Udacity Robotics Nanodegree Term 2

Robotic Inference

Robert Aleck, 19th May 2019

Abstract

In this paper, the author outlines an approach to training the well-known "GoogLeNet" network for image classification on a simple set of data with 3 classes, provided by Udacity, achieving greater than 75% accuracy. Further work was then carried out to capture a new 3-class dataset, and to train a model to differentiate a salt grinder, a pepper grinder, and a sugar pourer. This resultant model achieved accuracy no better than chance when trained.

Introduction

Object Classification is a well-known but ongoing challenge in computer science, aimed at identifying the content of an image. Using NVIDIA's DIGITS workflow, this project aimed to demonstrate the development of a classification model:

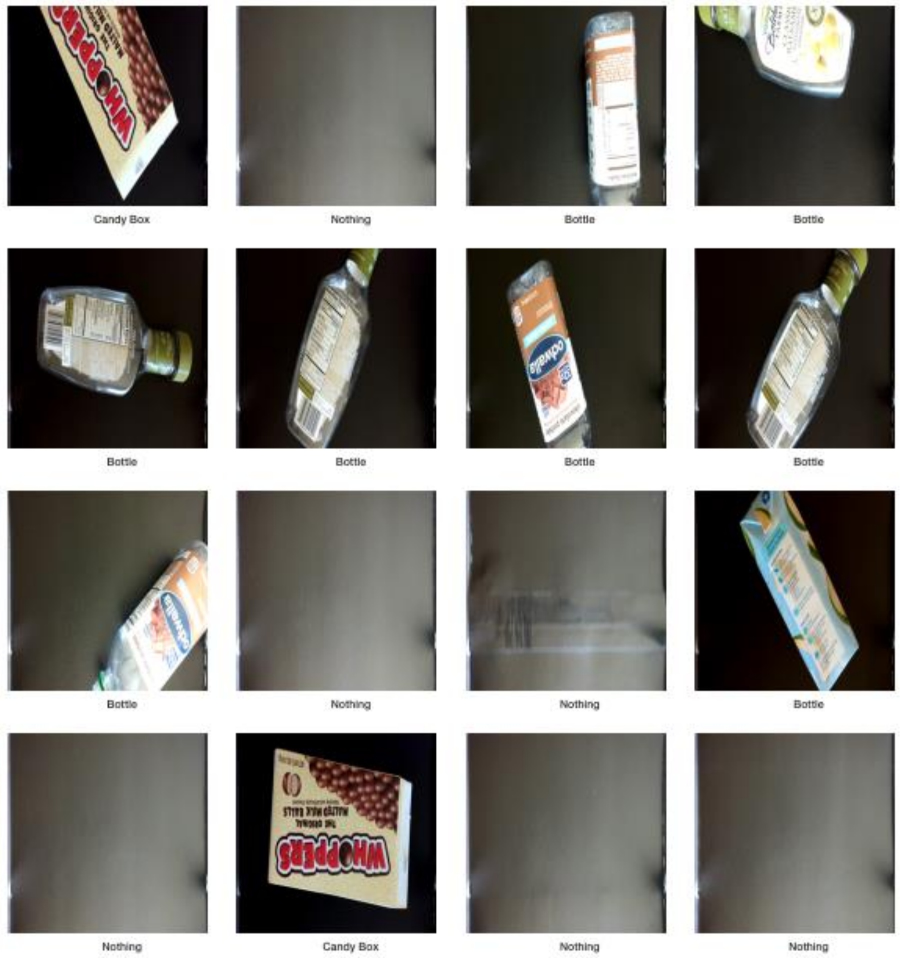
- Initially for a provided set of labelled images representing candy boxes, bottles, and an empty conveyor, and
- Later, for a set of images representing a salt-grinder, pepper grinder, and sugar pourer.

Ultimately, a model trained using these datasets could be deployed to the real world in situations such as object selection or sorting.

