SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I11P107

Volume 12 Issue 11, 75-85, November 2025
© 2025 Seventh Sense Research Group®

Original Article

A Robust Approach for Hair Contaminant Detection in Papadam Using Transfer Learning

Sarika Panwar[1] , Milind Gajare[2,] Monali Chaudhari[3] , Shravani Doke[4]

[1]*Department of Electronics and Telecommunication Engineering, MIT Academy of Engineering, Pune, Maharashtra, India.*
[2]*Department of Electronics and Telecommunication Engineering, AISSMS, Institute of Information Technology, Pune, Maharashtra, India.*
[3]*Electronics and Telecommunication Engineering, Vivekanand Education Society's Institute of Technology, Maharashtra, India.*
[4]*Analyst, Capgemini Technology Services India Ltd., Bengaluru, India.*

[2]*Corresponding Author : milind.gajare@aissmsioit.org*

Received: 05 September 2025 Revised: 07 October 2025 Accepted: 06 November 2025 Published: 28 November 2025

Abstract - *Hair contamination in food products is a significant problem that affects consumer confidence, safety, and quality. More sophisticated solutions are required because conventional hair detection methods frequently fail to identify hair particles. The goal of this study is to use deep learning and image processing techniques to create a reliable framework for accurately detecting hair in Papadam. Image masking is used to isolate areas of interest after a thorough preprocessing pipeline that includes greyscale conversion, Wiener filtering for noise reduction, Canny edge detection to highlight edges, and contour detection to extract borderline details. For classification, a pre-trained InceptionV3 transfer learning model is used; custom layers are optimised especially for hair detection, and the initial layers are frozen to retain learnt features. Global Average Pooling 2D and dense layers with ReLU activation are included in the model. Validated using real-world datasets, the suggested method achieved a 97.18% accuracy rate in identifying hair contaminants in Papadam. This study demonstrates how well deep learning and thorough preprocessing work together to improve quality, especially for automatic papadam hair detection.*

Keywords - *Hair Detection, Inception-V3, CNN, Transfer Learning.*

1. Introduction

The whole food product lifecycle, from production to consumption, is included in the food industry. A significant issue is the presence of foreign particles in food, including hair, glass, plastic, metal, and other contaminants. These contaminants result in costly recalls, significant health risks, legal ramifications, and damage to a brand's image [1, 2]. Given the health risks, allergen cross-contamination, and pathogen transmission potential, hair particles in particular represent a hazard to quality and safety. Current methods of quality analysis are able to identify contaminants such as stone, plastic, and metal, but they frequently miss hair particles. The proposed research primarily focuses on predicting hair strands in Papadam. Papadam is a snack that originated in India. It has a disc-like appearance. It is sold uncooked with proper packaging. These snacks are consumed as part of meals in India, especially in Gujarat and Rajasthan. A hair strand in Papadam is considered a contamination. It occurs before packaging during the production process. To improve quality control and maintain hygienic standards in food manufacturing, it is essential to predict the presence of hair in Papadam using techniques such as Convolutional Neural Networks (CNN) and computer vision [3]. Although several studies have acknowledged and researched the detection of contaminants such as metals, glass, and plastics in food, the detection of small macroscopic contaminants like hair strands remains an enigmatic task. Detection of hair strands is extremely onerous due to their thin structure, background camouflaging, and irregular shapes.

In small-scale papadam industries, still, conventional manual packaging is used. The upshot of it is that there are hygiene concerns and potential health risks. Hence, an automated and reliable detection method is needed. Despite advancements in computer vision and deep learning for food safety, the studies that have focused on the detection of hair contaminants in food are very few. Most of the existing systems target rigid or reflective materials, limiting their efficacy for detecting soft and thin contaminants like hair. This research meticulously bridges the gap by developing a deep learning-based framework using InceptionV3 with advanced preprocessing for the accurate identification of hair strands in Papadam, thereby leading to improvements in food quality and safety standards. Many researchers have focused on the detection and prevention of foreign material in food due to the substantial health risks, brand reputation, and legal responsibilities. Geueke et al. (2018 proposed the chemical safety aspects of commonly used materials in food packaging within a circular economy framework, highlighting the need for safety evaluations [4]. With an emphasis on their potential to improve food safety and cut waste, Chen et al. created colorimetric sensors and smartphone-based platforms for tracking food quality and identifying spoilage. [5].

