Capstone project

Machine Learning Engineer Nanodegree

Inventory Monitoring at Distribution Centers

Domain background:

In the last few decades, advancements in machine learning have led to start-of-the-art approaches for many challenging problems like many complex scenarios in robotics where the techniques based on first-principles may fail to capture the complexity of unstructured real-world environments. One of complex scenarios in robotics is how can a robot be programmed to pick up objects with an infinite number of variations in size, shape, color, and texture [1]. Inventory Monitoring at Distribution Centers project often use robots to move objects as a part of their operations but our robots task in this project is to classify the object class in each pin. So this project is underlying Robotics which is one of Machine Learning applications.

Problem statement:

The goal is to build a model that can count or classify the number of objects in each bin into one of 5 classes; class 1, class 2, class 3, class 4 and class 5. The tasks involved are:

1- Download and process a subset of Amazon Bin Image Dataset and upload it to S3 bucket.
2- Train the model using AWS sagemaker.
3- Fine tune the model to improve its performance by using the extracted best hyperparameters of hyperparameters tuning.
4- Perform a prediction on the model.

Datasets and inputs:

The Amazon Bin Image Dataset is the used dataset in training, validation and testing process of the model that is used to perform object classification. Amazon Bin Image Dataset contains over 500,000 images, sample given in Fig.1, 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 [2]. Amazon Bin Image Datasets are available in the aft-vbi-pds S3 bucket in the us-east-1 AWS Region[3].

A small subset of Amazon Bin Image Dataset will be downloaded and used in this project as recommended. The subset data is about 10446 bin images divided into 5 classes as shown in Fig.2 it's clear that dataset amount per class is not equal.

Fig1 bin object image.

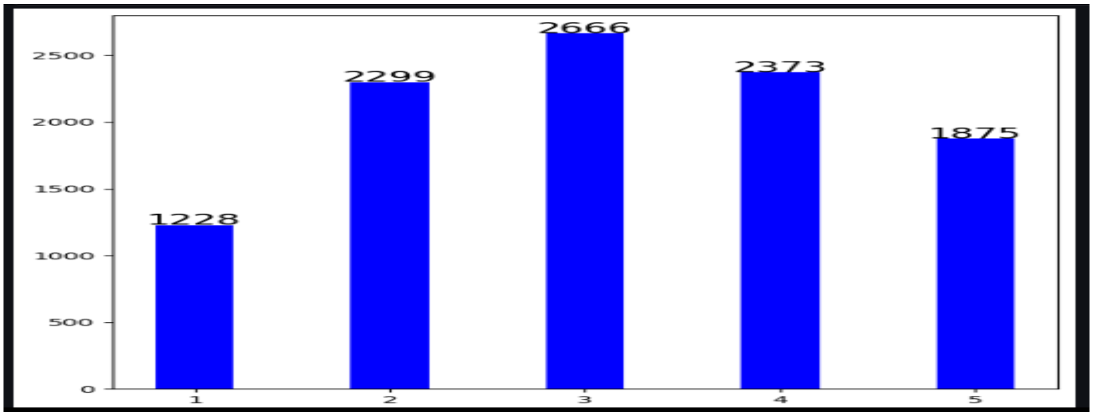

Fig.2 *Subset* of Amazon Bin Image Dataset.

The data subset will be randomly split over each individual object class as shown in Fig.3 and divided into training, validation and testing datasets with 80%, 5% and 15% percentage of the whole subset respectively. And this datasets will be uploaded to S3 bucket so sagemaker be able to use it to train, valid and test models.

