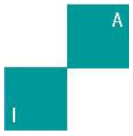
Inteligencia Artificial 27(74),48-61
doi: 10.4114/intartif.vol27iss74pp47-61



INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

FRESHNets: Highly Accurate and Efficient Food Freshness Assessment Based on Deep Convolutional Neural Networks

Jorge Felix Martínez Pazos [1,2,*], Jorge Gulín González [1], David Batard Lorenzo [1], Arturo Orellana García [2]
[1] CEMC – Study Center on Computational Mathematics, at University of Informatics Science, Havana, Cuba
[2] CESIM – Medical Informatics Center, at University of Informatics Science, Havana, Cuba
[*] jorgefmp.mle@gmail.com

Abstract Food freshness classification is a growing concern in the food industry, mainly to protect consumer health and prevent illness and poisoning from consuming spoiled food. Intending to take a significant step towards improving food safety and quality control measures in the industry, this study presents two models based on deep learning for the classification of fruit and vegetable freshness: a robust model and an efficient model. Models' performance evaluation shows remarkable results; in terms of accuracy, the robust model and the efficient model achieved 97.6% and 94.0% respectively, while in terms of Area Under the Curve (AUC) score, both models achieved more than 99%, with the difference in inference time between each model over 844 images being 13 seconds.

Keywords: Convolutional Neural Networks, Deep Learning, Food Freshness Classification, Xception.

1 Introduction

The classification of freshness in fruits and vegetables is a pivotal concern in the food industry, influencing consumer health, buying habits, and market pricing [1]. The advent of computer vision and machine learning has facilitated the creation of algorithms for automated object detection and recognition. These techniques have found applications in the fruit processing industry, where the categorization and grading of fruit freshness are crucial for delivering superior-quality products [2-4]. Fruits are vulnerable to viral and fungal infections, which exert economic strain on the agricultural sector. Manual sorting of fruit based on quality is labor-intensive [5]. Several studies have been conducted on the application of Artificial Intelligence (AI) in fruit identification and quality assessment. Goyal & Verma (2023) developed an AI-based system for fruit identification and quality detection using the YOLOv5 object detection system, which works in two stages: fruit identification and quality assessment. The dataset used in this research consists of 10,545 images of four different fruits (apples, bananas, oranges, and tomatoes) categorized based on their quality. Anupriya (2022) applied support vector machines (SVM) and the VGG-16 architecture to an apple fruit image dataset to predict fruit quality, where results showed that the VGG-16 architecture outperformed the SVM in terms of accuracy, confirming that deep learning can be superior to machine learning in computer vision tasks. These studies highlight the potential of AI to revolutionize the process of fruit quality assessment. Ren. X et al. (2023) developed a Convolutional Neural Network (CNN)--based electronic nose system for food freshness classification. The system, which consisted of a sensitive gas sensor array and a CMOS integrated circuit, took fixed exposures at specified intervals under varying gas conditions, allowing the extraction of time-series features from the sensor signals that were used to identify subtle differences in food odors at different freshness levels. The system achieved 97.3% classification accuracy for 20 types of food, with a 6.5% improvement after implementing time-series feature extraction.

Recognizing the significant contributions of AI to food freshness classification, which has profound implications for various industries and directly affects consumer benefits and safety, this study aims to develop a solution for the

ISSN: 1137-3601 (print), 1988-3064 (on-line)

multi-class classification of fruits and vegetables based on their freshness using deep learning techniques. The dataset used in this study consists of 18 different classes, including nine categories of fresh fruits and vegetables and nine categories of their spoiled counterparts [9]. The research will focus on two different approaches:

- Leveraging Transfer Learning Through the Xception Architecture: TL Model (Robust).
- Designing a Deep Convolutional Neural Network from Scratch, Drawing Inspiration from the Xception Architecture: XICNN Model (Efficient).

The following is an outline of the contributions that have been provided to draw attention to the relevance of the work that will be presented by this study:

- A robust deep learning model using transfer learning via the Xception architecture, designed for high performance in precision-critical use cases.
- An efficient deep learning model using separable convolutional blocks with residual layers, characterized by low inference time, ideal for real-time applications.
- Implement class balancing to avoid bias and improve model reliability and accuracy.
- The Use of learning rate reduction to accelerate gradient descent convergence, optimizing the learning process and improving model performance.

2 Materials & Methods

The proposed solutions in this study employ a variety of state-of-the-art technologies and methodologies to achieve the highest possible performance in terms of accuracy and efficiency. Python serves as the primary programming language, with Tensorflow and Keras utilized for model development. Data manipulation is handled using Pandas and Numpy, while visualization is accomplished through Matplotlib and Seaborn. Despite the use of other libraries and frameworks, these constitute the core tools employed in this work. The primary algorithms and techniques applied in this research are Convolutional Neural Networks, Transfer Learning, and Fine Tuning, which are the main techniques for achieving better performance in image classification tasks. The comprehensive workflow step-by-step for constructing both solutions is detailed in Figure 1.

