Table of Contents

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

Artificial Intelligence (AI)-for-Good: partnering with Habitat for Canines (H4C) Dog Shelter to assist caring citizens

In this document, we specifically focus on responses to the reviewer of the initial submission. Our sincere gratitude for the feedback.

Metrics

Metrics used to measure the performance of a model

Thank you for your feedback. We agree that there needs to be a justification for the accuracy metric. This justification was indeed provided in the original proposal (see *proposal.pdf*) and it was accepted during the proposal submission. In short, we choose accuracy because (1) the original authors of the ImageNet canine subset dataset discuss 'mean accuracy' as a metric on this task, (2) it allows us to benchmark our results to their global accuracy, and above all (3) it is the customer requirement that we achieve "at or above 70%" accuracy[1,2].

Analysis

We are incredibly grateful for this suggestion and we, therefore, note the following statistics.

### Features and calculated statistics relevant to the problem

In total there are 8,351 dog images in the dataset. There are 6,680 images in the training, 836 in the test, and 835 in the validation sets. There are 133 dog breeds in the total dataset as well as the training, test, and validation datasets.

---

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

Top 20 most frequent images in all dataset
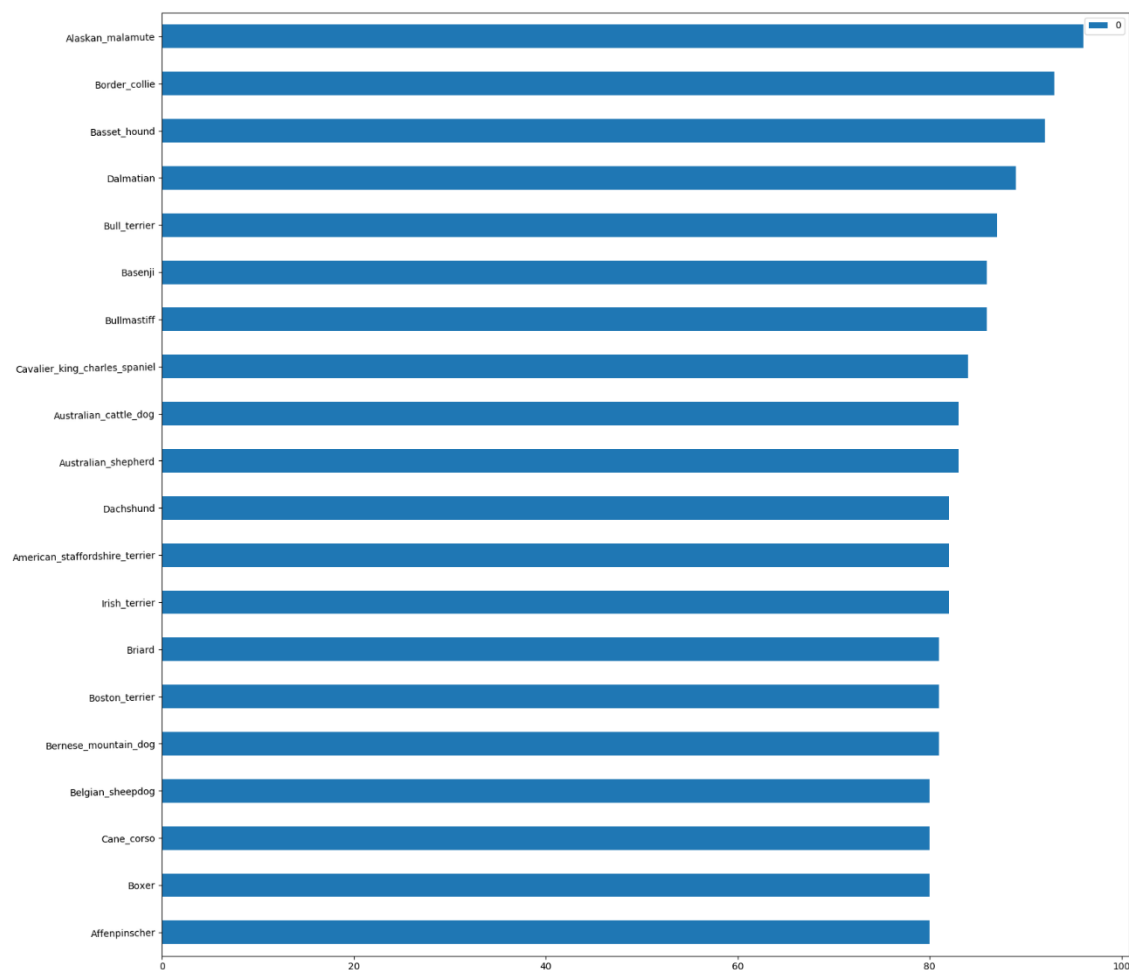
[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization.
First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition
(CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark
versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9
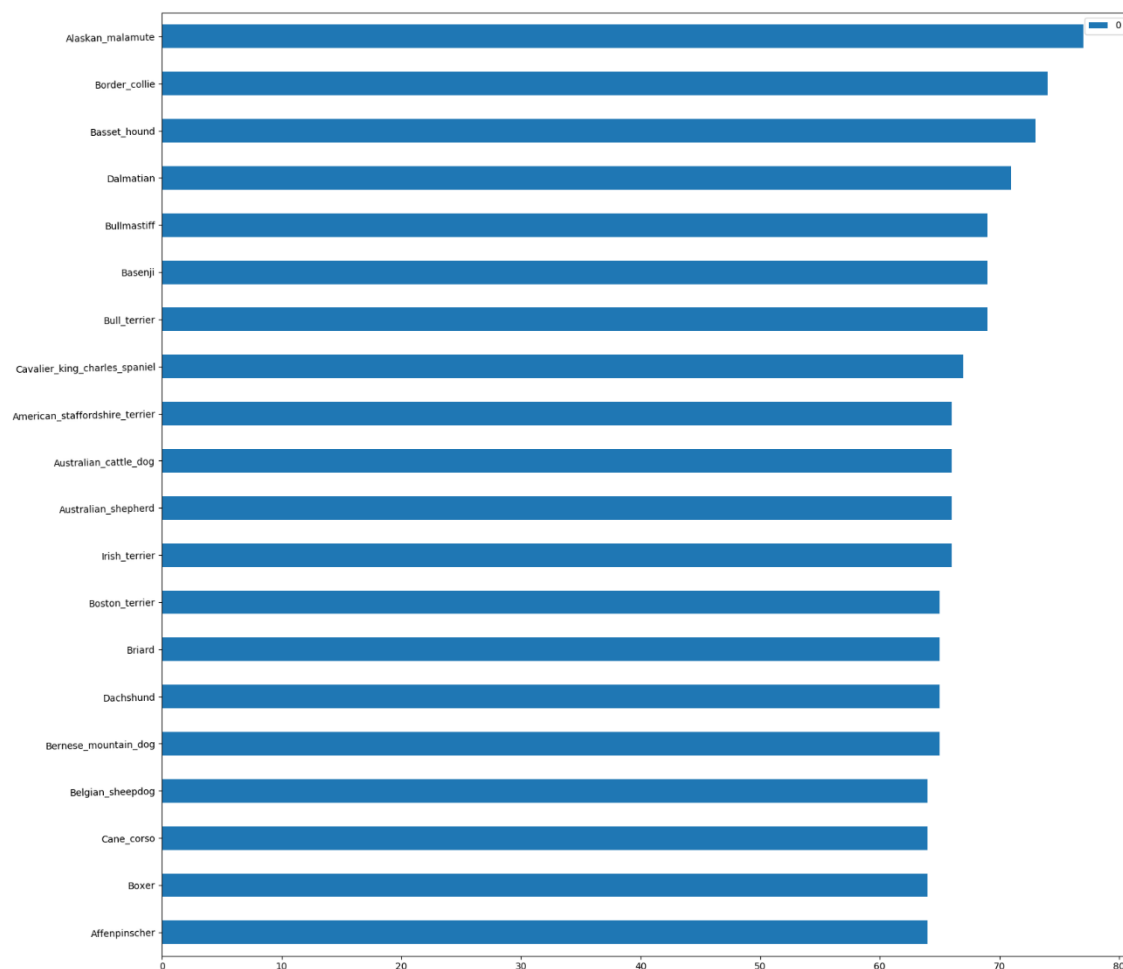
Training set