With better accuracy than traditional X-ray methods, Einarsdottir et al. concentrated on advanced X-ray imaging techniques, especially grating-based multimodal approaches, to improve the detection of foreign objects in food products. [6]. In order to improve the classification accuracy of food images with different textures and colours, Subhi et al. investigated a CNN [7]. Improved food quality control has been made possible by contemporary image processing tools. Pan et al. proposed a contour detection technique based on image saliency.

It aided in identifying contaminants' asymmetrical forms and contours. It improved food quality check detection capabilities [8]. A microscopic image filtering and noise reduction technique was proposed by Devi et al. It improved the clarity and accuracy of microscopic image analysis in food quality control [9]. Pu et al. proposed a novel edge detector based on a transformer. It helped to extract clear and crisp boundaries and also the accuracy of object boundary detection [10]. These developments in image processing technologies ensured their use in food quality assurance, offering more precise and effective ways to detect impurities in food products.

Many researchers successfully demonstrated the use of machine learning techniques such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, Random Forests, and Gradient Boosting algorithms in detecting and identifying contaminants in food products. Zhang et al. proposed an SVM-based foreign body recognition method for food images [11]. Goel et al. have proclaimed the role of artificial intelligence in detecting food adulteration [12]. Hansen et al. explored the Principal Component Analysis (PCA) and supervised classification techniques to discriminate milk samples based on physicochemical properties. [13]. The potential of the K-Nearest Neighbours (KNN) algorithm to enhance quality control and inventory management in the agricultural industry has been supported by Sudipa et al.'s demonstration of the algorithm's efficacy in agricultural applications, specifically for fruit classification and quality assessment [14].

Inglis et al. addressed how machine learning is increasingly being used to identify mycotoxins in food, providing a quicker and more scalable option than conventional laboratory techniques. Although convolutional neural networks are the most widely used architecture for mycotoxin detection, the study points out that reproducibility issues arise because many studies lack open-source code and comprehensive hyperparameter reporting [15]. Soltani et al. emphasised how machine vision and image processing work together to improve agricultural product defect detection, tackling problems like decay and insect damage. This method increases the effectiveness of defect detection and quality inspection by taking pictures of products in ideal conditions and using classification algorithms, which benefits the global agriculture sector [16]. A comprehensive review of 98 studies on the use of computer vision in agriculture, with a special emphasis on fruits and vegetables, was carried out by Tripathi et al. It draws attention to the gaps in the literature on mathematical frameworks and defect detection, suggests a generalised framework for quality grading, and comes to the conclusion that Support Vector Machine (SVM) performs better in classification across a range of datasets [17].

To determine the ripeness stages of tomatoes and bell peppers, Elhariri et al. proposed a classification system that makes use of colour features, Principal Component Analysis, and machine learning algorithms like Support Vector Machine and Random Forest. Based on actual farm photos, the experimental results demonstrated that SVM with a linear kernel performed better than RF in terms of ripeness detection classification accuracy [17]. For the purpose of nondestructive classification and tea adulteration detection, Zou et al. integrated Fluorescence Hyperspectral Imaging (FHSI) with machine learning models such as Random Forest and CatBoost. The accuracy and performance of these approaches have been greatly enhanced by preprocessing techniques like ensemble learning models and Median Filtering (MF) [19].

The literature review emphasizes important developments in the use of machine learning algorithms for food foreign particle detection, which is essential for guaranteeing food safety and quality. The detection capabilities of Convolutional Neural Networks (CNNs) and other image processing methods have been improved. K-Nearest Neighbours (KNN) and Support Vector Machines (SVM) have proven successful in identifying contaminants and adulterants in food items. The potential of these techniques to enhance food quality control is demonstrated by the effectiveness of decision trees, ensemble approaches like Random Forests, and gradient boosting algorithms like XGBoost and LightGBM in identifying contaminants in a variety of food items.

