INVENTORY MONITORING AT DISTRIBUTION CENTER PROJECT PROPOSAL

Domain Background

The purpose of the project is to build a model that can classify accurately the number of objects in a bin. This system is used to track inventory and ensure that delivery consignment have the correct number of items.

In big retail facilities such as Walmart and Amazon, the process of inventory and monitoring is very labor intensive, involve high risk, require multiple people and it is prone to error. The process can be improved by automation whereby a vision-based method using deep learning to count number of items in a bin before delivery.

Reference:

1. https://www.researchgate.net/publication/342529859_Inventory_management_using_Machine_Learning :In this research paper, a major requirement for small/medium-sized businesses is Inventory Management since a lot of money and skilled labor has to be invested to do so. E-commerce giants use Machine Learning models to maintain their inventory based on demand for a particular item. This is related to the inventory monitoring at distribution center
2. https://ieeexplore.ieee.org/document/9268394 :This research paper presents a vision-based method using a deep convolutional neural network to classify different items stored in a warehouse for the purpose of inventory management. The proposed method uses residual learning and employs ResNet-50 network architecture. It achieves a high accuracy of 98.94% on the dataset

Problem Statement

The distribution centers need to ensure the correct number of items are placed in the consignments for delivery. Since robots is used as part of their operations to move object, a model is needed to be built. Which can be integrated to the robot for it to count objects correctly as it moves it to the bin. This system is used to track inventory and ensure consignments have correct number of items.

Datasets and Inputs

The Amazon Bin Image Dataset 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. For each image there is a metadata file containing information about the image like the number of objects, it's dimension and the type of object.

The total number of images is 535,234 with an average quantity in a bin of 5.1 and number of object categories of459,476. Typical images in the dataset have 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 is dependent 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.

A typical example of a bin image and its metadata file is shown below;


```
{
    "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
}
```


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".

The dataset to be used is the subset provided by Udacity

Images are located in the bin-images directory, and metadata for each image is located in the metadatadirectory. Images and their associated metadata share simple numerical unique identifiers.

- How are the images formatted?

  There are two set of inputs for the model training

  1. Images for the model, which is available in the source as JPEG file
  2. JSON format with meta data for the image

- Are the dimensions consistent?

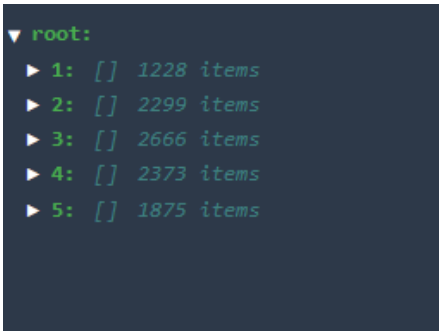
  The dimension of the images is consistent

- Are there color layers?

  There are color layers in the images

- How are the dataset classes balanced?

  The dataset classes are not balanced. The picture below shows the count of dataset for each class



Data available at: https://registry.opendata.aws/amazon-bin-imagery/

Solution Statement

The approach to be used to tackle this problem is to build a model that can classify number of objects in a bin. A model type and architecture will be used to train the model. A pre-trained convolution neural network is a possible choice for the model architecture.

Benchmark Model

The benchmark model will be based on a different pre-trained model, VGG-16. This will serve as threshold/goal I am trying to beat for my project solution

Evaluation Metric

The evaluation metric that will be used for this task is the cross-entropy loss function. Cross Entropy is a good loss function for Classification Problems, because it minimizes the distance between two probability distributions - predicted and actual. Other evaluation metric that can be considered for the classification task includes: Accuracy and F1 score

Project Design

Intended workflow:

1. Upload Training Data: I will upload the training data to an S3 bucket.
2. Read and preprocess data: I will read, load and preprocess my training, testing and validation data.
    - The preprocessing task will include re-sizing the images to have a uniform size while inputting the data into our model
    - The data will be divided into train, test and validation set
3. Model Training Script: I will write a script to train a model on the dataset.
    - I plan to use a pretrained model of Resnet for training my model. The pretrained model will beResNet34 model with a three Fully connected Linear NN layers in order to flatten the result before classifying it.
4. Train in SageMaker:
    - I will use SageMaker to run the training script and train my model
    - I will install necessary dependencies
    - I will setup the training estimator
    - I will submit job
5. I will evaluate my model
6. I will improve my project by hyper parameter tuning and multi-instance training
7. My overall implementation will be tune by performing different hyperparameter tuning to help pick best hyperparameters that helps increase the accuracy of the classifying model
8. Finally, I will create a read me file that explains the project