Top 20 most frequent images

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
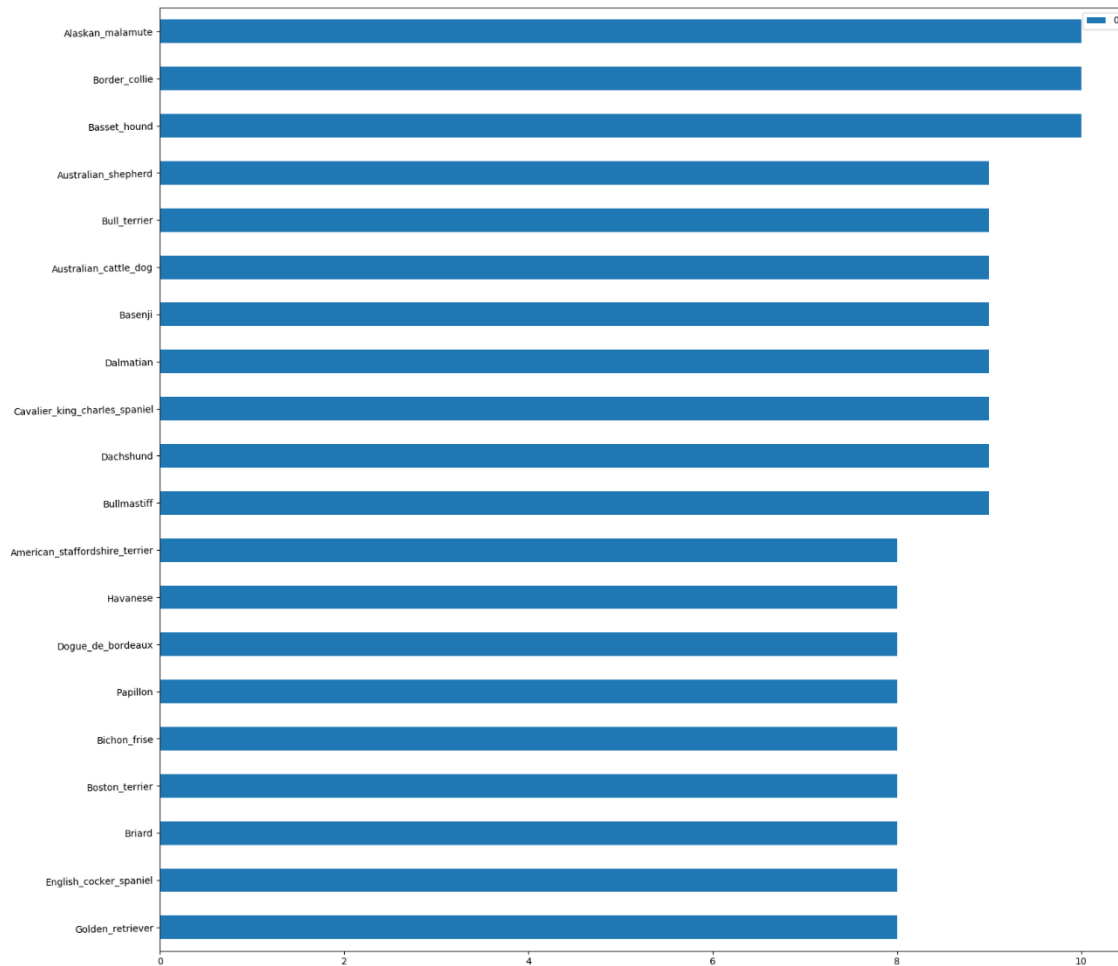[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

