Maryam Ahmed

Machine Learning 1 - CS 07556

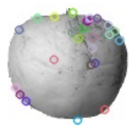
Programming Assignment

Questions 1 to 5

The dataset comprises images of fruits categorized into 10 classes: Apple Golden 1, Apple Red 1, Apple Red Delicious, Avocado, Ginger Root, Limes, Maracuja, Papaya, Potato Red, and Rambutan. These classes represent specific types of fruits. The dataset is intended for tasks such as image classification, where the objective is to develop models capable of accurately categorizing fruit images into one of these 10 classes. [2]

In this first step, we perform feature extraction from each image in the training data using the scale-invariant feature transform (SIFT) algorithm. SIFT helps in identifying and describing local features in an image, which are invariant to scale and rotation changes. Each keypoint detected by SIFT is described by a 128-dimensional feature vector. We obtain multiple keypoints from each image, which collectively represent its unique visual characteristics.

To visualize the keypoints, I plot them on one image from the Apple Golden 1 class. This visualization helps in understanding how SIFT detects distinctive points in an image, used for object recognition and image matching.



The 100-dimensional vector obtained after performing K-means clustering on the keypoints dataset represents a compact encoding of the visual information contained in the training images. It is like summarizing each image with a list of 100 key characteristics that represent what makes each image unique.

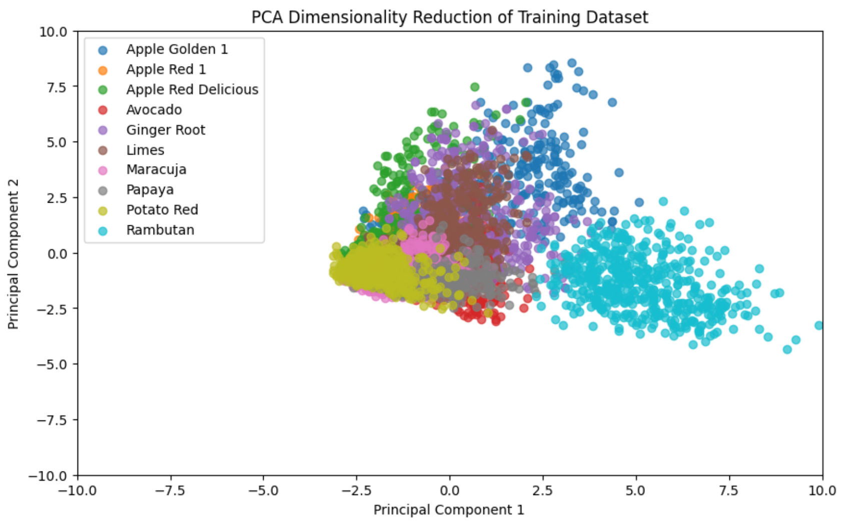
Example vector representing a fruit image from the training dataset: [0 0 1 3 0 0 3 2 1 0 0 1 0 0 0 1 0 0 3 0 0 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 1 0 1 1 0 0 2 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 4 3 0 0 0 0 2 0 2 1 1 1 0 1 0 2 0 2 0 2 0 0 0 0 2 1 0 1 0 1 1 0 1 3 0 1 0]

Each element of the vector corresponds to the average position of all the data points assigned to that cluster, obtained from the K-means algorithm, capturing the most significant features shared among the keypoints across the entire dataset. By quantizing the keypoints into clusters, we reduce the dimensionality of the data while preserving the essential characteristics necessary for distinguishing between different images.

In summary, this vector serves as a condensed representation of the image content, for efficient processing and analysis in subsequent stages of the SVM model.

Shape of training dataset D: (4665, 100)

The shape of the training dataset is (4665, 100), indicating that it contains 4665 sample fruit images and each sample is represented by a 100-dimensional vector. This shape suggests that there are 4665 instances in the dataset, with each instance having 100 features.



