Neural Network-based Fruit Classification

Aiden Harris
hrraid003@myuct.ac.za
University of Cape Town
South Africa

Stuart Heath
hthstu002@myuct.ac.za
University of Cape Town
South Africa

Joe Whales
whljos001@myuct.ac.za
University of Cape Town
South Africa

1 INTRODUCTION

This project explores the development of a neural network-based system for fruit classification, addressing the challenge of accurately categorizing 9 distinct fruit classes using image data. We implement and compare two architectures: a simple feedforward neural network as a baseline and an advanced Convolutional Neural Network (CNN). Using a dataset sourced from Kaggle and split into training (70%), validation (15%) and test (15%) sets, we employ systematic approaches including hyperparameter optimization, learning rate scheduling, data augmentation and regularization techniques to enhance model performance. Our evaluation uses comprehensive metrics such as precision, recall, F1-score and confusion matrices, while also considering ethical implications related to data collection, environmental impact and potential socioeconomic consequences of automated fruit classification systems.

1.1 Problem Formulation

We formulate the task as a multi-class classification problem, where the goal is to correctly identify and categorize fruits into one of 9 classes: Apple, Orange, Kiwi, Mango, Pineapple, Strawberries, Banana, Cherry, and Watermelon. This involves processing image data to predict the correct fruit category, a task that requires the model to discern subtle differences in color, texture and shape among various fruits.

1.2 Dataset Description

The dataset, sourced from Kaggle [1], consists of images representing 9 different fruit classes. To ensure robust model evaluation and prevent overfitting, we have split the data into training (70%), validation (15%), and test (15%) sets. This partition allows for effective model training, hyperparameter tuning and unbiased performance assessment.

1.3 Usefulness and Difficulty

The fruit classification task presents several challenges that contribute to its difficulty, with examples that can be seen in Figure 2:

- Limited Dataset Size: The relatively small dataset increases the risk of overfitting, requiring careful model design and regularization techniques.
- Data Quality: As the images are web-scraped, they may vary in quality and consistency, presenting challenges for accurate classification.
- Image Variability: The dataset includes images with various settings, color variations, and multiple fruits per image, increasing the complexity of the classification task.
- Cross-contamination: The presence of multiple fruit types in single images can lead to mislabeling and confusion during classification.

Figure 1: A Sample of the 9 different categories: Apple, Banana, Cherry, Kiwi, Mango, Oranges, Avocados, Pineapples, Strawberries and Watermelons

- Class Imbalance: Cleaner, more recognizable classes may dominate the model's predictions, potentially leading to misclassification of less distinct fruit types.

Despite these challenges, the fruit classification system has several potential applications that underscore its usefulness:

- Agricultural Automation: The system could be employed in automated harvesting, sorting, and quality control processes in the agricultural sector.
- Image Segmentation and Labeling: The techniques developed for this project could be extended to more complex image segmentation and labeling tasks.
- Quality Control: In food processing and packaging industries, such a system could automate quality control processes, ensuring consistency and reducing human error.
- Research and Education: The project serves as a valuable case study for machine learning and computer vision research, particularly in dealing with real-world, imperfect datasets.

1.4 Ethical Considerations

While fruit classification may appear benign at first glance, several ethical considerations warrant attention:

Figure 2: A subset of the cross-contamination, variability and quality of the data

- Data Collection and Bias: The dataset was created by web scraping, as such, there could be errors or discrepancies within the dataset and could also make use of license-able content. This method could also include potential biases in the dataset could impact model performance across different fruit varieties. .
- Environmental Impact: The substantial computational resources required for training deep learning models raise concerns about energy consumption and carbon footprint as these large models require large amounts of electricity to be trained, which can be seen by some of the largest models like ChatGPT [4].
- Socioeconomic Implications: Widespread adoption of automated fruit classification systems could potentially impact agricultural workers, necessitating careful consideration of socioeconomic consequences as it could result in major job losses [3].

