Promoting Sustainable Food Systems: A Comparative Study of Random Forest, SVM, and CNN for Fresh and Rotten Fruit Classification

Exam paper

Data Mining, Machine Learning, and Deep Learning

[CDSCO1004U]

MSc Business Administration and Data Science

Group:

M3-ML

Authors:

Marie Liljegren Gam (maga15ab) (102778)

Mathilde Lundsberg-Nielsen (malu22ae) (157498)

Michelle Judith Sara von Huth (mivo22ab) (158397)

Promoting Sustainable Food Systems: A Comparative Study of Random Forest, SVM, and CNN for Fresh and Rotten Fruit Classification

Marie Liljegren Gam, Mathilde Lundsberg-Nielsen & Michelle Judith Sara von Huth

Copenhagen Business School
MSc. Business Administration and Data Science

Abstract

This study investigates the application of machine learning for distinguishing fresh and rotten fruits, with the goal of addressing inefficiencies and waste associated with manual quality inspections. To solve the study issue, three different machine learning models were used: a Random Forest (RF), a Support Vector Machine (SVM), and a Convolutional Neural Network (CNN). The dataset used for this study comprises roughly 12,000 augmented images of fresh and rotten fruit varieties. The models were trained and tested based on their ability to accurately classify these images. The CNN model showed the highest F1-score, however, with regards to time optimization, the RF model proved the better option. The study concludes that the choice of model for practical implementation is determined by industry-specific goals and circumstances.

Keywords*: Fruit Freshness Classification, Quality Management, Random Forest, Support Vector Machine, Convolutional Neural Network, Image Classification.*

1 Introduction

The development of a novel classification model that can effectively identify fruit defects while reducing human labour, expenses, and production duration has the potential to not only revolutionize the fruit production industry but also promote sustainability efforts. Manual methods for identifying fresh and spoiled fruits are inefficient and time-consuming for agricultural producers and sellers, resulting in significant financial losses and food waste (Ishangulyyev et al., 2019). By employing machine learning algorithms such as Random Forest (RF), a Support Vector Machine (SVM) and a Convolutional Neural Network (CNN), as illustrated in Figure 1, this study emphasizes the utilization of diverse techniques to support sustainable food systems, taking into account their distinct strengths and limitations.

One potential use case for a fruit quality classification model is in the food retail industry, where grocery stores, and restaurants could utilize such a model to sort and categorize fruits based on quality factors like ripeness, colour, and texture. This would ensure that only the freshest products are sold to consumers, improving the quality of the products available while simultaneously reducing food waste by preventing the discarding of fresh fruits. In addition, agricultural producers could also leverage such algorithm to identify fruit defects early in the production process, allowing them to take corrective action before the fruits are harvested and sold.

By employing distinct algorithms to support automated fruit quality management, the present study highlights the transformative potential of machine learning in mitigating food waste and promoting sustainable practices.

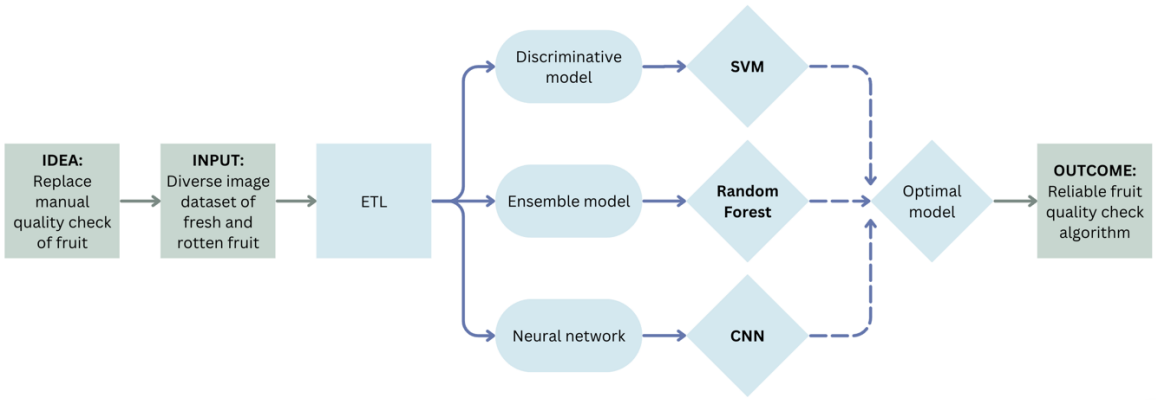


***Figure 1**: Conceptual framework, flowing from the initial idea to the outcome of the study.*

2 Related Work

Several recent publications have focused on identifying fresh and rotten fruits using machine learning algorithms for different purposes. However, the common goal of these works is to improve fruit production and reduce waste as a final means.

In 2022, Kumar et al. introduced a CNN model to identify apples, bananas, and oranges within a refrigerator, pinpointing households as significant contributors to food waste. Their research presents an innovative recycling approach for traditional refrigerators, which involves equipping compartments with cameras and employing machine learning models to ascertain the freshness of fruit. The researchers propose the concept of CNNs to effectively train their dataset in discerning food photographs, while also mentioning the possible use of pre-trained models, such as ALexa, Google Net, and ImageNet instead.

Shaikh et al. (2021) also proposed a model capable of detecting and classifying fruits based on their surface condition, using the Faster R-CNN. This algorithm provides a real-time solution for detecting fruit quality with multi-class classification, resulting in high accuracy across classes.

Another noteworthy and recent approach was proposed by Rincón et al. (2022), who employed machine learning algorithms to enhance the quality of red raspberries in particular. The scholars were able to detect fruit diseases and defects using a CNN based on a pre-trained Fast R-CNN approach as well.

While numerous studies have demonstrated the potential of machine learning algorithms in identifying fresh and rotten fruits, this study aims to compare the performance of three models - RF, SVM, and CNN - representing three distinct modelling approaches under the supervised learning paradigm: ensemble models, discriminative models, and neural networks respectively. In light of this, our research question can be stated as follows:

"How does the choice of supervised machine learning approach impact the performance of classifying fresh and rotten fruit?"

The research problem at hand will be addressed by means of gathering comparative insights to determine the approach that models the data better, thereby expanding the current perspectives of the classification task within the field. However, the implications extend beyond the realm of just machine learning. Developing a novel method for classifying fruits based on quality have the potential to yield positive impacts in various settings including minimization of waste and optimization of stocking and inventory management; all of which ultimately contribute to resource conservation and lowering of greenhouse gas emissions associated with the production, transportation, and disposal of wasted fruits (Kilkenny & Robinson, 2018).

3 Methodology

3.1 Conceptual Framework

To answer the proposed research question *"How does the choice of supervised machine learning approach impact the performance of classifying fresh and rotten fruit?"* the conceptual framework in Figure 1 was developed and followed. The appropriate data was selected, processed, and utilized for the development and assessment of three separate models from different subfields of supervised learning, culminating in the identification of an optimized model as the final output. In the following methodology section, the approaches followed during data pre-processing and the chosen training strategy, as also depicted in Figure 2, will be outlined and justified.

