UNIVERSITY OF WEST LONDON
The Career University

School of Computing and Engineering
CP70011E

2023

Dissertation

DEEP LEARNING OBJECT DETECTION MODEL FOR RETAIL SELF-CHECKOUT SYSTEM

Course: Artificial Intelligence

Name: Hubert Kacper Luszczyszyn

Supervisor Name: Professor Massoud Zolgharni

Student ID: 21415922

UNIVERSITY OF WEST LONDON

Abstract

MSc Artificial Intelligence
School of Computing and Engineering

This report research focuses on the development of deep learning model which is capable of distinguishing 131 different fruits and vegetables based on picture. The research is divided in two parts: available self-checkout systems and object classification which is focused on Convolutional Neural Network and Residual Neural Network. Agile methodology has been chosen due to short amount of time given for the project and thanks to that methodology, I was able to adjust my project based on the time left. Moreover, in order to show more insights of the model performance, model implementation has been divided into three sections where each section focuses on different issue: eight varieties of the same object, ten different object and full dataset which includes both first and second section and extends it up to 131 different objects. Model results include accuracy, number of misclassified labels and charts which include confusion matrix showing all classes and epoch plots.

Table of Content

List of Figures

List of Tables

1 Introduction

Based on surveys, about 60%[1] of people finds long queues irritating, so as an answer to that self-checkout systems have become increasingly popular in recent years as they provide a convenient and efficient alternative to traditional checkout processes in retail stores. Instead of requiring a cashier to ring up transactions, these technologies enable customers to scan and bag their own things.

Artificial intelligence (AI) and its numerous applications in various industries have attracted increasing attention in recent years. One example of this is the creation of checkout-free systems in shops, which was made possible by the deployment of deep learning object detection methods. Incorporating AI technology into checkout-free systems has the potential to completely transform the shopping experience by eliminating lines, increasing transaction accuracy, and increasing overall store productivity. Amazon Fresh is one of the stores using a blend of computer vision, sensor data, and deep learning to track the things that consumers take and restock. This approach's complexity, implementation costs, and maintainability are the key problems.

[1] Kats, Rimma (February 14, 2020).

1.1 Object classification use case

Healthcare: Healthcare: Object categorization algorithms are revolutionising the analysis of medical pictures in the field of healthcare. These algorithms, for instance, may recognise patterns in CT, MRI, and X-ray images to find fractures, tumours, and other abnormalities in radiology. These aids medical professionals in choosing the best therapies and accurate diagnosis. When identifying malignant cells and tissues from biopsy samples, object categorization in pathology helps to increase the speed and precision of cancer detection. Wearable technology that has object classification capabilities can also monitor patients' vital signs and notify medical staff of potential health problems.[2]

Autonomous Vehicles: Object classification is crucial to the ability of self-driving cars to detect and navigate their environment. These vehicles have cameras, LiDAR, and sensors that collect data about their surrounds, and object classification algorithms that recognise cars, bikes, pedestrians, and road signs. Informed judgments can then be made by the automobiles, allowing them to stop for pedestrians at crosswalks and steer clear of other cars. The safety and dependability of autonomous cars are guaranteed by accurate and real-time environmental analysis.[3]

Agriculture: Using precision agriculture, object classification is transforming contemporary farming techniques. Drones flying over fields take pictures with cameras and multispectral sensors, which are subsequently processed with object categorization algorithms. With the use of these algorithms, pests and diseases can be found, and the efficiency of fertilisation and irrigation can be evaluated. Farmers can apply interventions precisely where they are required with this data-driven methodology, which lowers costs, has a smaller negative impact on the environment, and boosts crop output.[4]

---

[2] NHS (2023) *Artificial Intelligence*, *NHS choices*. (Accessed: 4 August 2023).

[3] *H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao and D. Li (2018) Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment. (Accessed: 8 August 2023)*

[4] *A. K. Dwivedi, A. K. Singh and D. Singh An Object Based Image Analysis of Multispectral Satellite and Drone Images for Precision Agriculture Monitoring, (Accessed: 10 August 2023)*

Environmental Conservation: Object classification is used to track wildlife and keep an eye on ecosystems in the context of environmental conservation. Animal photos are captured by remote cameras and sensor networks positioned in forests and wildlife reserves. These images are then analysed by algorithms to determine the species and monitor population changes. Making informed judgments about conservation and comprehending how human activity affects wildlife habitats both require the knowledge provided by this material. Marine biologists can more easily analyse the health of undersea ecosystems with the help of underwater drones with object categorization capabilities that monitor marine life.[5]

Retail and e-commerce: Object classification has changed the retail environment by making shopping more individualised. Algorithms are used by recommendation systems to analyse user preferences and past purchases in order to provide personalised product recommendations. With the aid of visual search technology, customers can take images of the goods they prefer, and computer algorithms will locate equivalent items in online shops. By responding to individual interests, this boosts user engagement, boosts sales, and develops customer loyalty.[6]

Although object categorization has a wide range of possible uses, there are still certain issues that need to be resolved. Inequalities already present can be reinforced by biased training data, biased algorithms, and discriminatory results. Careful selection of various and representative datasets is necessary to strike a balance between the advantages of object categorization and resolving these biases. Additionally, the widespread usage of surveillance cameras presents issues with data security and personal privacy, calling for moral considerations and laws to safeguard private information.

[5] *Rutten, A. (2018) Assessing agricultural damage by wild boar using drones. (Accessed: 20 August 2023).*

[6] *Bawack, R.E., Wamba, S.F., Carillo, K.D.A. et al. (2022). Artificial intelligence in E-Commerce: a bibliometric study and literature review (Accessed: 22 August 2023).*

1.2 Aim and Objectives

Aim

The primary goal is to develop a deep learning classification model that can differentiate between various fruits and vegetables, such as apples, corn and lemons, as well as between different kinds of apples, such as Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red and Red Delicious. Ideally, I would like to build my own Convolutional Neural Network, but as part of the research, I will also evaluate the performance of existing models like ResNet. Additionally, I will evaluate the performance of several models by comparing them to the same dataset.

Objectives

- Find Proper dataset and prepare it for training
- Create my own CNN which will achieve accuracy, at least, at 85%
- Check performance of available models

1.3 Research question

Can we create a CNN model and optimise it to properly classify different varieties of the same object (e.g. apple red delicious, apple granny smith etc.)?

2 Literature Review

2.1 Research about available self-checkout systems (1300)

In addition to self-checkout kiosks, there are also mobile scanners that allow customers to scan items as they shop and pay for them at a designated station before leaving the store. Smart carts equipped with scanners and payment technology are another option for customers to scan and pay for items as they shop. Checkout-free shops take the concept a step further by using advanced technology such as computer vision and artificial intelligence to automatically detect and track items as they are removed from shelves and charge customers' accounts upon exit, eliminating the need for any checkout process[7].

The only thing which might be very problematic, mainly for small store chains, is the amount of money that has to be spent on this kind of solution. Out of those three, checkout-free stores and smart shopping cart solutions are typically more expensive than traditional self-checkout systems. This is because these advanced technologies require sophisticated hardware, software, and infrastructure to operate, which can be costly to develop and maintain. In contrast, self-checkout systems are relatively simple and straightforward, consisting of a scanner, scale, and payment terminal.

I have found one research paper about implementing object detection camera to self- checkout system which turned out to be working pretty well, but the only issue is still item overlapping which causes some errors. The researchers used Mask R-CNN and data augmentation (DA) techniques to increase the accuracy of object detection in the self-checkout system. The highest mean average precision (mAP) was achieved by them using ResNet-50 and ResNet-101 as the foundation. However, the accuracy of object detection was significantly decreased when objects overlapped or had similar colours. Mask R-CNN and DA techniques were applied to increase the mAP in order to solve this problem.

---

[7] Schmidt-Jacobsen, Advantages and disadvantages of 6 retail self-checkout systems (2023)

Before training, the researchers resized a sizable dataset of images of snacks available at supermarkets to 640 x 640. They used Sharpen, GaussianBlur, Add, Multiply, Cutout, CoarseDropout, and ContrastNormalization as seven different DA methods to prevent overfitting. Both semantic segmentation and instance segmentation was carried out using Mask R-CNN. The learning rate was set to 0.002 and the number of steps per epoch was set to 500 by the researchers using TensorFlow 1.15.2 and Keras 2.2.5.

After running Mask R-CNN with different backbone models, the researchers chose ResNet-101 as the backbone, due to its low error. They used Labelme to label images of grocery store items in a dataset they called "Snacks," which they created. The dataset contained 23 distinct categories, 13 of which were different snack categories, and the remaining eight were different doughnut flavour categories. There were 12,499 training images and 2,693 testing images available after data augmentation.

Precision, recall, and accuracy were computed by the researchers using a confusion matrix, and the results of the method were assessed. They used mAP to determine the average precision for each category, and to compare their findings to those of other researchers, they took the mean of all the average precision values[8].

The study's findings demonstrated how Mask R-CNN and data augmentation techniques can be used to increase the accuracy of object detection in a self-checkout system. The researchers created a sizable dataset called "Snacks" to train their model and proposed various applications of Mask R-CNN. They showed the high accuracy at 98.92% and potency of their method by assessing the outcomes using precision, recall, and accuracy in addition to mAP.

[8] Kuo et al., Study on mask R-CNN with data augmentation for retail product detection (2021)

2.2 General research about object classification algorithms available

The challenging computer vision problem of object detection involves finding and identifying items in images or videos. Traditional methods of object detection rely on hand-made features that might not be resistant to changes in the shape, posture, and lighting of the things being detected. Deep learning can automatically learn features from data, making it more flexible and an effective tool for object detection.

2.2.1 Introduction to Convolutional Neural Networks (CNNs) and Residual Networks (ResNets)

The development of deep learning has elevated computer vision to new heights in the dynamic field of artificial intelligence. Convolutional neural networks (CNNs) and residual networks (ResNets), among the ground-breaking developments, have emerged as pillars, altering how we comprehend object classification in visual data. CNNs are useful for image processing jobs because they are the standard approach for object detection. RNNs are effective for object detection in videos when the appearance of an object may change over time.

Convolutional Neural Networks (CNNs): Convolutional Neural Networks have completely changed the field of image processing and object classification. They were inspired by the complex mechanisms of the human visual system. CNNs are precisely designed to recognise complex patterns within images by extracting hierarchical features using a series of convolutional, pooling, and fully connected layers. CNNs represent a major shift from conventional neural networks. Local filters are used in the convolutional layers to capture spatial information, allowing the network to distinguish between distinguishing characteristics like edges, corners, and textures.

The development of CNNs goes beyond their proficiency in item classification; they are now essential for a number of computer vision tasks, such as object detection, image segmentation, and even image generation. The development of CNN architectures over time, from LeNet to the influential VGGNet and the introduction of deeper architectures like ResNets, has been characterised by a trajectory of improvement, resulting in state-of-the-art performance and motivating other architectural innovations.

Residual Networks (ResNets): In the effort to train deeper neural networks, Residual Networks (ResNets) developed as an innovative solution to the vanishing gradient problem, which had previously limited the depth of networks. Through skip connections, residual blocks introduced by ResNets enable information to obstruct one or more layers. The gradient signal is preserved by this clever architectural layout, allowing for the training of networks with previously unheard-of depth while preserving convergence efficiency.