2 MODELS

Our fruit classification project uses two neural networks: a basic neural network as our baseline and an advanced Convolutional Neural Network (CNN). This section explains both models' structures and our design choices.

2.1 Data representation/preparation

When reading in our data/images, we pre-process our images in order to make our images consistent with one another. For both the training and test set, we normalize, resize and covert all images to tensors. Resizing is important as cnn models require fixed input sizes as varying sizes can lead to issues during the training phase [2]. Resizing simplifies data handling, avoids size distortion and improves general performance. Other machine Learning projects involving image recognition tend to use 224×224 image size for resizing. Afterwards, we convert the images in its re-scaled format to a PyTorch tensor which is required for the package. From here, we normalise these images using the mean and standard deviation values that were obtained from the complete ImageNet dataset, which is common practice among machine learning problems with images and the default values used by the PyTorch package.

We also manually cleaned the dataset to remove some of the cross contaminated images, however this was only done after the initial testing and evaluation of the model. This cleaned version of the dataset was only tested on the best model. This was done to investigate the impact of using clean and high quality data on model performance.

2.2 Baseline: Simple Neural Network

We implemented a feedforward neural network as our baseline model. This architecture provides a benchmark to measure improvements achieved by more complex models in later experiments.

The baseline neural network consists of three fully connected layers with ReLU activations. It processes standard RGB images with dimensions 224x224x3. To prepare these images for the network, we flatten the 3-channel data into a 1D vector of 150,528 elements. This allows the Neural Network to treat each pixel as an individual and independent feature to be identified by the Neural Network. This vector is then processed through the following layers:

- Input layer: 150,528 neurons (flattened 224x224x3 image)
- Hidden layer 1: 512 neurons with ReLU activation
- Hidden layer 2: 256 neurons with ReLU activation
- Output layer: 10 neurons with softmax activation (one for each fruit class)

This simple architecture, while not optimized for image data, provides a baseline performance metric. It allows us to understand the complexity of the classification task and sets a lower bound for expected performance.

2.3 Advanced Model: Convolutional Neural Network (CNN)

Our advanced model is a Convolutional Neural Network (CNN), which improves upon the baseline. We chose a CNN because it excels at image classification, effectively capturing spatial relationships and local features in images.

To further enhance our CNN's performance, we implemented several additional techniques:

- Data augmentation to increase our training set and improve generalization
- Regularization methods like dropout and batch normalization to reduce overfitting
- Learning rate scheduling to optimize the training process

These improvements help our model learn more effectively from the available data and achieve better classification results.

2.3.1 ***Model Architecture.*** Our CNN architecture consists of the following key components:

- Convolutional Layers: Three layers with increasing depth (64, 128, and 256 filters), each followed by ReLU activation. These learn hierarchical features from the images.

- Max Pooling Layers: Applied after each convolutional layer to reduce spatial dimensions and increase the receptive field.
- Adaptive Average Pooling: Adapts the output to a fixed size, allowing flexibility in input image dimensions.
- Fully Connected Layer: Maps the learned features to the 10 fruit classes for final classification.

2.3.2 ***Additional Improvements.*** To enhance the model's performance and generalization capabilities, we incorporated several advanced techniques:

- Data Augmentation: We applied random rotations, horizontal flips and color jittering to the training images. This helps the model learn invariant features and reduces overfitting.
- Learning Rate Scheduling: We implemented Cosine Annealing with Warm Restarts as well as Cosine Annealing LR, however we found that not using a learning rate scheduler gave the best results.
- Regularization: Techniques such as dropout and L_2 regularization were employed to further combat overfitting and improve generalization. Dropout prevents neurons from having similar behaviour where they would be highly correlated with each other while L_2 regularisation discourages large weights dominating by balancing out the weights across the network.
- Batch Normalization: We incorporated batch normalization layers in our model architecture. Batch normalization normalizes the inputs to each layer for every mini-batch, which helps to address the internal covariate shift problem. This technique not only accelerates training by allowing higher learning rates but also provides a slight regularization effect, further improving the model's generalization capabilities.

