Automated Retail Product Checkout
Project Presentation

By Team I

Krishna Vamsi Rokkam
Sharanya Akkenapally
Arul Kiruthika Raghupathi
Ajay Ramasubramanian

Problem Statement

In the present days, the automated checkout (ACO) is booming in retail company sectors such as Amazon GO. So we wanted to explore this field as the data requirements may be changing with additional new products coming in the market very frequently. Therefore, we felt that this domain would help us explore different CNN algorithms and deep learning techniques because of various datasets that are readily available and also because of its recent innovation that is yet to generate an impact on the market in the retailing industry. Henceforth, we implemented Convolution Neural Network (CNN) models to pre-train the images from 3 different datasets in an attempt to automatically categorise the products. Because, it not only reduces the manual work but also can effectively track all the products being sold to the customers which makes tasks easier, efficient and convenient for the retailers.

Deep CNNs have proven to perform exceptionally well in classifying images; thus, it seems to be an excellent fit for resolving this problem. To evaluate the performance of the existing datasets like the Vegetable Image dataset, Freiburg groceries dataset, and Grocery store image dataset by implying three classification models for each using robust CNN architectures like Resnet18, Googlenet, and Alexnet. We also performed image augmentation techniques where input images are transformed using different transformations like horizontal flip, vertical flip, rotation, and normalization to increase the number of images passed through the network, and then feature extraction, map pooling, non-linear functionality, batch normalization are performed to get an output from a CNN architecture. Finally, we have used Adam Optimiser, a combination of RMS Prop and SGD momentum, to back-propagate the error gradients.

Dataset selection Process

- For our Project we have chose Three datasets namely:

Freiburg Groceries Dataset:

The dataset consists of 4749 images covering 25 different classes of groceries (such as beans, cakes, candy, coffee, corn, etc), with at least 97 images per class and the image resolution is about 256 x256px with all the images having JPEG format. We have to split the dataset into train, test and validation as we don't have any predefined paths and images separately. So we splitted the data into train ,test and val with 60 percent, 20 percent and 20 percent respectively. We collected this dataset from a github repository owned by PhilJd. Please refer to figure 1 for a generalized view into this dataset.

Vegetable Image Dataset:

In this dataset there are 21000 images from 15 classes, where each class contains a total of 1400 images. Each class has an equal proportion and image resolution of 224×224 and the image format is JPEG. The dataset was already distributed into 70 percent(approx.) around 15000 images for training and 15 percent(approx.) 3000 images for testing, and the rest 15 percent(approx.) 3000 images for validation.This dataset is collected from kaggle. Please refer to figure 3 for a generalized view into this dataset.

Grocery store dataset:

This dataset was collected from a github repository owned by Marcus Klasson. All the images are taken using a smartphone camera.The dataset contains 5125 natural images with 39 classes of fruits, vegetables, and carton items (e.g. juice, milk, yogurt) . As we already had images for train, test and validation separately, we didn't perform any splitting of the dataset. So there were around 2640 images under the Train set, 2485 images under the Test set, and about 672 images under the Validation set. The resolution of the image is 348x348 and the image format is JPEG. Please refer to figure 2 for a generalized view into this dataset.

Dataset selection Process

Freiburg dataset-1 Grocery dataset-2 Vegetable image dataset -3

Dataset Specifications	1	2	3
No of classes	25	39	15
No of images in Training set	4749	2640	15000
No of images in validation set	-	672	3000
No of images in testing set	-	2485	3000
Image Resolution	256x256	348x348	224x224
Image Format	JPEG	JPEG	JPEG
Image type	RGB	RGB	RGB

Methodologies

CNN Models :

1. ResNet-18:

It has the robust architecture when compared to other models in CNN. In ResNet models, all convolutional layers apply the same convolutional window of size 3 × 3, the number of filters increases following the depth of networks, from 64 to 512. At the end of all models, the average pooling layer is applied to replace fully connected layers. This replacement has some advantages. Firstly, there is no parameter to optimize in this layer, hence it helps to reduce the model complexity. Secondly, this layer is more native to enforce the correspondences between feature maps and categories . As we have a moderate amount of data, we implemented ResNet18 implementation which contains 18 convolutional layers. The another reason for choosing this model is because of its identity shortcut connection that is skip connections which can eliminate vanishing gradient issue which we faced earlier while training with alexnet architecture.

Methodologies

2. GoogleNet:

Researchers discovered that an increase of layers within a network led to a significant performance gain. But this has also few disadvantages which came at a cost. Large networks are prone to overfitting and suffer from either exploding or vanishing gradient problem.The GoogLeNet architecture solved most of the problems that large networks faced, mainly through the Inception module's utilisation. The main reason for selecting this architecture is it performs well by reducing the size of input image by retaining most useful spatial information. This makes it a powerful model with increased computational power. It contains 22 convolutional layers and 7 pooling layers but the number of parameters are reduced from 60 million (AlexNet) to 4 million. To prevent overfitting, GoogleNet uses auxiliary classifiers which can avoid vanishing gradient problems by adding it to intermediate layers namely third and sixth layers. Therefore, this fits well with our data requirements and as previously stated in the introduction part we are solving the problem of vanishing gradient problem with help of these sturdy architectures that help our model's gradient reach global minima.

Methodologies

3. AlexNet:

As previously stated, the main reason why we chose AlexNet is that it was the first model to be developed in previous decades where it had a basic network of layers of what we have today. So we wanted to compare the performance of this model with respect to more robust architectures such as Resnet18 and GoogleNet. Talking about the architecture, AlexNet has a total of eight layers in which it contains 5 convolutional layers and 3 fully connected layers which use ReLu as a non-linear activation function.

