Inventory Monitoring at Distribution Centers

Capstone Project Proposal

1. Introduction

Inventory monitoring is an essential process at distribution centers. It involves counting inventory objects and ensures that the correct quantities are maintained to meet supply chain demands. There has been a great need to automate the process to meet the growing increase in consumer demand and eliminate errors.

Distribution centers use robots to move objects in bins that can contain multiple objects. Convolutional Neural Networks (CNNs) have achieved excellent results for object detection in images [1-4], text detection [5] vehicle detection [6] and many other applications.

In this project, machine learning techniques will be used to build a model that can classify the number of objects in each bin to assist in automating the inventory monitoring process. This system can be used to track inventory and ensure that delivery consignments have the correct number of items [16].

2. Domain background

Inventory monitoring has evolved throughout the years to save time, lower costs and increase efficiency. Ancient methods involved using tally sticks (50,000 years ago) and clay tokens (4000 years ago) to count objects. Herman Hollerith developed the first modern automatic computation machine (1880). It recorded information using punch cards. The barcode was invented by Norman Woodland (1940). Lasers were later used to scan barcodes to track inventory (1960). Inventory tracking became even more efficient with more advanced computers and software (1980s and 1990s). Barcode readers could instantly update business databases therefore it wasn't necessary to input data by hand (2000s). RFID (Radio-frequency identification) tags (patented in 1970s) became widely used in warehouses, factories, and retail stores (2000s) [7, 8].

Research in computer vision and machine learning, seeks to improve and further automate inventory monitoring. Approaches to count objects in images include using shape information [9], background subtraction in vehicle detection and counting systems [10], fuzzy color histograms [11], visual features, clustering and artificial neural networks [12], support vector machines [13] and CNNs [13-15, 19].

3. Problem statement

Inventory monitoring systems face the following problems: employee errors, stock shortages, excess inventory, misplaced inventory and lack of optimization [15, 17]. Physically counting or scanning objects is a tedious and time consuming process that may lead to human error, fatigue, reduced efficiency and ultimately monetary losses.

The automation of inventory monitoring systems by leveraging computer vision and machine learning techniques will assist to minimize these problems and eliminate human error. This will in turn assist in saving costs, time and increase efficiency in inventory monitoring.

The problem of classifying the number of objects could be replicated by using the Amazon Bin Image Dataset (ABID) [18]. The use of CNNs to solve this problem is a growing research area [13-15, 19]. The results could be measured by accuracy and Root Mean Square Error (RMSE), section 7.

4. Datasets and inputs

The Amazon Bin Image Dataset (ABID) [18] will be used to classify the number of objects in each bin. It contains over 500 000 JPEG images and the corresponding JSON metadata from bins of a pod in an Amazon Fulfillment Center (AFC). The bin images are captured as robots transport pods during normal AFC operations [18].

Images are found in the bin-images directory, and metadata for each image is found in the metadata directory. Images and their corresponding metadata share numerical unique identifiers [20]. An example image and its corresponding metadata are shown in Figures 1 and 2, respectively.

The metadata contains information about the objects in the bin such as the Amazon Standard Identification Number (ASIN), name, normalized name, quantity, weight and size [19]. It contains the expected quantity of objects in the bin. The images and their corresponding metadata will serve as inputs to the training algorithm.

A random storage scheme is used where objects are stored in accessible bins with available space, therefore each bin's contents is random. Each bin image may show only one type of object or many different types of objects. If objects are misplaced the contents of some bin images may not match the inventory record [20].

Figure 1: Bin image 777.jpg [22].

{ "BIN_FCSKU_DATA":
{ "B0067EQF9I":
{ "asin": "B0067EQF9I",
"height": { "unit": "IN", "value": 2.2 },
"length": { "unit": "IN", "value": 4.9 },
"name": "PUR Gum Aspartame Free Wintergreen Gum, 9 Count (Pack of 12)",
"quantity": 3,
"weight": { "unit": "pounds", "value": 0.6 },
"width": { "unit": "IN", "value": 3.8 } } },
"EXPECTED_QUANTITY": 3,
"image_fname": "777.jpg"}

Figure 2: JSON metadata for image 777.jpg [23].

