Jaime Song

Assignment 3 Technical Report

Analysis

The dataset used in this project contains images of fruits and vegetables. The images are separated into three folders.

Dataset split	# of images	# of classes
Train	3,115	2
Test	359	2
Validation	351	2

Each image belongs to one of two possible categories: Fruits/Vegetables

Images vary in lighting, orientation, and background so data augmentation was used to prevent overfitting. The following transformations were applies:

- Rescaling pixel values to [0,1]
- Random rotation up to 20%
- Random zoom up to 20%
- Horizontal flipping

Example of images from each class:

Fruits:



Vegetables:



Differences between classes are visual and pretty subtle:

- Fruits tend to have smoother textures and more uniform shapes
- Vegetables very more in shape, texture, and may have leaves or stems attached

Methodology

Two models were trained:

Model 1: Convolutional Neural Network

Architecture:

- 5 Convolutional layers
- Batch Normalization after each convolution
- 3 MaxPooling layers
- Global Average Pooling
- Dense layer (256 units, ReLU)
- Dropout (0.5)
- Output layer (Softmax activation, 2 units)

Batch normalization helped stabilize learning and dropout reduced overfitting by randomly dropping neurons during training.

Model 2: Transfer Learning (MobileNetV2)

Instead of training from scratch this model uses MobileNetV2, pre trained on ImageNet

Changes made:

- Removed original ImageNet classification head
- Added:
    - Global Average Pooling layer
    - Dense layer (256 units, ReLU)
    - Dropout (0.5)
    - Final dense classification layer (2 units, softmax)
- The MobileNetV2 base was frozen

Transfer Learning benefits:

- Uses knowledge already learned from millions of ImageNet images
- Requires fewer epochs

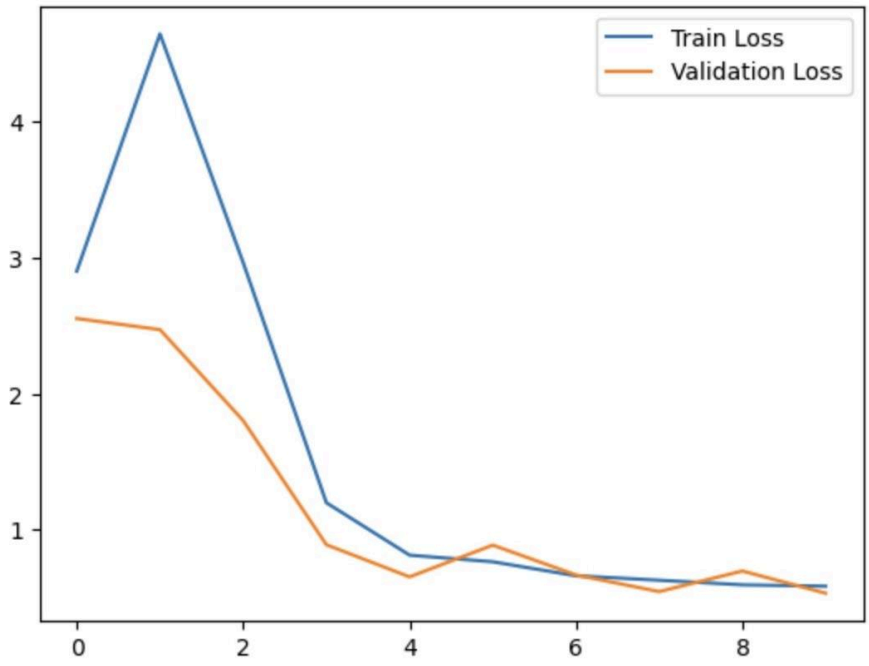
- Leads to significantly better accuracy