The CNN model improved upon the baseline performance through its ability to capture spatial features in images. The implementation of convolutional layers, data augmentation, and regularization techniques enhanced the model's accuracy and generalization capabilities. While effective for this fruit classification task, further optimization may yield additional improvements

3 MODEL VALIDATION AND OPTIMIZATION

3.1 Hyperparameter Tuning

We employ a systematic approach to hyperparameter optimization, exploring key parameters such as learning rate, batch size and weigh decay. Our implementation allows for easy experimentation with different values. We generate several configuration files based on several sets of adjustable parameters, a bat file is then used to run the model with each of these configuration files in succession. Our final parameters chosen were: 32 batch size, with an initial learning rate of 0.003, dropout rate of 0.5 and a weight decay of 0.0001.

3.2 Learning Rate Scheduling

To further optimize training, we implement learning rate scheduling using Cosine Annealing. This technique gradually decreases the learning rate over time, following a cosine curve, which can

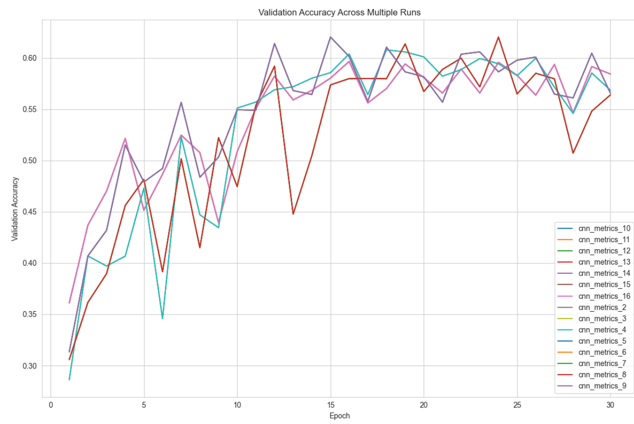

Figure 3: Validation Accuracy of Different Parameter Setups

help improve convergence and potentially lead to better final performance.

The addition of learning rate scheduling resulted in a final test f1 score of 0.7626, an improvement over using the model with a static learning rate which gave a final f1 score of 0.7344. We also tested using Cosine Annealing with warmup restarts, however this performed worse over 200 epochs than both a static learning rate and the normal cosine annealing scheduler.

- small dataset therefore lots of regularisation - train test and validation sets

4 EVALUATION

4.1 Evaluation Metrics

We employ a comprehensive set of metrics to evaluate our models:

- Precision: The total amount of fruit predicted correctly over the total number of fruit that were predicted at that type. For example, this would be the number of apples that were correctly identified as apples over the total number of images that were identified as apples.
- Recall: Measures the total number of fruit precicted correctly over the total number of that fruit their actually is. For example, this would be the number of apples identified correctly over the number of apple images there actually are.
- F1-score: Measures the overall performance of our networks prediction capabilities.
- Confusion Matrix: Offers a detailed view of class-wise predictions

These metrics are calculated using sklearn's precision recall f1 score support function, providing a holistic view of model performance across all classes.

For each model that was tested we collected training and validation scores at each epoch. We then used a simple python script to compile these results in a markdown report for each set of results obtained.

4.2 Results

Based on the provided reports, we can summarize the performance of our models as follows:

- Simple Neural Network:
  - Test Loss: 8.1413
  - Test Precision: 0.4734
  - Test F1 Score: 0.4216
- Convolutional Neural Network (CNN):
  - Test Loss: 0.7813
  - Test Precision: 0.7782
  - Test F1 Score: 0.7626
- CNN Using Cleaned Dataset:
  - Test Loss: 0.7811
  - Test Precision: 0.7977
  - Test F1 Score: 0.7800

These results demonstrate a significant improvement in performance when moving from the simple neural network to the CNN architecture, with the CNN achieving higher precision and F1 score, and lower loss on the test set. Our CNN correctly identified 78% of all fruits in our test set while our simple neural network only predicted less than 50%. The F1 score indicates that the CNN performed well both in terms of precision and recall. This means that it is better than our simple neural network at making correct positive predictions and identifying the actual fruits. The increase in scores resulting from using a cleaned version of our original dataset also demonstrates the importance of having clean and high quality data.