Due to the large size of the data, a small subset of the data will be used to stay within the allocated budget. The data subset contains 10441 bin images. The distribution of the number of products per bin in the data subset is shown in Figure 3. The range of the number of products is from one to five. The most common number of products per bin is three.

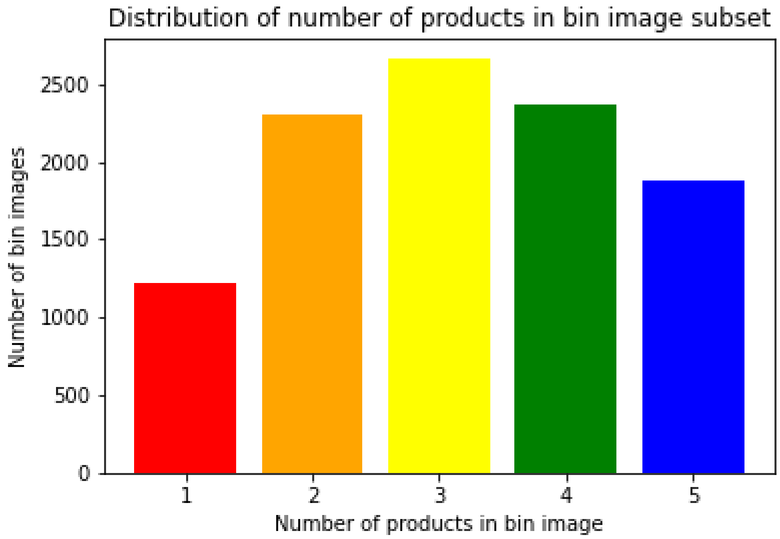

Figure 3: Distribution of the number of products in bin image subset.

5. Solution statement

To provide a solution to this project, AWS SageMaker and good machine learning engineering practices will be used to:

1. Fetch data from a database and preprocess it.

2. Perform hyperparameter tuning.

3. Train a CNN to count the number of objects in an image and measure its accuracy and RMSE.

4. Deploy an endpoint and make a prediction.

The project will serve as a demonstration of the end-to-end machine learning engineering skills learned in the Nanodegree [16]. The steps that will be taken are detailed in section 8.

6. Benchmark model

The ABID Challenge by Eunbyung Park [19] will be used as a benchmark to compare results. ResNet34 was trained and a validation accuracy of 55.67% was obtained.

7. Evaluation metrics

Accuracy and Root Mean Square Error (RMSE) are standard metrics used to evaluate model performance. Accuracy and RMSE are defined below, where *1* is an indicator function, p is the prediction, g is the ground truth and N is the number of samples in the dataset [19].

$$\text{Accuracy: } \frac{1}{N}\sum_{i=1}^{N} 1[p_i == g_i] \qquad \text{RMSE: } \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - g_i)^2}$$

Figure 4: Accuracy and RMSE equations [19].

8. Project design

The aims of this project are to train a machine learning model to classify the number of objects in a bin and demonstrate a machine learning engineering pipeline. To complete the project, the following steps will be taken:

1. Data: Data will be obtained from a database and uploaded to Amazon S3 for training the model. The data will be split into training (60%), validation (20%), and testing (20%) sets. The data will be preprocessed (i.e., resized, randomly flipped, converted to a tensor and normalized) before being passed to the model.

2. Model selection: A pretrained PyTorch CNN model (e.g., ResNet, VGG, Inception) [21] will be selected to perform classification based on accuracy, model size and speed.

3. Instance selection: Instances will be chosen based on cost and computing power.

4. Hyperparameter Tuning: SageMaker's hyperparameter tuning will be used to search through a hyperparameter space and find the best hyperparameters to improve model performance.

5. Training: The model will be fine-tuned and trained with the best hyperparameters using SageMaker and a training script. The trained model will be stored in S3.

6. Debugger and profiler: SageMaker's debugger and profiler will be used to provide insights for improvements and log metrics.