3.2 Dataset Description

The dataset used in this study was created and published by researchers at Jahangirnagar University, with the intention of being extensive enough to enable successful recognition and classification of a diverse spectrum of fresh and rotten fruit (Sultana et al., 2022). The dataset comprises sixteen distinct fruit categories, including fresh and rotten grape, guava, jujube, pomegranate, strawberry, apple, banana, and orange. The dataset consists of 3,200 original images and 12,335 augmented images, with an approximately even distribution between the sixteen categories (see Appendix A for exact counts). The original and augmented image dimensions are 512x512 pixels.

3.3 Data Pre-processing

In order to artificially increase the volume of the dataset from 3,200 to over 12,000 images, Sultana et al. (2022) performed data augmentation to create multiple realistic yet different variants of each existing image. This technique reduces the risk of overfitting as it enables models to learn on, and classify, images that appear to be taken, for example, from different angles or in different lighting (Géron, 2022). As described

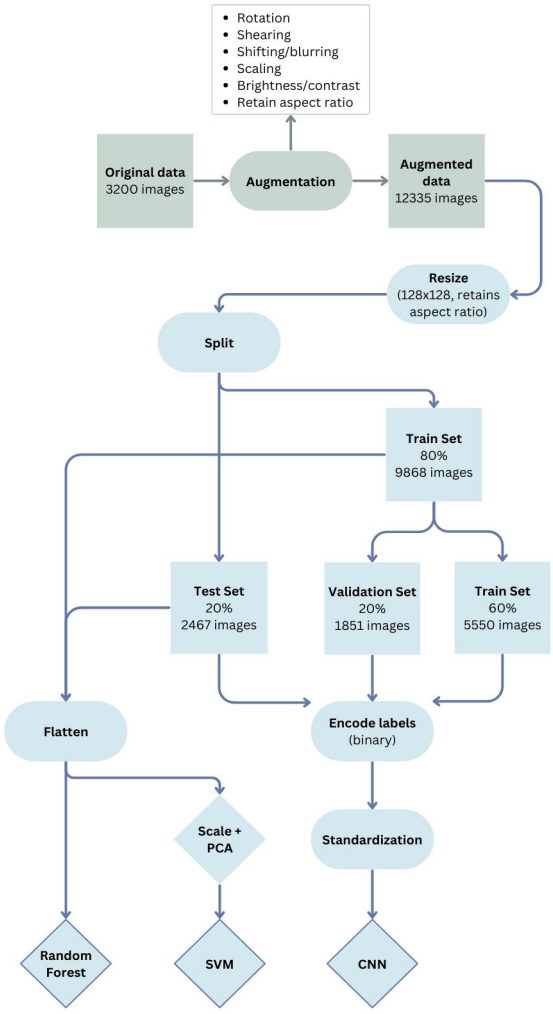


***Figure 2**: Data processing and training strategy flow.*

in Sultana et al. (2022)'s paper accompanying the data, the augmentation process involved several techniques such as random rotation, scaling, shearing, cropping, shifting, as well as adjustments in brightness, contrast, saturation, and hue. Rotation angles of 45°, 60°, and 90°, as well as re-scaling techniques like Nearest-Neighbour Interpolation, Bicubic Interpolation, and Bilinear Interpolation, were used. Histogram equalization to enhance image contrast was also applied. The augmentation parameters included a width shift range, height shift range, and shear range of 0.2. Figure 3 shows a sample of original and respective augmented images from the dataset (Sultana et al., 2022).

The augmented images were loaded using Python and resized from their original 512x512 pixels to 128x128 pixels, allowing the three

models to run faster while maintaining the images' 1:1 aspect ratio (Pedregosa et al., 2011). The aspect ratio of the images must be maintained throughout the dataset in order to keep image matrices' dimensions equal, as many machine learning algorithms cannot handle the irregularly sized matrices that would occur from having varying aspect ratios (Pedregosa et al., 2011). Despite the smaller size, the images retained sufficient detail for effective processing by our models, striking a balance between computational efficiency and data quality.

When downloaded, the images were stored in folders where the folder name indicates the respective class of the fruit (e.g. FreshApple, RottenGrape). When the images were loaded to be stored as a dataset, these folder names were used to assign the target label for each image. To have binary classification in alignment with the task of classifying freshness regardless of fruit type, just two labels 'Fresh' and 'Rotten' were assigned from the folder names, rather than the full 16 classes in the original published data.

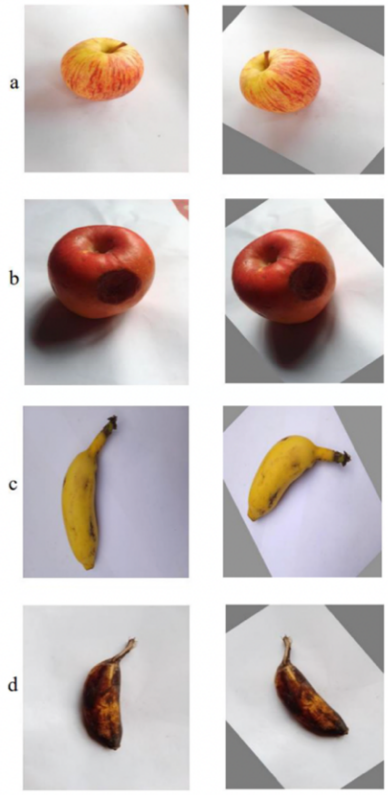


***Figure 3**: Examples of data augmentations on images of apples and bananas, both fresh and rotten (Sultana et al., 2022).*

Each image is represented by a 3D array (*height = 128*, *width = 128*, *n_colour_channels = 3*), where numerical values represent each pixel based on the intensity of the associated channel's colour (Géron, 2022). As RF and SVM only handles 2D inputs *(n_images x n_features)*, each image was then flattened into a 1D array (*n_features*) of 16,384 features. This flattened data was then passed directly to the RF model. For the SVM, the flattened data was passed into a pipeline containing a StandarScalar and Principal Component Analysis (PCA), before fitting the model. The PCA was performed to increase efficiency and performance of the SVM, as while SVM models are suitable for high-dimensional data, very large images and therefore large datasets can make the training process slow (Géron, 2022). The PCA compresses the images by extracting the most informative features from the dataset, i.e. those features that preserve the greatest variance in the images.

The CNN can process multidimensional image data by representing it as tensors with height, width, and colour channels (128, 128, 3) (Géron, 2022). The multidimensional nature of the input is in fact necessary for the network to effectively capture and exploit the spatial relationships and patterns of the data i.e., to extract the meaningful features of the images (Zhang et al., 2018). When the input has flowed through the convolutional and pooling layers of the network, the output feature maps are however also flattened before flowing through the dense and output layers to make the predictions.

3.4 Feature scaling and encoding

To boost image consistency specific to the CNN implementation, various feature scaling techniques were assessed on the performance of the model including normalisation and standardisation (Géron, 2022). Scaling pixel values by 255 to bring them within a normalized range of 0-1

led to a degradation in performance. Conversely, standardization, which involves transforming the data to have zero mean and unit variance, exhibited positive effects on the classification performance and the networks' generalization abilities (Géron, 2022). Thus, the standardized images were used as input for the CNN.

