Project: Inventory Monitoring at Distribution Centers

Domain Background

Artificial Intelligence has been integrated into almost every aspect of our daily lives, specially when it comes to commerce. Natural Language Processing (NLP) gave us advanced chat bots and translation tools that revolutionized the way customers interact with companies, while Computer Vision is responsible for skills like object identification and pattern recognition, that together can reduce production deviations and human errors in ways not imagined some decades ago [1].

In the early days of Computer Vision, in the late 1960s, the motivation for the development of the technology was to serve as a stepping stone leading to provide systems with intelligent behaviour. The first attempt to do so was attaching a camera to a computer and having it describe what it could perceive. [2]

Fortunately, Computer Vision has advanced a lot since then. One of the results of Computer Vision can be seen in robots used in stock management. This trend tends to be a part of logistic planning and grows constantly in distribution centers' daily operations.

Goods are moved by robots in bins that can contain from one to several items. This project takes that into consideration and uses Computer Vision to propose a usage of Machine Learning (ML) to optimize operations involving robots.

A robot that is able to count the number of items in each bin it moves would improve stock management. And the ML engineer with experience with that kind on computer vision skill can be an asset for companies.

Problem Statement

The problem I intend to solve in this project is the optimization of storage management applying Computer Vision to robots used in storage operations.

The problem can be broken down into categories:

Quantification

The quantitative aspect of my problem is defined by the number of items present in each image that are identified by the ML model. These quantities of items are the classes in which the images will be classified.

Measurability

One way to provide a measurable aspect to this problem is to take into consideration the difference between the number of items shown in an image and the quantity accused by the ML model.

Replicability

This problem can be tackled by anyone that has access to a large number of images where different number of items are portrayed. After labelling these images, the same method proposed by my in this work can be used.

Solution Statement

The chosen method to solve the proposed problem is to give robots, and consequently the controlling system, the ability to identify the number of items being transported in bins used for storage.

It is assumed that this capability could reduce time and possible mistakes in storage management processes.

The method I intend to use to develop my solution for this problem is to use a pretrained Computer Vision model called ResNet and finetune it to best fit the images in my dataset.

The platform used for developing this solution will be Amazon Web Services (AWS), more specifically Amazon Sagemaker. The notebook and python scripts will be run in its instances.

As observed during the Problem Statement section, the solution can also be detailed into categories:

Quantification

The quantitative aspect of the solution is summarized by the number images that are correctly classified regarding the number of items being portrayed in each of the images.

Measurability

In order to provide measurability to the solution, the concept of accuracy is used for evaluating the results output by the ML classification model created in this project. That is, the ratio between corrected classified images and all images examined.

Replicability

The dataset used for the development of the solution can be easily downloaded and all of the steps taken during the execution of this project will be recorded in a jupyter notebook and a python training script. Additionally, all the packages used in this project are already public and commonly used by the community. Therefore, anyone interested in replicating the solution should be able to do so without problems.

Datasets and Inputs

The dataset used for this project is the Amazon Bin Image Dataset. The dataset can be found here.

This dataset holds 500,000 images and metadata from bins of a pod in an operating Amazon Center. The images were captured as the bins were being transported by robots.

The images are not divided into labelled folders. In order to use them a prior division of train images would be necessary using the metadata JSON files by Amazon in the dataset S3 folder. That step is not necessary for this project since Udacity has kindly provided a JSON file that labelled 10,441 images into 5 different folders ranging from 1 item to 5 items in each picture. Therefore, it was decided that the files to be used for training, validation and testing would be the ones listed in the forementioned JSON file.

This reduced dataset is also convenient since the AWS resources used for training in this project are limited.

Benchmark Model

Since I intend to modify the fully connected layers and the default hyperparameters of the pretrained ResNet50 network, the benchmark model for this project will be the original ResNet50 model as provided by the torchvision API.

The ResNet architecture was a response to a problem that appeared as Convolutional Neural Networks (CNNs) were rising in popularity in the 2010s: as more studies were performed where the number on convolutional layers was being gradually increased, it was observed that the training and testing errors were growing instead of diminishing, as observed in the following Figure.

