Capstone Project

Inventory Monitoring at Amazon Distribution Centers

Eyup Acar
April 19, 2023

1. Domain Background:

The Amazon Bin Image Dataset is a comprehensive collection of images of products that are stored in Amazon's fulfillment centers. These images are captured by mobile robots that move around the fulfillment centers, taking pictures of various product packages as they go.

The primary objective of this dataset is to provide an opportunity to develop an object recognition system. This system would be designed to identify various products based on images of their packaging. This would be of great benefit to Amazon's fulfillment centers as it would help automate their processes, making them more efficient and accurate.

By using this dataset, we can develop a machine learning model that can accurately recognize and classify different products based on their images. This would help Amazon reduce the amount of time and effort required to sort and handle packages, as well as reduce the potential for errors.

Overall, the Amazon Bin Image Dataset provides an excellent opportunity to develop machine learning models for object recognition, which has numerous real-world applications beyond Amazon's fulfillment centers.

2. Problem Statement:

The problem to be investigated is to develop an accurate object recognition system that can identify different products based on images of their packaging, using machine learning algorithms and techniques. This problem is important for improving the efficiency of Amazon's fulfillment centers, as it can reduce errors in the identification and handling of products, leading to faster and more accurate delivery to customers.

3. Problem Solution and Evaluation Metrics

Algorithm

To fine-tune the Resnet50 pre-trained CNN, a hyperparameter optimization technique will be utilized to avoid overfitting and produce the desired results. The performance of the

model is highly dependent on hyperparameters such as batch size and learning rate, and choosing suboptimal values can lead to poor results. Thus, it is crucial to use a hyperparameter optimization technique to find the best combination of hyperparameters that will result in optimal performance on the given dataset. The training process will be evaluated using standard metrics such as precision (accuracy) and cross-entropy loss function to ensure repeatability and facilitate further improvements.

Evaluation Metrics

The Cross Entropy Loss function is a commonly used loss function in classification tasks, which measures the difference between the predicted probability distribution and the true probability distribution of the target class labels. By minimizing the loss function, we can optimize the model parameters to make accurate predictions on the given dataset.

Precision is a metric that measures the proportion of true positive predictions among all positive predictions. It is a useful metric when the cost of false positives is high. In the context of the proposed solution, precision will be used to evaluate how well the model can correctly identify products based on their packaging images.

The effectiveness of the training process will be assessed using these two parameters. *A high accuracy value for the Cross Entropy Loss function and precision means that the model is performing well and accurately identifying the products based on their packaging images.* The repeatability of these parameters will enable us to compare the performance of different models and track the progress of the project over time.

Benchmark

ABID Challenge could be a good benchmark to consider for the project. The challenge provides a pre-defined evaluation metric that measures the accuracy of object counting in the bins. (55.67%) The metric calculates the percentage of bins for which the absolute difference between the predicted and ground truth counts is within a specified tolerance.

By using this benchmark, comparison of the performance of the model with the results reported by other researchers and participants in the challenge can be made. It can also provide a standardized way to evaluate the performance of the model, which can help in identifying areas for improvement and optimization.