ResNet importance goes beyond only how they affect picture classification right away. Their inventive architecture has stimulated fresh ideas in other sectors, demonstrating the mul tidisciplinary nature of artificial intelligence. Furthermore, ResNets have cleared the way for i mproved training dynamics in domains other than classification, such as reinforcement learni ng and natural language processing.[9]

2.2.2 Improvements in CNN Technology

Deep learning models have the potential to revolutionise object detection as they grow more powerful and efficient, opening up new possibilities for applications like self- driving cars and surveillance systems. To enhance the effectiveness of deep learning object detection models, however, issues like data scarcity, overfitting, and real-time performance must be resolved.

The R-CNN, Fast R-CNN, YOLO, and SSD object detection techniques are some of the most significant deep learning techniques. One of the first techniques, R-CNN, creates a set of region proposals, extracts features, and categorises each region proposal as either an object or a background. Fast R-CNN is an improvement that utilises CNN for both object classification and region proposal generation.

[9] he et al., Deep residual learning for image recognition (no date)

2.3 Summary of literature review

2.3.1 Findings

Advances in Retail Checkout:

Self-checkout kiosks, mobile scanners, intelligent shopping carts, and checkout-free stores are a few examples of innovations in retail checkout operations. Although these improvements provide customers with greater ease, their implementation can be expensive, especially for small retail chains.

Improvements to Object Detection for Self-Checkout:

One of the research papers mentioned in the review focuses on enhancing object detection for self-checkout systems. This work used data augmentation methods and Mask R-CNN to get an excellent accuracy rate of 98.92%. Notably, ResNet-101 won the prize for best backbone model. This study emphasises how computer vision and deep learning have the potential to improve self-checkout procedures.

General Object Recognition Research:

Conventional object detection techniques were limited in their capacity to adapt to different object appearances since they depended on hand-crafted features. In contrast, deep learning has become a potent technique that enhances adaptation to many settings by automatically learning features from data.

Introduction to CNNs and ResNets:

The core structures of deep learning for computer vision are convolutional neural networks (CNNs) and residual networks (ResNets). CNNs are excellent at classifying objects from images and performing image analysis, while ResNets tackle problems with very deep neural network training. Beyond categorization, they also contribute to numerous computer vision tasks.

Advancements in CNN Technology:

The potential for CNN technology to revolutionise object identification is enormous, creating an opportunity for uses like self-driving automobiles and monitoring systems. R-CNN, Fast R-CNN, YOLO, and SSD techniques have been essential in improving object identification precision and real-time speed.

2.4 Conclusion

Despite significant progress in object detection technology for retail self-checkout systems, there is still a gap in the literature that can be filled. The most cutting-edge solutions available today have demonstrated encouraging results in accurately detecting a variety of products, but issues like item placement continue to exist. The effectiveness, accuracy, and robustness of object detection in retail self-checkout systems can thus still be improved through research and development. Future research might concentrate on creating more sophisticated algorithms that can handle these problems and investigating the incorporation of other technologies like computer vision and deep learning to improve the performance of these systems.

3 Research Methodology

The research methodology for this project entails a number of crucial procedures. I have done a thorough literature review before I began this research in order to accomplish the aims and objectives. This step will also involve a review of the literature on anything from object identification and categorization to neural network troubleshooting, with the goal of enhancing prediction performance. Second, a suitable dataset is selected, which comprises images of several apple varieties as well as images of fruits and vegetables. After the photographs have been scaled and normalised, the dataset is split into training, validation, and testing sets. There are various data augmentation procedures that may be applied to boost dataset diversity.

Next, a CNN architecture is developed to handle the classification of fruits and the identification of apple varieties. The design considers the number of layers, kernel sizes, pooling procedures, and activation functions to accurately reflect distinctive properties. As the proposed CNN model is trained using the training dataset, the model parameters are updated using a model optimization strategy. The training procedure is observed, and any required hyperparameter modifications are made, using the validation dataset.

After the model has completed training, its performance is evaluated using metrics from the testing dataset, such as accuracy, precision, recall, and F1 score. Performance evaluations are conducted using both the general fruit classification exam and the specialised apple variety identification job. The experiment findings are then contrasted and looked over. The performance of the CNN model for fruit categorization and apple variety identification is assessed in terms of accuracy and learning speed. Limitations and challenges encountered during the experimental process are noted. If the preliminary results are poor, the model could undertake optimization and fine-tuning. Adjusting the architecture, altering the hyperparameters, or utilising transfer learning approaches can all help to increase accuracy and generalisation abilities.

A summary of the results that highlights the performance of the CNN model in contrast to ResNet34 finishes the experimental procedure. Conclusions about the effectiveness of the suggested strategy will be presented, and potential topics for more research or development will be discussed. By employing this experimental approach, the research hopes to advance the field of computer vision in agriculture and the food sector by creating a strong CNN model that is capable of reliably classifying fruits and recognising apple kinds. Then CNN model will be compared to ResNet34 to decide which one is better and which will be used for the future implementation.

I am going to look at 3 types of development methodologies: Agile, Waterfall and V-model, and decide which one is the best for my type of project.

3.1 Types of methodologies

1. Agile is a practice-based technique for modelling and documenting software-based systems. It's intended to be a set of standards, guidelines, and practises for software modelling that may be used to software development projects with more adaptability than conventional modelling techniques.
2. The waterfall model, a sequential design method frequently used in software development processes, shows progress as flowing slowly downward through the phases (like a waterfall).
3. A software development approach that may be seen as an extension of the waterfall model is the V-model. After the coding stage, the process steps are twisted upwards to create the familiar V shape rather than progressing down the line.

3.2 Research approach and justification

Taking into consideration, that my idea is quite challenging for me, I have chosen the agile type of development since, due to many not specified information, I am not sure yet, if everything is possible to develop. Moreover, due to limited time, I might be forced to drastically change my project in order to finish before the deadline. The agile paradigm seems to be the most sensible thanks to its flexibility as well as easy adaptation in many situations, I will be still able to finish my project with a success, but with the not specified final effect.

3.3 System setup

For the entire model implementation and training process the following computer has been used:

Name	Specs
LAPTOP MODEL	MacBook pro 16 inch (2019)
SYSTEM	Windows 10
CPU	2.6 GHz 6-Core Intel Core i7
GPU	NVIDIA GEFORCE RTX 2070 SUPER
MEMORY	32GB DDR4 RAM
HARD DRIVE	SSD 1000GB

Table 1 - Computer specification

Entire implementation has been handled with this computer: EDA, model fitting, model training and model evaluation.

4 Design and Implementation

4.1 Environment setup

Type	Name
IDE	Jupyter Lab
Programming Language	Python
Deep Learning Library	Pytorch

Table 2 - Environment tools

4.2 Dataset Overview

Dataset used for this project is available under this link.

The dataset used in this research significantly expands the level of information about fruit classification. It provides 131 different fruits and vegetables and a wide range of samples, allowing me to look into a number of visual features. The dataset's samples have all been correctly labelled, enabling the successful use of supervised learning techniques.

Figure 2 - Sample image 100 by 100 pixels (Avocado)

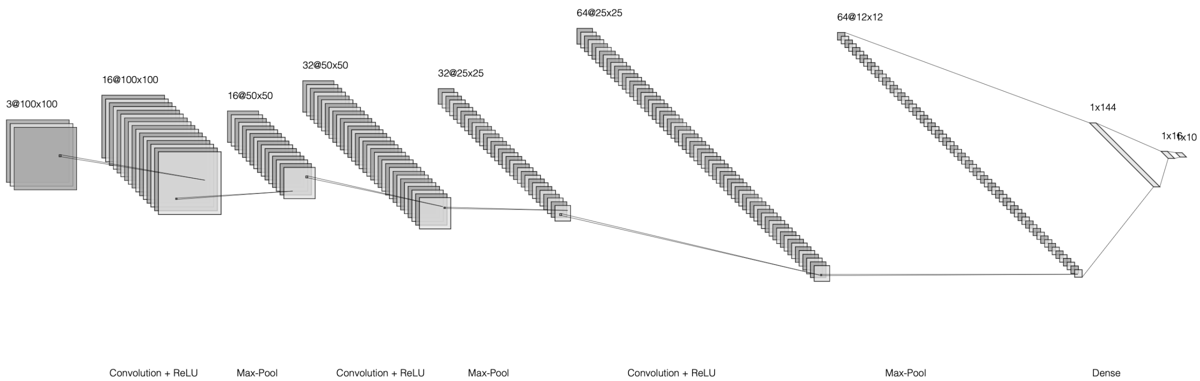
Figure 1 - Sample image 100 by 100 pixels (Corn)

Each image in the dataset has been scaled to 100 by 100 pixels in preparation for training and testing. The dataset is split into training and testing folders in order to better evaluate the model's performance. The training folder contains approximately 500 examples of each fruit or vegetable type, and the testing folder contains 160 samples of each.
Accurate representation and classification model evaluation are guaranteed by the sample set's uniform distribution across the dataset.

4.3 Deep Learning Model Architecture

4.3.1 CNN

Nine layers make up the CNN model for classifying fruits and vegetables. Convolutional layers extract features from the images at the beginning of the model, and then ReLU activation functions inject non-linearity. Max pooling layers shrink the feature maps' spatial size while still preserving crucial data. The output for the fully linked layers, which map the features to the 10 classes of fruits and vegetables, is reshaped by the flatten layer. ReLU activation is used once more by the model before the last completely connected layer. The design, activation functions, and regularisation methods of the model allow it to accurately categorise images and learn discriminative features.




```
Sequential(
  (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (1): ReLU()
  (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (4): ReLU()
  (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (6): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (7): ReLU()
  (8): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (9): Flatten(start_dim=1, end_dim=-1)
  (10): Dropout(p=0.25, inplace=False)
  (11): Linear(in_features=9216, out_features=1024, bias=True)
  (12): ReLU()
  (13): Dropout(p=0.25, inplace=False)
  (14): Linear(in_features=1024, out_features=131, bias=True)
)
```

Figure 3 - Model Architecture Diagram and Layer Description

4.3.2 ResNet34

ResNet-34, a member of the ResNet family, is well known for having a structure with 34 layers, which is deeper than typical CNNs. The key feature of ResNet-34 is the addition of skip connections, often referred to as residual connections, which enable the network to skip levels and quickly connect the input to further layers. This solves the vanishing gradient problem and facilitates the training of very deep networks. Because of the skip connections, ResNet-34 can develop residual representations that can capture complex data and intricate patterns. By generating outstanding results in a range of computer vision tests, this architecture has proven its value in picture categorization and other visual identification tasks.

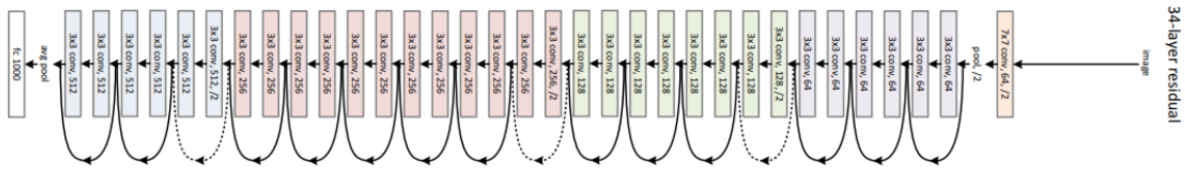


Figure 4 - ResNet34 Architecture Diagram

Hyperparameters used when training models:

- Learning Rate = 0.001
- Number of Epochs = 30
- Batch size = 64

4.4 Implementation

I divided this project into three independent portions to fully solve this challenge. Each section carefully analyses various aspects of fruit and vegetable classification, leading up to the final section's comprehensive evaluation of the overall findings. This method not only offers a thorough comprehension of the subject but also makes it easier to evaluate the results in a variety of ways.