As a final pre-processing step specific to the CNN input data, labels were encoded from the binary text labels assigned previously into numeric binary labels 0 (Fresh) and 1 (Rotten).

3.5 Training Strategy

Before initiating the training process, the images were split into training and testing sets with an 80/20 split (See Appendix B). To facilitate the training of the CNN model, the training set was further divided into training and validation sets, resulting in an overall split of 60% for training, 20% for validation, and 20% for testing (See counts in Appendix B). The CNN hyperparameter tuning was done by manually adjusting the relevant parameters and the network depth. The model's performance was monitored by evaluating learning curves for training and validation accuracy/loss over epochs (Géron, 2022). The SVM and RF models did not utilize a designated validation set, rather, k-fold cross-validation was implemented to divide the data into multiple validation folds to be able to continuously assess the models' performance (Géron, 2022). The ROC curves were subsequently plotted for each fold, to allow for visual performance comparisons across the multiple validation subsets.

The hyperparameter tuning process for the RF and SVM models followed a combination of grid search and the aptly nicknamed 'babysitting' or 'grad student descent' method (Yang & Shami, 2020). This approach was implemented to maintain full control over the training flow and to observe the impacts of tuning adjustments on the relevant parameters closely. Despite training time increasing relative to the width of the parameter space defined in the grid, the grid search method allowed exploration of several parameter combinations at once, thus, manual model configuration time was saved (Géron, 2022). However, as grid search lacks the capability to exploit other undiscovered well-performing regions further on its own, the 'babysitting' method was applied to manually adjust, shift, and narrow down the search space based on well-performing hyperparameter configurations found in previous search iterations (Breiman, 2001).

Finally, each model was benchmarked against a baseline model, which was built using only the default hyperparameters for RF and SVM, as defined by Scikit-Learn, and by adding only the minimum number of layers required for CNN (Géron, 2022). This was used to imitate the models and performance that one would get by putting in no manual configuration work, in order to analyse the difference in running time versus performance gained from tuning the models.

3.6 Modelling Framework

As priorly stated, this study employs three distinct machine learning models namely RF, SVM and CNN to address the classification task. Following implementation, a parallel evaluation of the models' performance will be conducted, to determine which algorithm is better suited to model the data at hand. The primary analysis as well as the study in general takes a supervised learning perspective with the objective of comparing various model approaches.

Benchmarking different model subtypes against each other becomes relevant for several reasons. For one part, evaluating models with different learning strategies and feature representations allows us to understand each model's strengths and limitations for the given research problem (Gontijo-Lopes et al., 2022). Secondly, the specific employment of the CNN allows us to determine whether the additional complexity of deep learning can be justified in terms of improved performance.

The coming sections seeks to broadly outline each model's algorithmic architecture including descriptions of the relevant hyperparameters utilized in the tuning process. For every section, the optimal parameters employed in the final model will additionally be stated and explained.

3.6.1 Random Forest (RF)

The initial model utilized for the classification task was a Random Forest (RF), an ensemble learning model. At its core, the RF is essentially an ensemble, or collection, of decision trees (Breiman, 2001). To construct each tree, a random subset of data features is employed at each split, imparting the model with a significant degree of diversity and randomness. This inherent randomness helps to reduce overfitting, making the RF model reliable and robust.

The procedure for constructing the trees in the RF model involves a series of splitting operations, which are decided based on certain criteria (Breiman, 2001). By default, our model relies on Gini impurity to determine the quality of its splits (Géron, 2022). While Entropy and Gini impurity lead to similar trees most of the time, the latter is slightly faster to compute, and therefore Gini impurity is a good default choice.

As previously stated, an iterative hyperparameter tuning process was used for the RF. This process involved an adapted "babysitting" technique, whereby the parameter grids were continually updated based on the outcomes of preceding grid searches (Yang & Shami, 2020). The relevant hyperparameters tuned during the training process were 'n_estimators', 'max_depth', 'min_samples_split', and 'max_features' - each with several specific ranges of values specified in the parameter grid. The values of the parameters were chosen based on their impact on the performance of the RF model. The 'n_estimators' parameter refers to the number of trees in the forest, the 'max_depth' parameter controls the maximum depth of each tree, the 'min_samples_split' sets the minimum number of samples needed to split an internal node, and lastly, the 'max_features' parameter prescribes the number of features to evaluate when searching for the optimal split. The values of the parameters leading to solid performance in the final model were found to be at a max_depth of 30 for each tree, a max_features of 'log2', a min_samples_split of 2 and a total of 800 trees in the forest specified by n_estimators.

As previously mentioned, the performance of different hyperparameter combinations was assessed using grid search with 5-fold cross-validation (Géron, 2022). The performance of the model on each of the validation folds was monitored by a ROC curve analysis. Ultimately, the hyperparameter combination that yielded the highest average cross-validation score was selected to be used as the final tuned model.

3.6.2 Support Vector Machine (SVM)

An SVM model was chosen as the discriminative model to be used for this study, as it works well when there is a clear separation of classes in data, including non-linear separations (Géron, 2022). As SVM works well on non-linear data, it is also particularly well suited for classifying image data. The intuition behind the SVM is finding the decision boundary that best separates the data, which is one that maximises the distance, or margin, between the decision boundary and all classes (Géron, 2022). To do this for non-linear data, more features can be added in order to map the data into a higher dimensional space such that good separation of the classes is generated. These features can be added by performing kernel functions on the existing data, such as polynomial or similarity functions which are applied by using the polynomial and RBF kernels, respectively (Géron, 2022). Once the kernel functions are applied, a decision boundary is found where the trade-off between accuracy (or training error) and generalisability is minimized. This trade-off can be controlled through the tuning of various hyperparameters, including C and gamma.

For SVM models, the hyperparameter C controls the margin around the decision boundary:

a larger C value specifies more regularisation, which sets a small margin on either side of the decision boundary (Géron, 2022). This will lead to fewer errors on the training set but is also likely to lead to overfitting where the model will not generalise as well. To improve generalisation, C can be reduced to a smaller value, giving less regularization and larger margins, but also more errors on the training set (Géron, 2022). Gamma is the kernel coefficient which controls the radius for polynomial and RBF kernels and helps to define the model's notion of distance between datapoints.

As with the Random Forest, a 'babysitting' grid search approach with 5-fold cross validation was used to tune both the PCA and the SVM models (Yang & Shami, 2020; Géron, 2022). The parameter grids included the number of PCA components, the SVM hyper parameters C and gamma and various SVM kernel functions. The iterative grid search process led to an optimal model with 100 PCA components, an RBF kernel, C = 40, and gamma = *scale,* being chosen as the final tuned model. The *scale* option for gamma sets the parameter scaled to the number of features in the data and their variance, as this defines a distance between datapoints that is relative to the actual dataset at hand (Pedregosa et al., 2011).

3.6.3 Convolutional Neural Network (CNN)

The final model utilized to address the research problem belongs to the subfield of deep learning, namely the CNN (Géron, 2022). CNNs have been shown to excel at solving classification tasks, with particularly superior performance in the domain of image classification for balanced as imbalanced data (Johnson & Khoshgoftaar, 2019). The noteworthy achievements of CNNs in tasks similar to the one undertaken in this study underscore the indispensability of this model in our use case and preclude any attempts to circumvent it.