Data Acquisition

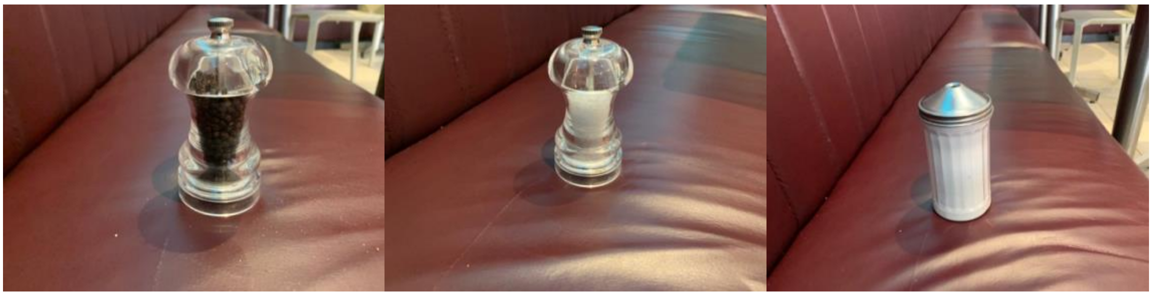
Provided dataset

On the provided Udacity workspace, the `/data/P1_data` folder contains some 10,094 images captured by an overhead camera of bottles and candy wrappers passing under the camera on a conveyor belt, including images captured with no object in the camera's field of vision. As objects pass under the camera, it is intended that a classifier would identify the object type, and trigger an action (e.g. a swing arm further down the belt) to sort/select items.



Additional Dataset

Approximately 1,000 photos were taken using an iPhone XS of three items commonly found in cafes and restaurants – a salt grinder, a pepper grinder, and a sugar pourer. The samples were spread roughly evenly across each set. A recognised limitation is that the objects are all of a single design, although care was taken to ensure that images were captured from a variety of angles and orientations. The images were then rescaled from approximately 4MP to 256x256 pixels (colour), and labelled. Approximately 10% of images were removed in to a test set.