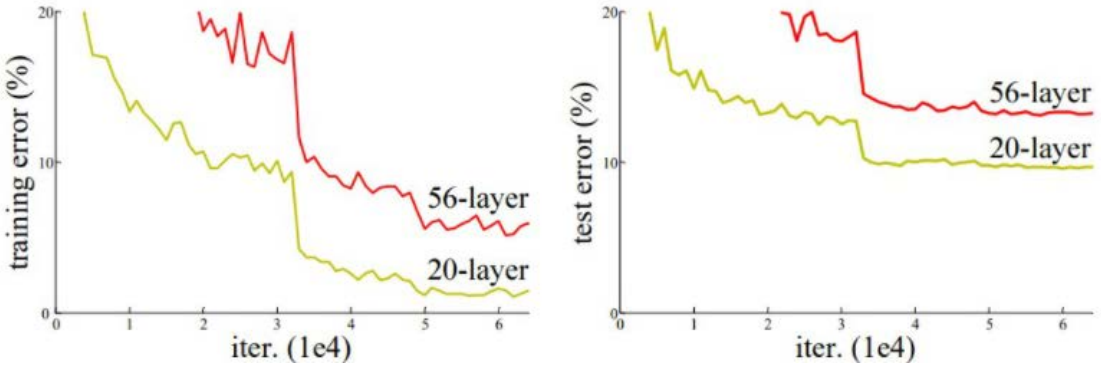

Figure 1 - Performance Comparison between CNNs with different number of layers (source: https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning)

The presented Figure also dismisses the idea of overfitting since it is not just the testing error that is greater for a deeper network, the same occurrence can be seen in training.

Research has found out that this was caused by vanishing/exploding gradients, whereas the gradients are back propagated to earlier layers, the repeated multiplications performed could turn said gradient either too small (vanishing gradient) or too big (exploding gradient).

The ResNet introduced the concept of the Residual Block, where the technique called skip connections allowed skipping training from a few layers and connecting directly to the output. Therefore, instead of learning from the underlying mapping, the network would benefit from fitting the residual mapping.

The advantage of this approach is to allow skipping by regularization of any layer that would hurt the network performance by generating vanishing/exploding gradients. [3, 4]

The next Figure shows ResNet architecture compared to previous networks that inspired it.

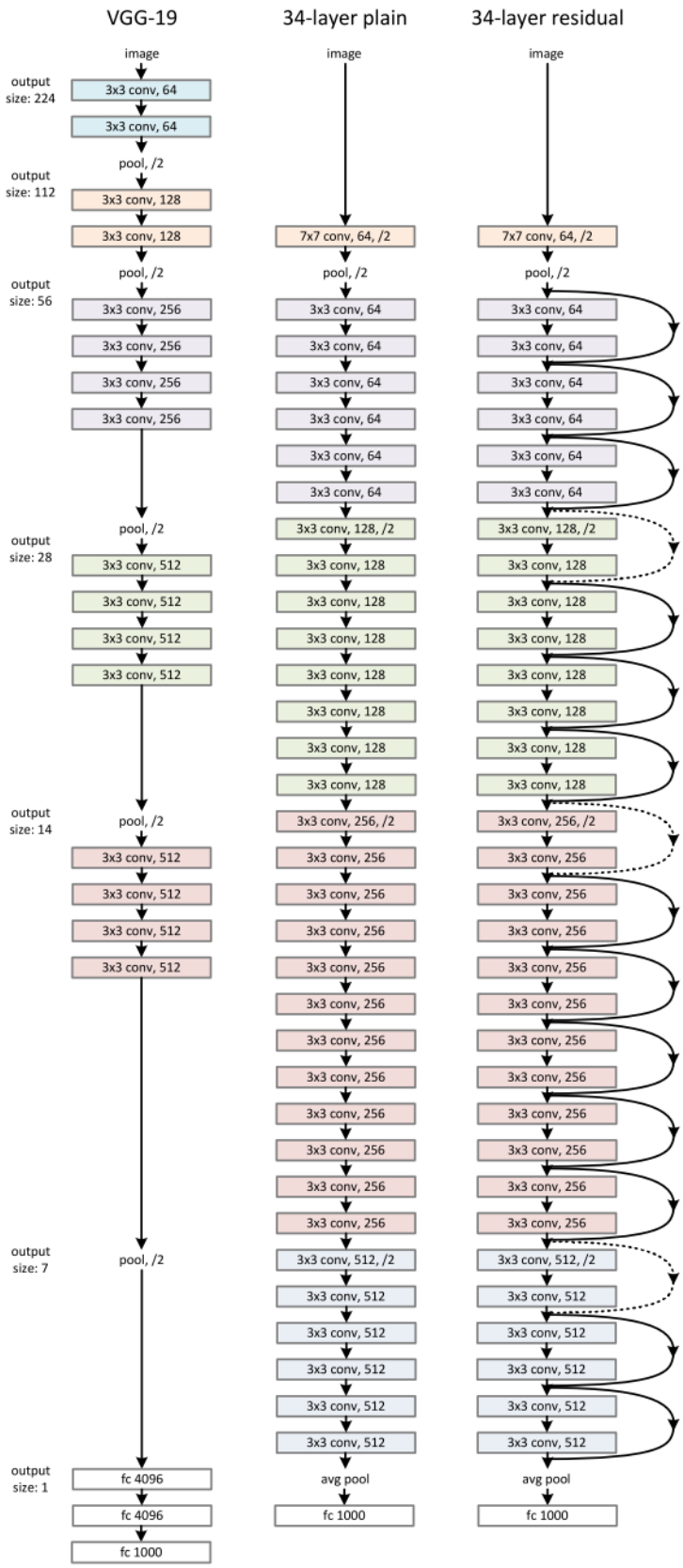

Figure 2 - The ResNet architecture (source: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035)

The paper presenting the original ResNet models can be found here.