A good tactic for gaining deeper understanding and enhancing the model's performance is to test several fruit and vegetable varieties separately before assessing the complete dataset on the model.

This strategy offers a more fine-grained study by focusing initially on specific categories like different varieties of apples or pears, revealing light on the model's capacity to discriminate closely related classes. This step aims to identify any potential difficulties the model might encounter when dealing with minute variances and offers information on potential areas for development in differentiating similar items.

Second, by evaluating various veggies separately, the technique enables the targeted addressing of issues that are exclusive to vegetables, such as variances in forms, sizes, and colours. This targeted approach makes it easier to improve the model's performance on particular categories and make precise adjustments that are specific to each category's characteristics.

Finally, as we proceed through various subsets of the dataset, valuable information about the model's advantages and disadvantages is revealed. For instance, it can become clear that the model is excellent at identifying apples but struggles to tell between different kinds of vegetables and fruits. These insights are crucial in directing our work on optimising the model and data pre-processing techniques. The need to train and test on the complete dataset for each iteration is reduced, which is an important aspect of this methodology that encourages resource efficiency. Instead, we can repeat quickly on smaller amounts of data, making necessary corrections more successfully and sparing computational time and resources.

4.4.1 Classification of 10 Different Fruits and Vegetables

In this section, the main focus is the classification of a diverse array of 10 distinct fruits and vegetables. This initial exploration serves as the foundation for subsequent evaluations. This section provides details of the unique characteristics of each item within the selected subset using a custom deep learning model as its foundation. The outcomes of this stage are meant to provide information about how well the model can capture the subtle differences between each fruit and vegetable. The foundation for the following sections is provided by this initial understanding of item-level classification.

From the larger dataset, which contains 131 fruits and vegetables, a smaller dataset designed to meet the requirements of this project was created. The smaller dataset focuses on ten different fruits and vegetables to increase classification accuracy. By keeping the same number of samples per class as in the larger dataset, the smaller dataset, which was important, preserved consistency in the sample distribution.

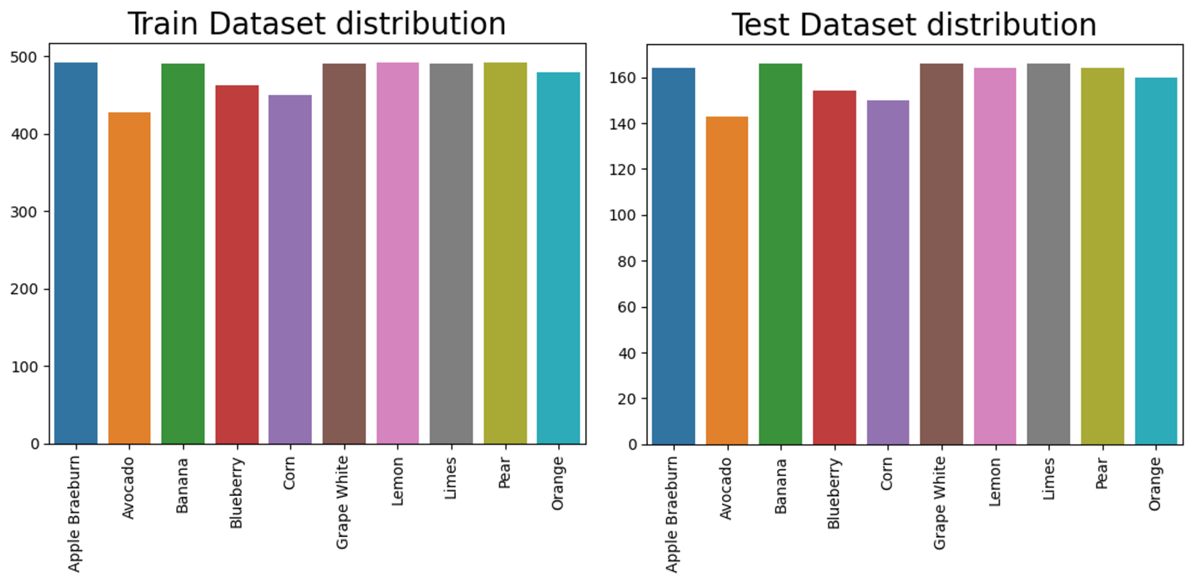


Figure 5 - Data Distribution Across Classes

A subset of the training dataset was set aside for the creation of the validation dataset. To be more precise, 20% of the initial training dataset was sacrificed for validation.

To evaluate the effectiveness of the classification models during training, a separate validation set was created by sacrificing a portion of the training data. This validation set makes it possible to track the model's generalisation and look for overfitting problems.

The models are trained using the remaining 80% of the initial training dataset, ensuring that there is enough data to enable the models to understand the patterns and traits of the fruits and vegetables. In order to evaluate the model's performance and make educated decisions about hyperparameter tuning or model selection, the 20% that was sacrificed creates a solid validation set.

It is crucial to remember that before the dataset was made available, the dataset supplier had already done some transformations to samples. To increase the accuracy of training, these modifications involve scaling the photos to a standard 100 by 100 pixel size and making minor colour adjustments. As an addition to transformation made by the author of the dataset, I have done data normalization as well as converted data into tensors. Transforming data into tensors makes data compatible with deep learning frameworks like Pytorch, which I have used for this project, and normalization ensures that data is in consistent range which optimizes neural network training.

```
# set random seed so we get the same sampling every time for reproducibility

random_seed = 100
torch.manual_seed(random_seed);

transformation = transform.Compose([transform.ToTensor(),
                                    transform.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])])

dataset = datasets.ImageFolder(train_path, transform=transformation)
test_dataset = datasets.ImageFolder(test_path, transform=transformation)

train_ratio = 0.8
train_size = int(train_ratio * len(dataset))
val_size = len(dataset) - train_size

train_dataset, val_dataset = random_split(dataset, [train_size, val_size])

# Create the data loaders
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=64, shuffle=True, num_workers=12, drop_last=True)
val_loader = torch.utils.data.DataLoader(val_dataset, batch_size=64, shuffle=True, num_workers=12, drop_last=True)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size=64, shuffle=True, num_workers=12, drop_last=True)

dataset.classes

['Apple Braeburn',
 'Avocado',
 'Banana',
 'Blueberry',
 'Corn',
 'Grape White',
 'Lemon',
 'Limes',
 'Orange',
 'Pear']
```

Figure 6 - Data Pre-Processing and Data Distribution Between Train, Test and Validation

As the dataset provider has already handled some transformations, the pre-processing I do will concentrate only on applying normalisation to the data, transforming it to tensors, and creating batches with 64 samples each.

```
def show_batch(dl):
    for images, labels in dl:
        fig, ax = plt.subplots(figsize=(12, 12))
        ax.set_xticks([]); ax.set_yticks([])
        ax.imshow(make_grid(images[:64], nrow=10).permute(1, 2, 0))
        break

show_batch(test_loader)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

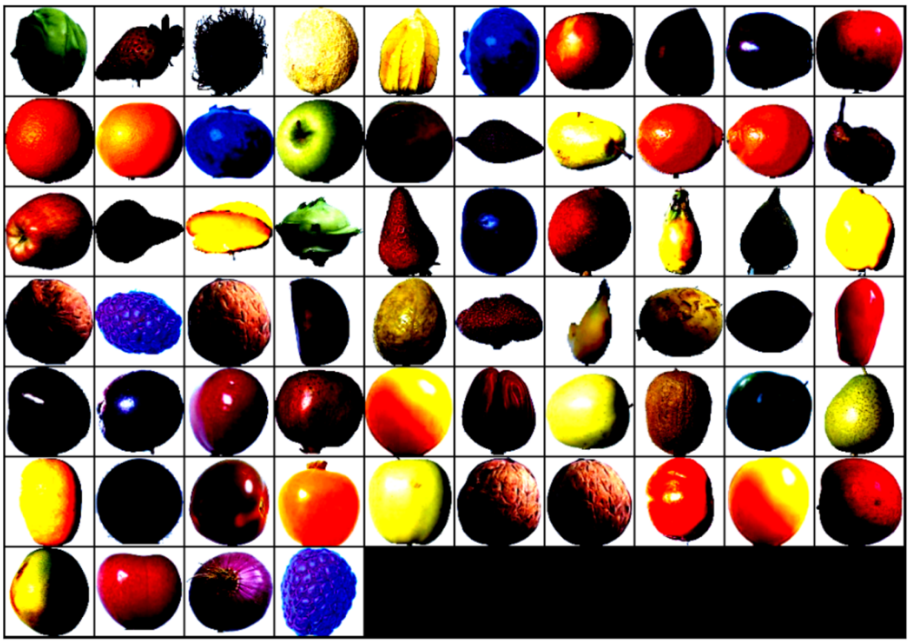


Figure 7 - Data Batch Visualization

4.4.2 Classification of 8 Apple Varieties

As we move on to the second section, the classification of 8 different apple varieties is the main topic. This section explores the model's ability to distinguish small variations within a single category as it delves into the finer granularity of classification. This phase aims to determine the degree to which each model can distinguish the subtleties of the same type of the object variety differentiation by contrasting the results of the self-designed deep learning model with the ResNet34 architecture. The findings of this investigation set the stage for the thorough evaluation conducted in the last section.

Data preparation is basically done the same way as it was done in the previous task, however I have included all available varieties of apples in this dataset which are:

- Apple Braeburn,
- Apple Crimson Snow,
- Apple Golden,
- Apple Granny Smith,
- Apple Pink Lady,
- Apple Red,
- Apple Red Delicious,
- Apple Red Yellow

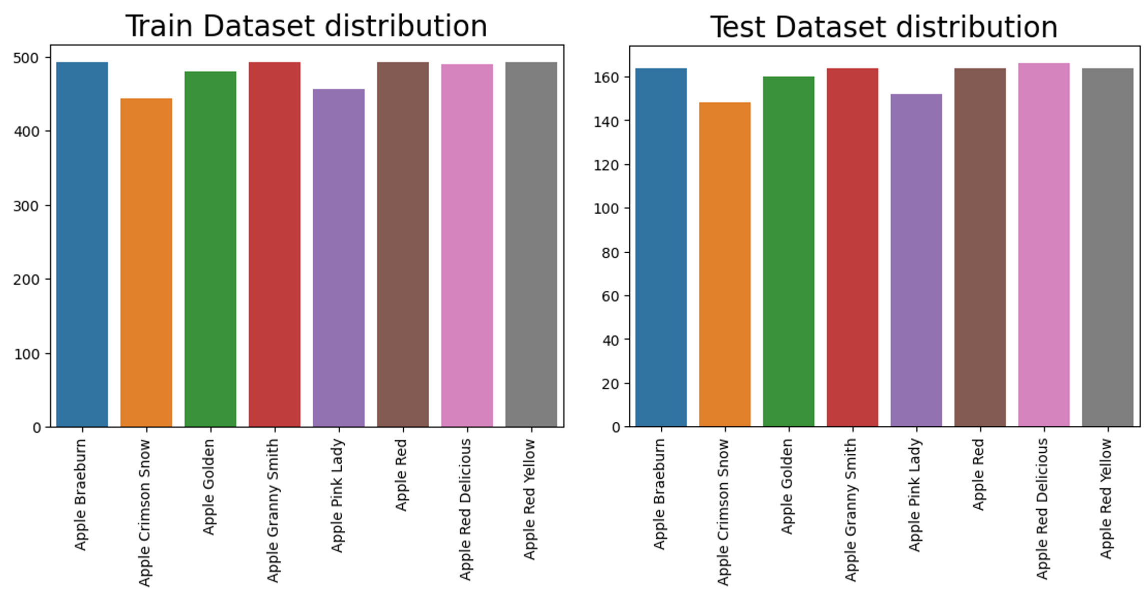


Figure 8 - Dataset Distribution (Varieties of Apples)

In order to create a validation dataset, a portion of the training dataset was allocated for validation purposes. Specifically, 20% of the original training dataset was set aside and sacrificed for validation the same way as it was done in previous task.