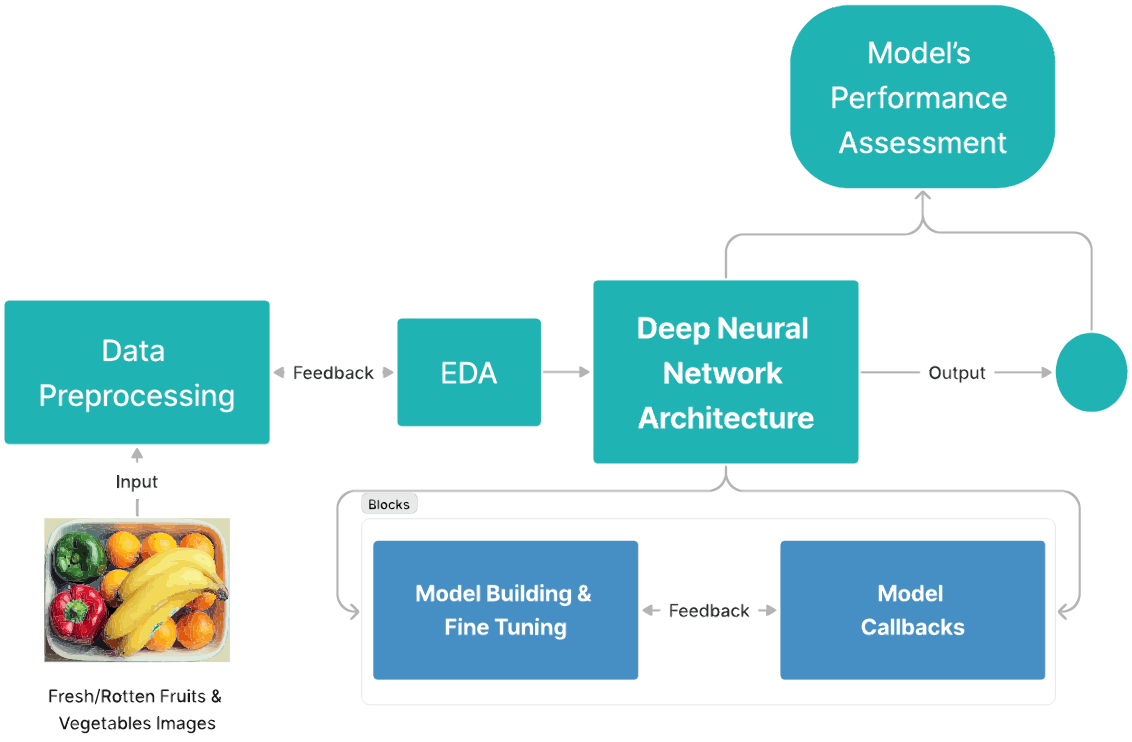


Figure 1: Workflow step-by-step of the proposed solutions. Source: The authors

Data Preprocessing: The first step in the process is to preprocess the input data to prepare it for use in the model. This involves resizing, normalizing, and augmenting the images.

Exploratory Data Analysis (EDA): The second step is to perform exploratory data analysis on the pre-processed data to understand its characteristics and distribution. This step may reveal additional insights about the data that could lead to further preprocessing to improve the models' performance.

Deep Neural Network Architecture: The Third step in the process involves designing the architecture of the deep neural network model. This step has two sections: Model Building & Fine Tuning and Model Callbacks. These sections are interrelated due to the presence of a feedback mechanism between them, mainly because callbacks directly affect the models' performance by reducing the learning rate, stopping early, and other techniques.

Model Building & Fine Tuning: As part of the deep neural network architecture design, the model is built and fine-tuned using the pre-processed data. This involves training the model on a subset of the data and validating its performance on another subset to optimize its hyperparameters and prevent overfitting.

Model Callbacks: During training, callbacks are used to monitor the models' performance and make necessary adjustments such as saving the best-performing model, stopping training early if the models' performance plateaus, or adjusting the learning rate.

Model Performance Assessment: The fourth step in the process is to assess the performance of the trained and fine-tuned model on a separate test dataset. This involves calculating various performance metrics to evaluate how well the model performs on unseen data. In the performance metrics discussed below, TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

- Accuracy: Indicates how close the measured value is to a known value.

$$\text{Accuracy} = \frac{(\text{TP}+\text{FN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})} \quad (1)$$

- Precision: Indicates how accurate the model is in terms of those predicted to be positive.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})} \quad (2)$$

- Recall: Calculates the number of real positives that the model was able to capture after labelling it as positive.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP}+\text{FN})} \quad (3)$$

- F1: Provides a balance between precision and recall.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{(\text{Precision}+\text{Recall})} \quad (4)$$

- ROC (Receiver Operating Characteristics) Curve: Graphical representation of the relationship between a classifier's sensitivity (true positive rate) and 1- its specificity (false positive rate).

- AUC (Area Under the Curve) Score: Measure of the degree to which a classifier can distinguish between classes. A higher AUC score indicates a better ability to differentiate between classes.

2.1 Data Preprocessing

The image format used is RGB with a size of 100x100, which is a balance between the minimum (71x71) and maximum (299x299) possible input values for the Xception architecture. Keeping the images at a resolution of around 100 pixels and in RGB scale means that more patterns in the pixel array can be examined by the model without increasing the processing time too much. This allows for optimal computation of the predictive model and its training process, as working with higher-resolution images would incur high processing costs and time.

For optimal processing, the images are rescaled from a range of 0-255 to a range of 0-1. This is a common preprocessing step in image analysis and machine learning, as it helps to normalize the data and improve the performance of the model. By rescaling the pixel values, the model can more easily learn the relationships between the different features in the image, leading to better predictions.

2.2 Exploratory Data Analysis & Further Preprocessing

It appears that augmentation techniques have been applied to the dataset. If this is the case, it may not be necessary to apply image data augmentation during the training process. To confirm this, the dataset is deeply analysed, Figure 2 shows a representative sample of 100 images extracted from the dataset. Upon closer inspection, it is evident that the dataset includes images that have undergone various transformations, such as rotation, zoom, and other forms of augmentation, which is a confirmation of the previously mentioned idea.