The novelty of this study comes from using transfer learning (InceptionV3) to detect microscopic contaminants in Papadam, a traditional Indian food. Previous studies mainly looked at defect detection in fruits or packaged goods. In contrast, this research specifically targets hair contamination by analyzing high-resolution images taken in actual production settings. The rest of this paper is structured as follows. The materials and methods are examined in Section 2, which also provides a thorough description of the dataset collection, the intricate preprocessing procedures used to prepare the images, and the model architecture. Experimentation results are presented in Section 3. Section 4 contains the discussion and suggests potential avenues for future research.

2. Materials and Methods

Figure 1 shows the block diagram for hair detection in the papadam industry. It begins with dataset collection, followed by a data pre-processing pipeline. After data pre-processing, the dataset is divided into a training and a testing set. CNN and Inception V3 (transfer learning model) were both trained on the available dataset. The system output is an image classified into two categories: with hair and without hair.

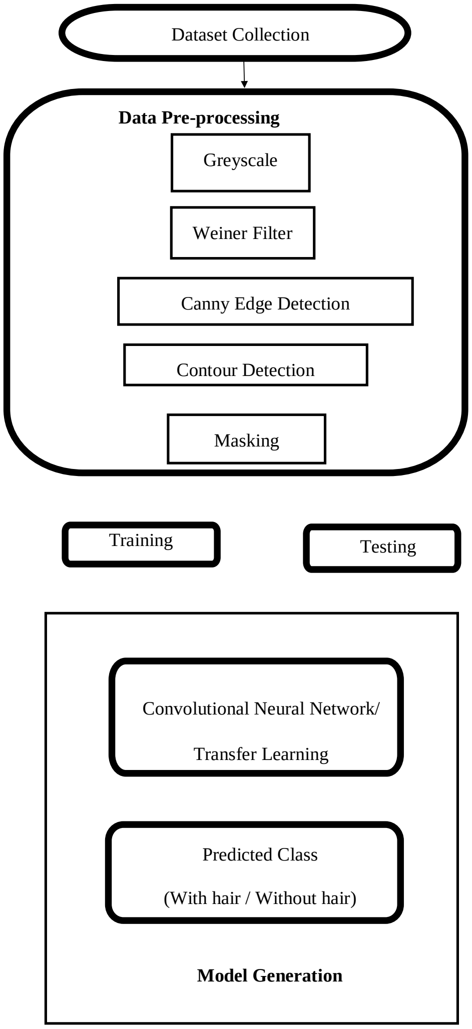

Fig. 1. System architecture

2.1. Data Collection

Papadam images were acquired from a local manufacturer in Pune, Maharashtra. A total of 1400 papadam images were acquired and collected for this study. The total images were evenly split into two classes: 700 images with hair and 700 images without hair. The images are acquired using a smartphone camera featuring a high-resolution main sensor of 64 MP, with an aperture of f/1.79 and a focal length of 4.73 mm. The image resolution is 6944 x 9248 pixels. The dataset was split into 75% for training and 25% for testing, with 1,050 images used for training and the remaining images for testing.. Even though the dataset offers a balanced view of both classes, it is limited because of the controlled environment and a single source of data collection. Images were taken under consistent indoor lighting with a white background to reduce shadows. Each image was captured from about 30 cm above the Papadam at a 90° angle. The papadams came from one local manufacturer, so texture and thickness variations might be limited. While the dataset offers a balanced view of both classes, it is constrained by the controlled setting and a single source of data collection. Although this study did not use data augmentation techniques like rotation, flipping, and contrast adjustment, adding them in future research could help improve model performance across different lighting and texture situations. Future research could also include images from various manufacturers, different lighting conditions, and various types of contamination, such as thread, dust, and insects, to enhance generalization.

