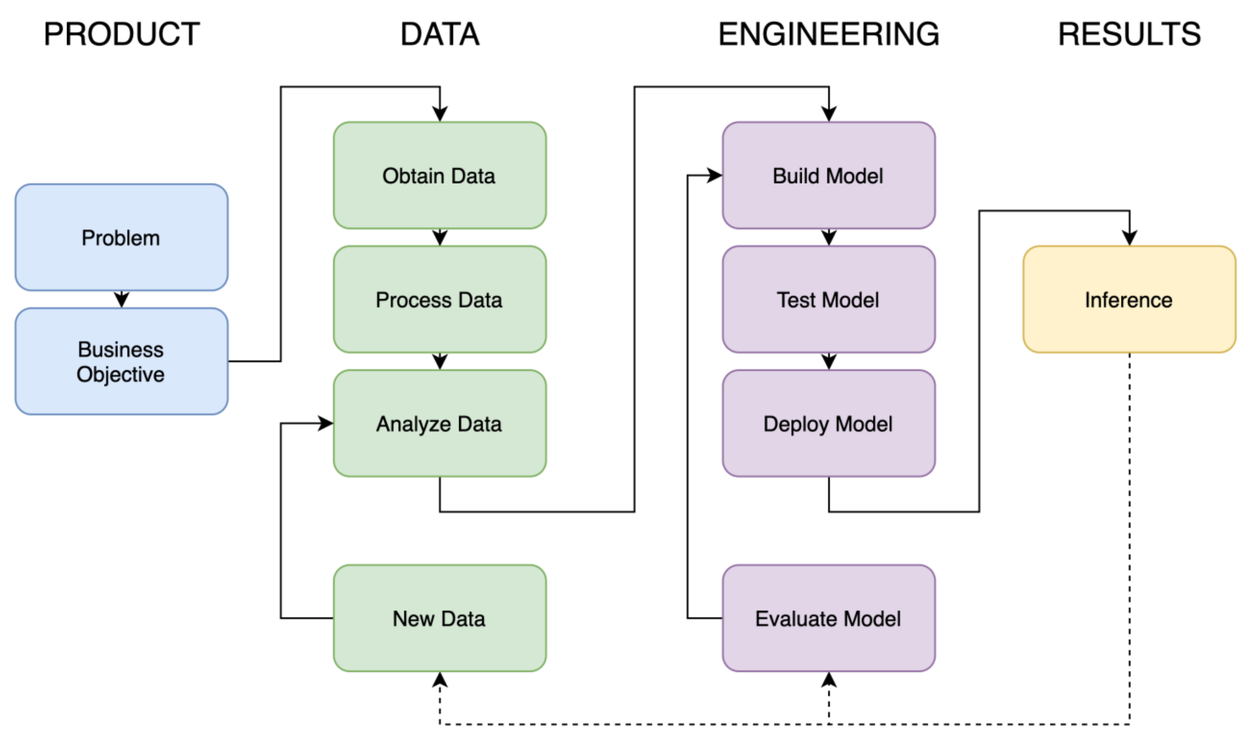
The object counting using convolutional neural network has been introduced with connected component analysis from the following research paper is used as a benchmark to achieve the results. The authors had produced relatedly good results.

N. K. Verma, T. Sharma, S. D. Rajurkar and A. Salour, "Object identification for inventory management using convolutional neural network," 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2016, pp. 1-6, doi: 10.1109/AIPR.2016.8010578.

5. Evaluation Metrics:

Since it is a classification problem, **the overall accuracy of the classification and F1 score** can be used to evaluate the performance of the trained model.