Plot of the 2D points using 10 different colors/symbols for data from the 10 classes: each point in the plot represents an instance from the dataset after performing Principal Component Analysis (PCA) dimensionality reduction. In other words, each point corresponds to a specific fruit image in the original dataset.

The PCA algorithm projects the original high-dimensional data onto a lower-dimensional space while preserving the maximum variance. Therefore, each point in the plot represents the same data instance as in the original dataset, but now represented in a two-dimensional space defined by the principal components.

The colors assigned to each point distinguishes the data from different fruit classes. This color mapping helps visualize how the different classes are distributed and separated in the reduced-dimensional space. Understanding the separation of classes helps to assess the effectiveness of the classification algorithm. If classes are well-separated in the reduced-dimensional space plot, it suggests that the data is easily separable, leading to better classification performance.

The clustering or overlapping of most classes indicates that they share similar feature representations in the reduced-dimensional space. This suggests that these classes may be closely related in terms of their features.

The Rambutan class (light blue) is positioned farther away from the clustered regions suggesting that it has distinct features compared to the other classes. This separation implies that the feature representations of Rambutan fruits are less similar to those of the other classes, leading to a more distinct cluster.

We perform the same steps on the testing dataset:

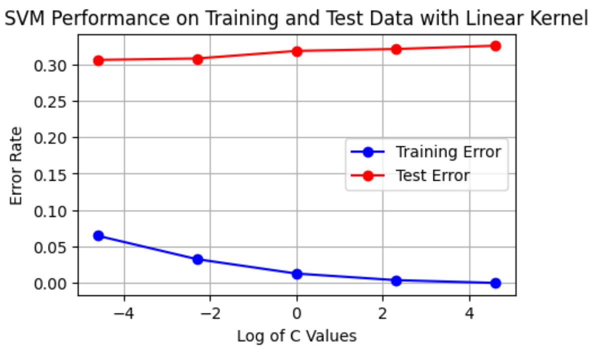
Example vector representing Image A for testing data: [0 5 0 0 0 0 0 0 2 1 0 0 0 0 0 1 0 1 0 0 3 0 0 0 0 4 0 0 0 0 0 2 1 1 2 2 1 0 1 3 1 0 0 2 0 0 0 0 0 2 1 4 0 2 0 2 0 0 0 1 6 1 0 2 1 0 0 0 0 0 0 2 10 0 1 0 1 2 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0]

Shape of testing dataset D: (1542, 100)

Linear SVM Model Performance with Varying C Values: the SVM model was trained using different C parameter values to determine their impact on model performance. Here are the results obtained:

Table 1: Linear SVM Model Performance with Varying C Values

C Value	Training Accuracy	Testing Accuracy	Train Error	Test Error
0.01	0.9353	0.6939	0.0647	0.3061
0.1	0.9674	0.6920	0.0326	0.3080
1.0	0.9871	0.6816	0.0129	0.3184
10	0.9961	0.6790	0.0039	0.3210
100	1.0000	0.6744	0.0000	0.3256



- C Value: 0.01

  Training Accuracy: 0.9353

  Testing Accuracy: 0.6939

  With a very small value of C, the model demonstrates relatively high training and testing accuracies. However, the model might be underfitting, as indicated by the significant difference between training and testing accuracies.

- C Value: 0.1

  Training Accuracy: 0.9674

  Testing Accuracy: 0.6920

  As the C value increases, the training accuracy improves slightly. However, the testing accuracy starts to decline suggesting the potential for overfitting.

- C Value: 1.0

  Training Accuracy: 0.9871

  Testing Accuracy: 0.6816

  With a moderate C value, the training accuracy continues to improve, but the testing accuracy further declines. This indicates that the model is likely overfitting to the training data, as it fails to generalize well to unseen data.

- C Value: 10

  Training Accuracy: 0.9961

  Testing Accuracy: 0.6790

  At higher C values, the training accuracy reaches near-perfection. However, the testing accuracy decreases further, indicating significant overfitting and poor generalization.