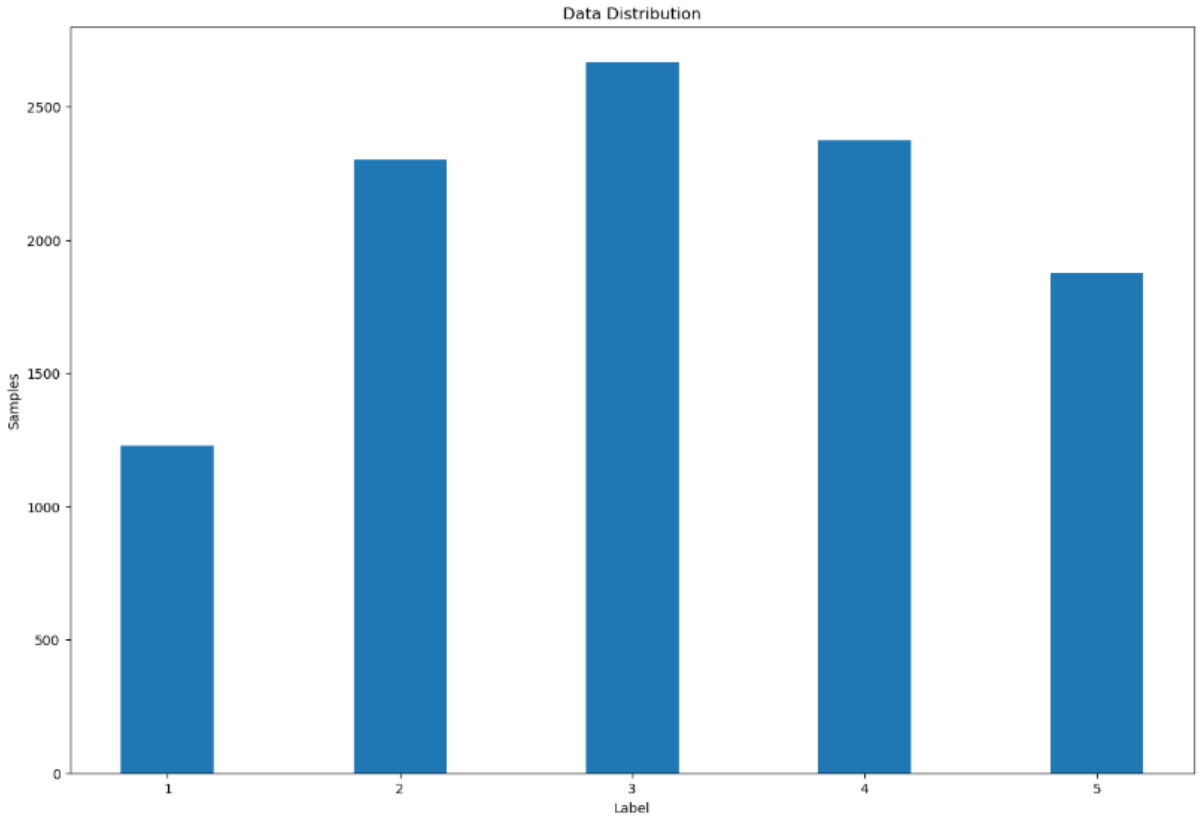
4. Data Exploration

The Amazon Bin Image Dataset is a large collection of images and metadata from bins of a pod in an operating Amazon Fulfillment Center. The dataset contains over 500,000 images of products stored in Amazon fulfillment centers, captured by mobile robots as they travel around the facilities. Each image in the dataset is labeled with information about the bin in which the product was stored, including the number of items inside the bin, the bin's location, and the

time at which the image was captured. Here the number of items in each bin is aimed to be predicted through machine learning algorithms.

4.1 **Data Distribution**

Number of files in each folder is plotted below.