The general composition of a neural network comprises a collection of neurons which are organised in layers with respective learnable weights and biases (Géron, 2022). In their most basic form (the perceptron and the multilayer perceptron, i.e., feed-forward networks), neural networks have an input layer, a single or several hidden layers, and an output layer. In addition to this structure, the CNN architecture retains additional convolutional and pooling layers, the former relying on convolution operations to, as relevant to this use case, reduce input images into a form that is more easily processed, without compromising features that are critical to correct identification and thus prediction (Géron, 2022).

The convolution procedure is a mathematical process that includes sliding a filter over the data that is inputted and computing the dot product between this filter and the local input data at every position (Géron, 2022). In this way, convolutional operations create feature maps that highlight specific patterns and structures in e.g., inputted images. After a convolutional operation, the CNN apply pooling through a pooling layer to be able to reduce dimensionality, whilst also limiting training time and preventing overfitting (Géron, 2022). More specifically, the pooling layers implement downsampling of each of the feature maps by reducing the image width and -height but retaining the image depth.

The convolution and pooling layers are subsequent to the input layer. When the data has flown through these initial levels, the resulting feature maps are usually flattened and fed into a single or several fully connected layers i.e., following the single or multilayer perceptron structure (Géron, 2022). The individual weights of the neurons' connections are learned here through the backpropagation optimization algorithm by propagating the error backwards through the network to try to minimize the difference between the predicted and true output (Géron, 2022).

The hyperparameter tuning process for the CNN was done manually to ensure full control over the training flow, and strict monitoring of

the effects of fine-tuning on the relevant parameters (Hasebrook et al., 2022). The final model demonstrating strong performance in-sample as out-of-sample consists of four convolutional layers including the input layer. The output of every convolutional layer is passed through a max pooling layer with a 2x2 window size to down-sample feature maps. In the initial convolutional layer, where the input layer is defined implicitly for 128x128 RGB images, 32 filters of size 3x3 are applied, relying on the rectified linear unit (ReLU) activation function. The following second, third and fourth convolutional layers with 64, 128 and 256 filters of size 3x3 respectively, also rely on ReLU activation. After the images have flown through the convolution and pooling layers, the output of the last pooling layer is flattened into a 1D vector and passed through two fully connected dense layers with 256 and 128 neurons respectively, also relying on ReLU activation. Here, the convolutionally processed features are used to make a prediction about whether the input image of a fruit represents the fresh or rotten label (Géron, 2022).

As a means to limit overfitting, dropout regularization with a rate of 0.5 was applied to the output of each of the dense layers (Géron, 2022). In the dense output layer, a single neuron was specified for binary classification, relying on sigmoid activation to produce a probability distribution over the two possible classes (Géron, 2022). The model was trained to minimize the binary cross-entropy loss function by means of the Adam optimizer with a 0.001 learning rate at its optimal state.

During training, the model processed a batch of 32 samples at a time before updating weights and biases and the entire dataset was evaluated over 20 epochs. At the end of each epoch, the model's performance was evaluated on the specified validation set to ensure that it was able to generalize to new, unseen data, and to prevent it from overfitting and memorizing the training data leading to strong but artificial performance.

3.7 Performance metrics

The main measure for evaluating the classifiers' performances was the F1-score. Secondary metrics count precision and recall. Precision denotes the accuracy of the positive predictions (Géron, 2022). In this case, how well each model performed in predicting the 'rotten' labels. High precision denotes a low rate of false positives. Conversely, recall measures the ratio of positive, rotten instances, that were correctly identified by the models (Géron, 2022). High recall implies a low rate of false negatives.

Precision and recall are closely related and exhibit a trade-off - when one increases, the other tend to decrease (Géron, 2022). Thus, depending on the objective of the classification task, the prioritised evaluation metric may vary. As the objective of this study is environmentally bound, precision would carry immediate precedence as misclassifying fresh fruit as rotten would lead to unnecessary food waste. Ensuring high precision will enable a minimized likelihood of discarding edible fruit, ultimately minimizing food waste and promoting environmental sustainability.

On the other hand, a classifier that comprises an optimal balance between the two will generally be preferred to ensure minimal waste, but also warrant that a substantial number of rotten fruits are not missed so the value of the classifier is not compromised. Hence, the F1-score, the harmonic mean of precision and recall, becomes a valuable metric to assess the models' overall performances (Géron, 2022).

4 Results

The CNN model demonstrated superior precision, recall, and F1-scores for both classes, outperforming the other models, as displayed in Table 1. For the fresh fruit class, the CNN model achieved a precision of .99, and a recall of .96, resulting in an F1-score of .98. For the rotten fruit class, the model achieved a precision of .96, and a recall of .99, leading to a .98 F1-score as well.

Table 1: Results.

		Precision	Recall	F1-score
RF	Fresh [0]	.92	.95	.94
	Rottten [1]	.94	.92	.93
SVM	Fresh [0]	.96	.95	.96
	Rotten [1]	.96	.96	.96
CNN	Fresh [0]	.99	.96	.98
	Rotten [1]	.96	.99	.98

The SVM model showed slightly lower F1-scores compared to the CNN model, however, still achieving a strong performance of .96 in F1 for both the fresh and rotten class. In addition, the SVM managed to achieve a precision in the rotten class identical to the precision of the CNN, hence, directly competing with strong performance in terms of accurate predictions on the positive, rotten class. The RF model showed the lowest precision, recall, and F1-scores, but still demonstrated solid performance with an F1 of 0.94 and 0.93 for the fresh and rotten classes, respectively.

The findings are supported by confusion matrices depicting the models' generalization performance on the test set, which can be found in Appendix D. Naturally, the CNN showed the lowest number of false positives and false negatives, however, it re-highlights that although the RF model had the relatively worst performance, it still only misclassified a total of 165 fruits out of the 2,467 in the test set. This indicates that the model is still capable of accurately classifying a high proportion of the images, despite having lower precision, recall, and F1-scores compared to the other models.

4.1 Training evaluation

Each model's training procedure was carefully examined in order to assess their individual performance and generalization abilities, and determine if overfitting would be a concern. The training and validation accuracy and loss curves for the CNN are displayed in Figure 4, whilst the ROC curve is plotted over validation folds in Figure 5 for the RF and SVM models respectively.

The performance of the CNN model was evaluated on the validation set over the course of 20 epochs, as the testing phase revealed no improvement on performance past this limit (Géron, 2022). Throughout the training phase, the validation and training accuracy followed a similar increasing pattern, somewhat stabilizing at a high level throughout epochs. The training and validation loss also maintained a relatively strong correlation with a downward trending pattern, suggesting minimal chances of overfitting (Géron, 2022). The validation loss however shows a brief spike at the 9th epoch and a smaller but increasing divergence between the curves thereafter. The latter confirms that the model has already reached convergence, and that only minimal chances are present that the model would improve over additional epochs (Géron, 2022).