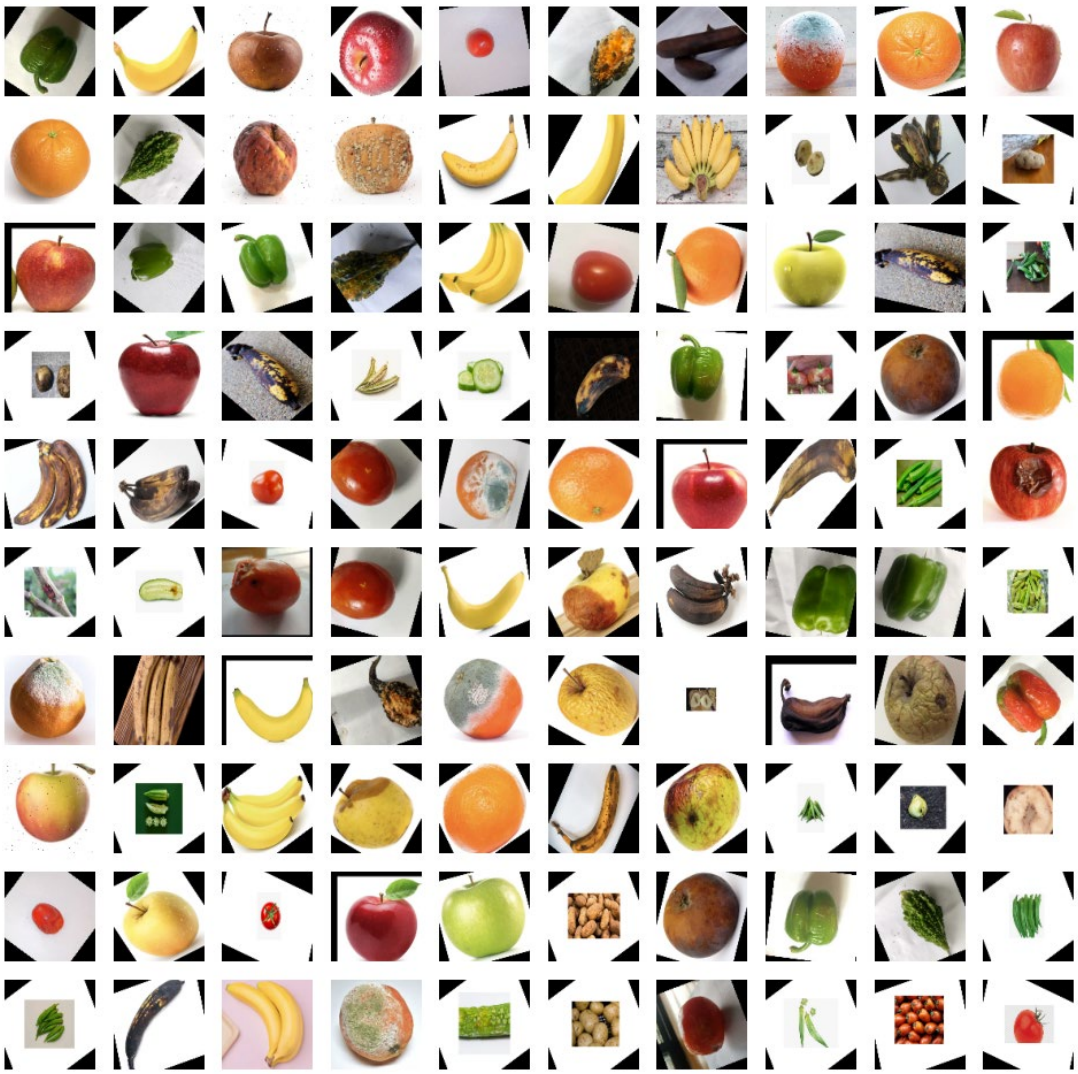


Figure 2: Sample of 100 images from the dataset. Source: The authors

2.2.1 Comparing Fresh & Rotten Conditions

When comparing the images of fresh and rotten fruits and vegetables in Figures 3 and 4, it can be seen that the biggest difference is in terms of brightness and high opacity, while fresh fruits and vegetables have more intense brightness and color, rotten fruits and vegetables have dark colors with less brightness and opacity. In certain predominant cases, it has been observed that rotten fruits and vegetables exhibit a more pronounced relief in comparison to their fresh counterparts. This is mainly due to the decomposition of their texture.

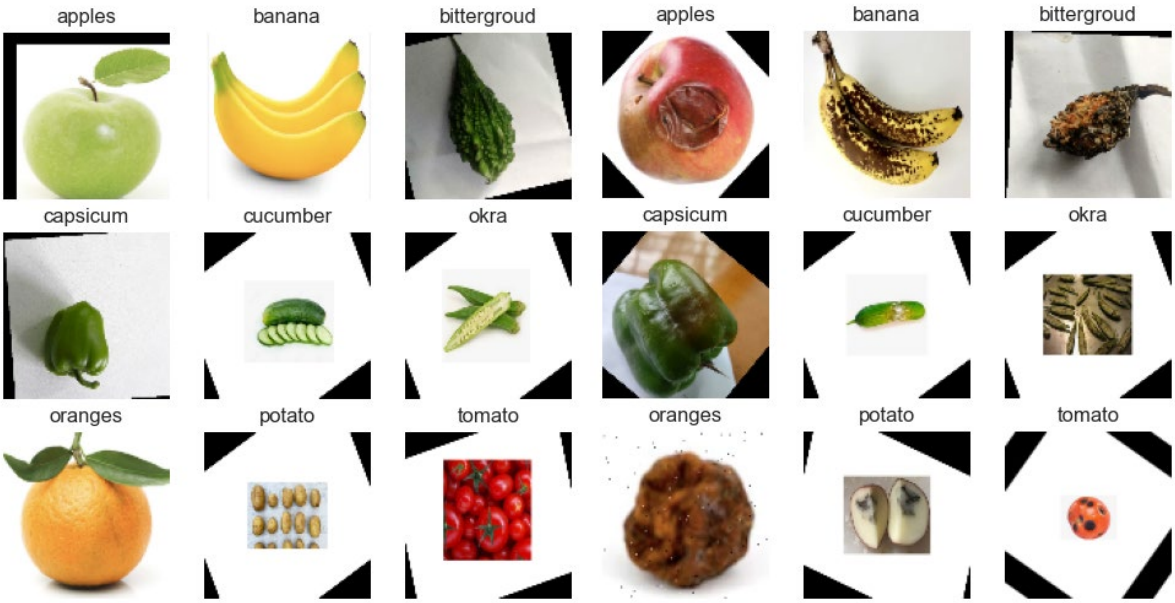


Figure 3: Left to mid fresh fruits and mid to right rotten fruits. Source: The authors

2.2.2 Dataset Distribution

The class distribution reveals a highly unbalanced dataset, where the most common class in the dataset is rottenapples, with a total of 3248 instances, while the least common class is freshbittergroud, with a total of 327 instances. The difference between these values is significant.

It is important to train supervised learning models with balanced data to avoid bias and improve model performance. If a dataset is unbalanced, meaning that one class has significantly more instances than another, which is the case, the model may be biased toward the majority class and not perform well on the minority class. Figure 4 provides a detailed representation of the distribution of classes within the dataset. Of the 18 total classes, the four classes representing rottenapples, rottenbananas, freshbananas, and freshapples comprise nearly 50% of the dataset. Despite the presence of a significant class imbalance within the dataset, the distribution of fresh and rotten states among fruits and vegetables is balanced.