5 ANALYSIS OF MODEL PERFORMANCE

5.1 Comparative Analysis

The CNN model demonstrates superior performance compared to the simple neural network baseline. This improvement can be attributed to the CNN's ability to capture spatial hierarchies and local patterns in the image data, which is crucial for distinguishing between different fruit classes. We will analyse the loss function and accuracy figures below.

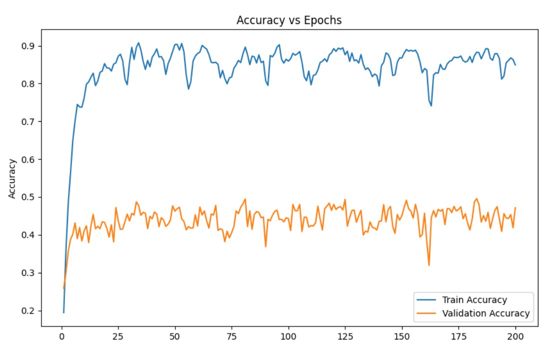

Figure 4: Neural Network Model Validation vs Training Accuracy

For the neural network model accuracy, we see that both the validation and training accuracy increases quite dramatically whereas

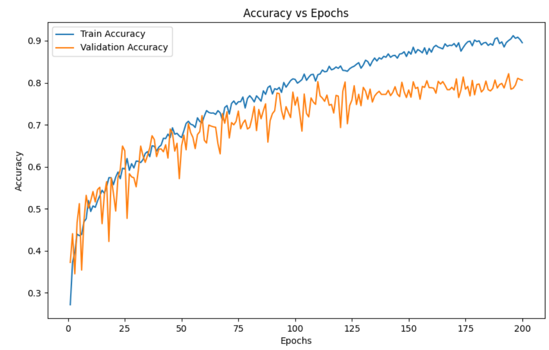

Figure 5: CNN Training vs Validation Error

the CNN has a more gradual and consistent increase in comparison. Both the training and validation accuracy appears to be quite inconsistent as their range within a 20 epoch span is 20 % which indicates that the model is over fitting the data as the results are quite inconsistent. This is also supported by the poor validation accuracy of less than 50 %. The CNN performs much better and does not overfit the data. This is seen in the steady increase of both accuracies while still maintaining great accuracy of 80% which is substantially better than our standard Feed Forward Neural Network.

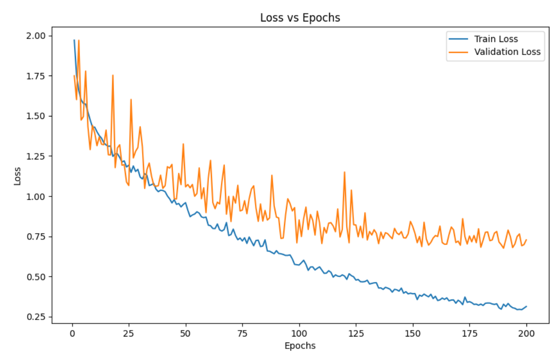

Figure 6: CNN Training vs Validation Loss Function

Similar to our Accuracy figures, we see that our CNN performs better in terms of analysing the loss function results. The Neural Network clearly overfits as seen in the high validation set loss where it ranges from 4-8 where our CNN has a much lower loss value of 0.75. This shows again that our methods of data augmentation, regularisation and dropout improves performance immensely of our fruit classifier with these methods being included in our CNN.