- C Value: 100

  Training Accuracy: 1.0000

  Testing Accuracy: 0.6744

  With the highest C value, the model achieves perfect training accuracy but performs poorly on the testing data. This extreme overfitting highlights the limitations of the linear kernel in capturing the complexities of the dataset.

In summary, the observed trend suggests that the linear kernel may not be the best fit for this dataset. The model struggles to find a balance between bias and variance, leading to poor generalization and overfitting as the C value increases. Alternative kernel functions or more complex models may be better suited to capture the underlying patterns in the data and improve performance.

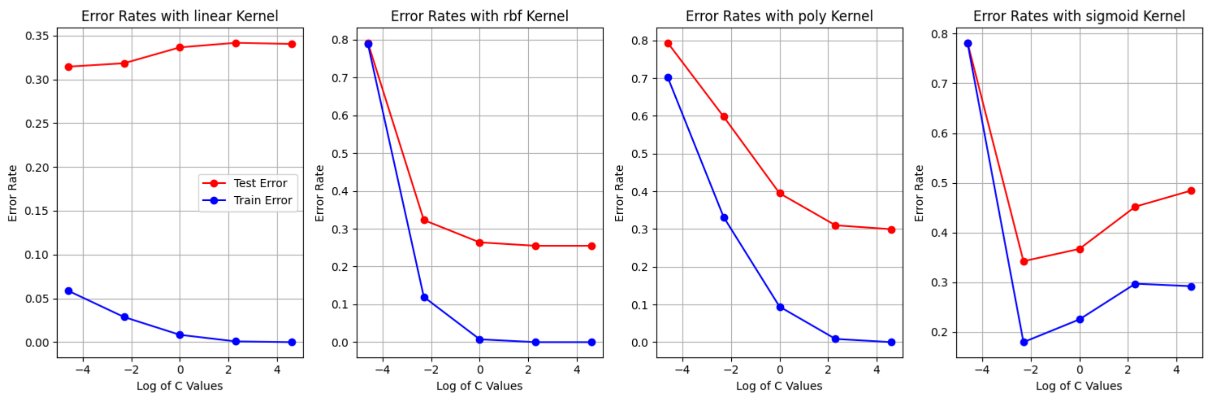
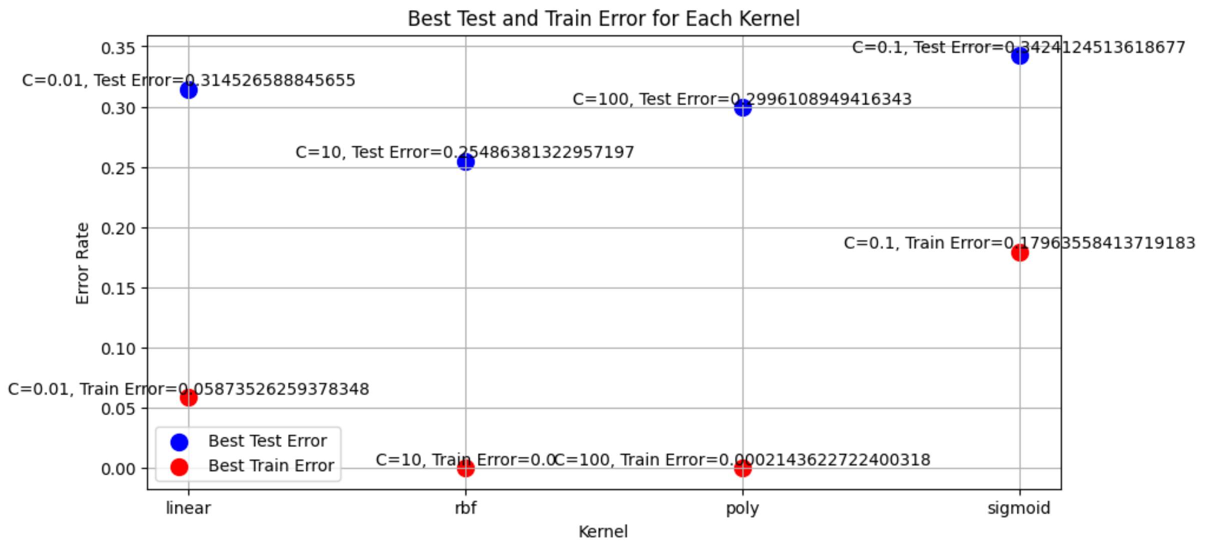