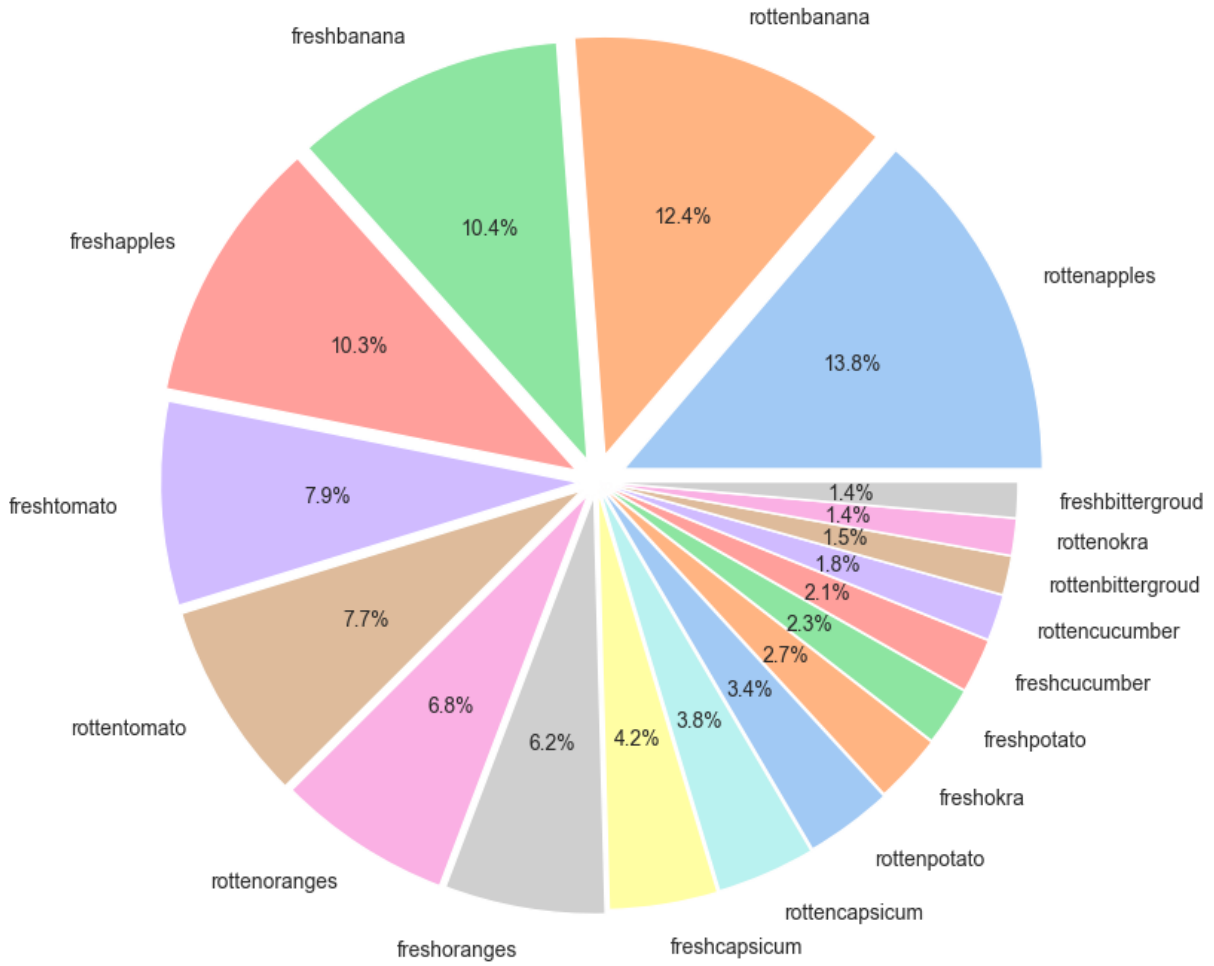


Figure 4: Class distribution before balancing. Source: The authors

By limiting each class to 500 instances, an optimal class balance is created, which can help improve the performance of the model over all classes. Resulting in a highly balanced dataset in terms of state and classes, which is beneficial for the training process and the model performance.

2.3 Model Building

Given the conclusion that the dataset already encompasses images with multiple types of augmentation, this technique is not employed during the model construction or training phase. This decision is based on the understanding that additional augmentation could potentially introduce redundancy and may not contribute to the enhancement of the model's performance. In terms of dataset partitioning, a three-way split is implemented resulting in training, validation, and test sets of 80%, 10%, and 10%, respectively:

- The training set comprises 80% of the total images in the dataset. This substantial portion is allocated to allow the model to learn and extract meaningful patterns effectively.
- The remaining 20% of the images from the validation set, which is used to fine-tune the model parameters and prevent overfitting.
- Furthermore, half of the images from the validation set are reserved for the test set. This set is utilized to evaluate the final models' performance, providing an unbiased assessment of its effectiveness in classifying unseen data. This rigorous partitioning strategy ensures a robust and comprehensive evaluation of the model's performance across various stages of the machine learning pipeline.

2.3.1 Transfer Learning Through Xception Architecture

Previous studies have shown that the Xception architecture generally outperforms other architectures in freshness classification tasks for fruits and vegetables [10-11], therefore this is the architecture we adopt to apply transfer learning and fine-tuning. Xception's architecture is tailored to specific configurations, including a 100x100x3 input layer, and frozen weights using pre-trained ones from ImageNet, a database structured according to the WordNet hierarchy [12]. Next, the output stream is connected to the output of the Xception engine, which includes a 2D Global Average Pooling layer that has a significantly high compression ratio, resulting in a 2D dimensionality of the form (batch_dimension, n_channels), as opposed to the Flatten layer typically used to feed fully connected layers, which simply reshapes the matrix into a single dimension [13]. Next, a 70% probability Dropout layer is added to counteract potential overfitting due to the extensive pre-trained model, along with a Batch Normalization layer to normalize the inputs, ensuring that the mean output remains close to 0 and the standard deviation of the output remains close to 1 [14, 15]. The model culminates with a dense output layer of 18 neurons corresponding to the domain classes, using the softmax activation function to obtain the class probabilistic vector.

The model is built using the Adaptive Moment Estimation (Adam) optimizer with a learning rate of 3e-4, the categorical cross-entropy loss function, the accuracy metric, and a batch size of 32. The learning rate and dropout values were determined from the tuning process using the Keras tuner, with the hyperparameter search performed at [0.03, 0.003 0.0003] and [0.65, 0.70, 0.75] for learning rate and dropout, respectively [16]. The batch size parameter was chosen based on recommendations for low-performance computing. The learning rate values were chosen considering that while the default value for the Adam optimizer is 0.001, a commonly used value is 0.0003 [17]. Therefore, an interval from 0.03 to 0.0003 was used, which is an order of magnitude reduction. High dropout values were chosen because the amplitude of the pre-trained model could cause the predictive model to overfit.

Callbacks

During the training process, four callbacks will be implemented, namely early stop, learning rate reduction, tensorboard, and checkpoint [13]:

- Early Stop: This callback halts the training process when no further improvements are observed in the loss metric. It serves as a mechanism to prevent unnecessary computations and potential overfitting.
- Learning Rate Reduction: This callback adjusts the learning rate of the model, initially set at 3e-4, by monitoring the loss on validation data. This allows for enhanced optimization and model accuracy on the validation data.
- Tensorboard: This is a tool incorporated in the Tensorflow Framework that provides real-time visualization of all variables and the model's behaviors. It facilitates the optimization of hyperparameters.
- Checkpoint: This callback enables the saving of the model's weights and biases at a certain state during training. In this case, it saves each time the model improves based on the validation accuracy.

The model undergoes training for 30 epochs using the validation set, the selected batch size, and the aforementioned callbacks. The techniques of learning rate reduction and dropout layer, along with real-time monitoring of hyperparameters through tensorboard, are specifically employed to mitigate overfitting of the predictive model.

2.3.2 Deep Convolutional Neural Network from Scratch Inspired in Xception Architecture

The Deep Convolutional Neural Network is inspired by the Xception Architecture and consists of three main components: an entry flow, a middle flow, and an exit flow. In the entry flow, the input data is processed through several layers of convolution, max pooling, and batch normalization to extract features from the data. A residual connection is also introduced at this stage to improve the flow of information through the network. The middle flow, which is repeated either once or twice depending on a hyperparameter choice, consists of separable convolution, max pooling, and batch normalization layers that further refine the extracted features. The output flow applies another set of separable convolution, max pooling, and batch normalization layers, followed by a global average pooling layer to reduce the dimensionality of the data. A dense layer with a hyperparameter choice for the number of units is then applied, followed by a dropout layer with a hyperparameter choice for the dropout rate to prevent overfitting. Another `batch normalization` layer is applied before the final dense layer with a softmax activation function to generate the output probabilities for each class. In each convolutional layer, the bias parameter is set to

false since the batch normalization layer is applied at the end of each convolutional block. Figure 5 shows a layered visualization of the XICNN architecture obtained using the visualkeras package [18].