7. Model deployment: The trained model will be deployed to a SageMaker endpoint and queried with an image to get a prediction. [16]

8. Evaluate performance: The accuracy and RMSE of the model will be measured.

References

[1] Redmon, J. & Farhadi, A. 2018. YOLOv3: An incremental improvement. arXiv preprint, arXiv1804.02767.

[2] Fu, C-Y., Liu, W., Ranga, A., Tyagi, A. & Berg, A.C. 2017. DSSD: Deconvolutional single shot detector. arXiv preprint, arXiv:1701.06659.

[3] https://github.com/ultralytics/yolov5, Accessed 18/1/22.

[4] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

[5] He, T., Huang, W., Qiao, Y. and Yao, J., 2016. Text-attentional convolutional neural network for scene text detection. IEEE transactions on image processing, 25(6), pp.2529-2541

[6] Chen, X., Xiang, S., Liu, C.L. and Pan, C.H., 2014. Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and remote sensing letters, 11(10), pp.1797-1801.

[7] https://www.citycleanandsimple.com/2018/07/16/history-of-inventory-management-technology/#:~:text=The%20earliest%20form%20of%20inventory,%E2%80%9Ctally%20sticks%E2%80%9D%20to%20count.&text=Over%20time%2C%20inventory%20management%20developed,ancient%20Greek%20and%20Egyptian%20societies , Accessed 18/1/22.

[8] https://www.barcodedirect.com/the-evolution-of-inventory-management/, Accessed 18/1/22.

[9] Wählby, C., Sintorn, I.M., Erlandsson, F., Borgefors, G. and Bengtsson, E., 2004. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. Journal of microscopy, 215(1), pp.67-76.

[10] Mandellos, N.A., Keramitsoglou, I. and Kiranoudis, C.T., 2011. A background subtraction algorithm for detecting and tracking vehicles. Expert Systems with Applications, 38(3), pp.1619-1631.

[11] Verma, N.K., Goyal, A., Chaman, A., Sevakula, R.K. and Salour, A., 2015, June. Template matching for inventory management using fuzzy color histogram and spatial filters. In 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 317-322.

[12] Sharma, T., Rajurkar, S.D., Molangur, N., Verma, N.K. and Salour, A., 2019. Multi-faced object recognition in an image for inventory counting. In Computational Intelligence: Theories, Applications and Future Directions-Volume II, pp. 333-346.

[13] Rodriguez Bertorello, P., Sripada, S. and Dendumrongsup, N., 2018. Amazon Inventory Reconciliation Using AI. Available at SSRN: https://ssrn.com/abstract=3311007, Accessed 18/1/22.

[14] Verma, N.K., Sharma, T., Rajurkar, S.D. and Salour, A., 2016, October. Object identification for inventory management using convolutional neural network. In 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1-6.

[15] Kalahiki, C.B., 2020. Computer Vision for Inventory Management, Master of Science Thesis, College of Engineering & Science Louisiana Tech University.

[16] https://classroom.udacity.com/nanodegrees/nd189/parts/cd0549/modules/864d3e12-dc8d-47c6-b443-d01d6c7aedde/lessons/d6ec6005-e421-455c-9b79-ebc2df44e1ac/concepts/ad3d3503-f9a9-4c18-86ae-c03d6fbb1470, Accessed 18/1/22.

[17] https://bizfluent.com/info-8596246-common-problems-inventory-systems.html, Accessed 18/1/22.

[18] https://registry.opendata.aws/amazon-bin-imagery/, Accessed 18/1/22.

[19] https://github.com/silverbottlep/abid_challenge, Accessed 18/1/22.

[20] https://github.com/awslabs/open-data-docs/tree/main/docs/aft-vbi-pds, Accessed 18/1/22.

[21] https://pytorch.org/vision/stable/models.html, Accessed 18/1/22.

[22] https://aft-vbi-pds.s3.amazonaws.com/bin-images/777.jpg, Accessed 22/1/22.

[23] https://aft-vbi-pds.s3.amazonaws.com/metadata/777.json, Accessed 22/1/22.