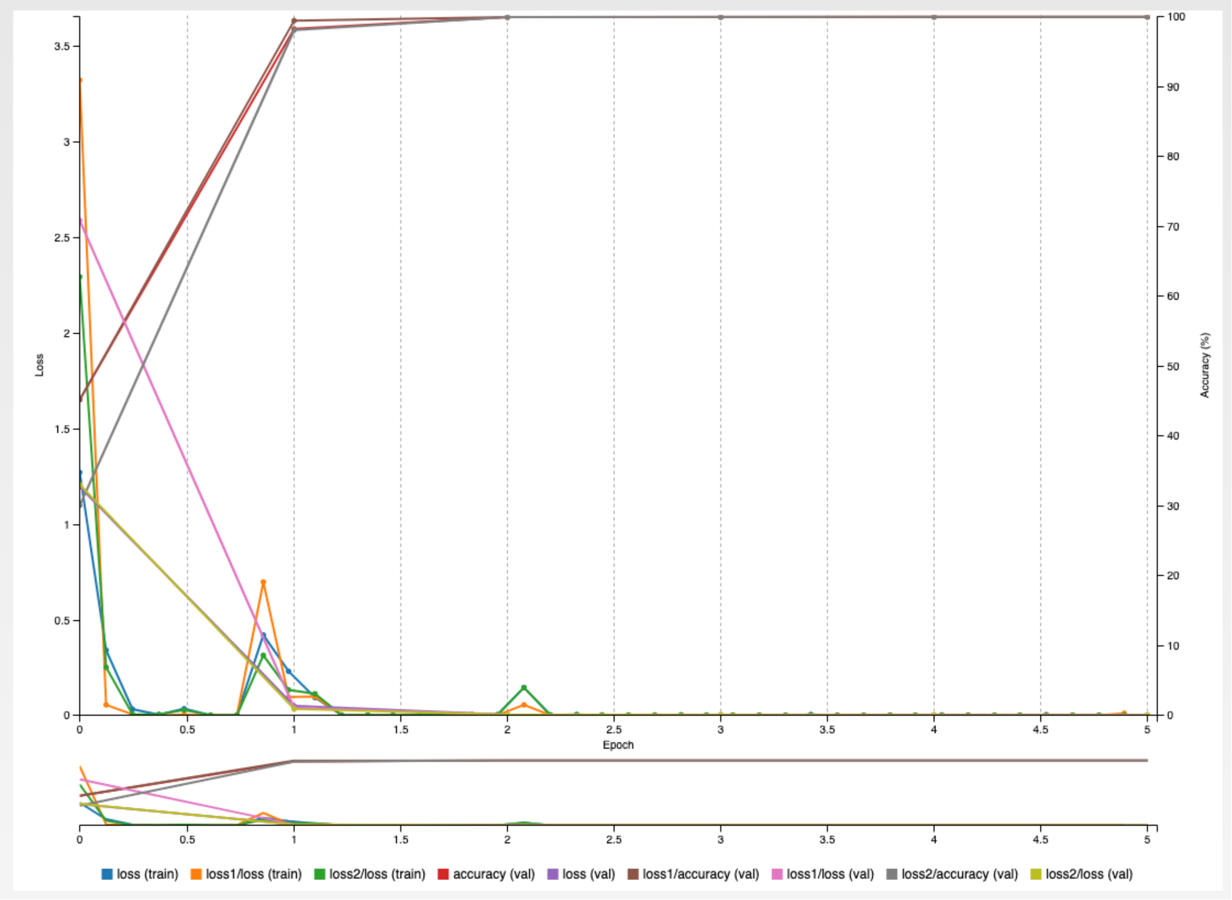
Training

Initial training focused on achieving a 75% accuracy on the provided dataset. Various models were trialled – including later versions of the Inception/GoogLeNet model to attempt to increase accuracy and response time.

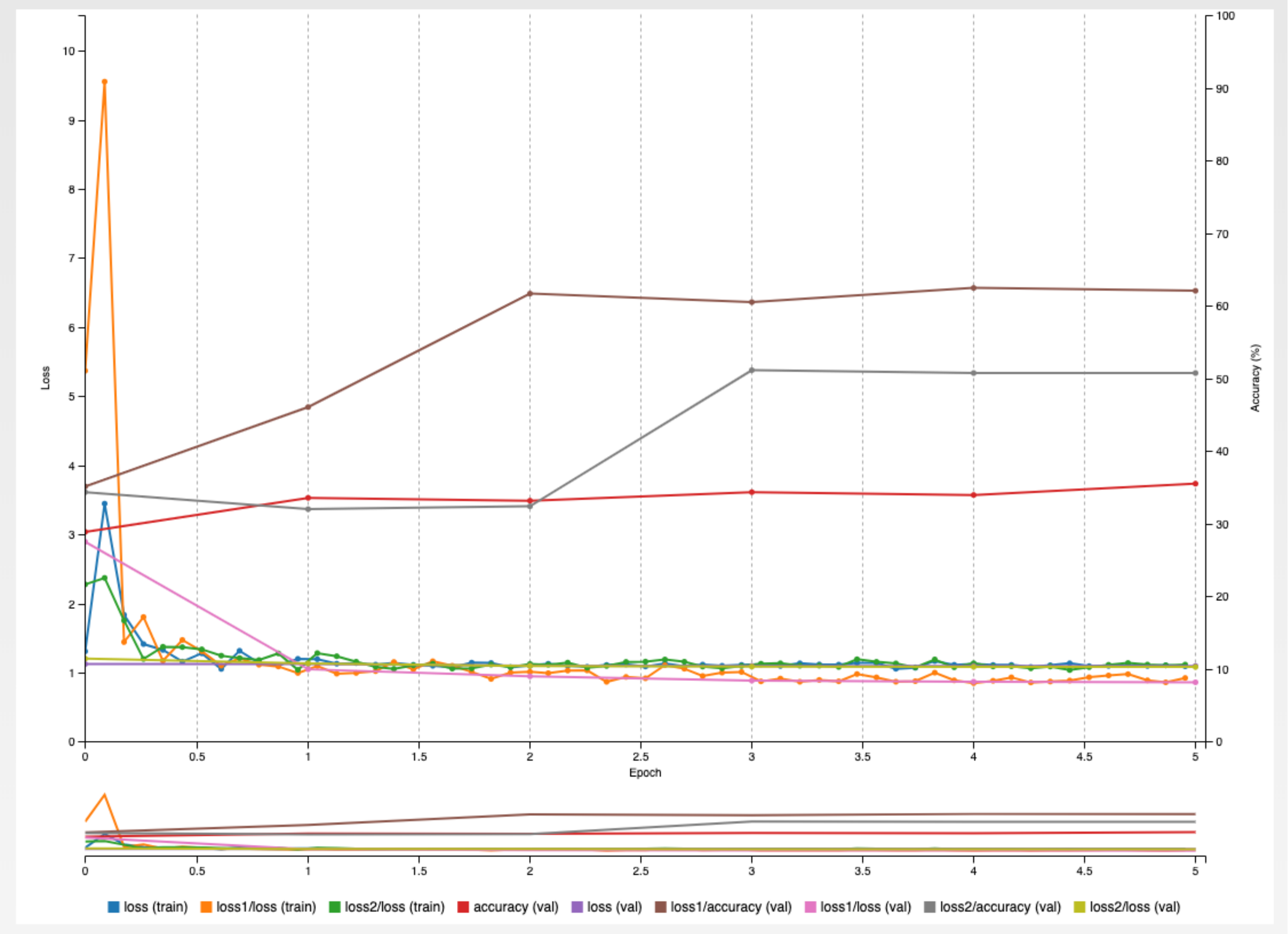
Model	optimizer	Training time	Accuracy
AlexNet	SGD	18 minutes	Approx. 68.55%[1]
GoogLeNet (inception v1)	ADAM	Approx. 20 minutes	75.41%
GoogLeNet (inception v1)	SGD	Approx. 20 minutes	74.11%
Inception v3[2]	SGD	90 minutes	30.43%
Inception v4[2]	SGD	130 minutes	33.98%

