Proposal for Developing model for Inventory Monitoring at Distribution Centers

Vinamra Sareen , vsareen24@gmail.com

Domain Background

Distribution centres often use robots to move objects as a part of their operations. Objects are carried in bins which can contain multiple objects. Sometimes items get misplaced and to prevent mismatch while maintaining inventory records an efficient system can be put in place.

In this project, we will have to build a model that can count the number of objects in each bin. Creating a system like this can be used to track inventory and make sure that delivery consignments have the correct number of items.

Problem Statement

As described above in the domain, robots carry out the job of moving objects in distribution centres and items might get misplaced. To tackle the problem of mismanagement we are introducing an object counting task. Developing a solution for this problem also has some real world applications such as crowd surveillance, traffic monitoring, wildlife conservation and inventory management.

For this task, we have images of bin and we know each bin can contain objects in range of 1-5. If we develop a model that can take a picture of the bin and accurately predict the number of objects present in the bin, we can solve a crucial problem in the inventory management system.

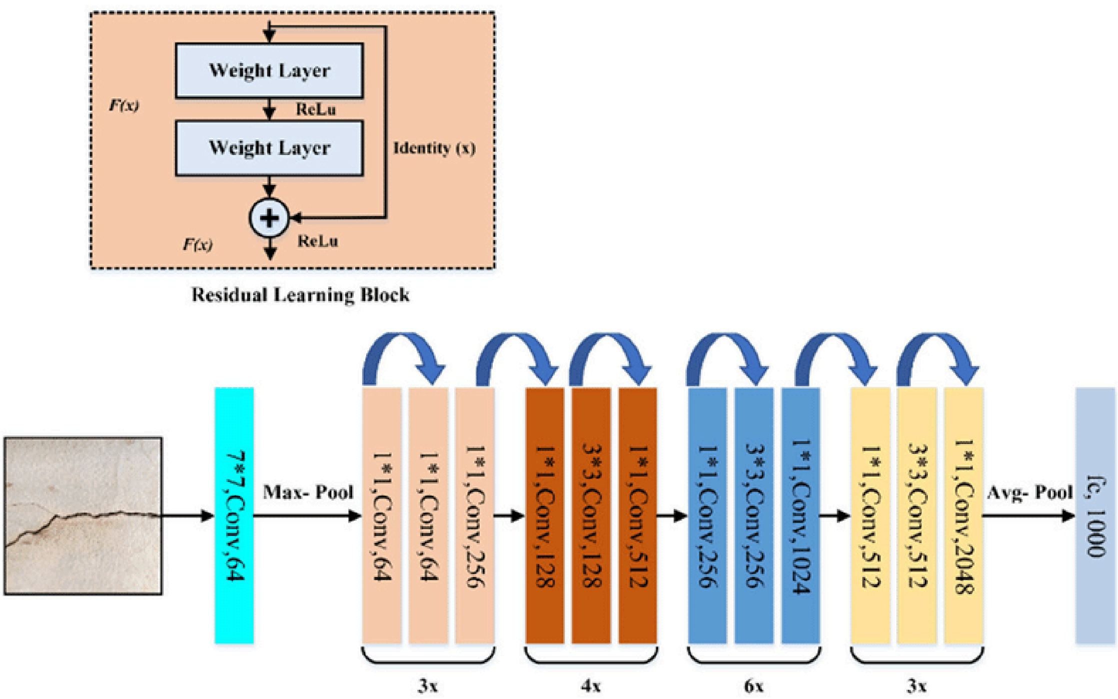
Solution Statement

This problem can be solved using a Convolutional Neural Network which is a state of art technique for image recognition tasks. We will leverage pretrained models to solve the problem.

For this task, we will be using a ResNet50 model, which performs well for small labelled datasets and does not require much computation power. It will be a multiclass classification task where given an image, we have to identify labels from 1-5, each number representing an individual class.

An detail on Resnet50 model architecture:

Comprising 50 layers, it addresses the challenge of vanishing gradients by introducing skip connections that facilitate direct flow of gradients through the network. The architecture consists of residual blocks, each containing convolutional layers and shortcut connections. These connections enable the network to learn residual functions, enhancing training efficiency and performance. ResNet-50 employs bottleneck blocks to reduce computational complexity while maintaining representational capacity. Widely adopted in computer vision applications, ResNet-50 has achieved state-of-the-art results on benchmarks such as ImageNet, demonstrating its versatility and effectiveness in various tasks.



We will be using AWS cloud platform for following services:

1. S3: To store the datasets such as train, valid, test
2. Sagemaker Studio: We will be using sagemaker studio to create our project, run notebook instances, train and deploy models.