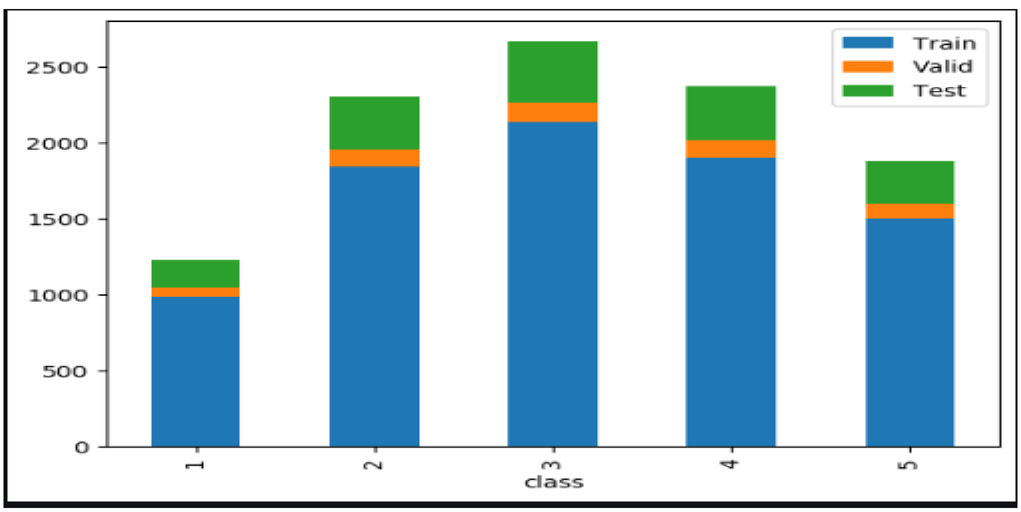

Fig3. Divided dataset; training, validation and testing dataset

Solution statement:

The project task is to classify an object image into one of 5 classes; class 1, class 2, class 3, class 4 or class 5. So chose a model of one of models with large neural network architecture where they show excellent performance when dealing with image classification tasks and then train the chosen model using the provided dataset. To guarantee high accuracy of object image classification the chosen model will be fine-tuned.

Benchmark model:

Since distribution centers are considered to be part of a supply chain network, I am going to use one of algorithms used for Supply Chain, Convolution Neural Networks (CNNs) are a type of algorithm that usually deals with image recognition. CNN is one of the most widely used algorithms in supply chain management [4]. CNNs is used to recognize the category of a given image. It's quite handy in the supply chain, as it can classify different products in an instance, and separate them accordingly [4].

Benchmark model:

- The benchmark model used is CNN model training_a_cnn_solution.py given at Udacity course>Common Model Architecture Types and Fine-Tuning> Exercise: Fine-Tuning a CNN Model [5].
- Model is finetuned by pre-trained model resnet18 model.
- The model is used as its but with some modification in it, create create_data_loaders () function and main () function.
- The used hyperparameters to train the model are same as the ones in the original model.
- The model will be trained using AWS sagemaker studio through benchmarak_model.ipynb file.

Evaluation metrics:

Test loss and test accuracy are the metrics used to measure the model performance which directly reflected on how well the used solution. Both matrices have a vice versa relationship, low test loss means high test accuracy.

Accuracy = $\frac{model\ prediction\ -actual\ prediction}{tolal\ dataset\ size}$

Loss = $loss\_criterion(model\ prediction,\ actual\ prediction)$

$loss\_criterion$, this criterion computes the cross entropy loss between input logits and target [6].

The *target* that this criterion expects should contain either:

- Class indices in the range $[0, C)$ where C is the number of classes; if *ignore_index* is specified, this loss also accepts this class index (this index may not necessarily be in the class range). The unreduced (i.e. with `reduction` set to `'none'`) loss for this case can be described as:

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \cdot 1\{y_n \neq \text{ignore}$$

Where x is the input, y is the target, w is the weight, C is the number of classes, and N spans the minibatch dimension as well as $1,\ldots,\ d1,\ldots,dk$ for the *K*-dimensional case[6]

Project design:

AWS SageMaker will be used to build this project and the solution steps will be as follow:

1- Download and Preprocessing the dataset:

- Download a subset of Amazon Bin Image Dataset.
- Divide dataset into training, validation and testing dataset.
- Upload the data to S3 bucket.

2- Train the model:

- Write a train.py script to train a model by implementing specific functions.
- Fine tune the hyperparameters.
- Train the model using best tuned hyperparameters by running debug_model.py.
- Monitor model performance using model debugging and profiling.

3- Perform a prediction on the model:

- Deploy the model to an endpoint.
- Test the endpoint using random set of object images.

References:

1- https://web.stanford.edu/class/cs237b/pdfs/lecture/lecture_1.pdf
2- https://registry.opendata.aws/amazon-bin-imagery/
3- https://github.com/awslabs/open-data-docs/tree/main/docs/aft-vbi-pds
4- https://neptune.ai/blog/use-cases-algorithms-tools-and-example-implementations-of-machine-learning-in-supply-chain
5- https://9f84d623cee9420aae7fbcf3e3c2bfea.udacity-student-workspaces.com/edit/finetune_a_cnn_solution.py
6- https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html