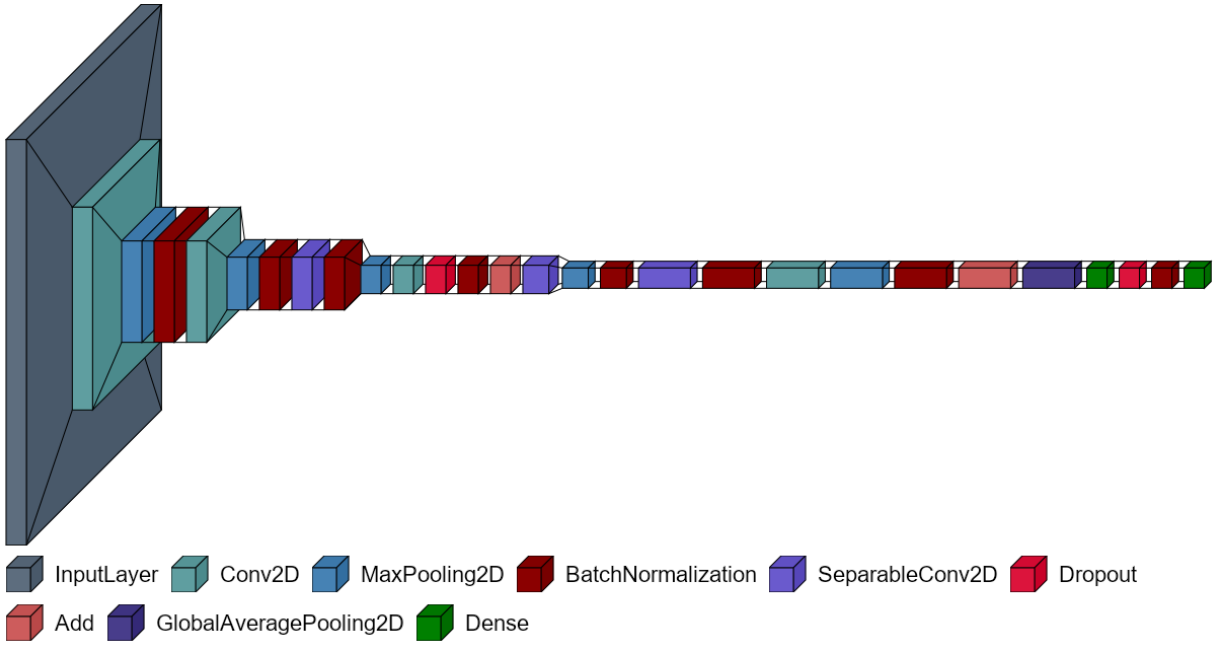


Figure 5: Architecture of FRESHNet: (XICNN). Source: The Authors

The model is compiled using the Adam optimizer with a hyperparameter choice for the learning rate. The loss function used is categorical cross-entropy, and accuracy is used as the evaluation metric. The following image is a graphical representation of the XICNN Model. The same callback configurations are used. The model is trained through 100 epochs using the validation set, the selected batch size, and the aforementioned callbacks, the patience value of early stopping is increased from 3 to 7 and the min lr of learning rate reduction is decreased from 0.00001 to 0.000001.

3 Results & Discussions

3.1 Models Performance Assessment

Evaluating predictive models on unseen data is crucial to ensure their real-world performance. Without thorough evaluation on data outside the training process, there is a risk that the model may not perform as well in real-world scenarios, leading to incorrect decision-making. By evaluating the model with new data, a more accurate understanding of its real-world performance can be gained, increasing confidence in its use and ensuring its reliability and safety for use in real-world solutions.

As detailed in the model history in Figures 6 and 7, both models exhibit robust generalization capabilities, as evidenced by their consistent performance across the training and validation datasets. The best values for each metric are obtained at epochs 9 and 31 for the TL Model and XICNN Model respectively, so this is the configuration of the models stored by the checkpoint callback.