2.2. Papadam Image Pre-Processing

Enhancement and isolation of the hair strands are part of the pre-processing steps for papadam images. Greyscale conversion, Canny edge recognition, contour detection, Wiener filtering for noise reduction, and masking to separate regions of interest are all part of the pre-processing pipeline. The purpose of this is to draw attention to Papadam's hair strand. First, the original papadam image is converted to grayscale to reduce it to a single channel, retaining essential structural and textural information using Equation (1).

$$I_{gray}\,(x,y) = 0.299 \times I_R(x,y) + 0.587 \times I_G(x,y) \; + \; 0.114 \times I_B \qquad (1)$$

Noise reduction is done using a Wiener filter. It improved overall image quality [20]. The Wiener filter with adaptive noise filtering capabilities based on local statistics is depicted by Equation (2).

$$I_{filtered}(x,y) = \mu(x,y) + \frac{\sigma^2(x,y)-\sigma_n^2}{\sigma^2(x,y)}\left(I_{gray}(x,y) - \mu(x,y)\right) \qquad (2)$$

Where:

$I_{filtered}(x,y)$ Is the filtered pixel value at position (x, y).

$I_{gray}(x,y)$ Is the grayscale pixel value at position? (x,y).

$\mu(x,y)$ Is the local mean around the pixel (x,y).

$\sigma^2(x,y)$ Is the local variance around the pixel (x,y).

σ_n^2 is the noise variance.

The local mean and variance are calculated over a window centered around each pixel (Equations (3) and (4)):

$$\mu(x,y) = \frac{1}{N}\sum_{i=-k}^{k}\sum_{j=-k}^{k} I_{gray}(x+i,y+j) \qquad (3)$$

$$\mu(x,y) = \frac{1}{N}\sum_{i=-k}^{k}\sum_{j=-k}^{k}(I_{gray}(x+i,y+j) - \mu(x,y))^2 \qquad (4)$$

Subsequently, the Canny edge detection algorithm [21] is used to identify potential hair edges within the pre-processed image. The gradient of the image intensity is calculated to find the edges using Equation (5).

$$G = G_x^2 + G_y^2 \qquad (5)$$

Where G_x and G_y are the gradients in the x and y directions, detected edges are subjected to contour detection, where contours that surpass a predetermined length threshold are retained [22]. This step helped remove small, unwanted details, allowing the hair strands to be highlighted. A binary mask is created using the filtered contours that shows continuous dotted lines, most likely depicting hair, along with the outline of the Papadam [23]. Binary mask used to apply a bitwise and operation to the original image, which effectively highlighted the hair by isolating the areas that match the continuous lines (Equation (6)).

$$I_{masked}\,(x,y) = I(x,y) \times M(x,y) \qquad (6)$$

Where M (x, y) is the mask value (0 or 1) at pixel (x, y).

For further analysis, the output image is resized to a desired scale. The dimensions are scaled down by 50% by using Equations (7) and (8).

$$width = \frac{(original\,width \times 50)}{100} \qquad (7)$$

$$height = \frac{(original\,height \times 50)}{100} \qquad (8)$$

By carefully adjusting brightness and contrast and utilising noise reduction methods like the Wiener filter, the suggested preprocessing pipeline maximises the clarity and definition of hair strands. The pre-processing pipeline guarantees that the analysis that follows can concentrate on precisely detecting and evaluating hair particles with greater precision and dependability.

2.3. Model Training

The classification of papadam images to identify hair strands is investigated using CNN and Inception-V3 (transfer learning model).

2.3.1. CNN Model

CNN is extremely effective at image classification tasks because it can automatically extract complicated features from picture input. CNN's hierarchical structures and convolutional layers enable it to capture intricate patterns and textures in photos more effectively than traditional machine learning algorithms. CNN performs exceptionally well by directly learning pertinent features from images [24, 25]. CNN is a model for deep learning. Using a variety of building blocks, including convolution layers, pooling layers, and fully connected layers, it is intended to automatically and adaptively learn spatial hierarchies of features by backpropagation [26]. CNN architecture is shown in Figure 2.