4. Optimization Algorithm:

For optimization we exclusively used Adam optimizer. The reason we chose Adam over other available optimizers is that as we learned in the course work that Adam is a combination of RMSprop and SGD Momentum as well as the results of the Adam optimizer are generally better than every other optimization algorithms, have faster computation time, and require fewer parameters for training. Another reason for choosing this optimizer is, it performs very well with a model which has a large number of learnable parameters. As we are using complex algorithms, optimization algorithms should be powerful in order to get better performance. Because of Adam, it required less tuning of the hyperparameters to achieve expected outputs and also as it uses less memory and time, it helped us in controlling the resources used by the model.

Results and Comparisons

Main Result:

After implementing all the deep CNN models—ResNet, Googlenet, and Alexnet—it can be concluded that the ResNet models provided superior performance followed by Googlenet by slightest of margins, whereas in case of Alexnet as expected the performance was low where the model is getting stuck at local minimas even after changing and applying hyperparameters, balancing out data and data augmentation techniques. Overall, Vegetables dataset gave good results when compared with other two datasets because from table 1 we can observe that it has more number of learning examples as well as from T-SNE visualization the data is well balanced. The evaluation metrics that are used for comparing the models' performances were the precision score, the recall/sensitivity score, the F1 score, confusion matrix, and accuracy. Now let us analyse individually for each architecture: The results are summarised in the following Table images respectively for each of the dataset with the representation of the evaluation metric scores.

Result and Comparisons

Vegetable Images dataset	Resnet from scratch	Googlenet from scratch	AlexNet from scratch
Accuracy	0.99	0.98	0.94
Precision	0.99	0.98	0.94
Recall	0.99	0.98	0.94
F1- Score	0.99	0.98	0.93
Train Accuracy	99.13%	98.13%	97.23%
Test Accuracy	99.1%	97.4%	92.86%
Validation Accuracy	98.37%	97.67%	92.74%

Grocery Dataset Observations	Resnet from scratch	Googlenet from scratch	AlexNet from scratch
Accuracy	0.75	0.72	0.73
Precision	0.77	0.72	0.72
Recall	0.75	0.72	0.71
F1- Score	0.74	0.71	0.72
Train Accuracy	91.65%	91%	90.12%
Test Accuracy	75.81%	72%	71%
Validation Accuracy	71.92%	70%	70.43%

Freiburg Dataset Observations	Resnet from scratch	Googlenet from scratch	AlexNet from scratch	Resnet Transfer Learning	Googlenet Transfer Learning
Accuracy	0.42	0.31	0.46	0.42	0.39
Precision	0.44	0.31	0.48	0.44	0.41
Recall	0.42	0.31	0.46	0.42	0.39
F1- Score	0.41	0.30	0.46	0.41	0.38
Train Accuracy	95.93%	88.05%	94.12%	87.62%	83%
Test Accuracy	42.82%	31.74%	45.39%	45.56%	33.28%
Validation Accuracy(approx)	58.00%	31.86%	47.94%	44.61%	38.10%

Results and Comparisons

Ablation study:

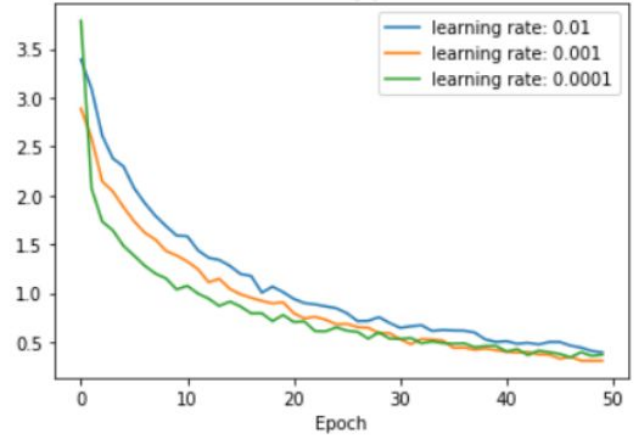

Figure 5. Comparision between different learning rates when trained using Resnet-18 model on Grocery store dataset The graph represents loss vs epochs for each learning rate

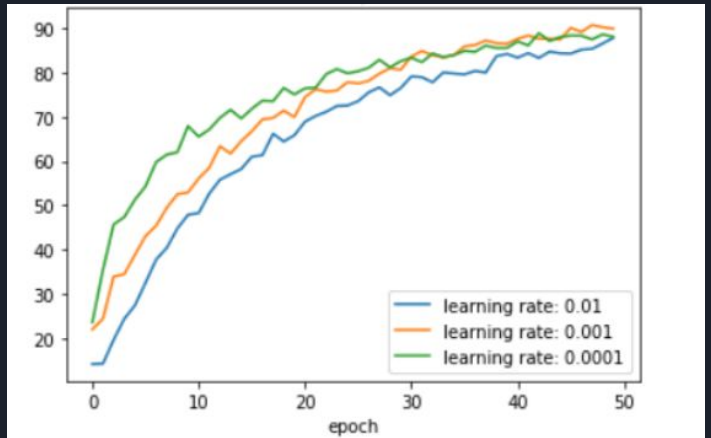

Figure 4. Comparision between different learning rates when trained using Resnet-18 model on Grocery store dataset The graph represents accuracy vs epochs for each learning rate