To assess the generalization performance and potential overfitting of the RF and SVM models, the Receiver Operating Characteristic

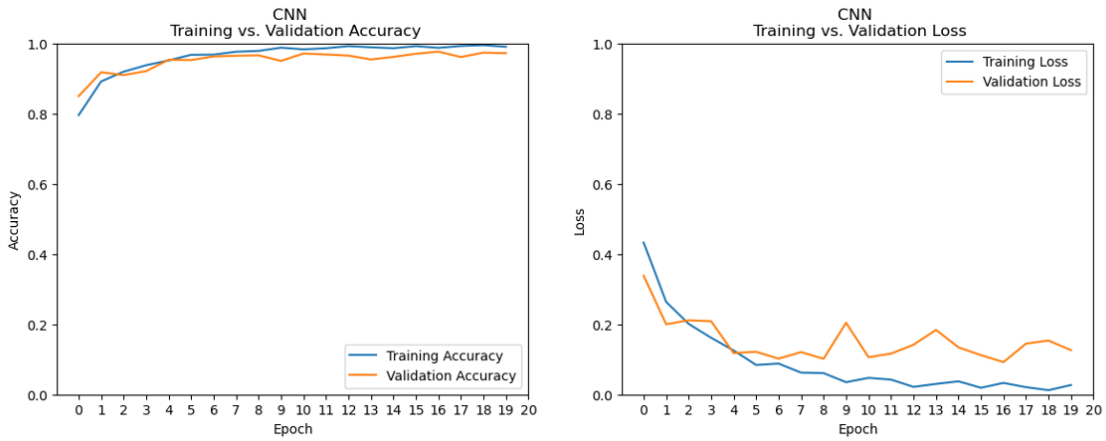


Figure 4: CNN training and validation results.

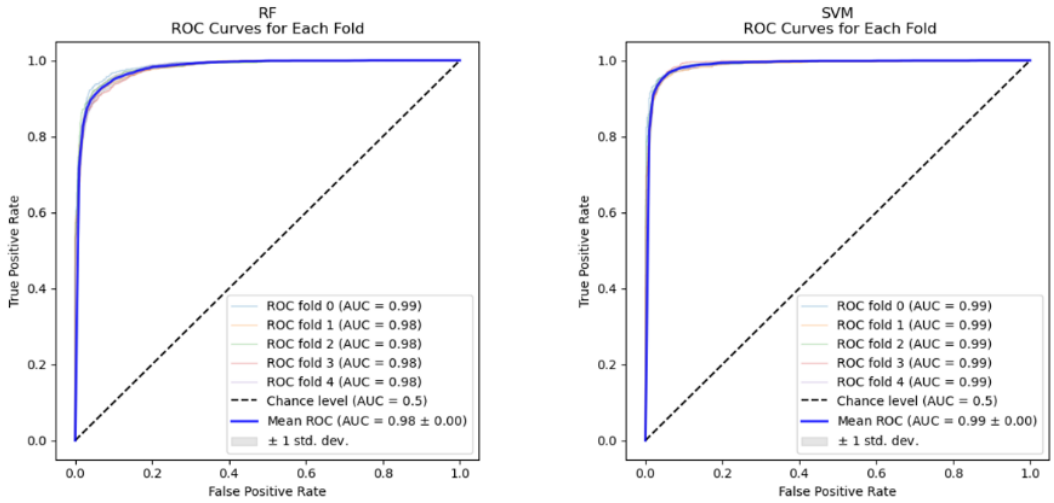


***Figure 5**: ROC curves for the optimal SVM and RF models.*

(ROC) curves were plotted over the cross-validation folds during training (Pedregosa et al., 2011). In this process, the training data was divided into subsets and cross-validated using a Stratified K-fold approach with 5 folds. For each fold, ROC curves were calculated to display the trade-off between the True Positive Rates (TPR) and the False Positive Rates (FPR) (Géron, 2022). The resulting graphic, as seen in Figure 4, shows the mean ROC curve, and individual ROC curves for each fold.

The RF model showed a consistent AUC score of .98-.99 across all five folds. This demonstrates that across diverse validation-subsets of the training data, the RF model continually demonstrated high discriminative power, and accurately distinguished between fresh and rotting fruit samples (Géron, 2022). Also the SVM model consistently performed well across all folds, with an AUC score of approximately 0.99 for all folds. The consistent and high AUC values for both the RF and SVM models not only highlight their reliability and effectiveness in the classification tasks but also provide evidence of strong generalization powers, and indicates a lowered risk that the models overfitted the training data (Géron, 2022).

4.2 Complexity & Running Time

While evaluating the quality of machine learning models, it is critical to consider both performance and training time, since there are occasions when a less accurate model may be chosen owing to a shorter running time, requiring fewer expensive resources. All three models were run on machines with 32 CPUs and 192 GB of memory. As previously stated, the CNN model performed best in terms of precision, recall, and F1-scores. Yet, as shown in Table 2 and Figure 6, the tuned CNN model had the longest CPU running time of all the models, at 3 hours 48 minutes. Despite the increased speed, the almost 20x longer running time may not be suitable for applications that need quicker processing, or where resources are limited.

The SVM model had only minutely lower performance metrics than the CNN model, but significantly shorter CPU running times. The tuned SVM model took only 9 minutes and 13

***Table 2**: CPU running times (baseline vs. tuned).*

		CPU
RF	RF Baseline	2min 40s
	RF Tuned	1min 19s
SVM	SVM Baseline	10h 13min 33s
	SVM + PCA Baseline	9min 31s
	SVM + PCA Tuned	9min 31s
CNN	CNN Baseline	1h 33min 22s
	CNN Tuned	3h 48min 19s

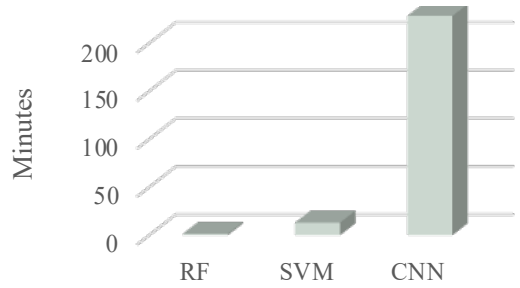


***Figure 6**: CPU running times (tuned optimal models).*

seconds to fit both PCA and SVM, providing a potential alternative for cases where a quicker or cheaper solution is desired without sacrificing performance. In contrast, the RF model had the lowest performance metrics out of the three models. Nevertheless, it still provided high scores and had the shortest operating time, with only 1 minute and 19 seconds for the optimized model. Therefore, it could be an attractive choice for applications that prioritize computational efficiency.

4.3 Trade-off: Performance vs. Complexity

All three models demonstrated strong performance and showed encouraging results in classifying fresh and rotten fruit samples. The CNN model exhibited promising behaviour throughout the training process, with a relatively correlated and downward going training and validation loss, converging around the 6th epoch (Géron, 2022). Similarly, the SVM and RF models consistently achieved high AUC values across different validation folds, demonstrating their excellent discrimination ability in categorizing the fruit samples (Géron, 2022). However, while choosing a suitable machine learning model, it is critical to consider the trade-offs between performance and running time. The CNN model provides the maximum performance but needs longer running periods, making it suited for applications requiring the utmost precision feasible. The SVM and RF models, on the other hand, strike a compromise between performance and computational efficiency, making them more suitable for situations requiring quicker processing without compromising on the performance.