Table 2: SVM Performance with Varying C Values

C value	Training Accuracy	Testing Accuracy	Train Error	Test Error
Kernel: linear				
0.01	0.941	0.685	0.059	0.315
0.1	0.971	0.682	0.029	0.318
1.0	0.992	0.663	0.008	0.337
10	0.999	0.658	0.001	0.342
100	1.000	0.660	0.000	0.340
Kernel: rbf				
0.01	0.212	0.208	0.788	0.792
0.1	0.880	0.677	0.120	0.323
1.0	0.992	0.736	0.008	0.264
10	1.000	0.745	0.000	0.255
100	1.000	0.745	0.000	0.255
Kernel: poly				
0.01	0.298	0.206	0.702	0.794
0.1	0.668	0.402	0.332	0.598
1.0	0.906	0.605	0.094	0.395
10	0.991	0.690	0.009	0.310
100	0.999	0.700	0.000	0.300
Kernel: sigmoid				
0.01	0.219	0.219	0.781	0.781
0.1	0.820	0.658	0.180	0.342
1.0	0.775	0.633	0.225	0.367
10	0.703	0.548	0.297	0.452
100	0.708	0.516	0.292	0.484

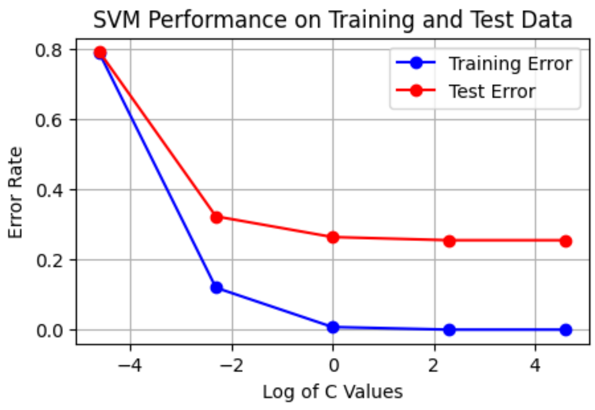
The best results for each kernel based on the least test error are: Linear kernel with C value of 0.01 (Test Error: 0.315), RBF kernel with C values of 10 and 100 (Test Error: 0.255), Poly kernel with C value of 100 (Test Error: 0.300), Sigmoid kernel with C value of 0.1 (Test Error: 0.342).

Plot of the best performance (lowest error rate) for each kernel. It identifies the best performing C value for each kernel based on the lowest error rate achieved.



The graph illustrates the performance comparison of SVM using four different kernels: linear, rbf, poly, and sigmoid. The best error rate refers to the test error, which is the error rate calculated on the testing dataset.

The results indicate that the RBF kernel with C=10 provides the best overall performance, achieving the highest testing accuracy of approximately 74.5%. Additionally, it has the lowest test error rate 0.255 among all kernel-C combinations, indicating its effectiveness in generalizing to unseen data. The RBF kernel with C=10 also exhibits perfect training accuracy, suggesting a good balance between bias and variance, avoiding overfitting while capturing the underlying patterns in the data.



- C Value: 0.01:

  Testing accuracy: 0.2082

  This is the lowest testing accuracy observed in the table, indicating poor generalization performance of the SVM model.

  The model trained with this C value likely suffers from high bias, meaning it is underfitting the training data and failing to capture the underlying patterns in the data.