Top 20 most frequent images

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

Validation set

Top 20 most frequent images

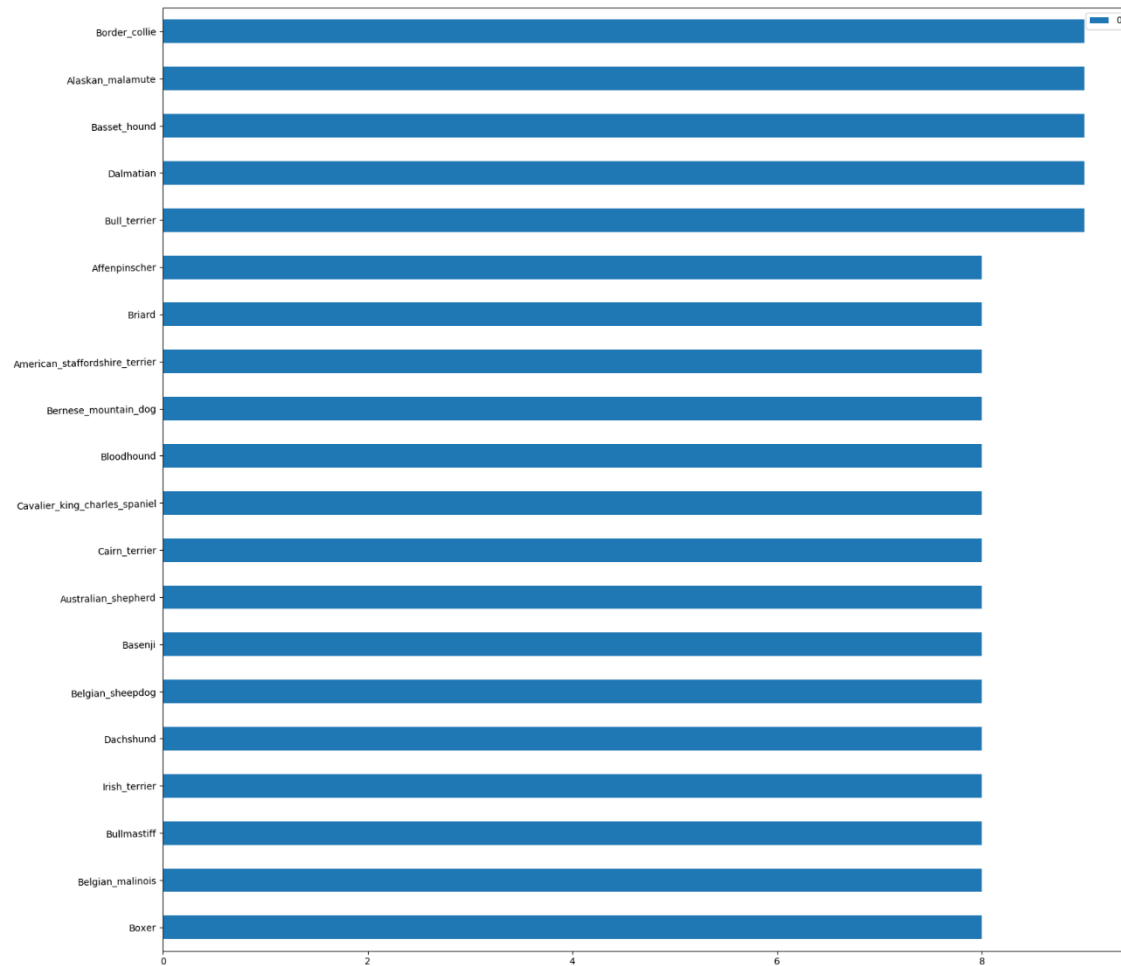[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

Abnormalities or characteristics of the data

The dataset is imbalanced. For instance, we can see that the least number of images available for the top 20 most frequent dog breeds (in the entire dataset) is 80, but in the bottom 20, the highest number of images is around 40. Thus, we have a class imbalance in the training set that was given to our model. Data augmentation, such as discussed later, is one approach to resolving this issue.

