# COMPUTER VISION

# PROJEC 2

**1.Dataset**

The dataset is stored on Google Drive and organized into three main subsets:

- Training dataset: /apple/training
- Validation dataset: /apple/validation
- Test dataset: /apple/testing

Each subset contains class-specific folders, where each folder represents one apple category.

Image resolution: 224 × 224 pixels Batch
size: 32

Number of classes: 2

**2. Data Loading and Pipeline Optimization**

Images are loaded using TensorFlow's image_dataset_from_directory function, which automatically assigns labels based on directory names.

To improve training efficiency and performance:

- The training dataset is shuffled.
- Caching is applied to reduce disk I/O.
- Prefetching is enabled to optimize GPU utilization.

**3. Handling Class Imbalance**

To mitigate potential class imbalance within the training dataset, class weights are computed using the compute_class_weight function from scikit-learn.

This approach:

- Assigns higher importance to underrepresented classes.
- Reduces bias toward dominant classes.
- Improves fairness and robustness of the classifier.

Computed class weights: {0: np.float64(1.2047904191616767), 1: np.float64(0.8547153780798641)}

**4. Data Augmentation**

To enhance generalization and reduce overfitting, data augmentation is applied to the training data only. The following techniques are used:

- Random horizontal flipping
- Random rotation (up to 20%)
- Random zoom (up to 20%)

These transformations simulate real-world variations in apple images.

**5. Model Architecture**

The model is built using ResNet50 as a fixed feature extractor with the following configuration:

- Pre-trained on ImageNet
- Top classification layers removed
- Base model weights frozen during training
- Custom Classification Head
- Global Average Pooling layer
- Fully connected layer with 256 neurons and ReLU activation
- Dropout layer with a rate of 0.3
- Output layer with Softmax activation for multi-class classification
- Input preprocessing:
  Images are processed using preprocess_input, which matches the original ResNet50 training distribution and improves performance compared to simple normalization.

## 6. Model Compilation

The model is compiled with the following settings:

- Optimizer: Adam
- Learning rate: 0.0001
- Loss function: Sparse Categorical Crossentropy
- Evaluation metric: Accuracy

## 7. Training Strategy

The model is trained for a maximum of 30 epochs using an adaptive training strategy:

- Early Stopping: Stops training if validation loss does not improve for several epochs and restores the best weights.
- Model Checkpoint: Automatically saves the best-performing model based on validation accuracy.
- Class weights are applied during training to address dataset imbalance.

Number of epochs completed: 30

## 8. Training Results

```
...
Epoch 28/30
126/126 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9990 - loss: 0.0032
Epoch 28: val_accuracy did not improve from 1.00000
126/126 ━━━━━━━━━━━━━━━━━━━━ 789s 6s/step - accuracy: 0.9990 - loss: 0.0032 - val_accuracy: 1.0000 - val_loss: 0.0117
Epoch 29/30
126/126 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9990 - loss: 0.0040
Epoch 29: val_accuracy did not improve from 1.00000
126/126 ━━━━━━━━━━━━━━━━━━━━ 851s 7s/step - accuracy: 0.9990 - loss: 0.0040 - val_accuracy: 1.0000 - val_loss: 0.0075
Epoch 30/30
126/126 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9982 - loss: 0.0061
Epoch 30: val_accuracy did not improve from 1.00000
126/126 ━━━━━━━━━━━━━━━━━━━━ 855s 7s/step - accuracy: 0.9982 - loss: 0.0060 - val_accuracy: 0.9851 - val_loss: 0.0251
```