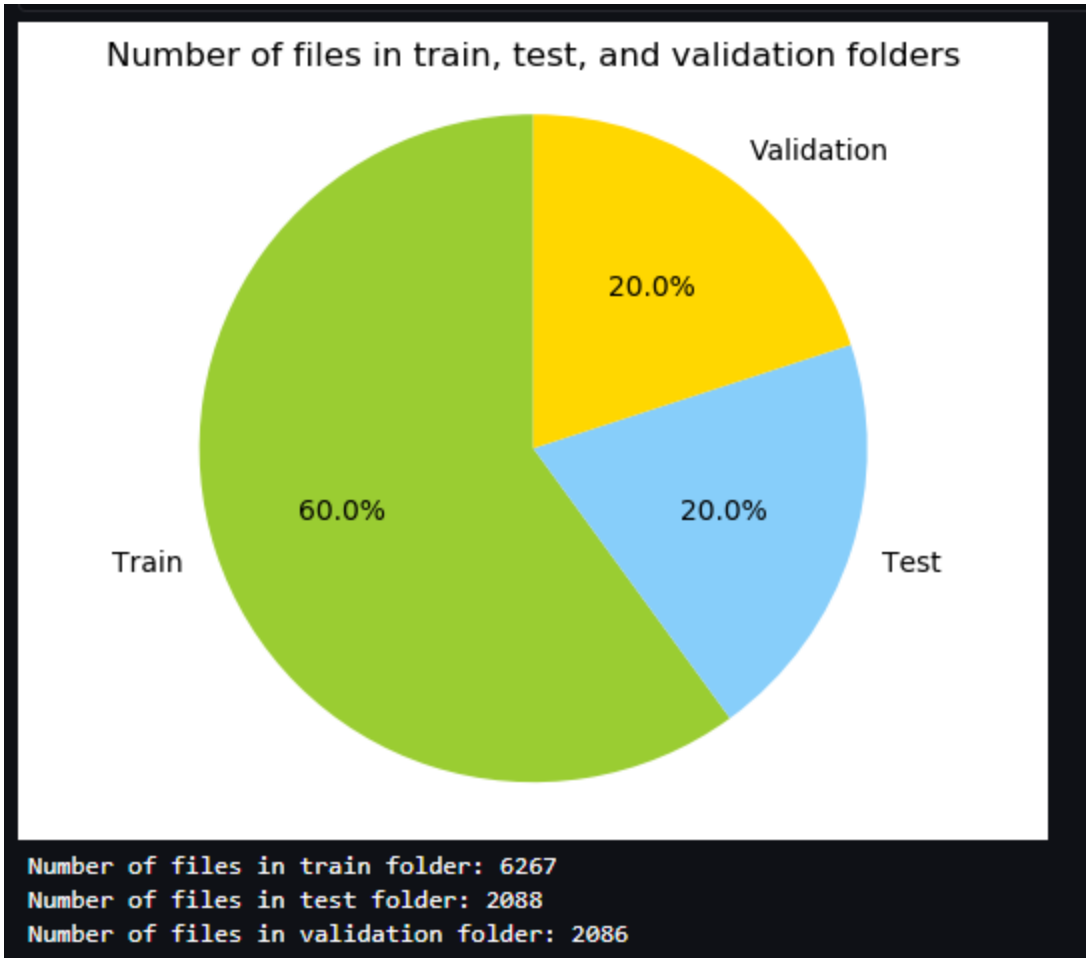
Distribution of files in each folder

4.2 **Image Resize**

ResNet50 is a convolutional neural network architecture that takes input images of a fixed size, usually 224x224 or 256x256 pixels. Images in the dataset are about 600px x 600px. so we need to resize them to get consistent results.

4.3 **Data Split**

To make the training process more efficient, only a subset of around 50,000 images labeled with the number of items in each bin will be used for hyperparameter tuning. For model training the whole dataset will be used. The dataset will be post-processed and divided into train, validation, and test subsets with a 60/20/20% split ratio. The number of images per class will be balanced to ensure that there are similar numbers of test, train, and validation images for each class.



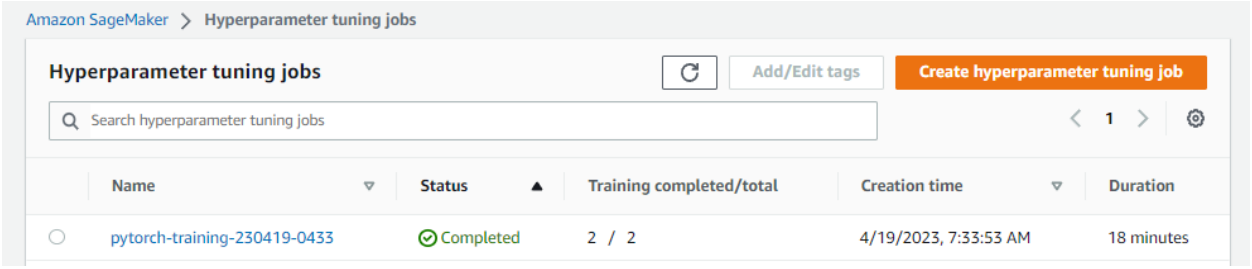
5. Hyperparameter Tuning:

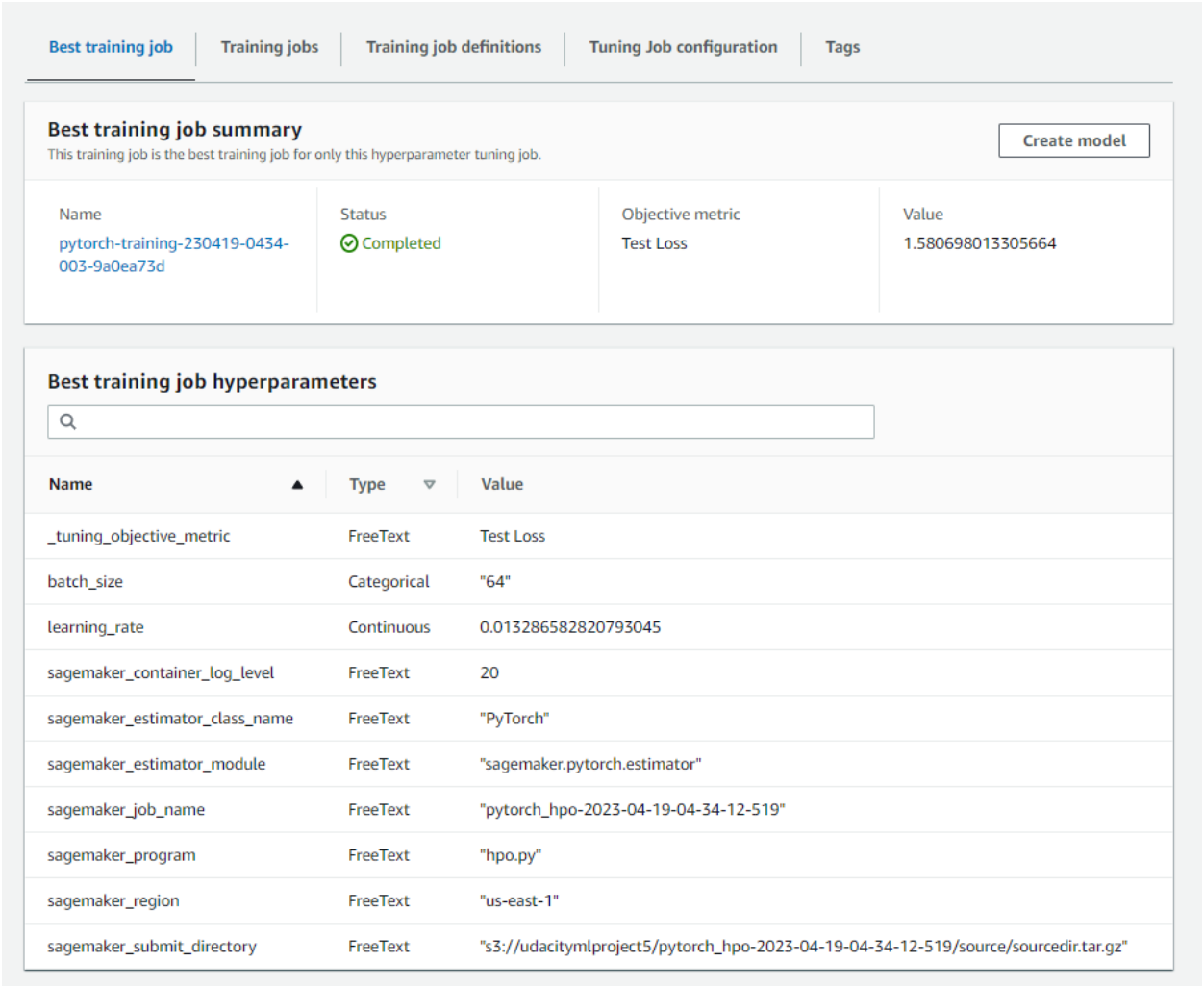
In order to identify the optimal hyperparameters for the training process, a hyperparameter tuning phase is necessary. The AWS Sagemaker Hyperparameter Tuner tool was utilized to train the model with various combinations of hyperparameters and determine the best-performing combination for use in the actual training phase.

Two hyperparameters were selected for tuning: the learning rate and batch size. The learning rate is a hyperparameter that controls the extent to which the model is modified in response to the estimated error during weight updates. Optimization of the learning rate could improve the training process beyond the default settings. The batch size hyperparameter, which determines the number of samples that are processed before updating the model parameters, was also selected for optimization. Optimizing the batch size could aid in optimizing computation speed and convergence.

```
hyperparameter_ranges = {
    "learning_rate": ContinuousParameter(0.001, 0.1),
    "batch_size": CategoricalParameter([32, 64, 128, 256, 512]),
}
```