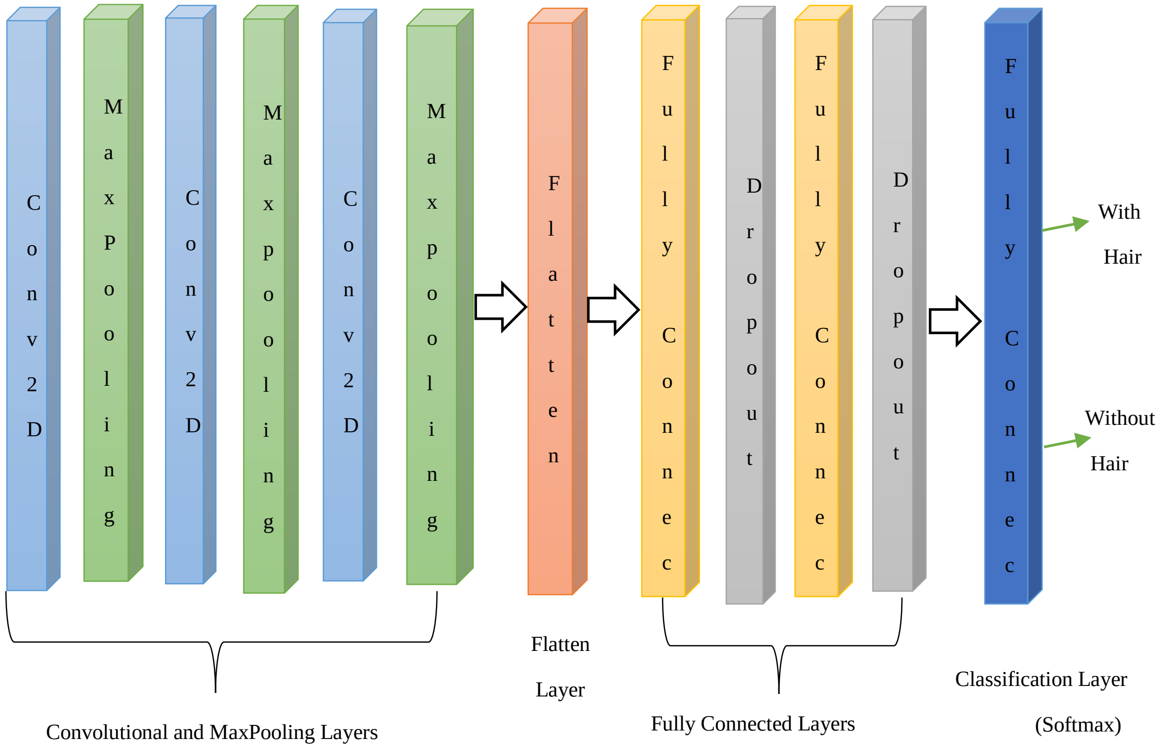

Fig. 2 CNN architecture

Each of the three convolutional layers in the proposed CNN architecture utilizes Rectified Linear Unit (ReLU) activation functions to enhance non-linearity. The filters used by the convolutional layers are (32, 3x3), (64, 3x3), and (128, 3x3), in that order. This arrangement is designed to efficiently identify intricate spatial patterns in the input papadam photos. Each convolutional layer is followed by max-pooling layers with a pool size of (2x2) to minimise overfitting and effectively downsample the feature maps. The combination of pooling and convolutional layers ensures robust feature extraction from the input images. The design consists of two dense (fully connected) layers of 512 and 256 units, respectively, following the convolutional and pooling layers. To enable precise final predictions, these dense layers combine the acquired information.

To improve model generalisation and avoid overfitting, dropout layers with a dropout rate of 0.5 are purposefully introduced after each dense layer. The model can efficiently generate class probabilities for binary classification problems thanks to the final dense layer's usage of a softmax activation function. The design and salient features of the suggested CNN model for hair detection are compiled in Table 1. The optimiser and loss function are carefully chosen during model compilation and evaluation. Due to its effectiveness in deep neural network training, the Adam optimizer was selected. The model's performance is enhanced by utilizing the categorical cross-entropy loss function, which accurately measures the discrepancy between the actual and predicted class distributions.