5.2 Error Analysis

If we look at the confusion matrices, fruits with distinct features such as kiwis, pineapples and cherry appear to be predicted well across both models, but it is clear that the convoluted neural network performs better than the neural network. The Neural network appears to predict cherries for a lot of the fruits with at least 8

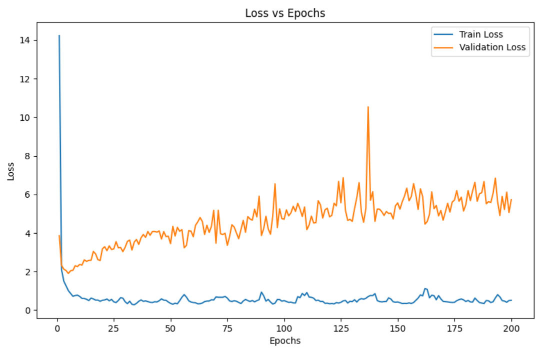

Figure 7: NN Loss Function

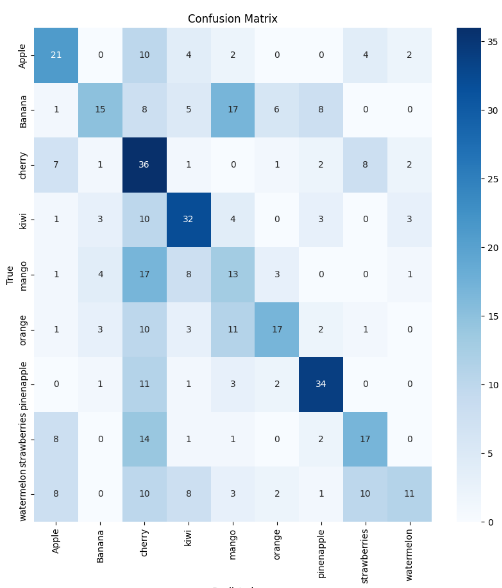

Figure 8: Neural Network Confusion Matrix

incorrect cherry predictions per fruit, The CNN performs performs better in this regard as only the precision was 78% compared to the neural networks precision of 29% for cherries. Watermelons were not being predicted in the first model nearly enough as it had a recall of 81% in the CNN model compared to 21% in the NN. Therefore, it is clear to see that the CNN is overall a better model than the NN, as it has better precision, recall and overall better accuracy than the NN.

5.3 Reproducing results

6 LIMITATIONS AND FUTURE WORK

Our CNN model for fruit classification demonstrated significant improvements over the baseline neural network, achieving a test precision of 0.7977 and an F1 score of 0.7800. The model's strengths

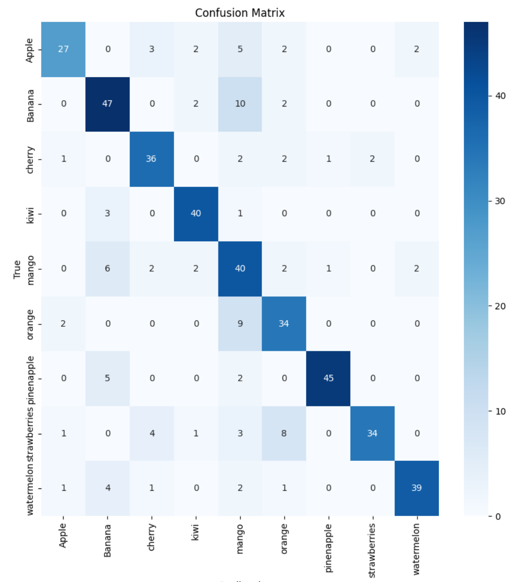

Figure 9: CNN Confusion Matrix

include effective capture of spatial features, successful implementation of data augmentation and regularization techniques and substantial performance gains over the baseline.

However, there are still several limitations to the current implementation. The relatively small, web-scraped dataset limits generalization. Class imbalance affects performance across categories, while image quality variability and cross-contamination in multi-fruit images lead to misclassifications. Despite applying several regularization techniques, we still see some overfitting.