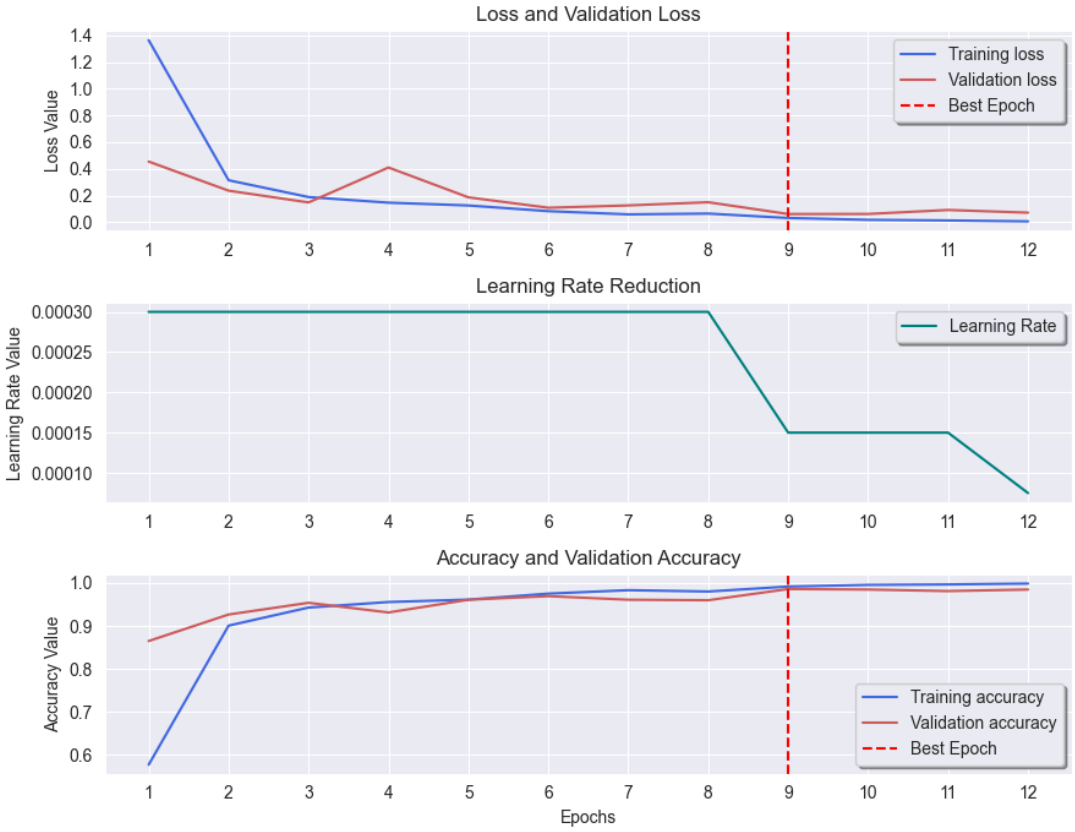


Figure 6: Model History of FRESHNet (TL Model). Source: The authors

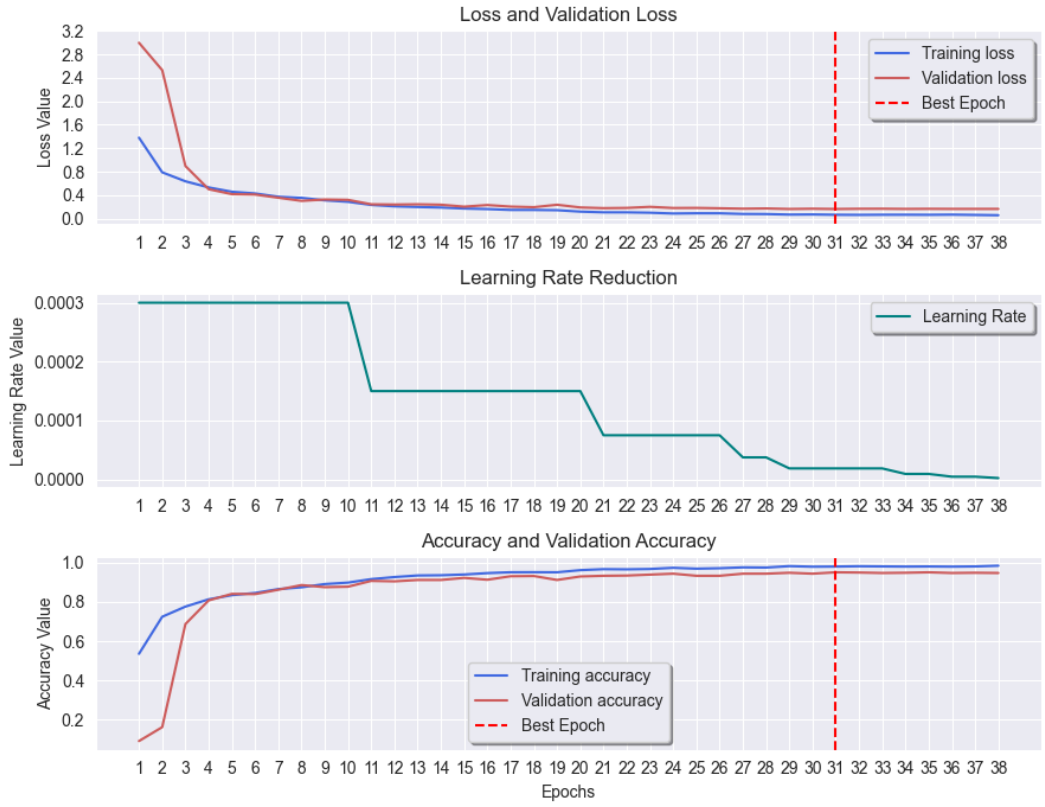


Figure 7: Model History of FRESHNet (XICNN Model). Source. The authors

The accuracy values achieved by the models at their best epochs were of 98.57% and 95.17% for the TL Model and XICNN Model respectively. In comparison to the performance of the models on the validation set, both models exhibited similarly high performance on the test set for each assessed metric, with a minimal decrease of approximately 1% for each model. Both models demonstrated a remarkably low misclassification rate, with the TL Model and XICNN Model misclassifying only 20 and 42 images, respectively, out of a total of 844 images.

As depicted in Figure 8, both models exhibit maximum AUC values for more than half of the classes within the domain using the One versus Rest (OvR) approach, indicating their ability to accurately classify fresh/rotten fruits and vegetables. Overall, both models demonstrate exceptional performance, with minimum AUC values of 99.27 and 99.15 for the TL Model and XICNN Model, respectively.

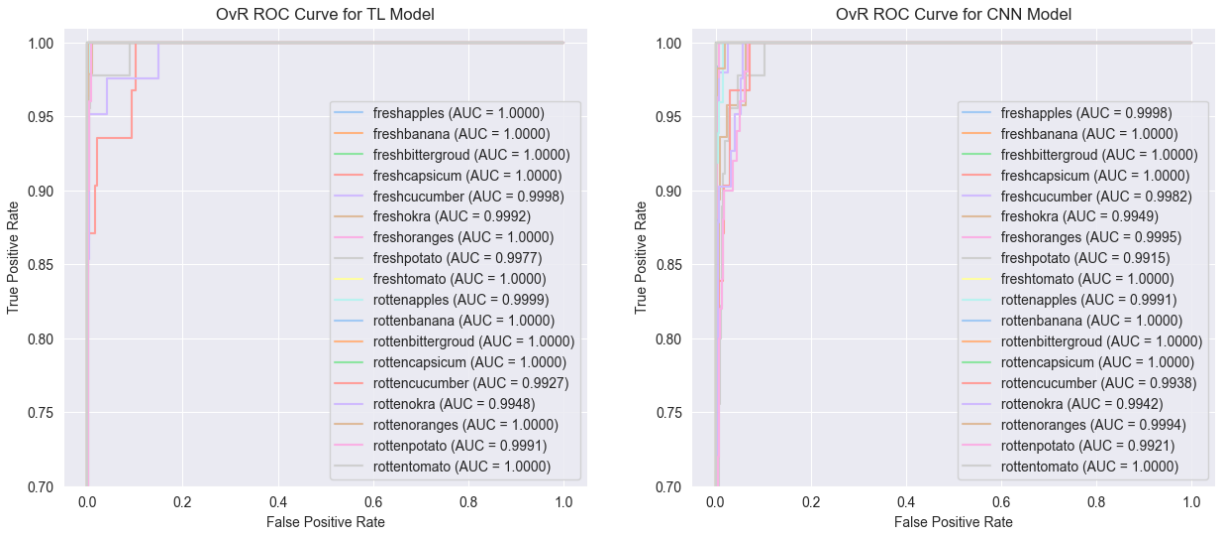


Figure 8: ROC Curve and AUC Scores for both FreshNets. Source: The authors

Figure 9 details the performance metrics of both models TL Model & XICNN Model evaluated over a test set of 844 images. TL Model shows superior performance in terms of accuracy, precision, recall, and F1 score, outperforming the XICNN Model by less than 4%, while both models achieve an AUC score above 99%. In contrast, the XICNN Model shows significantly higher efficiency in terms of processing time compared to the TL Model.