Top 20 least frequent images in the total dataset

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
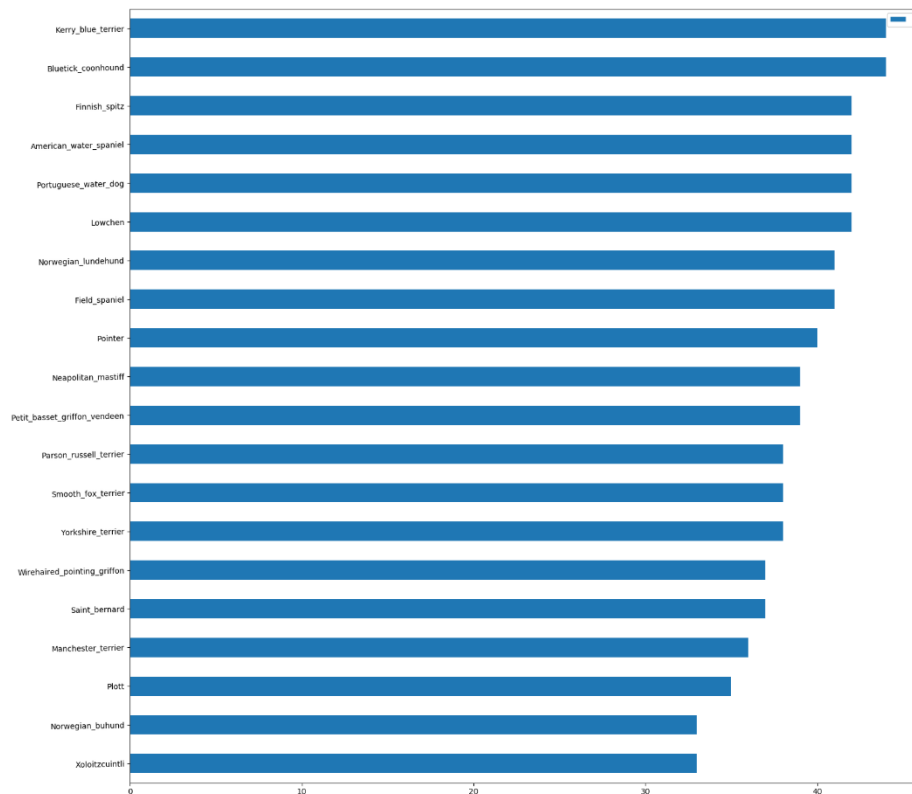[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

A visualization has been provided that summarizes or extracts a relevant characteristic

Please see above for visualization of top and bottom 20 classes. For the sake of visualization (there are 133 classes), we limit our charts to top and bottom 20 classes.

Algorithms and techniques used in the project are thoroughly discussed

We are adopting a transfer learning strategy with this task. Specifically, we used a pretrained *resnet*50 PyTorch model because it has already been trained on the ImageNet dataset[3]. This is the same dataset from which our subset of canine data were extracted[2]. Of note is that we relied on the features the resnet model has already learned from ImageNet by freezing its initial layers and appending a fully connected (FC) layer on that with ReLU activation to better handle non-linearity in our model. Only the weights / parameters of the last FC layer are trained.