This project will use the convolutional portion of ResNet and finetune its last section (the fully connected layers) as an attempt to optimize accuracy for the Amazon Bin Image Dataset.

Evaluation Metrics

The main evaluation metric for comparison between the benchmark and the solution models will be result accuracy, that is, the number of correct inferences provided by the model divided by the total number of inferences made.

This metric makes sense since it encompasses the main goal of the project: to obtain a model capable of identifying the number of items in the images in the best possible way.

Project Design

My planned workflow for the resolution of this problem is:

- Creation of an ETL Pipeline in order to obtain data from the Amazon Bin Dataset, process and load it to an S3 bucket;
- Train a ML vision model using the database previously created;
- Using AWS tools, make sure the best practices are being employed and the processes are cost effective.

The ETL Pipeline will load data, separate it into train, validation and test sets and load them into separate S3 folders.

The ML model responsible for identifying the number of items in each image will be based on a pretrained ResNet50 model from torchvision that will have its fully connected layers finetuned looking to obtain better results.

After finetuning, an attempt of model improvement via hyperparameter optimization will be performed.

Finally, after arriving at a satisfactory model, the training script will be modified in order to insert debugging and profiling mechanisms. That final step aims to provide better code and instance monitoring.

All steps will be recorded in a jupyter notebook and a python training script to ensure replicability.

A workflow representing the planned steps is shown in the following Figure.

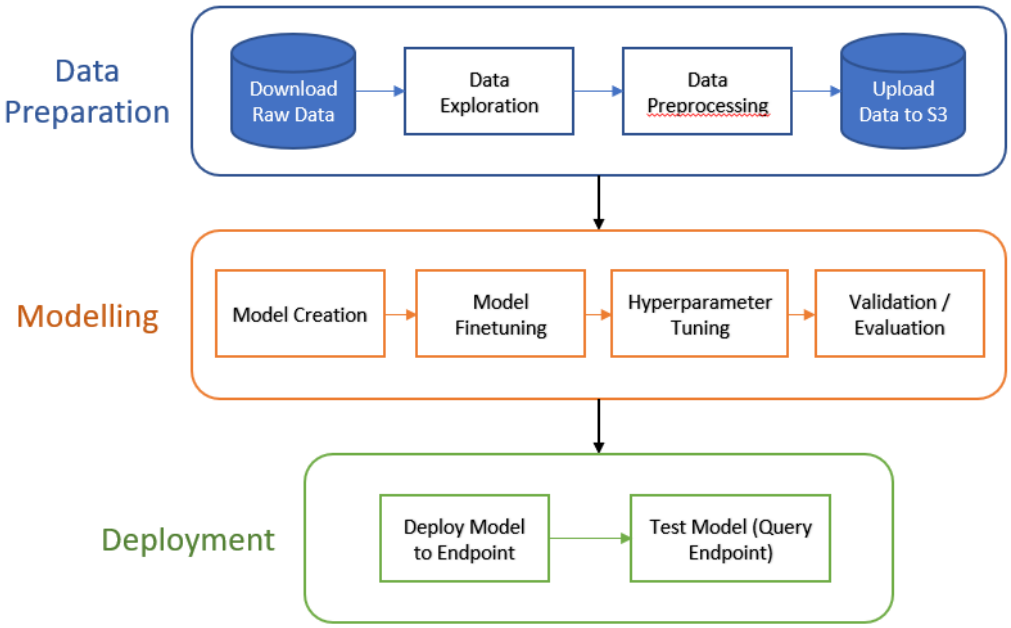

Figure 3 - Project Workflow

References

[1] *How Artificial Intelligence Revolutionized Computer Vision: A Brief History*. (2019, May 16). Motion Metrics. https://www.motionmetrics.com/how-artificial-intelligence-revolutionized-computer-vision-a-brief-history/#:~:text=Computer%20vision%20began%20in%20earnest,that%20could%20transform%20the%20world.

[2] *Computer Vision*. (n.d.). Wikipedia. https://en.wikipedia.org/wiki/Computer_vision

[3] *Residual Networks (ResNet) – Deep Learning*. (2022, January 27). Geeks for Geeks. https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/

[4] Feng, V. *An Overview of ResNet and its Variants*. (2017, July 15). Towards Data Science. https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035