Figure 9: Performance and Efficiency Comparison of both FRESHNets. Source: The authors

Eight images of real-world scenarios were selected from the internet for classification using the proposed models. A visual representation of each image is presented in Figure 10 and Figure 11, along with its respective classification, which is indicated by the label on the image title. The previous results demonstrate the remarkable performance of both models in correctly classifying the fresh/rotted fruit and vegetable domain even in real scenarios.

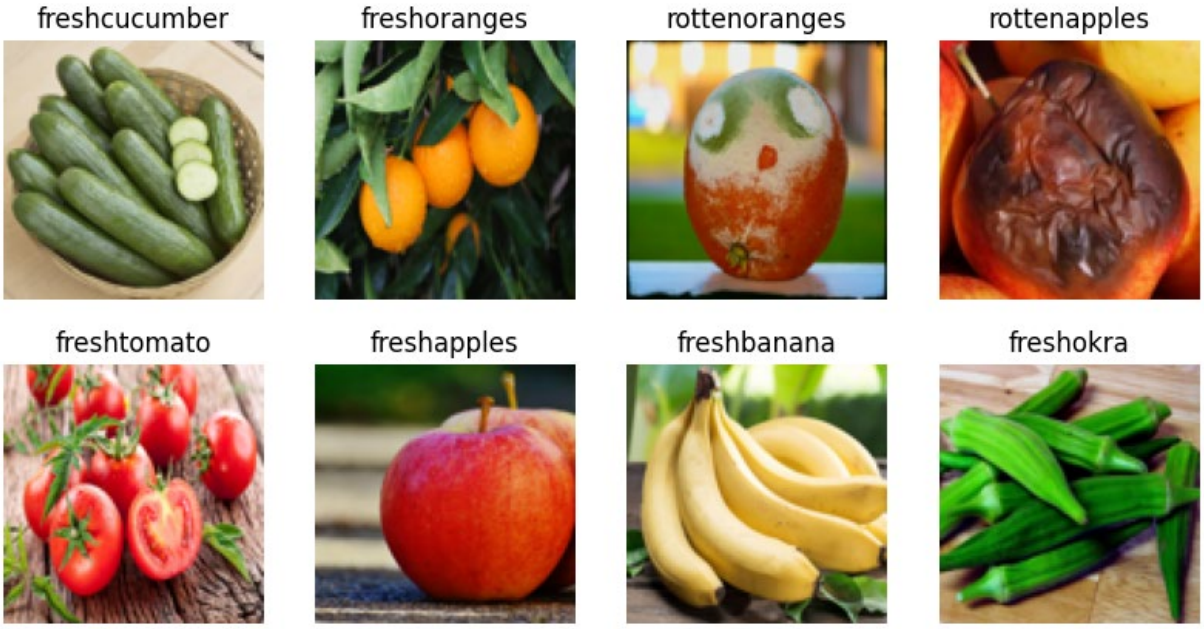


Figure 10: TL Model Predictions. Source: The authors

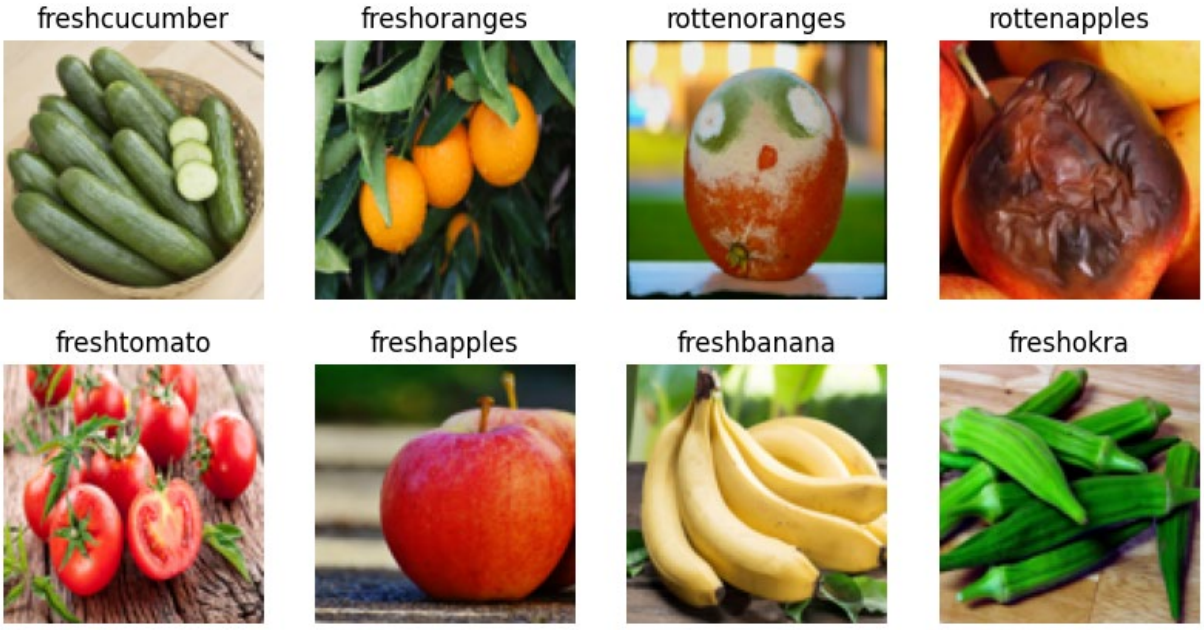


Figure 11: XICNN Model Predictions. Source: The authors

A comparative analysis was performed between the robustly proposed model, called the TL model, and the most similar models found in the reviewed literature. It is important to note that these models use different datasets for training and evaluation, in addition, the model proposed by Miah et al. uses two classification heads: one for fruit classification and another for freshness classification; to obtain the overall accuracy, we calculated the average of the results obtained from both classification heads. The proposed model evaluates a significantly larger number of fruit and vegetable classes compared to other models. With a difference of 0.57%, the model proposed by Amin et al. surpassed the performance of our model, being the model with the highest accuracy, but using only 6 classes against the 18 classes of our study. In terms of the relationship between the number of classes and accuracy, the closest model to ours is the one proposed by Kang & Gwak with 0.32% superior accuracy, but with 4 classes less than the proposed model.

Despite a generally slightly lower accuracy, the proposed model outperforms others when considering the ratio of the number of classes and accuracy, which demonstrates the superior capability of the proposed model in the comparative analysis detailed in Table 1.

Table 1: Comparison of the proposed models with the reviewed literature.

Model	Accuracy	No. Classes
Robust Proposed Model	**97.63**	**18**
Palakodati, S. S. S., et al [4]	97.8	6
Amin U., et al [5]	98.2	6
Kang, J., & Gwak, J [2]	97.95	14
Miah, M. S et al [11]	97.34	10

3.2 Strengths & Limitations

Although the TL Model demonstrated marginally superior performance compared to the XICNN Model, the latter may be the optimal choice for software solutions that require real-time processing due to its high efficiency, utilizing approximately 10% of the time required by the TL Model to process images. However, if the solution prioritizes precision over efficiency, then the TL Model would be the preferred choice.