- C Value: 0.1:

  Testing accuracy: 0.6770

  There is a significant improvement in testing accuracy compared to the previous C value.

  While the model's performance has improved, it still exhibits some degree of underfitting, as indicated by the noticeable gap between training and testing accuracies.

- C Value: 1.0:

  Testing accuracy: 0.7361

  This represents a substantial increase in testing accuracy compared to the previous values of C.

  The model seems to have found a better balance between bias and variance, leading to improved generalization performance.

- C Value: 10 and 100:

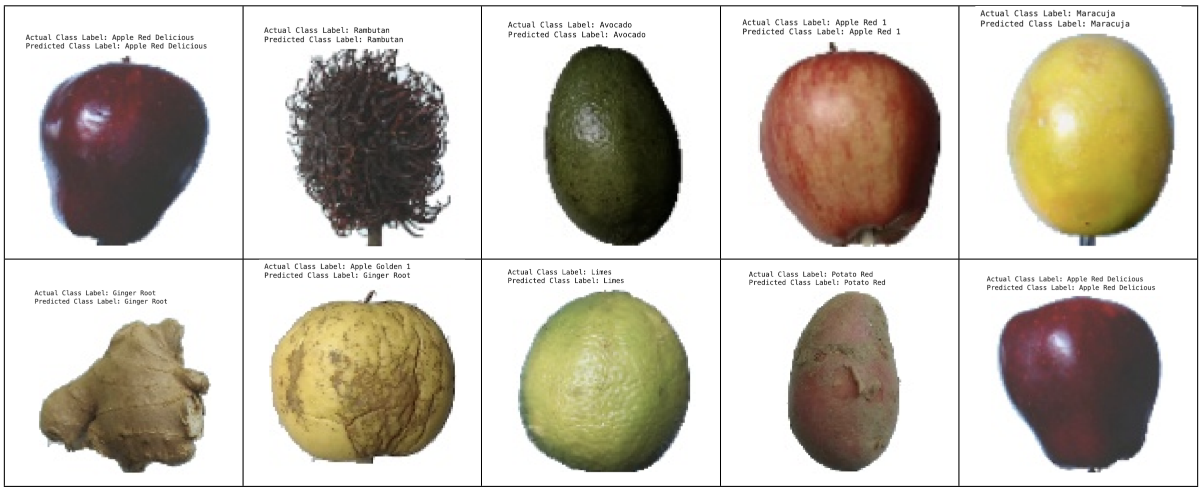
  Testing accuracy: 0.7451 (identical for both C values)

  These two values of C yield the highest testing accuracy in the table. The model performs consistently well on unseen test data, suggesting that further increasing the complexity of the decision boundary (by reducing regularization) does not provide significant additional benefits.

In the context of Support Vector Machines (SVM), C values represent the regularization parameter. Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function. In SVM, the parameter C controls the trade-off between maximizing the margin (decision boundary) and minimizing the classification error. A small C value allows for a larger margin but may misclassify some points (soft margin). A large C value penalizes misclassifications heavily, potentially resulting in a smaller margin (hard margin). In essence, C values determine the balance between achieving a low training error and generalizing well to unseen data.

In summary, the testing accuracies demonstrate the impact of the regularization parameter C on the SVM model's ability to generalize to unseen data. Lower values of C may lead to underfitting and poor performance, while higher values can improve performance up to a certain point before potentially overfitting the model to the training data. The optimal choice of C depends on balancing bias and variance to achieve the best generalization performance.