```
# set random seed so we get the same sampling every time for reproducibility

random_seed = 100
torch.manual_seed(random_seed);

transformation = transform.Compose([transform.ToTensor(),
                                    transform.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])])

dataset = datasets.ImageFolder(train_path, transform=transformation)
test_dataset = datasets.ImageFolder(test_path, transform=transformation)

train_ratio = 0.8
train_size = int(train_ratio * len(dataset))
val_size = len(dataset) - train_size

train_dataset, val_dataset = random_split(dataset, [train_size, val_size])

# Create the data loaders
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=64, shuffle=True, num_workers=12, drop_last=True)
val_loader = torch.utils.data.DataLoader(val_dataset, batch_size=64, shuffle=True, num_workers=12, drop_last=True)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size=64, shuffle=True, num_workers=12, drop_last=True)

dataset.classes

['Apple Braeburn',
 'Apple Crimson Snow',
 'Apple Golden',
 'Apple Granny Smith',
 'Apple Pink Lady',
 'Apple Red',
 'Apple Red Delicious',
 'Apple Red Yellow']
```

Figure 9 - Data Pre-Processing and Data Distribution Between Train, Test and Validation (Varieties of Apples)

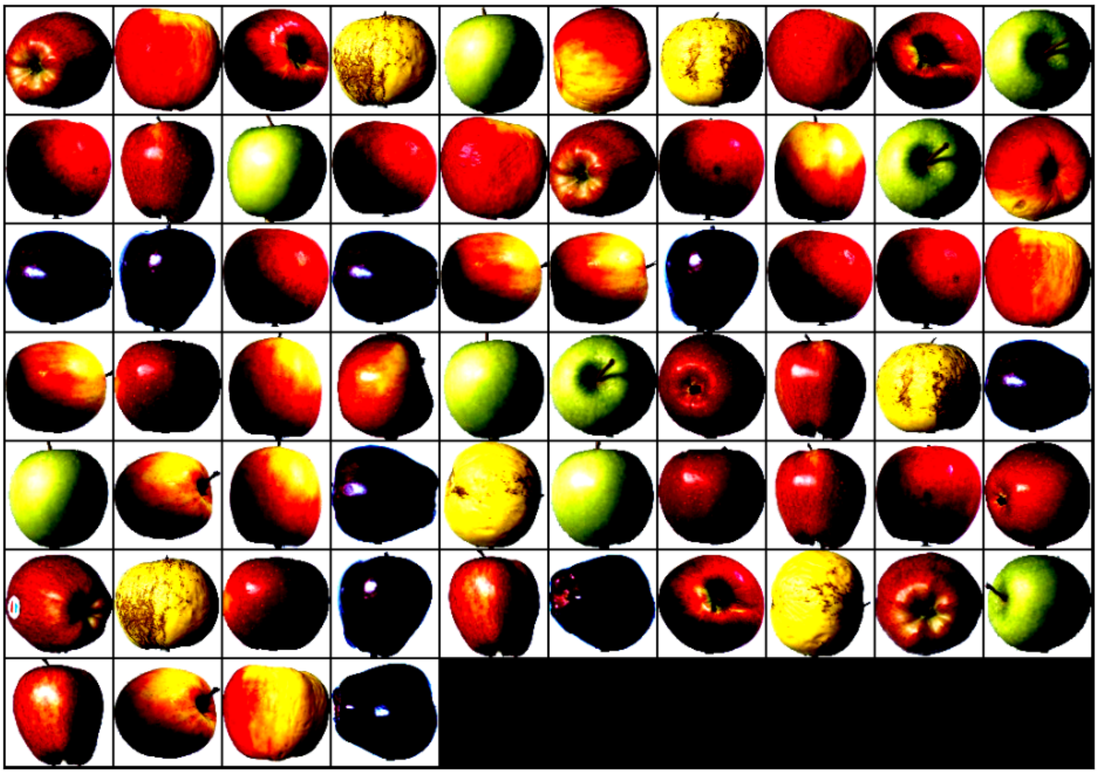


Figure 10 - Batch Visualization (Varieties of Apples)

4.4.3 Classification of Full Dataset (131 fruits and vegetables)

The concluding section encompasses the entirety of the dataset, comprising 131 fruits and vegetables. This comprehensive overview enables to combine both, analysis of different fruits and subtle variations of single category. Thanks to the size of the dataset I can test my model in more real-world example.

Final, extensive evaluation of the classification outcomes observed throughout the project. In this case, the evaluation goes beyond specific items and categories in favour of broad trends and difficulties. This section combines the knowledge gained from the earlier sections by investigating issues like model generalizability and imbalanced dataset distribution. The holistic evaluation highlights the project's real-world implications in addition to giving a macro view of its results.

The only difference in final section compared to previous two is the number of classes and samples taken to consider while training the model. Here, full dataset will be used which is 131 classes.

'Apple Braeburn',	'Cherry Rainier',	'Kiwi',	'Pear 2',	'Rambutan',
'Apple Crimson Snow',	'Cherry Wax Black',	'Kohlrabi',	'Pear Abate',	'Raspberry',
'Apple Golden 1',	'Cherry Wax Red',	'Kumquats',	'Pear Forelle',	'Redcurrant',
'Apple Golden 2',	'Cherry Wax Yellow',	'Lemon',	'Pear Kaiser',	'Salak',
'Apple Golden 3',	'Chestnut',	'Lemon Meyer',	'Pear Monster',	'Strawberry',
'Apple Granny Smith',	'Clementine',	'Limes',	'Pear Red',	'Strawberry Wedge',
'Apple Pink Lady',	'Cocos',	'Lychee',	'Pear Stone',	'Tamarillo',
'Apple Red 1',	'Corn',	'Mandarine',	'Pear Williams',	'Tangelo',
'Apple Red 2',	'Corn Husk',	'Mango',	'Pepino',	'Tomato 1',
'Apple Red 3',	'Cucumber Ripe',	'Mango Red',	'Pepper Green',	'Tomato 2',
'Apple Red Delicious',	'Cucumber Ripe 2',	'Mangostan',	'Pepper Orange',	'Tomato 3',
'Apple Red Yellow 1',	'Dates',	'Maracuja',	'Pepper Red',	'Tomato 4',
'Apple Red Yellow 2',	'Eggplant',	'Melon Piel de Sapo',	'Pepper Yellow',	'Tomato Cherry Red',
'Apricot',	'Fig',	'Mulberry',	'Physalis',	'Tomato Heart',
'Avocado',	'Ginger Root',	'Nectarine',	'Physalis with Husk',	'Tomato Maroon',
'Avocado ripe',	'Granadilla',	'Nectarine Flat',	'Pineapple',	'Tomato Yellow',
'Banana',	'Grape Blue',	'Nut Forest',	'Pineapple Mini',	'Tomato not Ripened',
'Banana Lady Finger',	'Grape Pink',	'Nut Pecan',	'Pitahaya Red',	'Walnut',
'Banana Red',	'Grape White',	'Onion Red',	'Plum',	'Watermelon'
'Beetroot',	'Grape White 2',	'Onion Red Peeled',	'Plum 2',	'Potato Red Washed',
'Blueberry',	'Grape White 3',	'Onion White',	'Plum 3',	'Potato Sweet',
'Cactus fruit',	'Grape White 4',	'Orange',	'Pomegranate',	'Potato White',
'Cantaloupe 1',	'Grapefruit Pink',	'Papaya',	'Pomelo Sweetie',	'Quince',
'Cantaloupe 2',	'Grapefruit White',	'Passion Fruit',	'Potato Red',	'Peach Flat',
'Carambula,	'Guava',	'Peach',	'Huckleberry',	'Pear',
'Cauliflower',	'Hazelnut',	'Peach 2',	'Kaki',	'Cherry 2',
				'Cherry 1',

Table 3 - List of 131 Classes in Dataset

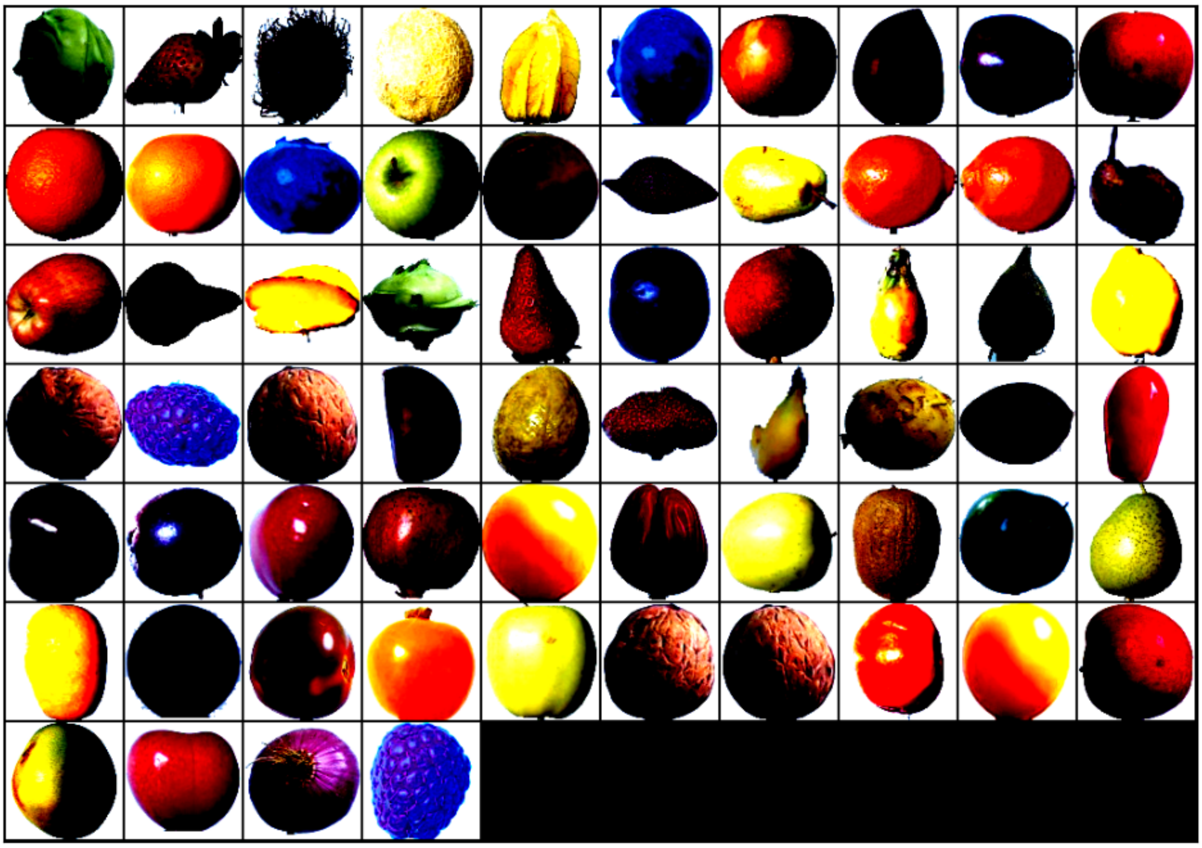


Table 4 - Batch Visualization (131 classes)

5 Result and Analysis

5.1 10 Fruits

5.1.1 CNN

In 30 Epochs model was able to achieve accuracy and F1score at 97.14%.