When training new models with limited computational and human labour resources while using the aforementioned combination of grid search and 'babysitting', it is also significant to evaluate the trade-off between time spent manually tuning and running models, and performance increases. As can be seen in Table 2, SVM has practically no change in CPU running time if PCA has been applied, so the trade-off here for getting an increase in performance is only in manual tuning time. For RF there is an interesting trend of decreasing running time with manual tuning, which is likely due to the definition of the 'max_depth' parameter, that will act as a computational cut-off for branches once the depth threshold is reached (Pedregosa et al., 2011). Due to the nature of the RF, many of the hyperparameters can quickly increase or decrease computational complexity, when set to higher or lower values, which should be taken into account when evaluating the trade-off. Finally, for CNN there is around a 2.5x increase in CPU running time from the baseline model to the tuned model. Therefore, if a baseline or intermediate model has good performance, it should be considered whether spending time to manually tune and fit models further will pay off in model performance or not.

5 Discussion

The following discussion will be structured into five smaller sections: 5.1 Error Diagnostics, 5.2 Feature Importance, 5.3 Practical Implications, 5.4 Ethics, and 5.5 Limitations. Due to the superior performance demonstrated by the neural network, the initial sections will primarily rely on the classifications provided by the CNN.

5.1 Error Diagnostics

Based on the confusion matrix structure, Figure 7 displays a sample image from each of the true positives (positive denoting "Rotten"), true

negatives, false positives, and false negatives classes for the CNN test-set predictions. Out of a total of 2,467 images, the model wrongfully classified only 55 images, compared to 98 by the SVM and 165 by the RF.

The images of the actual fresh and rotten classes in the true negative and true positive classes show clear signs of freshness and spoilage respectively. The banana has a healthy yellow tint with few small brown marks as can be expected from natural ripeness. Meanwhile, the spoilage of the apple in the true positive class has not only a brown colour but also a larger break, and what looks to be a mushy surface.

When inspecting the false negative sample, the actual rotten state of the orange is also visible but definitively less clear compared to the apple in the true positive sample. This can be argued to have limited the model's performance and may be further reasoned by the spoilage being indicated by other senses than visual decay, such as bad smell or a soft feel (Sultana et al., 2022).

The strawberries portrayed in the false positive class are likewise visibly fresh with relatively firm and plump-looking textures, all securely attached to the stems. However, whether the dark shadows from overlapping, the fact that the strawberries are grouped closer compared to other fruit types in the dataset, or whether a third arbitrary feature has led to the misclassification remains unknown so far.

Commenting on the general errors of the models we see a pattern of misclassified images on the fruit types that have a higher likelihood of non-visual spoilage (Sultana et al., 2022). Where bananas typically enter a process of enzymatic browning, strawberries spoil more in smell and texture before eventually moulding, thus diminishing the visual dissimilarity between fresh and rotten (Sultana et al., 2022) (Appendix F). In the SVM and RF models' misclassifications, we saw an additional number of fresh apples classified as rotten. This could possibly be due to a particularly bright and round

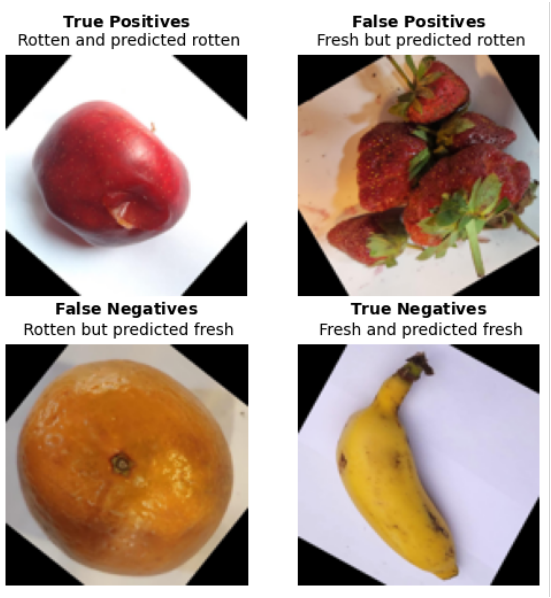


***Figure** 7: CNN sample images by classification in the confusion matrix structure.*

reflection on the surface which may conceivably have interfered with the models' analysis. Finally, a number of errors could also be attributed to questionable labelling, like the strawberries in Figure 7, confirming the 'garbage in, garbage out' principle (Kilkenny & Robinson, 2018).

Based on the general strong performance of all models, and the marginal performance differences from baseline to tuned models (See also baseline performance in Appendix E), it can, however, be reasoned that the dataset utilized for modelling was of general high quality with well-separated data points, strong feature signals, and sufficient quantity for the given setting. In section 5.5, Limitations, we are however to note how it can limit our models' performance in a real-life setting, that it has only been trained on data from a single source, hence, introducing limited data variability.

5.2 Feature Importance (Grad-CAM)

To attempt to further inspect how the CNN model classified the various images, and understand where and why errors occurred, Grad-CAM visualisations were produced. This technique utilizes the gradients from the CNN

model to understand which features in the images that the model found most important or salient when deciding which class to predict for an image (Selvaraju et al., 2016). The gradients are visualised as a heat map, which is then overlayed on the original image – here yellow indicates greater salience while a more purple hue indicates less. Figure 8 shows the sample images from Figure 7 as Grad-CAM visualisations. See Appendix C for choice of colourmap.

Visual inspection of hundreds of these images concluded that the model often looks at where there are colour differences on the surface of the fruit, which is in line with what we as humans would also look at to decide. For True Positives and True Negatives, we see that salience is given to darker areas for rottenness and lighter areas for freshness respectively.

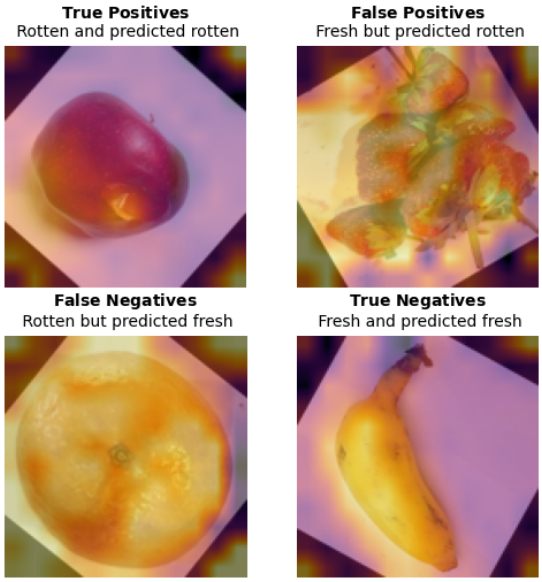


***Figure 8**: Examples of Grad-CAM visualisations in the confusion matrix structure (more yellow indicates greater salience).*

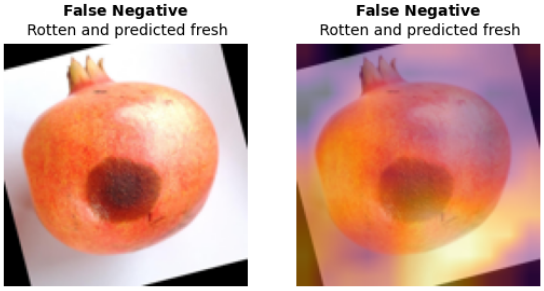


***Figure 9**: Original image and Grad-CAM of a misclassified pomegranate.*

False positive misclassifications often appeared to be where the fruit had dark edges that were mistaken for rotten areas or where multiple fruits overlap and create shadows as can be seen in the strawberries in Figure 7 and Figure 8. Interestingly, in the case of the false negative orange, the system appears to focus on the light brown hues and classifies it as fresh despite it being rotten. This could be attributed to the system's focus on the texture of the orange, which bears resemblance to that of a fresh orange, coupled with the fact that the discoloured spots are fairly light in colour.