Table with Predicted Class (kernel: RBF, C=10)


Actual Class Label: Apple Red Delicious
Predicted Class Label: Apple Red Delicious
Actual Class Label: Rambutan
Predicted Class Label: Rambutan
Actual Class Label: Avocado
Predicted Class Label: Avocado
Actual Class Label: Apple Red 1
Predicted Class Label: Apple Red 1
Actual Class Label: Maracuja
Predicted Class Label: Maracuja
Actual Class Label: Ginger Root
Predicted Class Label: Ginger Root
Actual Class Label: Apple Golden 1
Predicted Class Label: Ginger Root
Actual Class Label: Limes
Predicted Class Label: Limes
Actual Class Label: Potato Red
Predicted Class Label: Potato Red
Actual Class Label: Apple Red Delicious
Predicted Class Label: Apple Red Delicious

In the table above, the predicted classes for a selection of six fruit images align with their actual classes, with one exception. Among these images, there was a discrepancy: an image labeled as Apple Golden 1 was instead predicted to be Ginger Root.

Questions 6 to 8

The validation loss decreases consistently with each epoch (graph 6(a)), indicating that the model is learning effectively from the training data. Simultaneously, the training accuracy increases steadily, reaching 100% accuracy by the 4th epoch. This suggests that the model is fitting the training data very well, achieving perfect accuracy.

Table 3: Performance Metrics for Simple Convolutional Neural Network

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.6253	0.7912	0.1014	0.9418
2	0.0557	0.9850	0.0454	0.9741
3	0.0071	0.9998	0.0182	1.0000
4	0.0024	1.0000	0.0120	1.0000
5	0.0013	1.0000	0.0060	1.0000
6	0.0009	1.0000	0.0059	1.0000
7	0.0006	1.0000	0.0054	1.0000
8	0.0005	1.0000	0.0034	1.0000
9	0.0003	1.0000	0.0031	1.0000
10	0.0003	1.0000	0.0033	1.0000
11	0.0002	1.0000	0.0030	1.0000
12	0.0002	1.0000	0.0027	1.0000
13	0.0002	1.0000	0.0026	1.0000
14	0.0001	1.0000	0.0027	1.0000
15	0.0001	1.0000	0.0024	1.0000
16	0.0001	1.0000	0.0019	1.0000
17	0.0001	1.0000	0.0013	1.0000
18	0.0001	1.0000	0.0015	1.0000
19	0.0001	1.0000	0.0016	1.0000
20	0.0001	1.0000	0.0015	1.0000

The validation accuracy, evaluated on the test data, improves significantly with each epoch. It reaches 100% (graph 6(a))accuracy by the 3rd epoch and maintains this perfect accuracy throughout the remaining epochs. This indicates that the model generalizes well to unseen data, achieving optimal performance on both the training and test datasets.

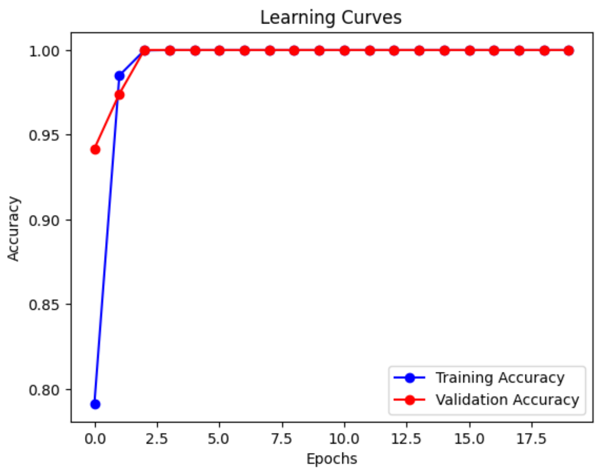


Figure 1: 6(a)

Given the perfect accuracy achieved, the learning curves appear as two flat lines at the top of the graph after a few epochs (graph 6(a)). Overall, the CNN model demonstrates excellent learning capabilities, effectively capturing patterns in the data and achieving perfect classification accuracy on both training and test datasets.

Throughout the 10 epochs of training, the training loss consistently decreases (graph 7(b)), indicating effective learning from the training data. The training accuracy quickly reaches 100%, demonstrating that the model is successfully fitting the training data.

