CAPSTONE PROPOSAL

Inventory Monitoring at Distribution Centers

Udacity AWS Machine Learning Engineer Nanodegree

Author: Ishfaq Malik

1. Domain Background

Inventory management is a critical function in modern distribution centers (DCs). It ensures the correct quantity of goods is available, enabling smooth operations across the supply chain. Traditionally, inventory counting relied on manual processes, which are labor-intensive, error-prone, and not scalable for high-throughput fulfillment centers.

With advancements in automation and computer vision, inventory tracking using image-based methods offers a scalable and efficient solution. During high-demand periods (e.g., global pandemic scenarios), where manual labor is limited, the ability to automate such processes has proven essential to maintaining supply chain integrity.

This project focuses on utilizing machine learning and computer vision to automate item counting in storage bins. The goal is to replace manual inventory checks with a robust model that processes bin images and predicts object counts reliably.

2. Problem Statement

Manual inventory counting is inefficient and error-prone, especially when bins contain various object types, occlusions, or inconsistent layouts. Miscounts can lead to stock discrepancies, delayed shipments, or financial losses. There is a need for a scalable solution to automate this process, thereby reducing human dependency and operational cost.

By leveraging machine learning and computer vision, we aim to develop a model that classifies the number of items in a bin image accurately. This will enhance inventory accuracy and improve the efficiency of warehouse operations.

CAPSTONE PROPOSAL

3. Solution Statement

To address the problem, we propose the development of a supervised deep learning model using Convolutional Neural Networks (CNNs) to classify the number of items in bin images. The solution will involve:

- Training a CNN model (e.g., ResNet50 or EfficientNet) using transfer learning.
- Processing and augmenting the dataset for robust learning.
- Hyperparameter tuning to improve accuracy.
- Deploying the model using AWS SageMaker for scalable inference.

This image classification system will take bin images as input and output the predicted object count class, making it suitable for real-time inventory applications.

4. Datasets and Inputs

Dataset: Amazon Bin Image Dataset

Source: Open-source dataset available through AWS resources.

Description:

- Over 500,000 labeled images of bins from a live Amazon Fulfillment Center.
- Each image has an associated JSON metadata file with:
  - ASINs (Amazon product codes)
  - Object quantity
  - Size, weight, and dimension attributes.

Example:

"EXPECTED_QUANTITY": 3

This rich dataset is ideal for training object-counting classifiers. It also includes variations in object size, bin shape, and occlusions.

CAPSTONE PROPOSAL

5. Benchmark Model

The benchmark model will be based on techniques introduced in:

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

This baseline uses simple CNN architecture with connected component analysis. While it delivers acceptable performance, its ability to generalize is limited due to hand-crafted features and shallow depth.

Our proposed solution aims to outperform this baseline by leveraging deeper architectures and data augmentation techniques.

6. Evaluation Metrics

Since the task is classification-based (predicting discrete item counts), we will use the following metrics:

- Accuracy: Proportion of correctly predicted item counts.
- F1 Score: Balance between precision and recall, especially important in imbalanced classes.
- Confusion Matrix: To visualize misclassifications.
- Mean Absolute Error (MAE): Secondary metric to understand how far off predictions are numerically.

7. Workflow

The end-to-end workflow is outlined below:

Step 1: Data Preparation
- Download and unzip dataset from AWS.
- Parse image and metadata JSON pairs.
- Resize images, normalize pixel values, augment data.

CAPSTONE PROPOSAL

Step 2: Model Development

- Choose a CNN model (e.g., ResNet50).
- Apply transfer learning with a pre-trained backbone.
- Fine-tune using training data.
- Validate using hold-out dataset.

Step 3: Training and Tuning

- Train model using AWS SageMaker.
- Tune hyperparameters using SageMaker hyperparameter tuning jobs.
- Store model artifacts in S3.

Step 4: Model Deployment

- Deploy trained model on SageMaker Endpoint.
- Run predictions on test samples.
- Evaluate metrics.

Step 5: Inference and Scaling

- Implement batch or real-time inference.
- Explore integration with a warehouse monitoring dashboard (optional stretch goal).