To address these limitations and enhance the models performance further, future work could focus on:

- Expanding and refining the dataset, including targeted collection for the underrepresented classes
- Exploring advanced architectures such as EfficientNet or Vision Transformers
- Implementing ensemble methods to improve the robustness and accuracy of the system
- Using transfer learning from large-scale datasets to potentially improve performance

7 CONCLUSION

Our CNN model for fruit classification demonstrated significant improvements over the baseline neural network, achieving a test precision of 0.7977 and an F1 score of 0.7800. The project highlighted the effectiveness of CNNs in capturing spatial features in images and the benefits of data augmentation and regularization techniques. Through systematic hyperparameter tuning and architecture optimization, we developed a model capable of distinguishing between 9 different fruit classes with reasonable accuracy, despite the challenges posed by our dataset. Notably, when we used a cleaned version of our dataset, we observed a substantial increase in model accuracy, demonstrating the importance of high-quality, clean data in improving classification performance.

Our experiments also revealed interesting insights into the model's behavior. The confusion matrices showed that fruits with distinct features, such as kiwis and pineapples, were consistently well-classified across both models. The implementation of Cosine Annealing learning rate scheduling improved our final test F1 score from 0.7344 to 0.7626, demonstrating the impact of adaptive learning rates on model performance. While limitations in dataset size and quality persist, our results provide a solid foundation for future work. Potential areas for improvement include more advanced architectures, implementing ensemble methods and using transfer learning from large-scale datasets. These enhancements could lead to a more robust and reliable fruit classification system, better suited for real-world agricultural automation and quality control applications.

REFERENCES

[1] Karim Abdulnabi. 2022. Fruit Classification (10 Classes). https://www.kaggle.com/datasets/karimabdulnabi/fruit-classification10-class Accessed: 2024-10-12.

[2] EITCA Academy. n.d.. Why is it necessary to resize the images to a square shape? https://eitca.org/artificial-intelligence/eitc-ai-dltf-deep-learning-with-tensorflow/using-convolutional-neural-network-to-identify-dogs-vs-cats/introduction-and-preprocessing/examination-review-introduction-and-preprocessing/why-is-it-necessary-to-resize-the-images-to-a-square-shape/#:~:text=CNN%20models%20require%20fixed%2Dsize,data%20handling%20and%20processing%20steps. Accessed: 2024-10-17.

[3] Conterra Ag Capital. 2024. How Will Artificial Intelligence (AI) Disrupt the Farming Industry? https://www.conterraag.com/artificial-intelligence-and-agriculture-will-ai-disrupt-the-farming-industry/ Accessed: 2024-10-12.

[4] World Economic Forum. 2024. Generative AI's Energy Consumption and Emissions Impact. https://www.weforum.org/agenda/2024/07/generative-ai-energy-emissions/ Accessed: 2024-10-12.

8 APPENDIX

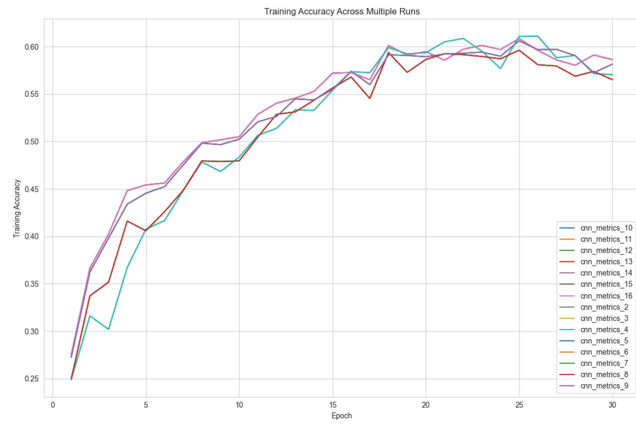

Figure 10: Train Accuracy of Different Parameter Setups