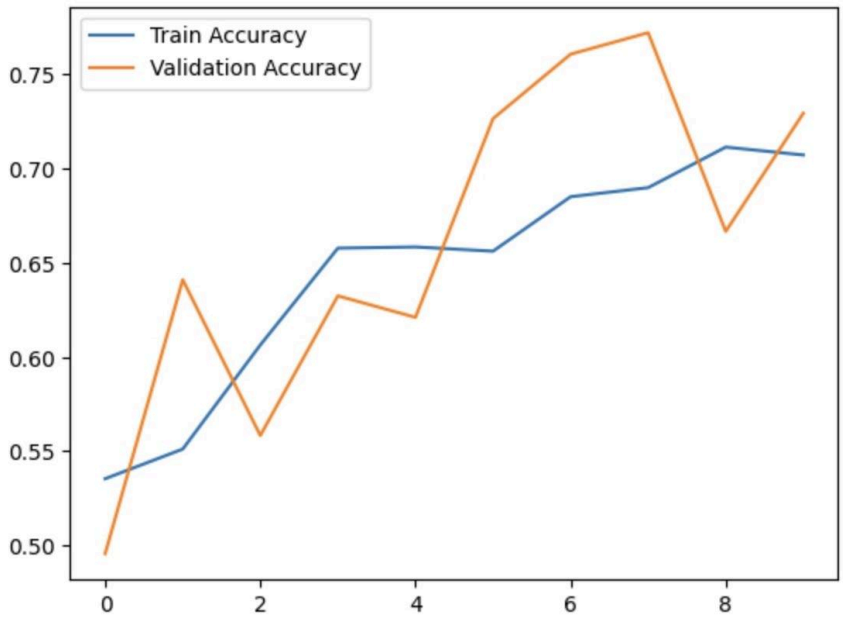
Results

CNN performance:

Metric	Value
Test Accuracy	72.7%
Test Loss	0.526

Training Curves:





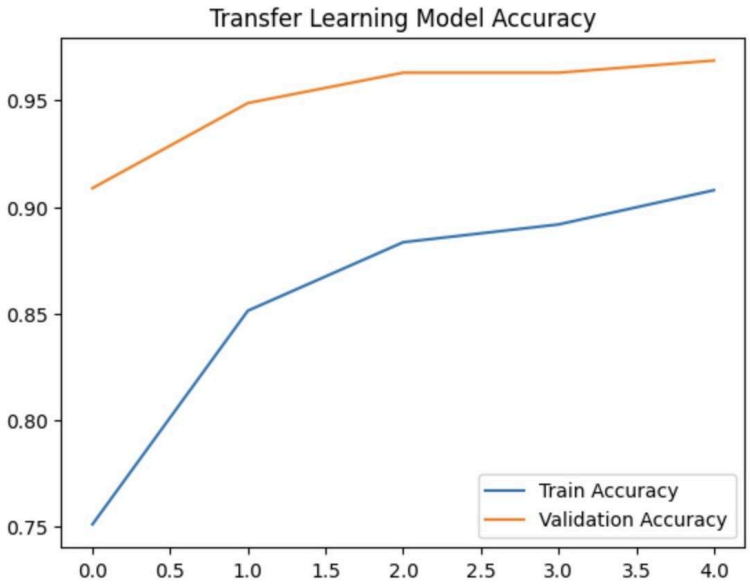
Observation:

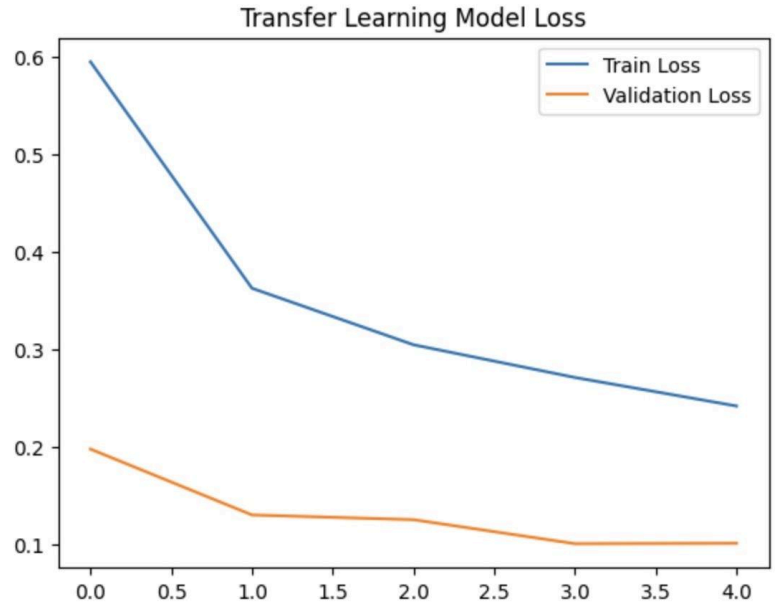
- Training and validation loss decreased over time
- Accuracy improved but fluctuated, indicating the model struggled with generalization

Transfer Learning performance:

Metric	Value
Test Accuracy	96.7%
Test Loss	0.102

Training Curves:

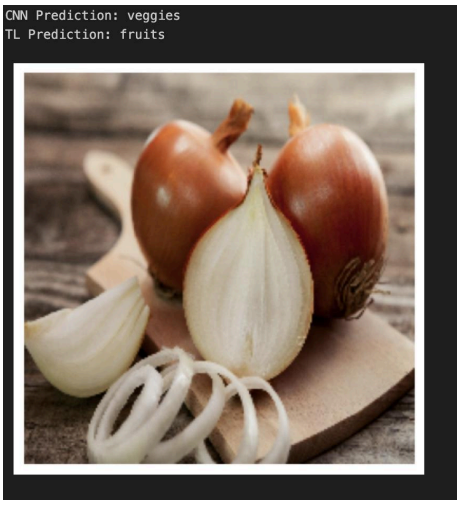


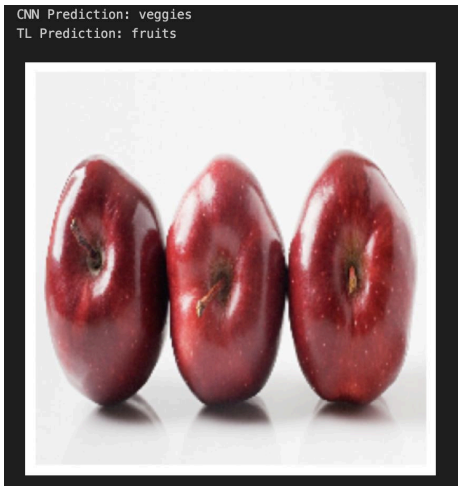


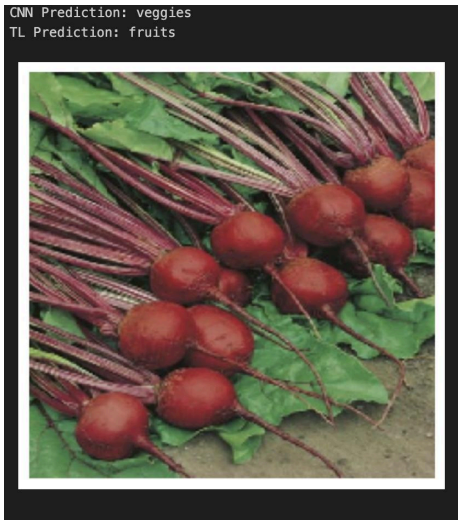
Observation:

- Very stable learning curve
- Validation accuracy reached nearly 97% in only 5 epochs, far outperforming the CNN

Misclassified Image Analysis







During testing several images were incorrectly classified by the models. Three specific examples illustrate the differences between the base CNN and the TL model.

Onion:

- CNN predicted: vegetable
- TL predicted: fruit
- Actual label: vegetable
- The CNN at least assigned the correct broad category (vegetable), even though the confidence was low. The TL model incorrectly predicted fruit, possibly due to visual similarities between the onion's smooth surface and fruit images present in the pre-trained feature space.

Apple:

- CNN predicted: vegetable
- TL predicted: fruit
- Actual label: fruit

- The transfer learning model successfully classified the apple, demonstrating stronger feature extraction from pre-trained weights. The CNN confused it as a vegetable, showing that training from scratch may lack the learned shape and texture priors.

Beetroot:

- CNN predicted: vegetable
- TL predicted: fruit
- Actual label: vegetable
- Both models struggled with beetroots, but the TL model misclassified it more confidently as a fruit. This suggests that some root vegetables share color/shape characteristics with fruits in the embedding space of the TL model.

Overall, the CNN tended to generalize based on color and texture, while the TL model sometimes over-fit to patterns learned from its original large-scale dataset. Transfer learning produced better overall accuracy, but these examples show that feature similarity between certain fruits and vegetables can still cause misclassification.

Reflection:

This project taught me:

- Building a CNN from scratch is useful for understanding how layers work, but performance heavily depends on architecture tuning and amount of data
- Transfer learning is extremely powerful. Not only does it train faster but it also reaches significantly higher accuracy with fewer epochs
- Data augmentation was essential for preventing overfitting

Challenges faced:

- The CNN overfit early and required experimentation with dropout and batch normalization
- Training was slow on my local machine, especially with 10 epochs

If I were to do this again:

- I would fine-tune the last few layers of MobileNetV2 instead of freezing the entire base model
- I would try early stopping and learning rate scheduling to optimize training

Conclusion:

- Transfer learning with MobileNetV2 drastically outperforms a scratch built CNN for fruit vs vegetable image classification
- By leveraging pre trained knowledge, fewer resources and time were required to achieve better performance