```
# Calculate the F1 score
f1 = f1_score(true_labels, predicted_labels, average='micro')
# Other average options: 'micro', 'weighted', or None (gives per-class F1 scores)

print("F1 score:", f1)
```

```
F1 score: 0.9713541666666666
```

```
print(f"Accuracy on the test set: {epoch_accuracy:.2%}")
```

```
Accuracy on the test set: 97.14%
```

Figure 11 - F1 and Accuracy Score

Based on chart below, we can see that model was able to learn very fast and later on it was making just some small adjustments Train Loss was below 0.0001 after 3rd epoch.

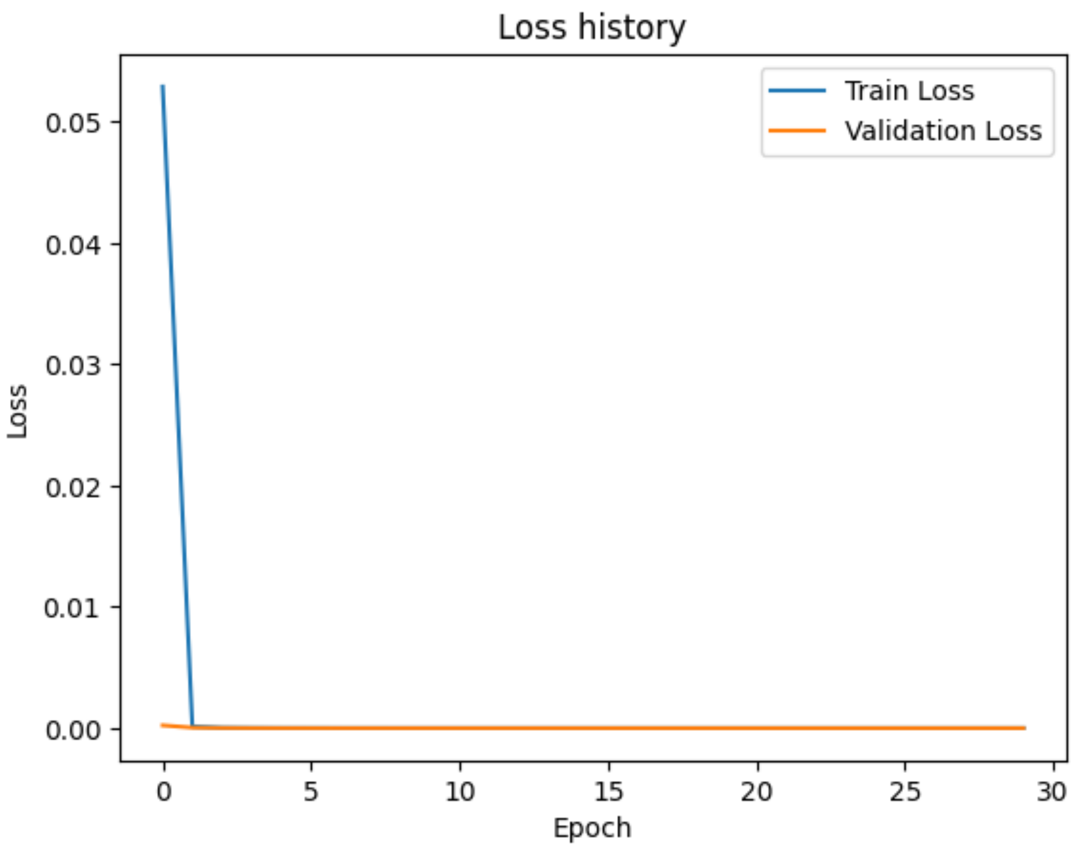


Figure 12 - Loss History

Confusion matrix Clearly shows where model is still struggling. We can see that corn is being misclassified with Lemon quite significantly. This is probably caused by the same colour of those two classes.

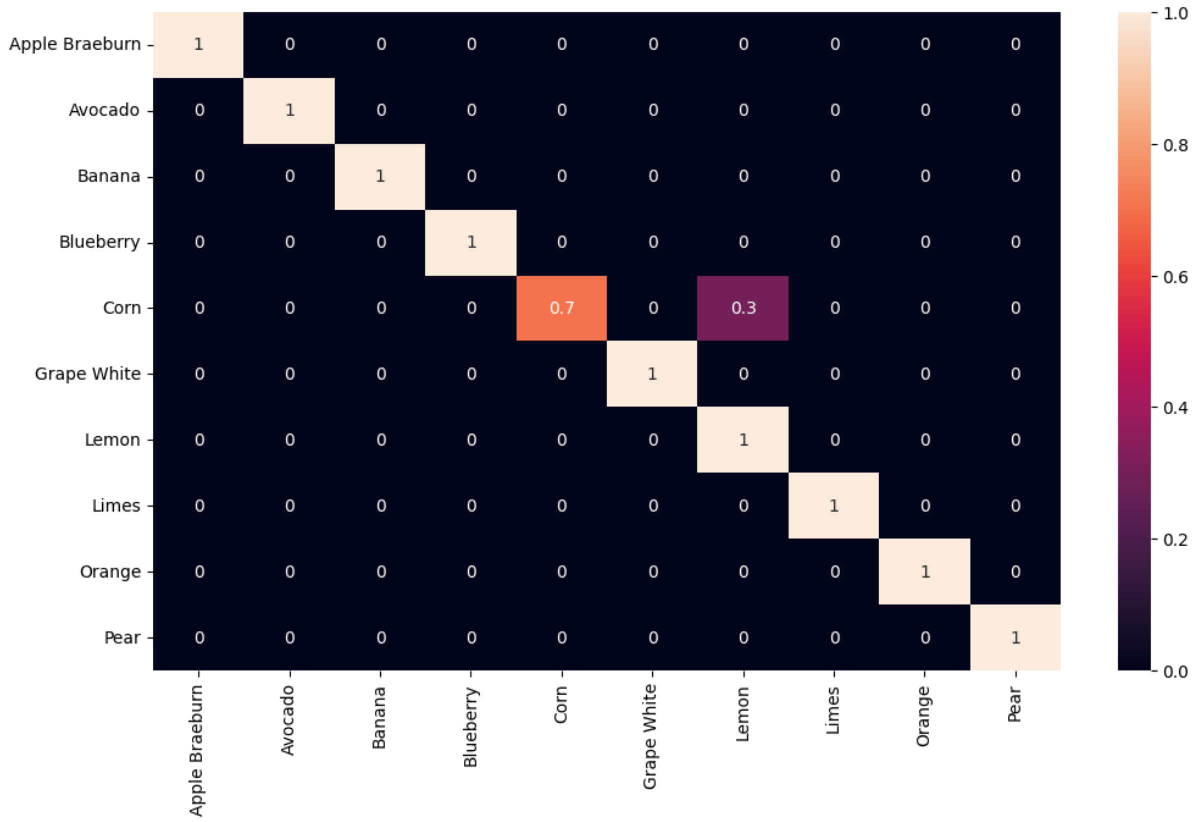


Figure 13 - Confusion Matrix Of 10 Fruits and Vegetables

5.1.2 ResNet34

In 30 Epochs model was able to achieve perfect accuracy and F1 score at 100.00% which is 2.86 better than my own approach.

```
# Calculate the F1 score
f1 = f1_score(true_labels, predicted_labels, average='micro')
# Other average options: 'micro', 'weighted', or None (gives per-class F1 scores)

print("F1 score:", f1)
```

```
F1 score: 1.0
```

```
print(f"Accuracy on the test set: {epoch_accuracy:.2%}")
```

```
Accuracy on the test set: 100.00%
```

Figure 14 - F1 and Accuracy Score (ResNet34)

Based on chart below, we can see that model was able to learn not as fast as my model due to its complexity, however it was able to achieve Train Loss equal to 0.001 at epoch 30, which turned out to have accuracy at 100.00%

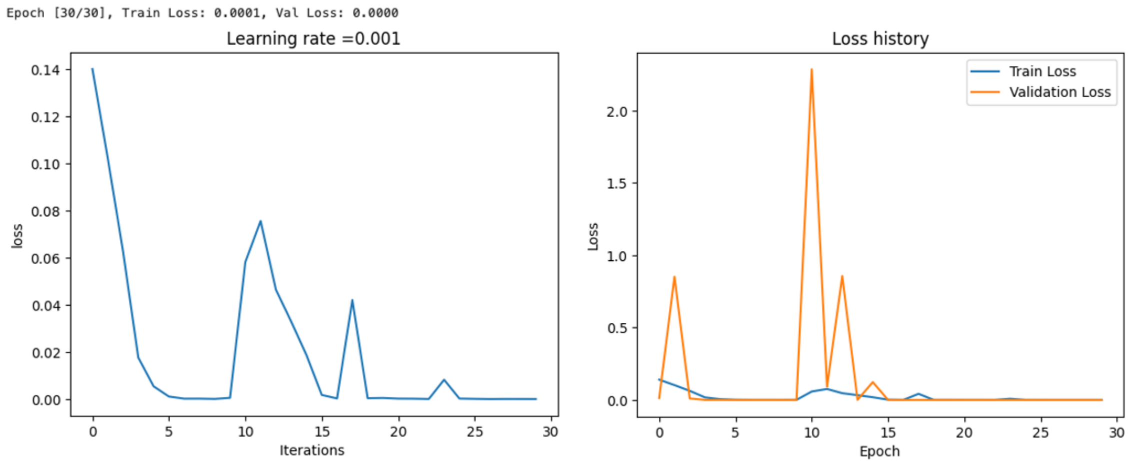


Figure 15 - Loss History (ResNet34)

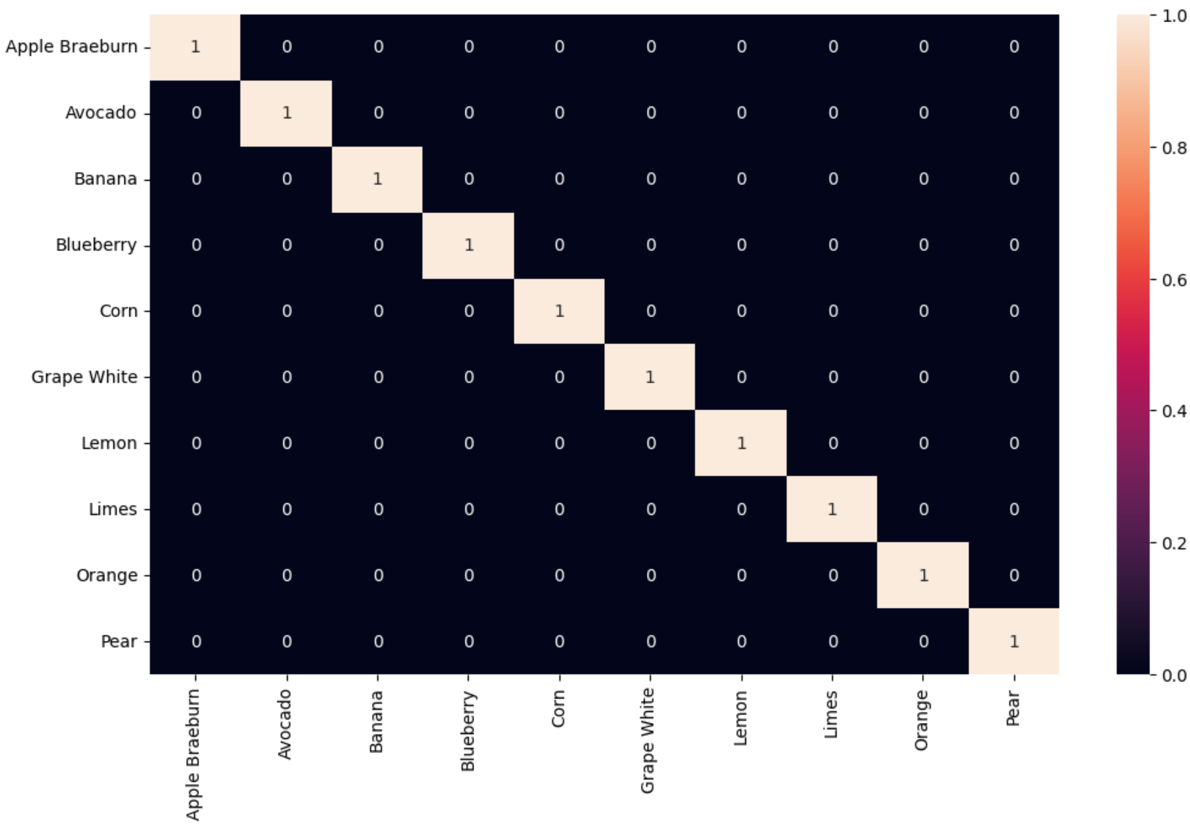


Figure 16 - Confusion Matrix (ResNet34)

5.2 8 varieties of apples

5.2.1 CNN

After 30 epochs of training model was able to achieve accuracy and F1 score at 97.73%.