**Accuracy**

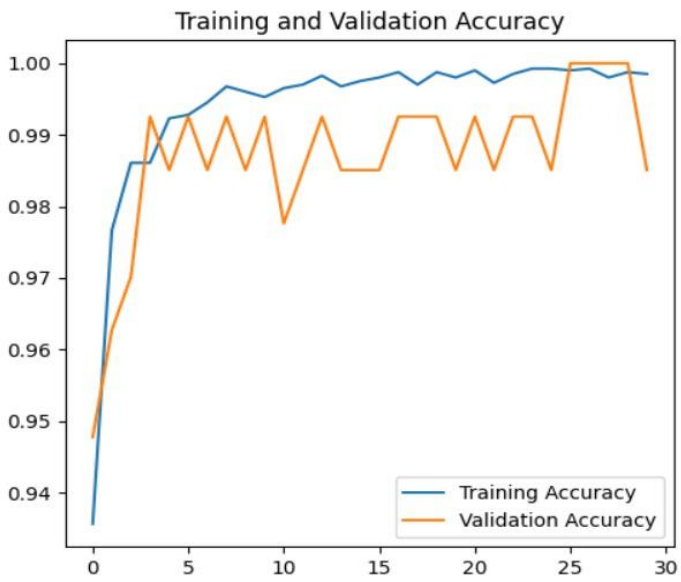
Final Training Accuracy: 0.9982

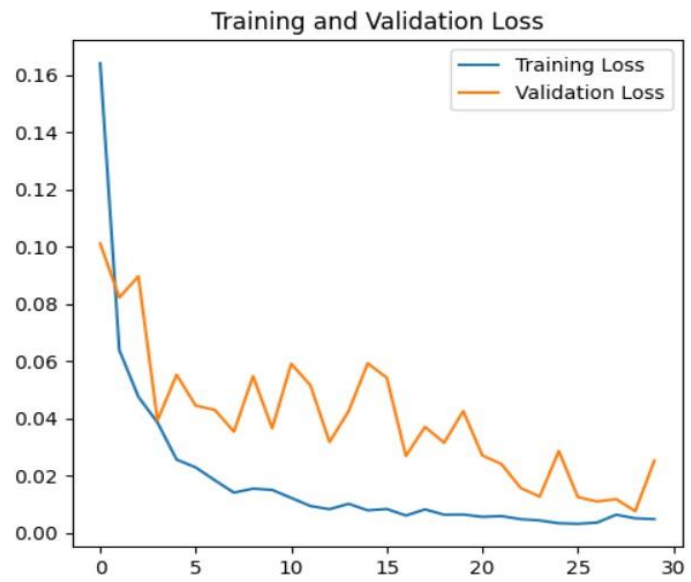
Final Validation Accuracy: 0.9851

**Loss**

Final Training Loss: 0.0060

Final Validation Loss: 0.0251



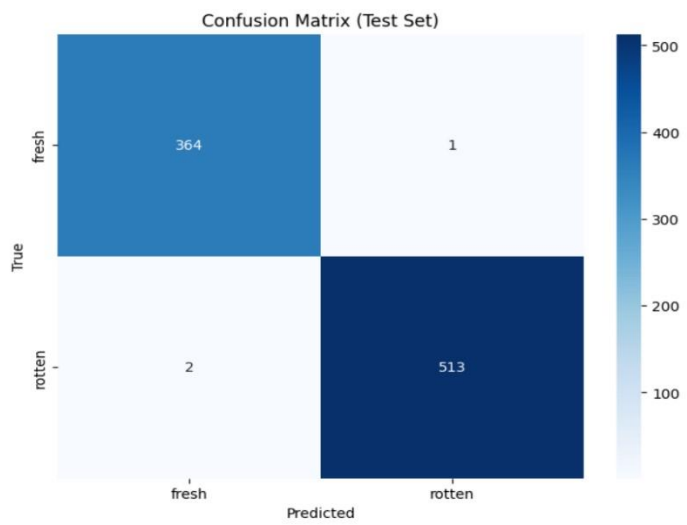




## 9. Model Evaluation on Test Dataset

The trained model is evaluated on a separate test dataset that was not used during training or validation.

**Confusion Matrix**

The confusion matrix illustrates the distribution of correct and incorrect predictions across all classes.




**Classification Report**

The classification report provides detailed metrics including precision, recall, and F1-score for each class.

| Classification Report: | precision | recall | f1-score | support |
|---|---|---|---|---|
| fresh | 0.99 | 1.00 | 1.00 | 365 |
| rotten | 1.00 | 1.00 | 1.00 | 515 |
| accuracy | | | 1.00 | 880 |
| macro avg | 1.00 | 1.00 | 1.00 | 880 |
| weighted avg | 1.00 | 1.00 | 1.00 | 880 |

**10. Conclusion**

This project demonstrates that transfer learning with ResNet50, combined with data augmentation and class imbalance handling, can achieve strong performance in apple image classification.

The model shows good generalization on unseen test data and is suitable for real-world applications.