Code defining our pretrained *resnet*50 model:

```python
def net():
    model = models.resnet50(pretrained=True)

    for param in model.parameters():
        param.requires_grad = False

    model.fc = nn.Sequential(
                nn.Linear(2048, 128),
                nn.ReLU(inplace=True),
                nn.Linear(128, 133))
    return model
```

To preprocess our data, we perform the following transformations. These transformations were aimed at augmenting our data (which might help resolve our imbalance issue) and allowing the model to handle different types of image inputs (e.g., an end user's dog image not centered or the dog being shown sideways etc.). These transformations provide a different view of our original images to our model over several epochs.

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

- **RandomResizedCrop**: Crops a random portion of our image and resizes it to a given size (in our case 224 X 224 regardless of the image size our end users input).

- **RandomHorizontalFlip**: Horizontally flips the given image randomly with a given probability.

- **Normalize**: Normalizes our tensor image with mean and standard deviation.

Our test and validation sets were only resized so we can have our 224 X 224 input size into our model. We only apply transformations to our training set since we are training on that set.

Methodology

Preprocessing steps

Please see above for a detailed discussion of our preprocessing steps. We believe our epoch size (50) also helps allow our model learn better for our less frequent dog classes in our imbalanced dataset.

Document process for which metrics, algorithms, and techniques were implemented

Kindly see earlier discussions on metrics and the fine-tuning of a pretrained *resnet*50 to achieve transfer learning on this task.

We first perform hyperparameter tuning on our model by specifying the following ranges

```
[4]: hyperparameter_ranges = {
         "lr": ContinuousParameter(0.001, 0.1),
         "batch-size": CategoricalParameter([32, 64, 128, 256, 512]),
     }
```

During the training process, we evaluate the trained model, with randomly chosen hyperparameters from our range (as shown above), on our validation set and exit our training at epoch 50.

---

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

```
if phase=='valid':
    if epoch_loss<best_loss:
        best_loss=epoch_loss
    else:
        loss_counter+=1
```

Then we validate our model on the test dataset

```
def test(model, test_loader, criterion):

    model.eval() #setting the model in evaluation mode
    running_correct = 0
    running_loss = 0.0

    with torch.no_grad():
        for batch_idx, (inputs, labels) in enumerate(test_loader):
            inputs, labels = inputs.to(device), labels.to(device)
            outputs = model(inputs)
            loss = criterion(outputs, labels)

            # Record the correct predictions for training data
            running_loss += loss.item() * inputs.size(0) # Accumulate the Loss
            _, predicted = torch.max(outputs, 1)
            running_correct += (predicted == labels).sum()

        # Record the testing loss and accuracy
        total_loss = running_loss // len(test_loader) #'//' is floor division
        total_acc = running_correct.double() // len(test_loader)

        logger.info(f"Testing Loss: {total_loss}")
        logger.info(f"Testing Accuracy: {total_acc}")
```

Results

The final model's qualities

The best performing scored 80% accuracy on the test. Hence, we surpass our user requirement for at least a 70% accuracy. The best model had the following hyperparameters:

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9

```
{
  learning rate: 0.004120510525842499
  batch_size: 512
}


epoch = 50
```

Validate the robustness of the model's solution

We believe our solution is robust for at least 3 reasons: (1) our model exceeds our user requirement of at least 70% accuracy; (2) it receives inputs of any shape but performs transformations on that image to crop it to a 224 X 224 image; and (3) we specified different transformations to our training images so as to allow the model better generalize when making its predictions regardless of the original image input by the end user.

The final results are compared to the benchmark result or threshold with some type of statistical analysis

Given the best model performance (on a test set) at 80% or more accuracy, we believe that our solution solves the business problem. The model is ready for production and has been deployed on online (hosted on an EC2 instance on AWS). It can be seen here for a limited time:

http://44.202.11.201:8501/

[1]Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[2]To simplify our work, we elect to use mean accuracy in alignment with the customer requirement and original benchmark versus multi class log loss, such as used in Kaggle, which might be a better evaluation metric. https://bit.ly/3ylyzsp
[3]ResNet-50 (PyTorch). https://bit.ly/3tINee9