1. The AlexNet models would not validate using the built in "evaluate" command; this validation was performed manually on a subset of the full test set.
2. Inception v3 and v4 were modified from the samples available at https://github.com/SnailTyan/caffe-model-zoo

The models tended to converge relatively quickly, with little improvement in validation loss beyond 2 or 3 epochs:

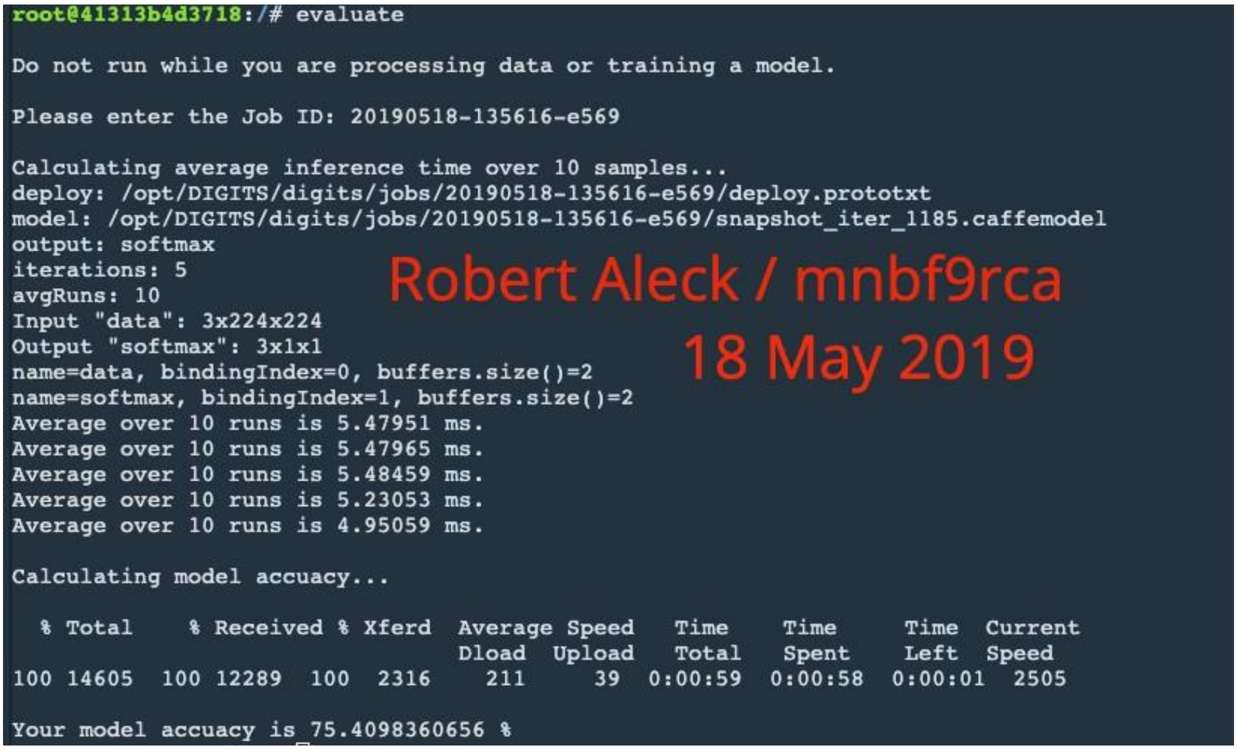


Having settled on GoogLeNet/Inception v1 with ADAM, this was then retrained on the newly acquired dataset of salt grinders, pepper grinders, and sugar pourers. Training times were roughly the same, but accuracy doesn't significantly improve with further epochs:



Results

As noted above, accuracy exceeding 75% was achieved with GoogLeNet and the ADAM optimiser, with an inference response below 10ms/image:



```
root@41313b4d3718:/# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20190518-135616-e569

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20190518-135616-e569/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20190518-135616-e569/snapshot_iter_1185.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x224x224
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 5.47951 ms.
Average over 10 runs is 5.47965 ms.
Average over 10 runs is 5.48459 ms.
Average over 10 runs is 5.23053 ms.
Average over 10 runs is 4.95059 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14605  100 12289  100  2316    211     39  0:00:59  0:00:58  0:00:01  2505

Your model accuacy is 75.4098360656 %
```

Acquired dataset

Using the test data set separated during data preparation, an accuracy of 35.58% was achieved; this is not materially better than pure chance. Indeed, looking at the probabilities assigned to each class for each test object, we can see that they all hover around the 1/3 mark:

image	label	1	1 likelihood	2	2 likelihood	3	3 likelihood	correct?
1	pepper	pepper	35.29%	sugar	32.91%	salt	31.80%	Yes
2	pepper	pepper	35.26%	sugar	32.94%	salt	31.81%	Yes
3	pepper	pepper	35.26%	sugar	32.93%	salt	31.81%	Yes
4	pepper	pepper	35.19%	sugar	32.98%	salt	31.83%	Yes
5	salt	pepper	35.21%	sugar	32.95%	salt	31.84%	No
6	salt	pepper	35.14%	sugar	33.02%	salt	31.84%	No
7	salt	pepper	35.21%	sugar	32.97%	salt	31.83%	No
8	salt	pepper	35.23%	sugar	32.95%	salt	31.82%	No
9	sugar	pepper	35.08%	sugar	33.07%	salt	31.85%	No
10	sugar	pepper	35.15%	sugar	33.01%	salt	31.84%	No

Discussion

Provided dataset

The original GoogLeNet (also called "Inception v1") model achieved best in class [1] image classification across 150,000 images of 1,000 classes [2] at ILSVRC2014. More recent versions of the GoogLeNet/Inception have been released offering greater speed and higher accuracy:

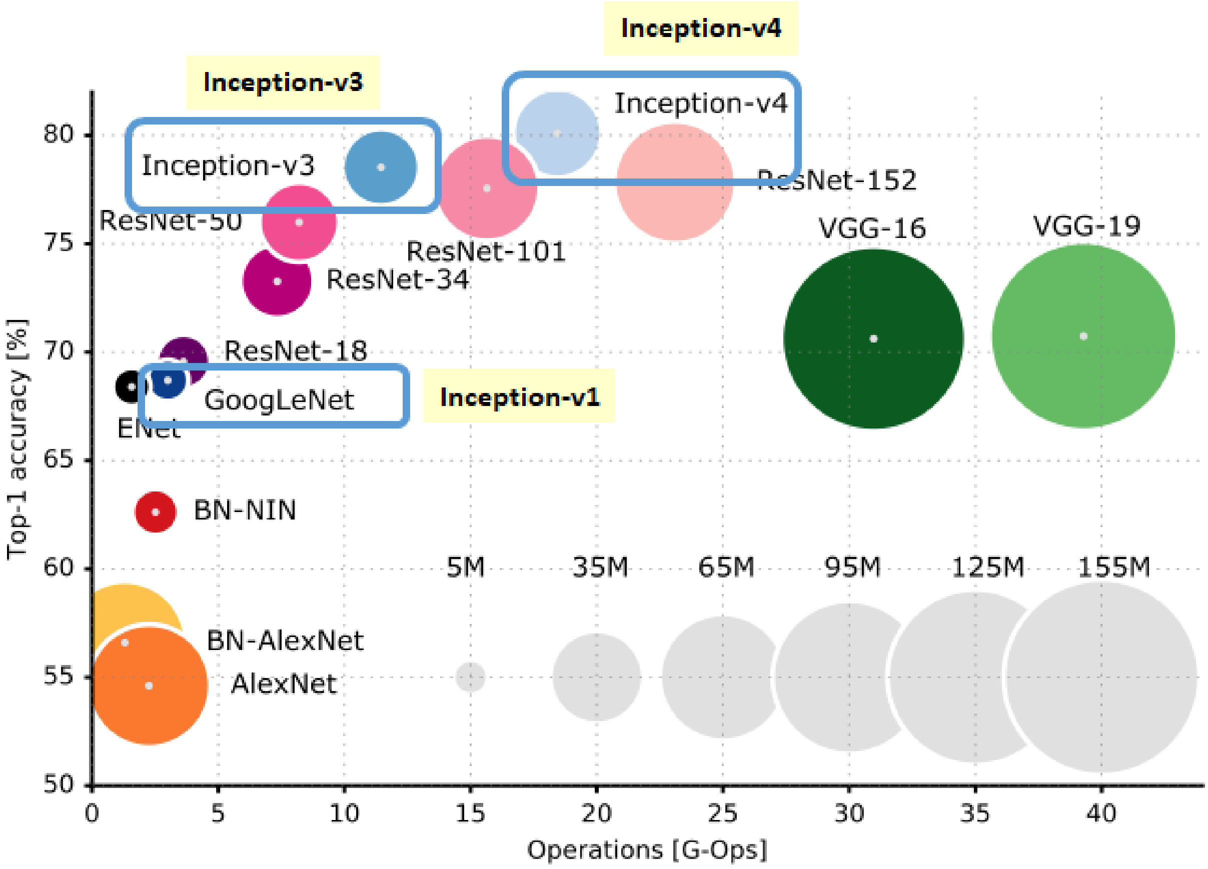


Figure 1 Top-1 Accuracy against Number of Operations (from [3])

Our results indicate that, while the more complex Inception v3 and v4 models may well provide significant performance and accuracy gains for large datasets with a significant number of classes, their complexity leads (and hence the number of trained variables) to significantly increased training time, and even when adjusted to account for the reduced number of classes, they appear to do no better than chance. In this specific project, considering the speed of travel of the object on the conveyor belt, effort should be made to improve accuracy to the point that inference time is reliably below the required activation time of the sorting device. An accuracy rate of 75% means that 1 in 4 video frames are being miscategorised – but without further analysis, it's not possible to tell exactly what is occurring:

- Bottles being miscategorised as empty space or candy boxes,
- Candy boxes being miscategorised as empty space or bottles
- Empty space being miscategorised as bottles or candy boxes

This can mean:

- Where objects are being misclassified, the introduction of contamination in the sorted object set (e.g. bottles in the candy wrappers), or
- The sorter mechanism being activated when there are no items to sort, potentially reducing its working life or increasing maintenance or operating costs.

Acquired dataset

The *provided* dataset is of high quality: images are captured from a static viewpoint, are well lit, and relatively clear. Examination of the *acquired* dataset shows that many images are blurred, and the number of different angles and orientations means that although there is a large number of images per class, there are relatively few images of the object from highly adjacent viewing angles (e.g. there are only 30-40 photos of the salt shaker grinder from the top), and the objects are relatively small in comparison to the background. This appears to have massively reduced the ability of the model to train effectively – in short, the training was unsuccessful.

Future Work

Future work should include:

- Capturing higher quality samples of the salt grinder, pepper grinder, and sugar pourer. Specifically, future datasets should include a larger number of images from across the range of capture angles and orientations.
- Introducing an object detection layer to automatically bound/crop the detected images, thereby ensuring that training is taking place on only the relevant subset of the images.
- If it is desired that the system can differentiate other models of salt grinder, pepper grinders, or sugar pourer, the captured images might be augmented with existing images from well-known datasets, such as ImageNet, or by searching sources such as flickr for images.

However, with the current accuracy rate no better than chance, it's highly unlikely that this classifier would ever be seen in a commercial project.

References

[1] ImageNet, "Results of ILSVRC2014," 2014. [Online]. Available: http://image-net.org/challenges/LSVRC/2014/results. [Accessed 18 05 2018].

[2] IMAGENET, "Imagenet Large Scale Visual Recognition Challenge 2014 (ILSVRC2014)," 2014. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2014/. [Accessed 18 05 2019].

[3] S.-H. Tsang, "Review: Inception-v4 — Evolved From GoogLeNet, Merged with ResNet Idea (Image Classification)," 27 09 2018. [Online]. Available: https://towardsdatascience.com/review-inception-v4-evolved-from-googlenet-merged-with-resnet-idea-image-classification-5e8c339d18bc. [Accessed 18 02 2019].