It is also important to note that the models were trained on a variety of fruits, which ripen and rot in various fashions (See examples in Appendix F). The effects of this could for example possibly be seen in some false negatives, such as in Figure 9, where an obvious rotten spot on a pomegranate has very similar colour and texture to a fresh apple, and thus could be mistaken for this.

5.3 Practical Implications

Based on the results presented in Table 1, it can be concluded that the added complexity of the neural network can be justified in terms of improved performance for the classification task. However, the ensemble and discriminative approaches are competing with only marginal differences of approximately 3-4 percentage points in F1s. This remark not only confirms the specialized capabilities of the CNN to learn hierarchical representations and patterns in images, but also highlights the enduring popularity, applicability, and value of RF and SVM models (Géron, 2022).

Consequently, the answer to the research question can be equal parts concise and multifaceted. The performance of fresh and rotten fruit classification may vary depending on the modelling approach, but the definition of performance in a practical setting can differ. In the

context of implementing such a classification system in grocery stores for sustainability purposes, where companies have clear objectives of economic and temporal optimization, the RF or SVM may provide satisfactory results, minding the reduced execution time. Conversely, for agricultural operations aiming at precise sorting mechanisms to ensure that only the freshest products are sent to the market, whilst limiting product waste, a CNN approach could be deemed of more value, with reference to the demonstrated performance of this research. By leveraging a CNN, optimal resource utilization can be achieved, avoiding wastage of packaging materials and transportation of spoiled fruits.

5.4 Ethical Considerations

While researching advanced technologies, it is always important to reflect on the ethical and moral considerations associated. One negative concern could be that of privacy, when greatly increased numbers of cameras in grocery stores would be needed to survey the fruit that is to be classified. One solution could be to ensure that cameras point away from the path of humans, and that footage and images are deleted after classification.

Furthermore, we must consider the issue of food waste if the implementer is most concerned with freshness, as this could lead to the discarding of incorrectly classified fruits. A solution to this could be manual spot-checks, which would also require the continued employment of workers. This potentially solves the final ethical concern of loss of jobs, which is always at the forefront when designing technology to replace manual work. This technology also cannot replace the manual labour needed to act on the decisions of the algorithm, such as discarding fruit, and thus human workers are still needed.

5.5 Limitations

While our study yielded promising results, there are several limitations that should be acknowledged. When inspecting the error-prone predictions, we noticed a few instances of label inconsistencies where seemingly rotten strawberries for instance were marked as fresh. This mislabelling most likely pertains to human error during data collection, but it may have introduced noise, and as witnessed, impacted the predictions of the models. Another noteworthy limitation is the high quality of our dataset, which predominantly pictures fruits under optimal conditions i.e., on plain white surfaces. In a real-life setting, fruits may be grouped on shelves or in boxes, thus complicating the assignment of classes by a model that has only been trained on data of a certain kind. The incorporation of data from multiple sources and diverse environments would introduce greater data variability into the training process and ultimately lead to models of more robust character.

Whilst our study represents a step in the right direction toward the development of machine learning algorithms to support automated fruit quality management, it is important to recognize that additional measures extending beyond what can be visually inspected are still necessary. As covered in earlier sections, certain factors that influence the spoilage process of different fruits may not be readily discernible through visual assessment alone.

6 Conclusion & Future Work

This study trained and employed three distinct machine-learning models to explore the classification task of predicting images of fruit as fresh or rotten. The primary objective was to identify and suggest a suitable method to support sustainable food quality monitoring in the agricultural and food retail industry. An RF and an SVM model were trained on a diverse dataset supplied by Sultana et al., (2022) for model development purposes, and the two were benchmarked against a custom CNN. The added complexity of the neural network demonstrated superior performance achieving a macro average F1-score of 0.98 compared to the other two

maximising at a range of 0.93-0.96. The appropriate approach for practical implementation may however differ based on the specific context, economic objectives, and time-related optimization goals of the company or sector intending to execute the measures. We encourage future scholars to utilize our models as pretrained, supplemented with additional data from diverse production- and store-specific sources. Experimenting with increasing data variability could possibly foster improved generalization capabilities and lead the current research in the domain one step closer to an all-encompassing algorithm supporting a sustainable direction in the industry.

7 References

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324/METRICS

Géron, A. (2022). Hands-on Machine Learning with Scikit-Learn, Keras, and Hands-On Machine Learning TensorFlow. In *O Reilly Media, Inc.* (3rd ed.). O'Reilly Media, Inc. https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/

Gontijo-Lopes, R., Dauphin, Y., & Cubuk, E. D. (2022). *No One Representation to Rule Them All: Overlapping Features of Training Methods*.

Hasebrook, N., Morsbach, F., Kannengießer, N., Org Franke, J. ¨, Hutter, F., & Sunyaev, A. (2022). *Why Do Machine Learning Practitioners Still Use Manual Tuning? A Qualitative Study*. https://arxiv.org/abs/2203.01717v1

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Ishangulyyev, R., Kim, S., & Lee, S. H. (2019). Understanding Food Loss and Waste—Why Are We Losing and Wasting Food? *Foods 2019, Vol. 8, Page 297*, *8*(8), 297. https://doi.org/10.3390/FOODS8080297

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 1–54. https://doi.org/10.1186/S40537-019-0192-5/TABLES/18

Kilkenny, M. F., & Robinson, K. M. (2018). Data quality: "Garbage in – garbage out." *Https://Doi.Org/10.1177/1833358318774357*, *47*(3), 103–105. https://doi.org/10.1177/1833358318774357

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net.