```
# Calculate the F1 score
f1 = f1_score(true_labels, predicted_labels, average='micro')
# Other average options: 'micro', 'weighted', or None (gives per-class F1 scores)

print("F1 score:", f1)
```

```
F1 score: 0.97734375
```

```
print(f"Accuracy on the test set: {epoch_accuracy:.2%}")
```

```
Accuracy on the test set: 97.73%
```

Figure 17 - Accuracy and F1 Score

Loss History is basically the same as in previous task where model was able to train very quickly but could not achieve 100% accuracy.

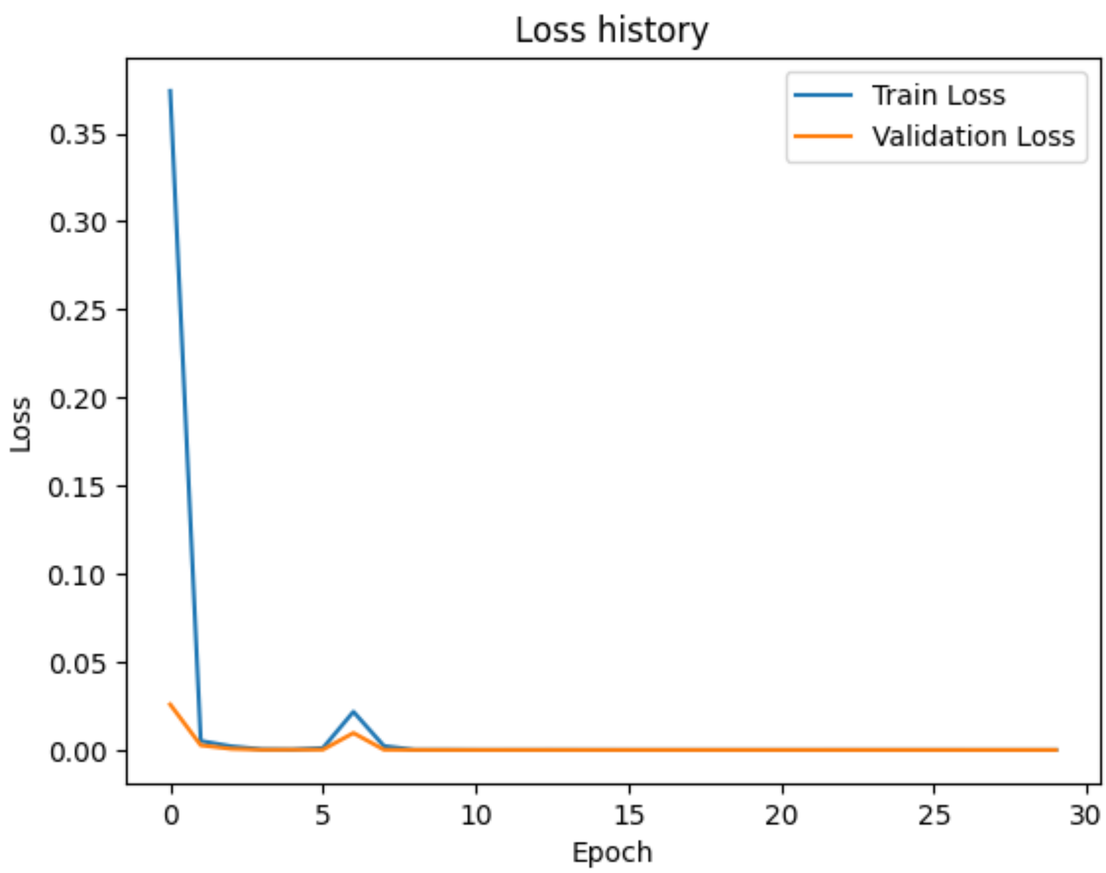


Figure 18 - Loss History

Confusion Matrix Visually represent perfect classification.

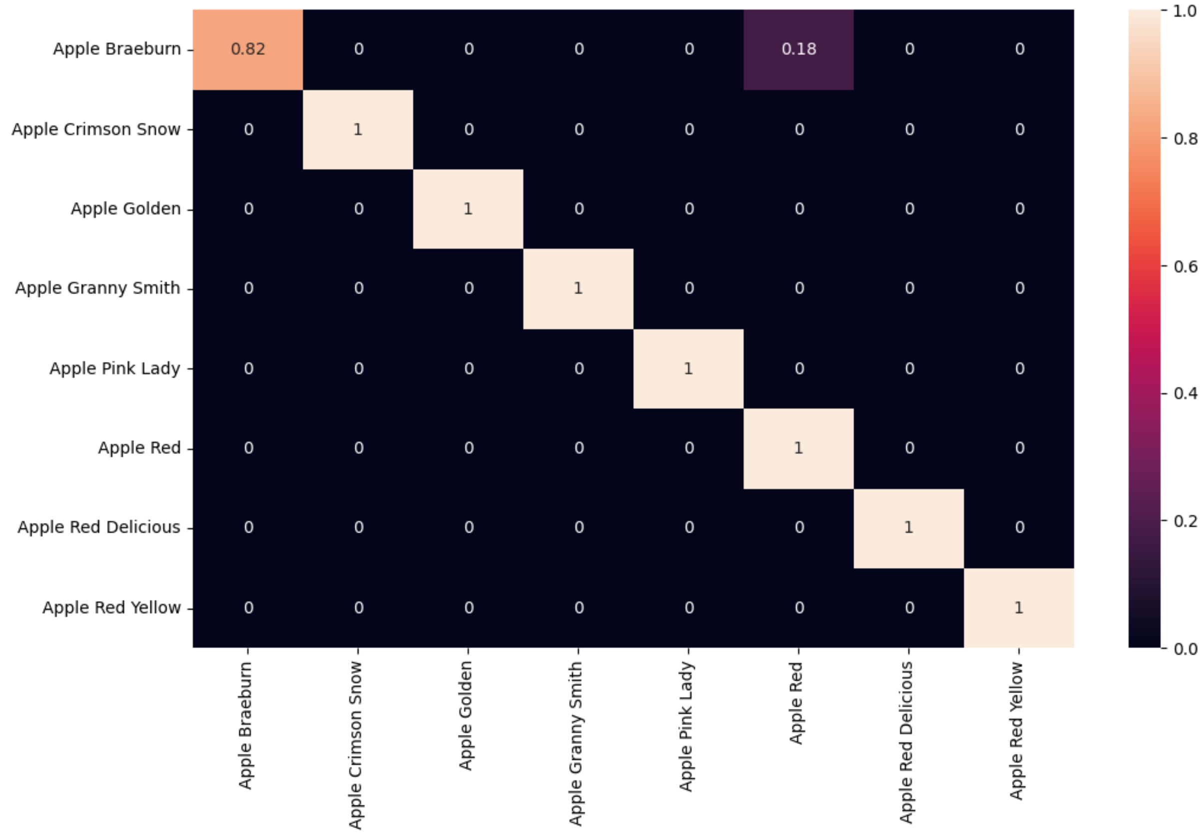


Figure 19 - Confusion Matrix

5.2.2 ResNet34

Accuracy and F1 score show perfect result as 100% with 0 misclassified samples.

```
# Calculate the F1 score
f1 = f1_score(true_labels, predicted_labels, average='micro')
# Other average options: 'micro', 'weighted', or None (gives per-class F1 scores)