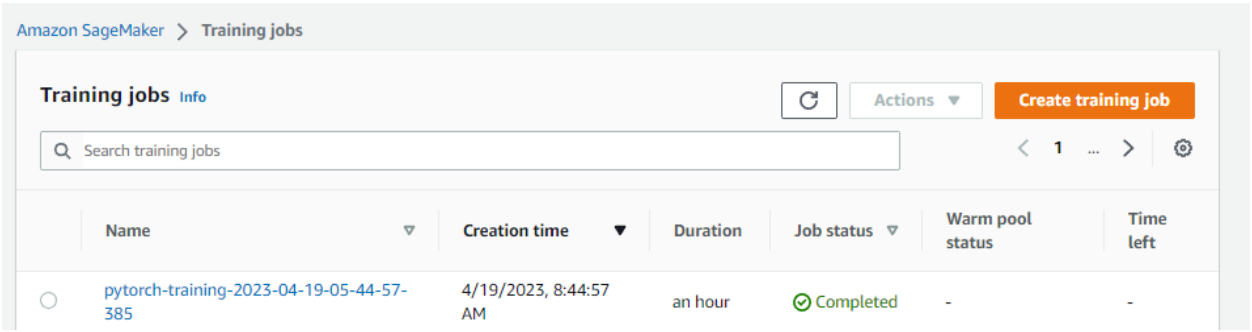
best hyperparameters = {'batch_size': 64, 'learning_rate': '0.013286582820793045'}

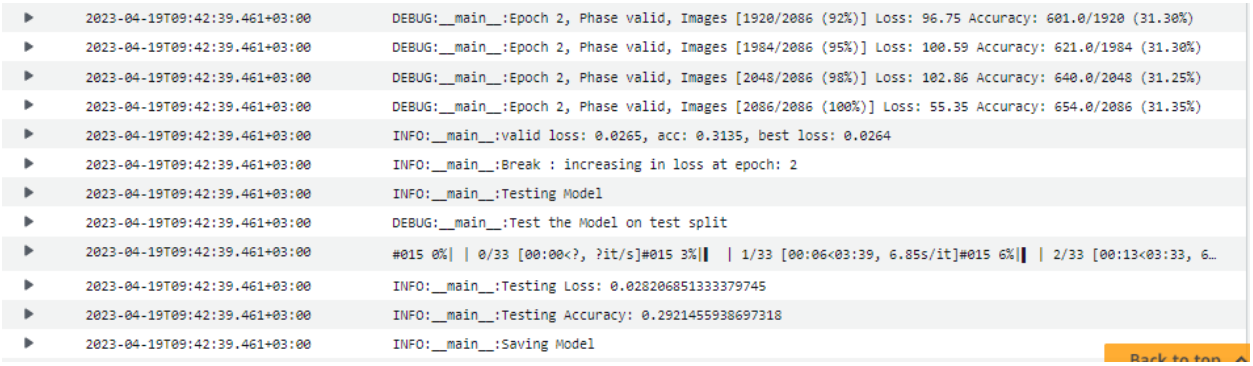




6. Model Training:

After the best hyperparameters were determined, the training process was initiated using the programming code in the *"train.py"* file. The file is designed to run as a standalone script on a computer, on inexpensive spot instances, or from within a Sagemaker notebook, with a sample usage provided in *"sagemaker.ipynb"*. The training was completed in 2 epochs, taking 1 hour to process the files with an accuracy of 29%. *'ml.g4dn.xlarge'* instance was used to speed up the process.