It should be recognized both models have certain limitations. The performance of the proposed models was not assessed on real-world data, but solely on a select set of eight images for verification. Therefore, future research aims to evaluate the efficacy of our models across diverse image datasets derived from real-world scenarios. This approach will provide a more comprehensive understanding of the models' applicability and robustness. Further, integrate the most efficient FRESHNet (XICNN) to a solution using image segmentation and detection for real-time food freshness classification.

The proposed solutions for classifying fruits and vegetables as fresh or rotten have the potential to revolutionize various industries, they can be applied in several ways to improve food safety, reduce waste, and increase efficiency. In supermarkets, this solution can be used to ensure that the fruits and vegetables sold are fresh and of high quality. By implementing this technology, supermarkets can improve customer satisfaction and reduce waste. In warehouses, inventory management can be enhanced by using this solution to monitor the freshness of stored fruits and vegetables. This allows for informed decisions about restocking or disposing of produce. Government agencies responsible for food safety can utilize this solution during routine inspections to assess the freshness of fruits and vegetables. This helps to ensure that consumers are not exposed to rotten produce. In agricultural research, this solution can be used to accurately classify produce as fresh or rotten. This provides valuable data for researchers studying the shelf life of fruits and vegetables. Smart home appliances, such as refrigerators, can also benefit from this technology by monitoring the freshness of stored fruits and vegetables. This helps consumers make informed decisions about when to consume or dispose of produce.

4 Conclusions

The advent of intelligent systems has revolutionized the food industry, improving its management, development, and sustainability while reducing costs. This study presents the development of two predictive models for the

classification of fresh and perishable fruits and vegetables, leveraging the power of deep convolutional neural networks, transfer learning, and the pre-trained Xception architecture.

The robust TL Model demonstrated remarkable performance, achieving approximately 97.6% accuracy, precision, recall, and F1 score when evaluated over 844 images with an inference time of 15 seconds. On the other hand, the efficient XICNN Model showed impressive efficiency, achieving approximately 94% for each evaluated metric within just 2 seconds of inference. Notably, both models achieved an AUC score of approximately 99%, indicating their excellent performance.

The observed performance of the models was made possible by hyperparameter tuning using the Keras Tuner, a learning rate reduction callback, and the incorporation of normalization techniques. Results on unseen data underscore the reliability of both models, with each model showing superiority in either performance or efficiency, demonstrating their potential for practical applications in the food industry**.**

Acknowledgments

The authors would like to express their profound gratitude to Natasquad and their dedicated team for fostering the advancement of research through their Artificial Intelligence Hackathon, which served as the birthplace of this study. Their commitment to promoting scientific exploration is deeply appreciated and was instrumental in the realization of this research.

References

[1] Mukhiddinov M, Muminov A, Cho J. Improved Classification Approach for Fruits and Vegetables Freshness Based on Deep Learning. Sensors. 2022; 22(21):8192. DOI: https://doi.org/10.3390/s22218192

[2] Kang, J., & Gwak, J. (2022). Ensemble of multi-task deep convolutional neural networks using transfer learning for fruit freshness classification. Multimedia Tools and Applications, 81(16), 22355-22377. DOI: https://doi.org/10.1007/s11042-021-11282-4

[3] Valentino, F., Cenggoro, T. W., & Pardamean, B. (2021, July). A design of deep learning experimentation for fruit freshness detection. In IOP Conference Series: Earth and Environmental Science (Vol. 794, No. 1, p. 012110). IOP Publishing. DOI: https://doi.org/10.1088/1755-1315/794/1/012110

[4] Palakodati, S. S. S., Chirra, V. R. R., Yakobu, D., & Bulla, S. (2020). Fresh and Rotten Fruits Classification Using CNN and Transfer Learning. Rev. d'Intelligence Artif., 34(5), 617-622. DOI: https://doi.org/10.18280/ria.340512

[5] Amin U, Shahzad MI, Shahzad A, Shahzad M, Khan U, Mahmood Z. Automatic Fruits Freshness Classification Using CNN and Transfer Learning. Applied Sciences. 2023; 13(14):8087. DOI: https://doi.org/10.3390/app13148087

[6] Goyal, K., Kumar, P. & Verma, K. AI-based fruit identification and quality detection system. Multimed Tools Appl 82, 24573–24604 (2023). DOI: https://doi.org/10.1007/s11042-022-14188-x

[7] Anupriya, K., Sri, G.M. (2022). Fruit Freshness Detection Using Machine Learning. In: Mallick, P.K., Bhoi, A.K., Barsocchi, P., de Albuquerque, V.H.C. (eds) Cognitive Informatics and Soft Computing. Lecture Notes in Networks and Systems, vol 375. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-16-8763-1_52

[8] Ren. X et al., (2023) A CNN-Based E-Nose Using Time Series Features for Food Freshness Classification. IEEE Sensors Journal, vol. 23, no. 6, pp. 6027-6038. DOI: https://doi.org/10.1109/JSEN.2023.3241842

[9] Siddharth, S.N. (2023). Fresh and Rotten Classification. Kaggle Datasets. Available online https://www.kaggle.com/datasets/swoyam2609/fresh-and-stale-classification

[10] Mamidi, S. S. R., Munaganuri, C. A., Gollapalli, T., Aditya, A. T. V. S., & B, R. C. (2022). Implementation of Machine Learning Algorithms to Identify Freshness of Fruits. In *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)* (pp. 1395-1399). Kannur, India. DOI: https://doi.org/10.1109/ICICICT54557.2022.9917989

[11] Miah, M. S., Tasnuva, T., Islam, M., Keya, M., Rahman, M. R., & Hossain, S. A. (2021, July). An advanced method of identification fresh and rotten fruits using different convolutional neural networks. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE. DOI: https://doi.org/10.1109/ICCCNT51525.2021.9580117

[12] Fei-Fei, Li., Deng Jia, Russakovsky, Olga., Berg, Alex., Li, Kai. (2009). ImageNet. Available online: https://www.image-net.org/index.php

[13] Keras. Keras API Reference. Available online: https://keras.io/api. Last Accessed 28/3/2023

[14] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580. DOI: https://doi.org/10.48550/arXiv.1207.0580

[15] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr. DOI: https://doi.org/10.48550/arXiv.1502.03167

[16] O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., & Invernizzi, L. (2019). Keras tuner. Available online https://github.com/keras-team/keras-tuner

[17] Sordello, M., He, H., & Su, W. (2019). Robust learning rate selection for stochastic optimization via splitting diagnostic. DOI: https://doi.org/10.48550/arXiv.1910.08597

[18] Gavrikov, P. (2020). VisualKeras. GitHub. Available online https://github.com/paulgavrikov/visualkeras