print("F1 score:", f1)
```

```
F1 score: 1.0
```

```
print(f"Accuracy on the test set: {epoch_accuracy:.2%}")
```

```
Accuracy on the test set: 100.00%
```

Figure 20 - Accuracy and F1 Score (ResNet34)

ResNet34 needed more time to learn optimal way to classify apples compared to my simpler model. However, it was able to find optimal path to classify all samples correctly.

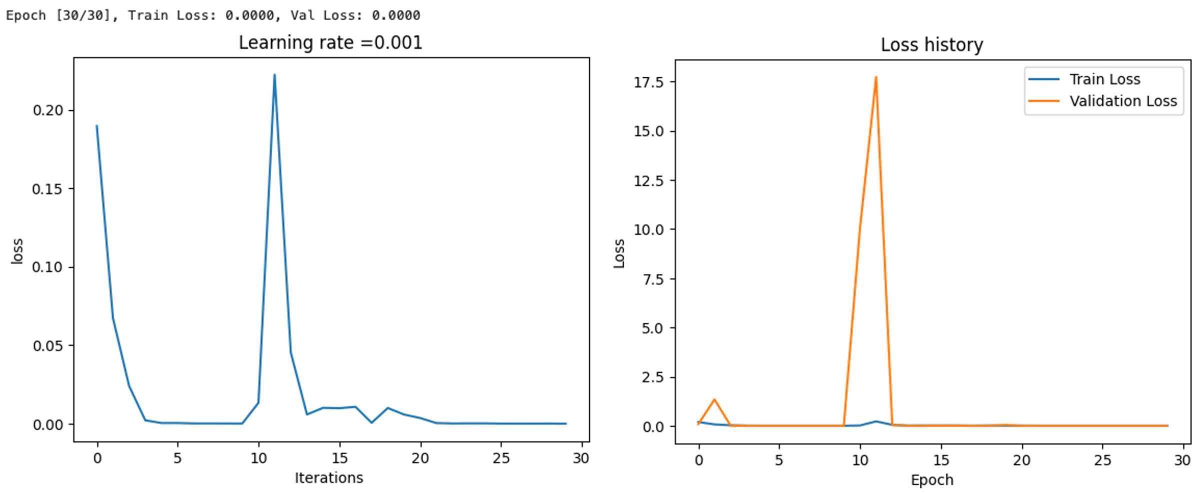


Figure 21 - Loss History (ResNet34)

Confusion Matrix Visually represent perfect classification as it has achieved 100% accuracy.

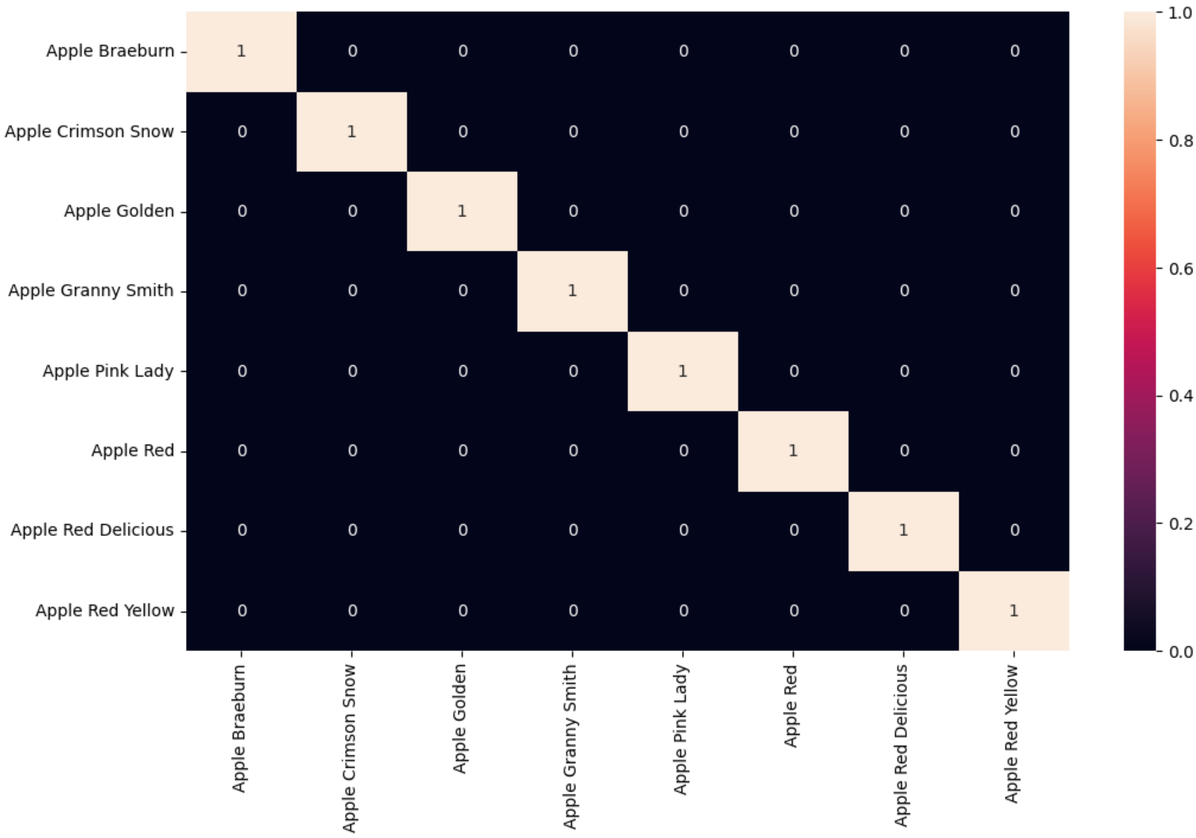


Figure 22 - Confusion Matrix (ResNet34)

5.3 Full Dataset

5.3.1 CNN

CNN approach was able to achieve 96.76% accuracy after 30 epochs. Model quickly was able to find optimal setting with minor changes throughout most of epochs. It again shows that CNN model is capable of fast learning, but cannot achieve as good results as ResNet34.

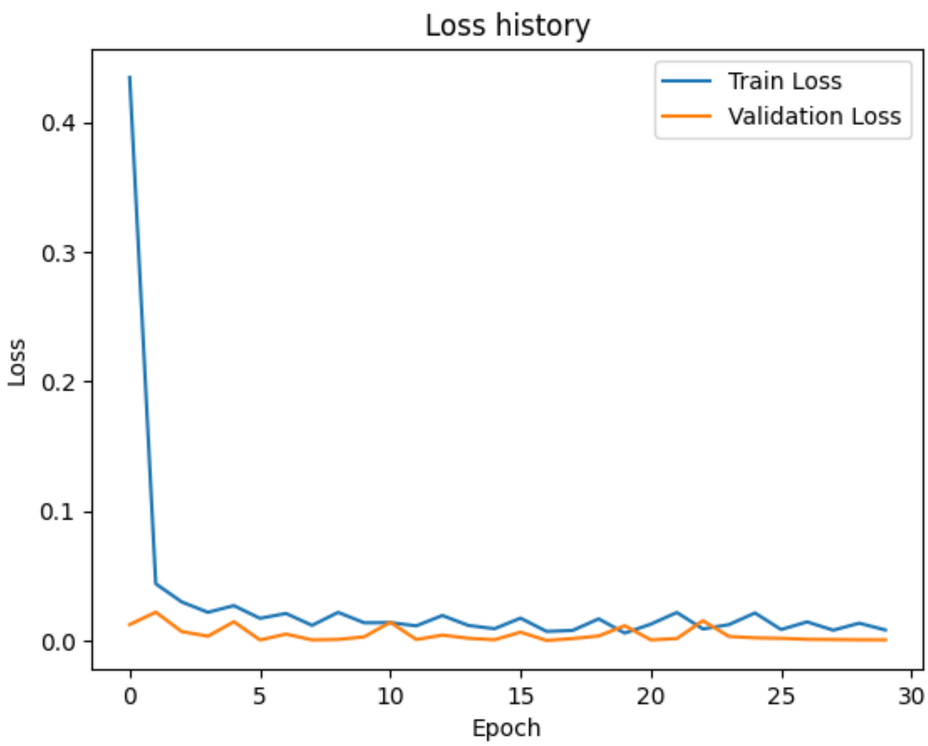
```
# Calculate the F1 score
f1 = f1_score(true_labels, predicted_labels, average='micro')
# Other average options: 'micro', 'weighted', or None (gives per-class F1 scores)

print("F1 score:", f1)
```

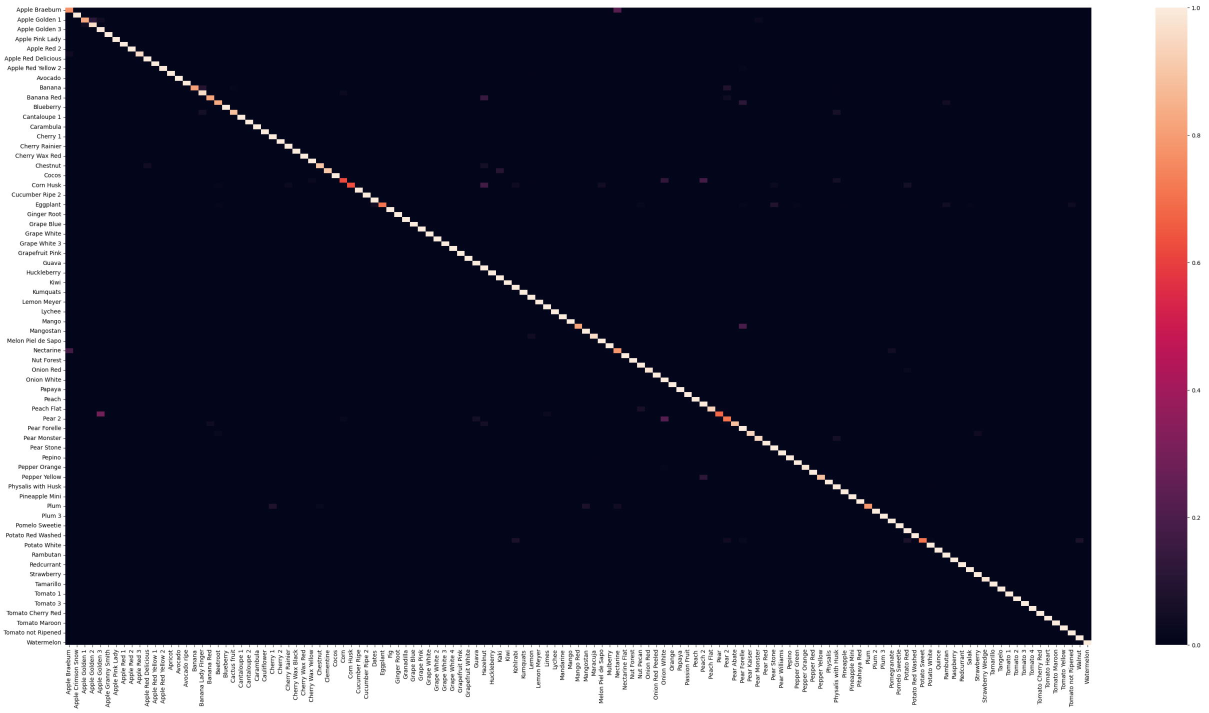
```
F1 score: 0.9676465395480226
```

```
print(f"Accuracy on the test set: {epoch_accuracy:.2%}")
```

```
Accuracy on the test set: 96.76%
```



Confusion Matrix shows that 36 labels have at least one misclassified sample.



Number of labels that have 100% accuracy

```
classified_labels = []
misclassified_labels = []
for column in df_cm.columns:
    x = df_cm[column]
    if 1.0 in df_cm[column].values:
        classified_labels.append(column)
    else:
        misclassified_labels.append(column)

print(f"Number of correctly classified labels: {len(classified_labels)}")
```

```
Number of correctly classified labels: 95
```

misclassified labels

```
: print(f"Number of incorrectly classified labels: {len(misclassified_labels)}")
  misclassified_labels
```

```
Number of incorrectly classified labels: 36
```

```
: ['Apple Braeburn',
   'Apple Golden 1',
   'Apple Golden 2',
   'Apple Golden 3',
   'Apple Pink Lady',
   'Apple Red 3',
   'Apple Red Yellow 2',
   'Apricot',
   'Banana',
   'Banana Lady Finger',
   'Banana Red',
   'Beetroot',
   'Cactus fruit',
   'Chestnut',
   'Clementine',
   'Corn',
   'Corn Husk',
   'Cucumber Ripe',
   'Eggplant',
   'Mango Red',
   'Maracuja',
   'Nectarine',
   'Onion Red',
   'Peach Flat',
   'Pear',
   'Pear 2',
   'Pear Abate',
   'Pear Kaiser',
   'Pear Monster',
   'Pepino',
   'Pepper Orange',
   'Pepper Yellow',
   'Physalis with Husk',
   'Plum',
   'Potato Sweet',
   'Strawberry Wedge']
```

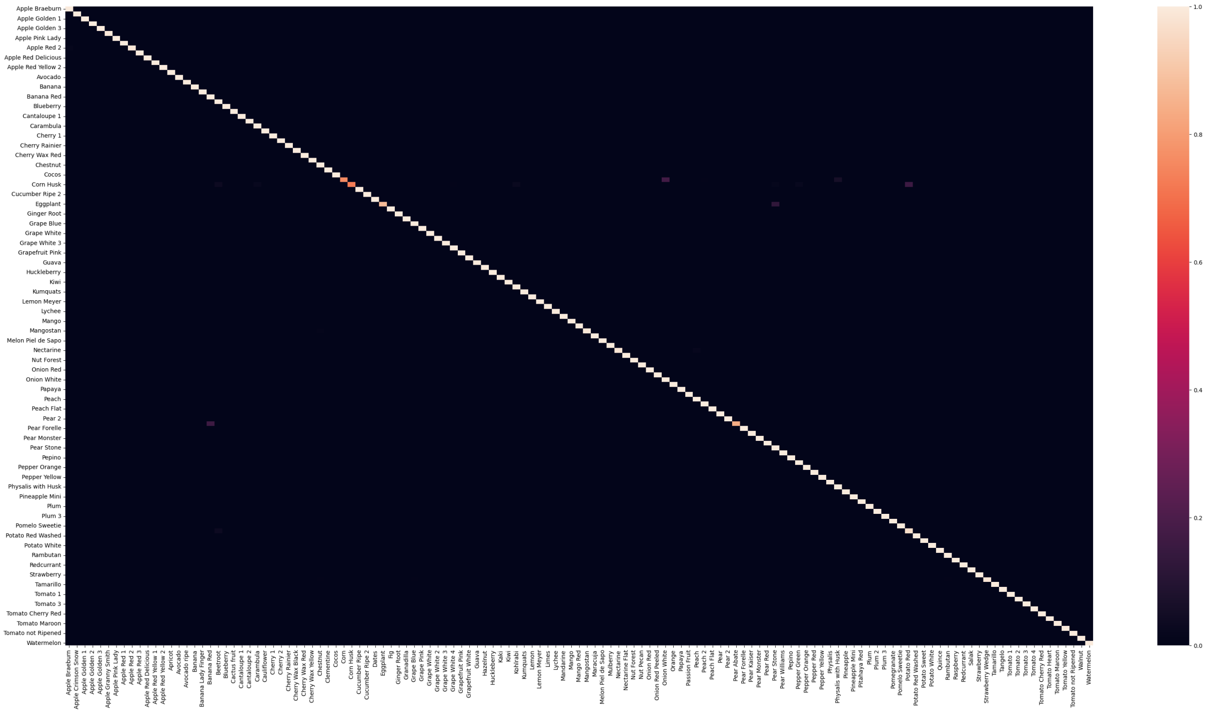
5.3.2 ResNet34

ResNet34 was able to achieve 99.38% accuracy, which is better by 2.63% than CNN approach.