Quintero Rincón, A., Mora, M., Naranjo-Torres, J., Fredes, C., & Valenzuela, A. (2022). Raspberries-LITRP Database: RGB Images Database for the Industrial Applications of Red Raspberries’ Automatic Quality Estimation. *Applied Sciences 2022, Vol. 12, Page 11586*, *12*(22), 11586. https://doi.org/10.3390/APP122211586

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2016). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Shaikh, H., Wagh, Y., Shinde, S., & Patil, S. M. (2021). Classification of Affected Fruits using Machine Learning. *International Journal of Engineering Research & Technology*, *9*(3).

https://doi.org/10.17577/IJERTCONV9IS03105

Sultana, N., Jahan, M., & Uddin, M. S. (2022). An extensive dataset for successful recognition of fresh and rotten fruits. *Data in Brief*, *44*, 108552. https://doi.org/10.1016/j.dib.2022.108552

Yang, L., & Shami, A. (2020). On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Zhang, Q., Cao, R., Shi, F., Wu, Y. N., & Zhu, S. C. (2018). Interpreting CNN Knowledge via an Explanatory Graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1), 4454–4463. https://doi.org/10.1609/AAAI.V32I1.11819

8 Appendices

Appendix A *- Number of images in the original and augmented datasets.*

Category	No. of Original Images	No. of Augmented Images
Fresh Apple	200	734
Rotten Apple	200	738
Fresh Banana	200	740
Rotten Banana	200	736
Fresh Orange	200	796
Rotten Orange	200	796
Fresh Grape	200	800
Rotten Grape	200	746
Fresh Guava	200	797
Rotten Guava	200	797
Fresh Jujube	200	793
Rotten Jujube	200	793
Fresh Pomegranate	200	797
Rotten Pomegranate	200	798
Fresh Strawberry	200	737
Rotten Strawberry	200	737
Total	**3,200**	**12,335**

Appendix B *– The number of images in each class after the dataset was split.*

Image 1. The number of images in the 80:20 train/test split

Data set split	Label 0: Fresh	Label 1: Rotten	Total
Train	4938	4930	9868
Test	1256	1211	2467
----------	----------	----------	----------
Total	6194	6141	12335

Image 2. The number of images in the 60:20:20 train/validation/test split

Data set split	Label 0: Fresh	Label 1: Rotten	Total
Train	3741	3660	7401
Validation	1197	1270	2467
Test	1256	1211	2467
----------	----------	----------	----------
Total	6194	6141	12335

Appendix C *- Matplotlib's 'Viridis' colormap used for Grad-CAM (Hunter, 2007).*

Appendix D – *Confusion matrices (test set predictions).*

Image 1. Random Forest

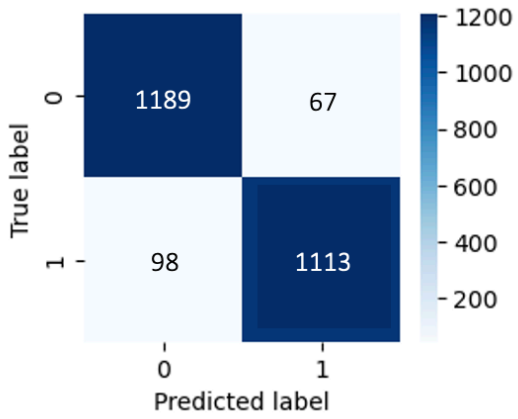


Image 2. SVM

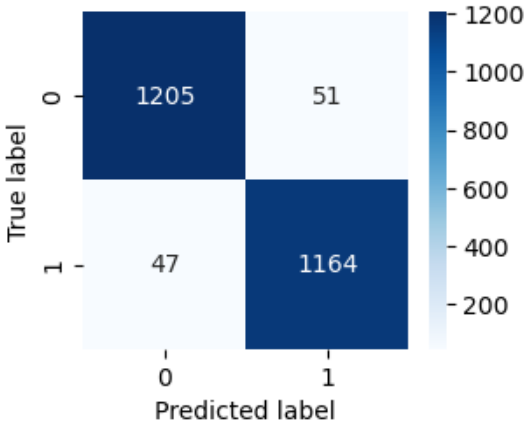
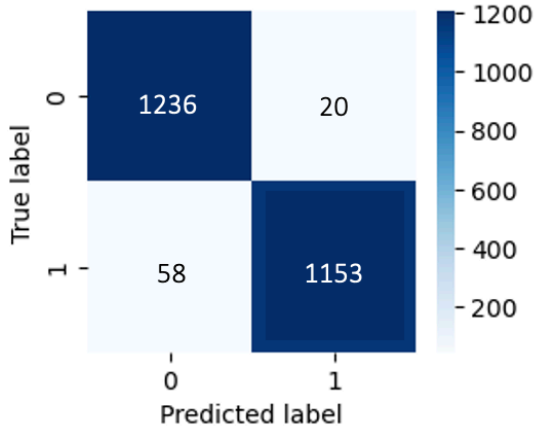


Image 3. CNN



Appendix E *–Performance of baseline models.*

Image 1. Random Forest

Performance Measure Table (Baseline Model, Test Data):

Performance Measure	Label 0: fresh	Label 1: rotten
Precision	0.7579	0.7523
Recall	0.7627	0.7473
F1	0.7603	0.7498
Support	1256	1211

Baseline Model Accuracy Score: 0.7552

Image 2. SVM

Performance Measure Table (SVM Baseline 1 Model (no PCA), Test Data):

Performance Measure	Label 0: fresh	Label 1: rotten
Precision	0.9174	0.9247
Recall	0.9283	0.9133
F1	0.9228	0.919
Support	1256	1211

SVM Baseline 1 Model Accuracy Score: 0.921

Image 3. SVM + PCA

Performance Measure Table (SVM Baseline 2 Model (with PCA), Test Data):

Performance Measure	Label 0: fresh	Label 1: rotten
Precision	0.9175	0.9263
Recall	0.9299	0.9133
F1	0.9237	0.9198
Support	1256	1211

SVM Baseline 2 Model Accuracy Score: 0.9218

Image 4. CNN

Performance Measure Table (CNN Baseline Model, Test Data):

Performance measure	Label 0: Fresh	Label 1: Rotten
Precision	0.9545	0.9681
Recall	0.9697	0.9521
F1	0.9621	0.96
Support	1256	1211

CNN Baseline Model Overall Accuracy Score: 0.9611

Appendix F *– Examples of Sultana et al., (2022)'s fresh and rotten fruits descriptions, taken from Table 2 in the original paper.*

Fresh Strawberry	Strawberries have a juicy texture and are soft, sweet, vivid red fruit. Fresh strawberries have a vivid red hue, a natural gloss, and green crowns that appear to be new. Strawberry seeds are tiny edible seeds that develop all over the top of fresh strawberries. It should have a firm consistency but not be crunchy. Overly ripe strawberries might be excessively soft.	
Rotten Strawberry	A moldy odor or feel defines a rotten strawberry. It has a rotten smell and a damaged appearance. If the color, flavor, or texture of a strawberry changes from its original tone, it can be categorized as rotten strawberry. If there are any stains, flaws, or other symptoms of degradation, the strawberry has already become rotten.	
Fresh Banana	A banana has a curved shape and a thick, sweet-tasting skin. Depending on ripeness, the hue should range from green to dark yellow with brownish flecks. A banana is a tropical fruit that is widely consumed around the world. Fresh bananas are those that are slightly but not excessively green, bright in color, full and plump, and have no depressed, wet, or black spots on the peel.	
Rotten Banana	In a process known as enzymatic browning, high levels of ethylene cause the yellow pigments in bananas to degrade into those distinctive brown marks. If the banana has a lot of brown or black areas inside the peel mildew appears, or if it has an awful odor, it's probably past its prime.