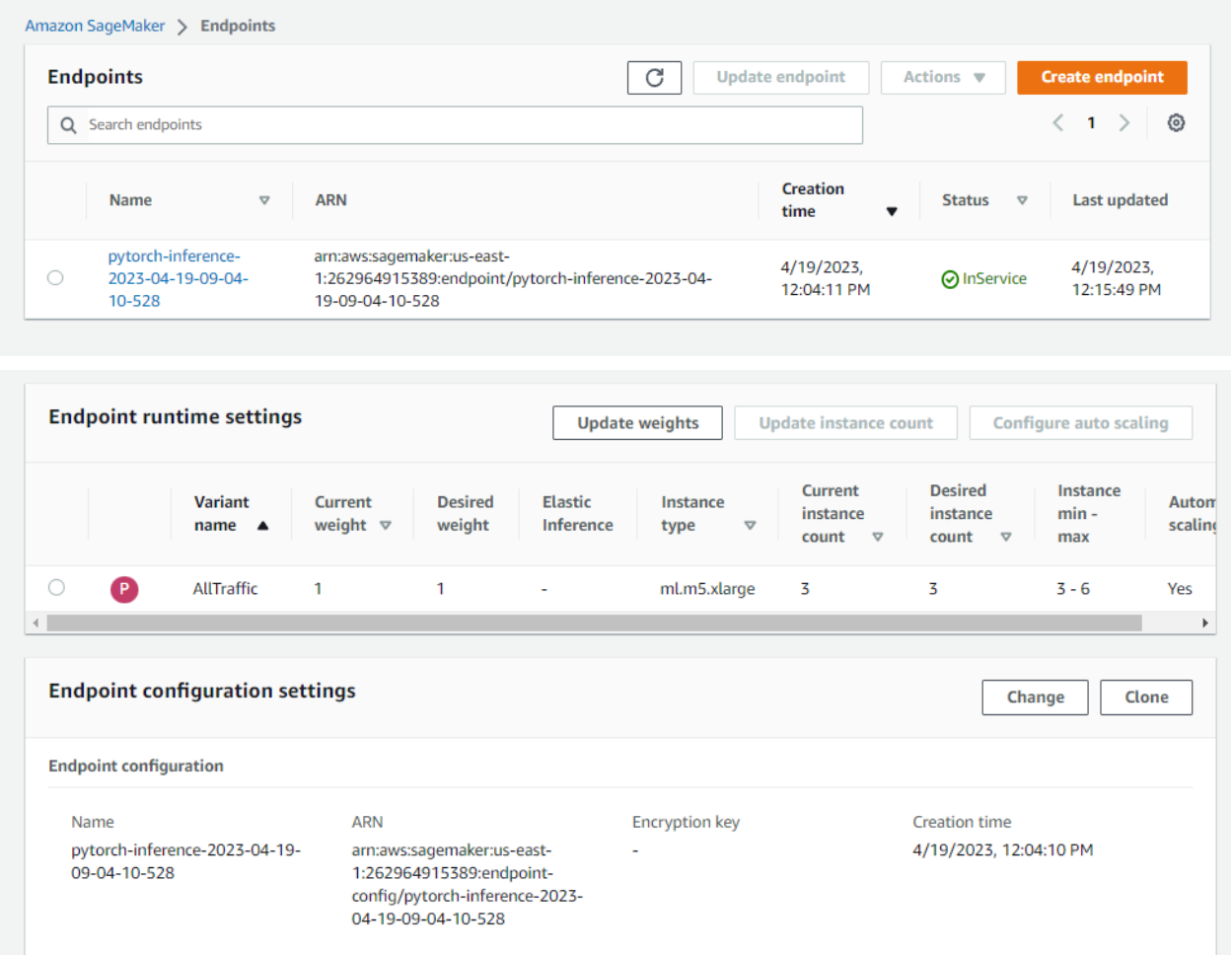
7. Refinement:

In comparison to the ABID challenge, our model achieved an accuracy of 29% and a loss of 0.0282. The ABID challenge used the Resnet34 architecture, trained from scratch with a batch size of 128 and ran for 40 epochs with a learning rate decay of 0.1 every 10 epochs. The validation accuracy of the ABID challenge peaked at 36 epochs.

To improve the model's performance, training on a larger portion of the dataset and increasing the number of epochs might be tried. Additionally, experimenting with different hyperparameters and model architectures may also yield better results.

8. Model Deployment:

At the end, the model is deployed to the endpoint with autoscale enabled to prevent possible bottlenecks.



9. Conclusion:

The trained model achieved an accuracy of 0.29, which is reasonable given that only 5% of the dataset was used and the training process was minimal. However, the model is insufficient for counting objects in a bin as only 29% of the measurements produced will be accurate.

Based on the ABID challenge leaderboard, the highest reported accuracy for the object counting task on the full dataset is around 0.8, while the lowest is around 0.5. However, it is worth noting that the challenge dataset is larger and more diverse than the subset used in this project, so direct comparison is not entirely fair.

The model may serve as a verification tool for complicated networks that characterize specific object properties. To improve the model, it should be trained on a larger portion of the

dataset and possibly allow training even while the loss function is expanding. During the project, these steps are followed: deploying ML solutions on AWS cloud, including downloading, preprocessing, and storing datasets in S3, training and tuning networks using Sagemaker, and deploying a model as an endpoint for inference.