```
: print(f"Accuracy on the test set: {epoch_accuracy:.2%}")
```

```
Accuracy on the test set: 99.38%
```

Confusion Matrix shows that 10 labels have at least one misclassified sample. This is 26 samples less than CNN approach.



Number of labels that have 100% accuracy

```python
classified_labels = []
misclassified_labels = []
for column in df_cm.columns:
    x = df_cm[column]
    if 1.0 in df_cm[column].values:
        classified_labels.append(column)
    else:
        misclassified_labels.append(column)

print(f"Number of correctly classified labels: {len(classified_labels)}")
```

```
Number of correctly classified labels: 121
```

misclassified labels

```python
print(f"Number of incorrectly classified labels: {len(misclassified_labels)}")
misclassified_labels
```

```
Number of incorrectly classified labels: 10
['Apple Granny Smith',
 'Apple Red 2',
 'Corn',
 'Corn Husk',
 'Eggplant',
 'Mangostan',
 'Nectarine',
 'Pear Abate',
 'Potato Red',
 'Strawberry Wedge']
```

5.4 Summary

In Each section ResNet34 outperform simple CNN model, by achieving 100% in both sections where dataset was narrowed down into one specific task. In section 3 which is supposed to show real-world example which contains 131 different classes, both different fruits and vegetables as well as different varieties of the same fruit or vegetable, it was able to achieve 99.38% compared to 96.76% accuracy of CNN simple model.

Section / Model	10 Fruits and Vegetables	8 varieties of Apples	Full Dataset
CNN	Accuracy = 97.14%	Accuracy = 97.73%	Accuracy = 96.76%
ResNet34	Accuracy = 100.0%	Accuracy = 100.0%	Accuracy = 99.38%

Table 5 - Accuracy Table

Simple CNN model had 1 misclassification in first and second section. In real-world example it misclassified 36 different fruits and vegetables which is big difference compared to ResNet34 approach which misclassified only 10 out of 131.

Section / Model	10 Fruits and Vegetables	8 varieties of Apples	Full Dataset
CNN	Misclassified = 1	Misclassified = 1	Misclassified = 36
ResNet34	Misclassified = 0	Misclassified = 0	Misclassified = 10

Table 6 - Misclassified Number of Labels

6 Conclusion

ResNet-34 offers the advantage of deeper network architectures, skip connections for better gradient flow, and residual learning capabilities, which can lead to improved performance on challenging tasks.

After comparing my own CNN model with ResNet34, it was observed that ResNet34 outperformed my model in each stage. The deeper architecture and skip connections of ResNet34 allowed it to capture more complex patterns and achieve higher accuracy in this task. However, my model demonstrated faster learning capabilities, suggesting that it required fewer training iterations to achieve optimal performance on this specific task. It was caused by a lot simpler architecture.

These results indicate that while ResNet34 excels in general fruit and vegetable classification. Further experimentation and fine-tuning may be necessary to explore the full potential of both models in different fruit classification scenarios.

6.1 Future implementation

As I analyse the project's development, it becomes clear that ResNet34's performance has surpassed my customised CNN model. As a result, my strategic approach for upcoming developments is focused on improving ResNet34 in order to maximise its efficiency in the domain of fruit and vegetable classification.

Exploration of the ResNet34 architecture's fine-tuning is a crucial area. This procedure involves the methodical adjusting of hyperparameters and learning rates, with the potential incorporation of transfer learning techniques utilising pre-trained models from substantial image datasets. Here, we aim to improve ResNet34 to better extract subtle features from our dataset.

The addition of data is a crucial aspect of improvement. I want to increase the model's adaptability across a range of real-world scenarios by enhancing the dataset through techniques like rotation, scaling, and the addition of synthetic noise. Additionally, incorporating high-resolution images or alternative modalities like hyperspectral data may improve classification accuracy.

Ensemble methods show promise as a way to improve classification performance. The predictions from various models are combined using this methodology, with ResNet34 acting as the primary building block. The potential for synergistic improvements in accuracy exists when different deep learning architectures or machine learning models are added to the ensemble.

Along with technological developments, the project's viability is a key concern. It is crucial to make sure the model is optimised for real-time deployment, especially when time-sensitive applications are taken into account. The model should also be modified to be compatible with edge computing hardware in order to increase its usefulness in scenarios where resources are limited.

The detailed tuning of ResNet34, achieved through fine-tuning, dataset augmentation, ensemble methods, and practical considerations, is the main focus of the roadmap for future implementations. This is done while maintaining the model's interpretability. These numerous improvements seek to improve the project's performance as well as its applicability and foster a deeper comprehension of the classification process within an academic context.

6.2 Research question

Can we create a CNN model and optimise it to properly classify different varieties of the same object (e.g. apple red delicious, apple granny smith etc.)?

I was successful in creating a CNN model that can distinguish between distinct apple varieties like "Red Delicious" and "Granny Smith," demonstrating how effectively it can classify different types of the same object. The model performed quite well, with an accuracy rate of 97.73 percent. However, among the several apple kinds, there was only one incident of misclassification, demonstrating the model's skill at picking up on small differences. This accomplishment highlights the potency of the CNN architecture used as well as the quality of the dataset used for the model's training and validation.

7 Definition of Terms

Artificial Intelligence (AI) - Machines that have been programmed to think and learn similarly to humans are referred to as AI systems. It includes a broad range of methods and tools aimed at giving computers the ability to carry out operations that ordinarily call for human intelligence, like comprehending natural language, spotting patterns, making judgments, and solving problems.

Machine Learning (ML) - The goal of machine learning (ML), a branch of artificial intelligence, is to create algorithms and models that enable computers to learn from data and make predictions or decisions based on it. It entails the development of systems that, without explicit programming, can learn to perform better at a given task over time.

Deep Learning - Artificial neural networks, particularly deep neural networks with multiple layers, are used in the subfield of machine learning known as deep learning (deep architectures). It works particularly well for tasks like speech and image recognition, natural language processing, and autonomous decision-making that require a lot of unstructured data. Deep learning algorithms can automatically extract complex features because they learn hierarchical data representations.

Neural Network - Artificial neural networks, particularly deep neural networks with multiple layers, are used in the subfield of machine learning known as deep learning (deep architectures). It works particularly well for tasks like speech and image recognition, natural language processing, and autonomous decision-making that require a lot of unstructured data. Deep learning algorithms can automatically extract complex features because they learn hierarchical data representations.

Supervised Learning - In supervised learning, a machine learning model is trained on a labelled dataset, where the correct output is provided for each input. The model learns to make predictions by finding patterns and relationships between the input and output data.

Unsupervised Learning - Unsupervised learning involves building a model on an unlabeled dataset and letting the algorithm find patterns, clusters, or structures on its own. Common applications include dimensionality reduction and clustering.

Reinforcement Learning - A type of machine learning called reinforcement learning teaches an agent how to decide how to maximise a reward signal. Through interaction with its environment, the agent discovers the best tactics by making mistakes.

Convolutional Neural Network (CNN) - A CNN is a particular kind of deep neural network made for handling grid-like data, like pictures and videos. Convolutional layers are used in this system to automatically learn and extract features from visual data, making it a good choice for jobs like object and image detection.

Recurrent Neural Network (RNN) - A type of neural network called an RNN is especially beneficial for sequential data, like time series or natural language. It can capture temporal dependencies because of connections that allow data to be passed from one step of the sequence to the next.

Natural Language Processing (NLP) - The goal of the AI and ML field of NLP is to make it possible for computers to comprehend, interpret, and produce human language. It includes activities like sentiment analysis, chatbot development, and language translation.

8 Bibliography

Mortimer , G. and Dootson, P. (2022) The economics of self-service checkouts, The Conversation. Available at: https://theconversation.com/the-economics-of-self-service-checkouts-78593 (Accessed: April 23, 2023).

Kats, R. (2020) For many consumers, the in-store shopping experience still has flaws, Insider Intelligence. Insider Intelligence. Available at: https://www.insiderintelligence.com/content/ for-many-consumers-the-in-store-shopping-experience-still-has-flaws (Accessed: April 23, 2023).

Schmidt-Jacobsen, T. (2023) Advantages and disadvantages of 6 retail self-checkout systems, Sprinting Retail. Available at: https://sprintingretail.com/blog/retail-self-checkout-systems/ (Accessed: April 23, 2023).

Zhao, Z.-Q. et al. (2019) Object detection with deep learning: A review . IEEE. Available at: https://ieeexplore.ieee.org/abstract/document/8627998/ (Accessed: April 25, 2023).

Kuo, M., Chan, H.-T. and Hsia, C.-H. (2021) Study on mask R-CNN with data augmentation for retail product detection . IEEE. Available at: https://ieeexplore.ieee.org/document/9651028/ (Accessed: April 29, 2023).

he, K., Zhang , X. and Ren, S. (no date) Deep residual learning for image recognition, arxiv.org. Available at: https://arxiv.org/pdf/1512.03385.pdf (Accessed: 15 June 2023).

NHS (2023) Artificial Intelligence, NHS choices. Available at: https://transform.england.nhs.uk/information-governance/guidance/artificial-intelligence/#:~:text=Examples%20of%20AI%20which%20are,numbers%20of%20people%20more%20quickly (Accessed: 04 August 2023).

H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao and D. Li,(2018) "Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment," in IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4224-423, doi: 10.1109/TII.2018.2822828.

Rutten, A. (2018) Assessing agricultural damage by wild boar using drones, wildlife. Available at: wildlife.onlinelibrary.wiley.com/doi/abs/10.1002/wsb.916 (Accessed: 20 August 2023).

A. K. Dwivedi, A. K. Singh and D. Singh,(2022) "An Object Based Image Analysis of Multispectral Satellite and Drone Images for Precision Agriculture Monitoring," IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, pp. 4899-4902, doi: 10.1109/IGARSS46834.2022.9884674. (Accessed: 20 August 2023).

Bawack, R.E., Wamba, S.F., Carillo, K.D.A. et al. Artificial intelligence in E-Commerce: a bibliometric study and literature review. Electron Markets 32, 297–338 (2022). https://doi.org/10.1007/s12525-022-00537-z (Accessed: 22 August 2023).