Datasets and Input

We can use the data provided by amazon for our model training and inference. The info about dataset can be found here:

[Amazon Bin Image Dataset - Registry of Open Data on AWS] ("Amazon Bin Image Dataset")

A bit of overview of data:

The Amazon Bin Image Dataset contains over 500,000 images and metadata from bins of a pod in an operating Amazon Fulfilment Centre. The bin images in this dataset are captured as robot units carry pods as part of normal Amazon Fulfilment Centre operations.

Example of metadata json file:

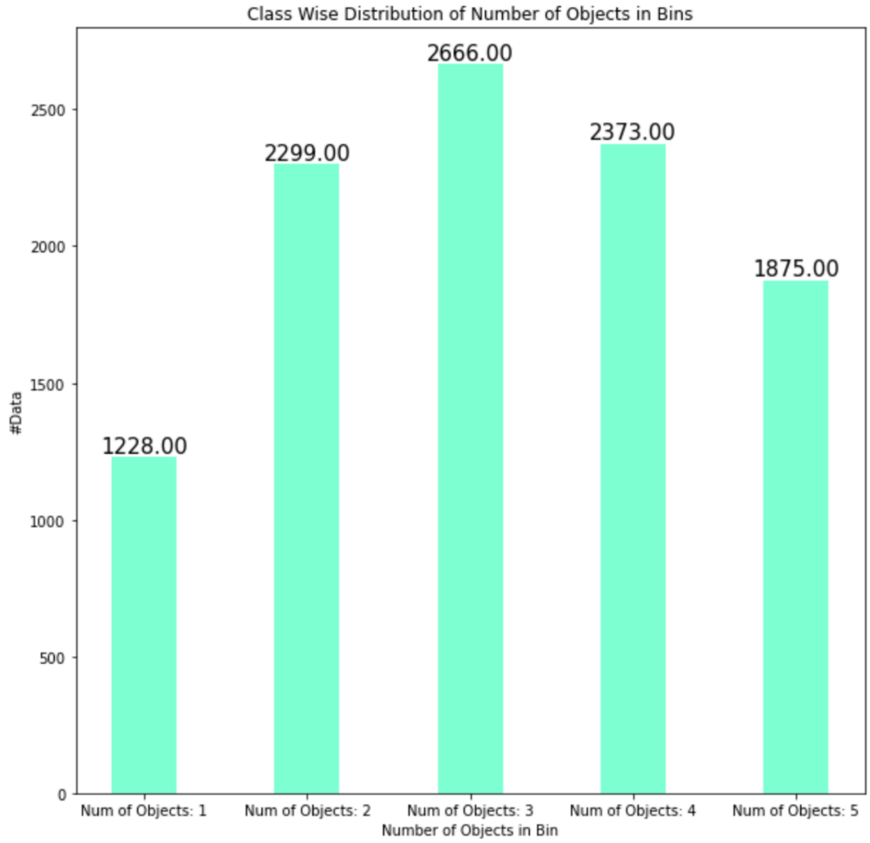

```
{
  "BIN_FCSKU_DATA": {
    "B000A8C5QE": {
      "asin": "B000A8C5QE",
      "height": {
        "unit": "IN",
        "value": 4.200000000000001
      },
      "length": {
        "unit": "IN",
        "value": 4.7
      },
      "name": "MSR PocketRocket Stove",
      "quantity": 1,
      "weight": {
        "unit": "pounds",
        "value": 0.45
      },
      "width": {
        "unit": "IN",
        "value": 4.4
      }
    },
    "B0064LIWVS": {
      "asin": "B0064LIWVS",
      "height": {
        "unit": "IN",
        "value": 1.2
```

```
            },
            "length": {
                "unit": "IN",
                "value": 5.799999999999999
            },
            "name": "Applied Nutrition Liquid Collagen Skin Revitalization, 10 Count 3.35 Fl Ounce",
            "quantity": 1,
            "weight": {
                "unit": "pounds",
                "value": 0.3499999999999999
            },
            "width": {
                "unit": "IN",
                "value": 4.7
            }
        }
    },
    "EXPECTED_QUANTITY": 2,
    "image_fname": "523.jpg"
}
```


Example of image for above metadata file:

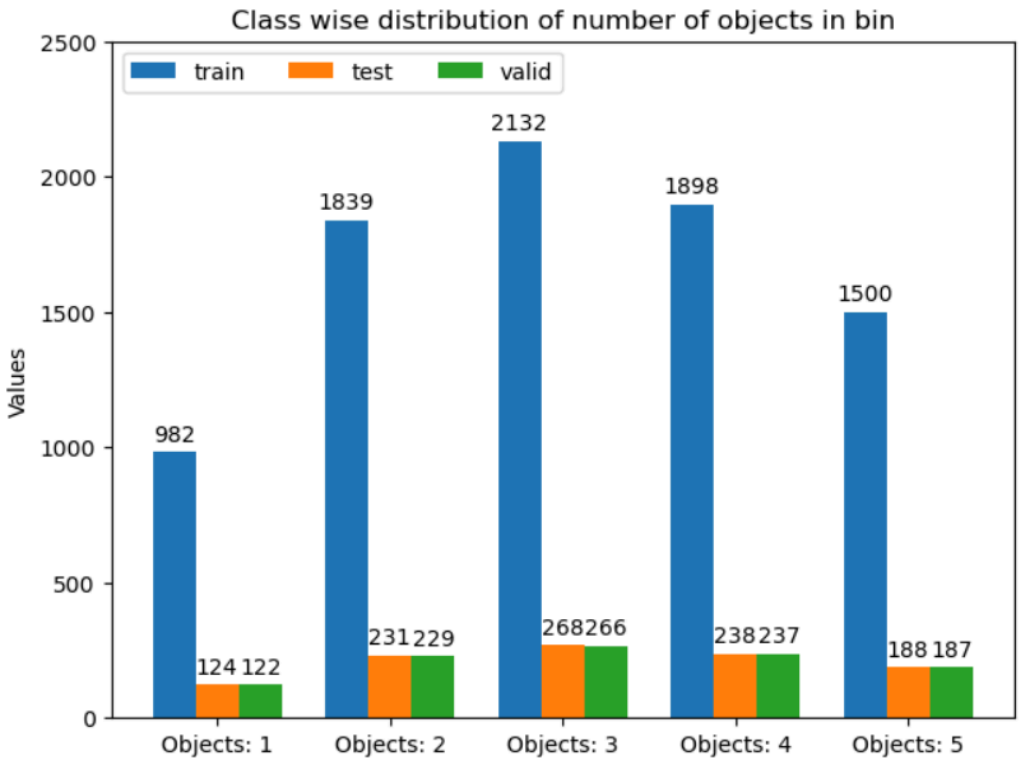
We have been provided with a **file_list.json (Udacity)** that has about 10,441 image json objects. Each being grouped with keys 1-5, representing labels for images. We will be preferring the **file_list.json (Udacity)** over the whole dataset of 500k+

records as it will be computationally expensive. Below is class wise distribution of



images.

The below image represents splitting of the whole dataset into train, test and validation dataset wrt to labels.



Benchmark Model

Benchmarks are used to compare performance of models. It can help us improve our model and see if it is close or how much far away from the established accuracy achieved so far in the same domain/problem.

The github repo from amazon challenge has an accuracy provided for the same problem, the accuracy specified there is about 55%.

[GitHub - silverbottlep/abid_challenge: Amazon Bin Image Dataset Challenge](#) (silverbottlep)

The above challenge was performed on a whole dataset of 535,234 images.

We have a benchmark accuracy of 55% achieved so far, if we are able to achieve anything above, it will be great.

***Note**: But keep in mind the quantity of the dataset as we are using just 10,441 images which will definitely affect the performance.*

Evaluation Metrics

The below metric will be used for this task:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Project Design

We are going to discuss practices we have learned throughout this program and apply them on this project.

We will follow the below steps:

1. Data Analysis and Preparation
    a. Downloading the dataset
    b. Performing Preprocessing step
    c. Upload to S3 bucket as Sagemaker only uses s3.
2. Split Data into training, validation and test dataset.
3. Choose a model and I will be using a pretrained model Resnet50.
4. Performing Hyperparameter tuning to find good values to begin the job, training for about 5 epochs.
5. Based on best values do training on a dataset for large epoch values but not more than 20 keeping in mind cost consumption.
6. Do Profiling and analysis from the report to understand what steps can be taken further to improve the model.
7. Deploy the model.
8. Perform inference and delete the endpoint once finished.

Works Cited

"Amazon Bin Image Dataset." *Registry of Open Data on AWS*, https://registry.opendata.aws/amazon-bin-imagery/. Accessed 29 February 2024.

silverbottlep. "silverbottlep/abid_challenge: Amazon Bin Image Dataset Challenge." *GitHub*, https://github.com/silverbottlep/abid_challenge. Accessed 29 February 2024.

Udacity. *Udacity*,

https://github.com/udacity/nd009t-capstone-starter/blob/master/starter/file_list.json. Accessed 29 February 2024.