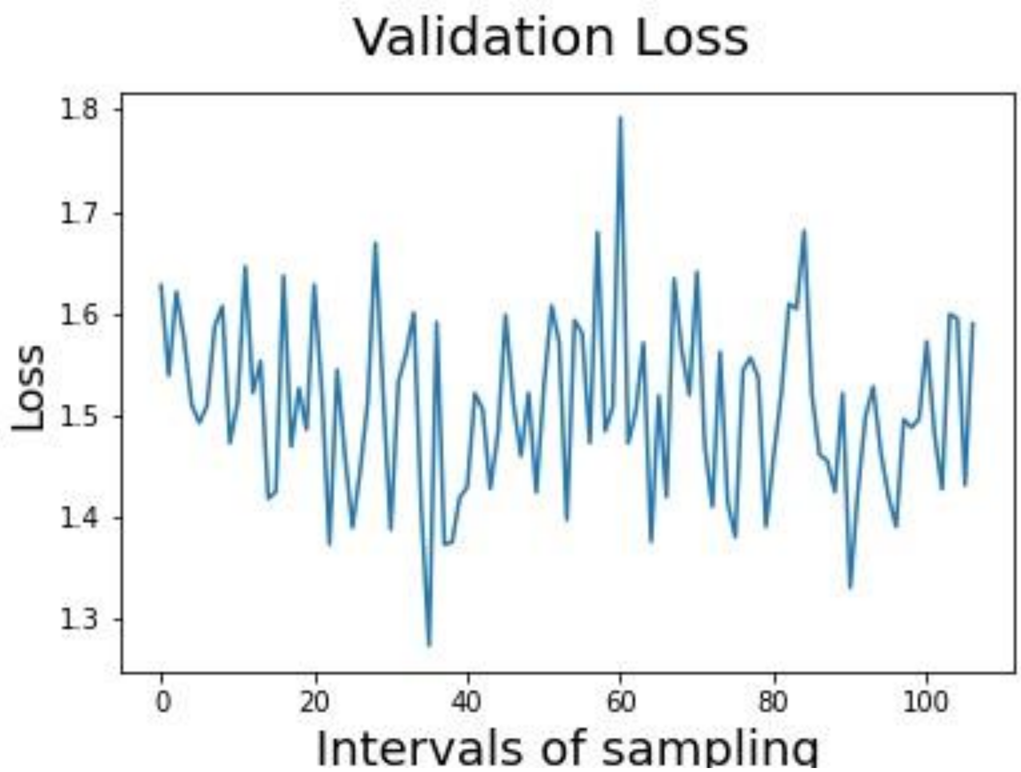
6. Workflow:



1. The given data could be transformed into a direct geometrical dimension without or minimal loosing of the data.
2. The data with X x Y dimension could be fed to the fully connected Neural Network for training.
3. The Data Augmentation techniques Geometric transformations, Color space transformations kernel filters to sharpen or blur an image, Random Erasing, Mixing images – basically, mix images with one another might to used to improve the training and avoid overfitting of the model.
4. The trained model will be evaluated based on the evaluation metrics defined in section 5.
5. Upon satisfactory results of the evaluation, the model can be used for inference.

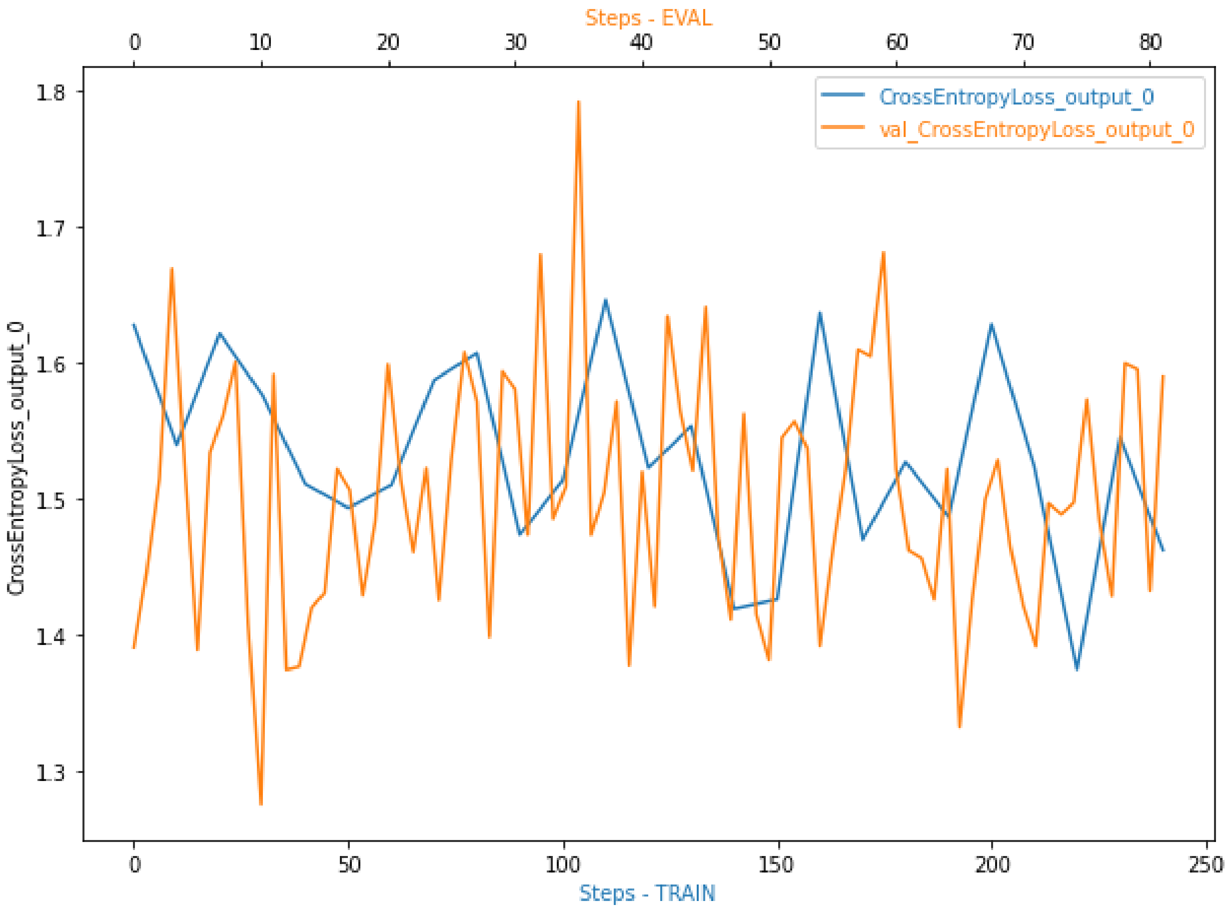
7. Model Evaluation Report:

Hyperparameter tunning is performed to find out the best hyperparameters to train the model.



The validation loss of the training process remains almost constant and there are frequent highs and lows, when compared with the training loss, it seems to fit along with it as shown below.

The accuracy of the model is not that great compared to the benchmark model chosen, so as next steps, have to work more on the data and it's transformation.



8. Cost Analysis:

The cost incurred for the Sage Maker is shown in the below image.

SageMaker		$3.30
US East (N. Virginia)		**$3.30**
Amazon SageMaker CreateVolume-Gp2		$0.02
$0.00 for SageMaker Debugger Built-in Rule Volume	78.383 GB-Mo	$0.00
$0.14 per GB-Mo of Endpoint ML storage	0.066 GB-Mo	$0.01
$0.14 per GB-Mo of Training Job ML storage	0.080 GB-Mo	$0.01
Amazon SageMaker Invoke-Endpoint		$0.00
$0.016 per GB for Endpoint Data IN	0.000000130 GB	$0.00
$0.016 per GB for Endpoint Data OUT	0.000000120 GB	$0.00
Amazon SageMaker RunInstance		$3.28
$0.0 for SageMaker Debugger Built-in Rule Instance	1.354 Hrs	$0.00
$0.00 for SageMaker Debugger Built-in Rule Instance	0.590 Hrs	$0.00
$0.05 per Studio-Notebook ml.t3.medium hour in US East (N. Virginia)	2.876 Hrs	$0.14
$0.115 per Hosting ml.m5.large hour in US East (N. Virginia)	12.297 Hrs	$1.41
$0.23 per Training ml.m5.xlarge hour in US East (N. Virginia)	1.996 Hrs	$0.46
$0.7364 for SageMaker Studio Notebook Instance ml.g4dn.xlarge per hour	1.716 Hrs	$1.26

3.28 USD is charged for the model training. The instance used is Amazon E2 – PyTorch 1.8 GPU Optimized.

If the spot instances are used for the training, the cost would have been significantly lesser compared to the current incurred costs.

Cost incurred for the S3 is as below :

Simple Storage Service		$0.29
US East (N. Virginia)		**$0.29**
Amazon Simple Storage Service Requests-Tier1		$0.21
$0.005 per 1,000 PUT, COPY, POST, or LIST requests	42,684.000 Requests	$0.21
Amazon Simple Storage Service Requests-Tier2		$0.08
$0.004 per 10,000 GET and all other requests	192,708.000 Requests	$0.08
Amazon Simple Storage Service TimedStorage-ByteHrs		$0.00
$0.023 per GB - first 50 TB / month of storage used	0.185 GB-Mo	$0.00

9. References:

1. Sagemaker -Python SDK for PyTorch - https://sagemaker.readthedocs.io/en/stable/overview.html
2. N. K. Verma, T. Sharma, S. D. Rajurkar and A. Salour, "Object identification for inventory management using convolutional neural network," 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2016, pp. 1-6, doi: 10.1109/AIPR.2016.8010578.
3. Amazon Bin Image Dataset - https://github.com/awslabs/open-data-docs/tree/main/docs/aft-vbi-pds