Table 4: Performance Metrics for Transfer Learning with ResNet18

Epoch	**Training Loss**	**Training Accuracy**	**Validation Accuracy**
1	0.387	0.945	1.000
2	0.042	1.000	1.000
3	0.020	1.000	1.000
4	0.013	1.000	1.000
5	0.009	1.000	1.000
6	0.007	1.000	1.000
7	0.005	1.000	1.000
8	0.004	1.000	1.000
9	0.004	1.000	1.000
10	0.003	1.000	1.000

The validation accuracy, evaluated on the test set, remains consistently at 100% throughout the training process (graph 7(b)). This indicates that the model generalizes well to unseen data and maintains high accuracy on the test set, suggesting robust performance.

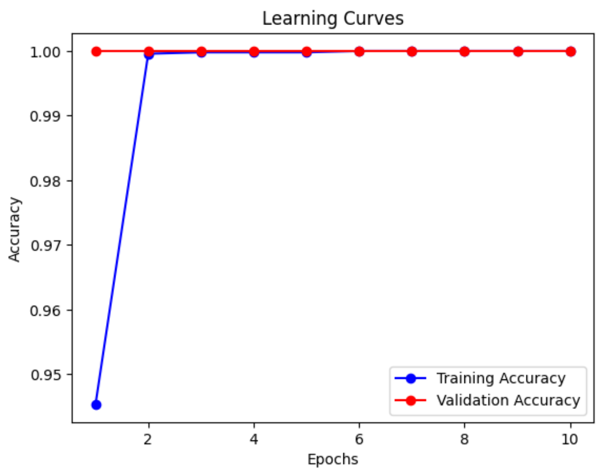


Figure 2: 7(b)

The validation accuracy curve shows as a flat line at the top of the graph, representing perfect validation accuracy. This demonstrates that the model achieves optimal performance without overfitting, as both training and validation accuracies converge to 100% (graph 7(b)).

```
SVM Training Accuracy: 1.0
SVM Test Accuracy: 1.0
```

The SVM prediction model trained, on the hyperparameters that resulted in best results (C=10, kernel: RBF), and using features extracted from the ResNet18 model, achieves perfect accuracy on both the training and test sets, with training accuracy and test accuracy both reaching 100%.

The SVM model achieves 100% accuracy on the training set, indicating that it perfectly separates the classes based on the extracted features. This suggests that the SVM effectively learns the decision boundary to classify the training data without any errors. Similarly, the SVM model achieves 100% accuracy on the test set, implying that it generalizes well to unseen data. This indicates that the model's performance is not limited to the training data and can accurately classify new instances based on the learned decision boundary.

Overall, the perfect training and test accuracies suggest that the SVM model, trained using features extracted from the ResNet18 model, effectively captures the discriminative information present in the data. This indicates the robustness and effectiveness of the feature extraction process and the SVM classifier in accurately classifying the 10 fruit classes.

The results of fine-tuning the pre-trained ResNet18 model show fluctuations over epochs. It starts high, reaches perfect accuracy (1.0) in the third epoch, but drops towards the second and fourth epochs. This behavior suggests that the model may be overfitting the training data, especially evident in the drop and fluctuations of validation accuracy. *"Low error rates and a high variance are good indicators of overfitting"* [1].

Table 5: Performance Metrics for Transfer Learning via Fine-Tuning with ResNet18

Epoch	Training Loss	Training Accuracy	Validation Accuracy
1	0.304	0.901	0.989
2	0.143	0.956	0.749
3	0.095	0.969	1.000
4	0.050	0.984	0.747
5	0.023	0.994	0.987
6	0.017	0.995	0.520
7	0.102	0.976	0.786
8	0.113	0.960	0.591
9	0.033	0.990	0.756
10	0.013	0.998	1.000

While the model achieves high validation accuracy in epoch 10/10, the fluctuations in validation accuracy indicate instability in model performance.

