Fruit Image Classification Using Deep Learning
Final Project Report

Mian Azan Farooq
Matriculation Number: 12433773

1 Introduction

Image classification is an important area of computer vision, and its applications are widely used in the real world, such as food identification, product quality control, and automated checkout systems. The objective of this project was to design, train, and deploy a deep learning model capable of classifying fruit images using the Fruits360 dataset.

The project was carried out in multiple stages, beginning with a basic convolutional neural network and evolving into a complete system that included training, performance evaluation, and a user-friendly demo application. This report describes the problem addressed, its importance, the proposed solution, results obtained, and the lessons learned during the project.

2 Problem Statement

The task addressed in this project is multi-class image classification, where the model must identify the correct fruit category given an input image. The dataset contains many visually similar classes, such as different apple varieties, making the task challenging even for advanced deep learning models.

This problem is significant because accurate fruit classification can support applications such as automated grocery checkout, agricultural monitoring, and food quality assessment. Manual classification is time-consuming and prone to errors, making automation essential.

3 Why This Is a Challenging Problem

Several factors make this task difficult:

- High visual similarity between different fruit varieties
- Sensitivity to lighting conditions, image angles, and image quality
- A large number of classes (over 130 categories)
- Risk of overconfident predictions on unseen or unfamiliar inputs

Traditional rule-based image processing methods struggle with these challenges, which makes deep learning a more flexible and effective solution.

4 Proposed Solution

To address the problem, a deep convolutional neural network based on the EfficientNetB0 architecture was implemented. EfficientNet was selected due to its strong performance and efficient use of parameters, as well as its proven success in image classification tasks.

The final solution includes:

- A training pipeline using TensorFlow and Keras
- A dataset loading and preprocessing module
- A trained EfficientNet-based classifier
- A Streamlit-based demo application for user interaction
- A confidence-based mechanism for out-of-distribution detection

5 Model Architecture

The model architecture consists of the following components:

- EfficientNetB0 backbone
- Global Average Pooling layer
- Fully connected softmax output layer

The model processes RGB images resized to 224×224 pixels. Pre-trained ImageNet weights were not used in order to maintain full control over the training process and avoid input-shape inconsistencies.

6 Training Pipeline

The Fruits360 dataset was used for both training and validation. The dataset was split into 80% training data and 20% validation data. All images were normalized to the range $[0, 1]$.

6.1 Error Metric and Target

Categorical accuracy was used as the evaluation metric, and categorical cross-entropy was used as the loss function. The initial target was to achieve at least 90% validation accuracy.

6.2 Training Details

- Optimizer: Adam
- Learning rate: 1×10^{-4}
- Batch size: 32
- Number of epochs: 10

The training process was successful, and the model achieved high validation accuracy for most fruit categories.

7 Inference and Demo Application

A demo application was developed using Streamlit to demonstrate the trained model. The application allows users to upload an image and receive the predicted fruit label along with a confidence score.

The inference process consists of:

1. Uploading an image

2. Resizing and preprocessing the image
3. Generating a prediction using the trained model
4. Evaluating the confidence score

8 Out-of-Distribution Detection

To prevent incorrect predictions on unseen inputs, an out-of-distribution detection mechanism based on prediction confidence was implemented. If the maximum confidence score is below 60%, the input is classified as an unknown object.

During testing:

- Fruit images from the dataset produced high-confidence predictions (e.g., Golden Apple with over 99% confidence)
- Non-fruit images such as playing cards resulted in low-confidence predictions

This approach significantly improves the reliability and robustness of the system.

9 Key Insights and Takeaways

Key lessons learned during the project include:

- Data preprocessing consistency is crucial for correct model performance
- Training and inference image sizes must match to ensure valid predictions
- Rejecting low-confidence predictions is essential for real-world deployment
- Debugging deep learning systems often takes more time than initial model development

10 What I Would Do Differently

If given the opportunity to redo the project, I would:

- Introduce automated testing earlier in the development process
- Experiment with additional data augmentation techniques
- Allocate more time to systematic hyperparameter tuning

These changes would further improve model robustness and performance.

11 Time Investment

The total time spent on the project was approximately 360 hours:

- Dataset exploration: 5 hours
- Model development: 272 hours
- Debugging and fixes: 72 hours
- Demo application development: 4 hours
- Evaluation and testing: 4.5 hours
- Documentation: 2 hours

12 Conclusion

This project successfully demonstrates the application of deep learning to a real-world image classification problem. An end-to-end system was designed, trained, evaluated, and deployed as an interactive demo application. The project highlights both the effectiveness of deep learning and the importance of careful system design, testing, and validation.

Overall, the project meets all course requirements and provides a solid foundation for future work in applied deep learning.