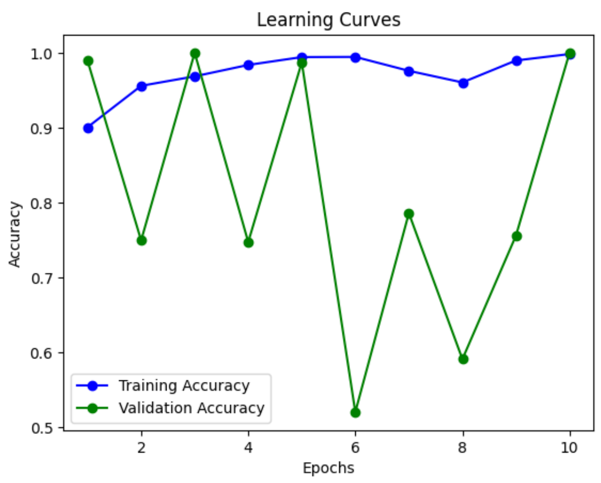


Figure 3: 8(b)

Question 9

(a) *"If the training data has a low error rate and the test data has a high error rate, it signals overfitting"* [1].

Overall, the results suggest that the three models have successfully learned the underlying patterns in the data and generalize well to new samples.

The CNN model achieves high training and validation accuracies throughout the training epochs.

Similar to the CNN model, the ResNet18 model also achieves high training and validation accuracies. Overfitting is not evident as the validation accuracy remains consistently high across epochs. Therefore, the test accuracy before overfitting occurs would correspond to the validation accuracy at the last epoch, which is 1.000.

Overfitting is indicated when the validation accuracy starts to deviate from the training accuracy or decreases. In this case, with the Transfer Learning via Fine-Tuning with ResNet18 model we observe a slight fluctuation in validation accuracy starting from epoch 2/10, suggesting potential overfitting. Therefore, the test accuracy before overfitting occurs would correspond to the validation accuracy at epoch 1/10, which is 0.989

(b) Comparing the best results for each model, we can see that CNN and transfer learning models achieved the highest test accuracies of 1.000, indicating perfect classification on the test dataset. Therefore, these approaches tie for the best performance in terms of test accuracy and low error rate.

Table 6: Comparison of Test Accuracy for Different Approaches

Approach	**Test Accuracy**
Deep Learning - CNN (epoch 20/20)	1.000
Transfer Learning - Feature Extraction (epoch 10/10)	1.000
Transfer Learning - Fine-Tuning (epoch 10/10)	1.000

We could instead consider the epochs before overfitting occurs for each model to provide a fair comparison of their test accuracies.

Table 7: Comparison of Test Accuracy Before Overfitting for Different Approaches

Approach	**Test Accuracy**
Deep Learning - CNN (epoch 20/20)	1.000
Transfer Learning - Feature Extraction (epoch 10/10)	1.000
Transfer Learning - Fine-Tuning (epoch 1/10)	0.989

Here the best performing model is the CNN and Transfer Learning - Feature Extraction models with an accuracy of 1.00

(c) The performance result that surprises me the most is the test accuracy of 1.000 (or 100%) achieved. This level of accuracy indicates that the models achieved perfect classification on the test data, which is quite remarkable.

The reason for my surprise is that achieving perfect accuracy, especially across all test samples, is rare in real-world scenarios. It suggests that the models have effectively learned the underlying patterns in the data and generalized well to unseen examples. However, such high accuracy rates may also indicate overfitting, where the models have memorized the training data too well, leading to poor performance on new data. This is mainly a cause of concern because the testing dataset is very similar to the training dataset in the case of the fruit images available with this project.

Therefore, while the perfect accuracy is impressive, it also prompts further investigation into the robustness and generalization capabilities of the models to ensure they perform well in diverse real-world scenarios.

References

[1] IBM. What is overfitting? `https://www.ibm.com/topics/overfitting`. Accessed: March 18, 2024.

[2] Horea Muresan and Mihai Oltean. Fruit recognition from images using deep learning. *Acta Univ. Sapientiae, Informatica*, 10(1):26–42, 2018.