Table 1. CNN model architecture summary

Layer Type	Layer Details
Convolution Layer 1	Filters: 32, Size: 3x3, Activation: ReLU
Max-Pooling Layer 1	Pool Size: 2x2
Convolutional Layer 2	Filters: 64, Size: 3x3, Activation: ReLU
Max-Pooling Layer 2	Pool Size: 2x2
Convolutional Layer 3	Filters: 128, Size: 3x3, Activation: ReLU
Max-Pooling Layer 3	Pool Size: 2x2
Dense Layer 1	Units: 512, Activation: ReLU
Dropout Layer 1	Dropout Rate: 0.5
Dense Layer 2	Units: 256, Activation: ReLU
Dropout Layer 2	Dropout Rate: 0.5
Output Layer	Units: Number of classes (Binary Classification), Activation: Softmax
Total Trainable Parameters	Approximately 13,070,658 (49.86 MB)

The main evaluation indicator, accuracy, provides a thorough assessment of the model's performance. This architecture contains approximately 13,070,658 trainable parameters, equivalent to roughly 49.86 MB.

2.3.2. Inception V3 Model

Google designs the InceptionV3 model. It is employed for its ability to achieve state-of-the-art results in image classification tasks [27]. It takes a pre-trained model into consideration, which has been previously trained on a large dataset, such asmageNet, and fine-tunes it for a specific task using a smaller dataset [28]. The already learned features of the pre-trained model significantly improve the performance and efficiency of training on new tasks. Figure 3 shows the architecture of the Inception V3 model.

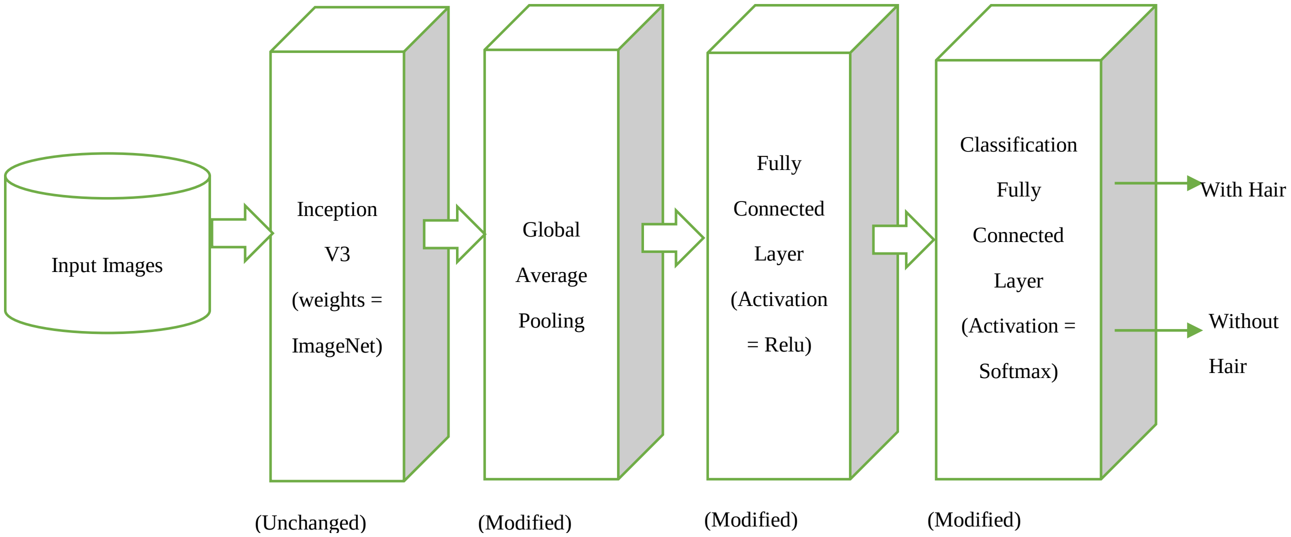
Input Images
Inception V3 (weights = ImageNet)
Global Average Pooling
Fully Connected Layer (Activation = Relu)
Classification Fully Connected Layer (Activation = Softmax)
With Hair
Without Hair
(Unchanged)
(Modified)
(Modified)
(Modified)

Fig. 3 Inception V3 architecture

In the Inception V3 architecture, the pre-trained layers are frozen to preserve their learned weights and prevent them from being modified during training. This approach confirms that training updates are only applied to the custom (modified) layers, which are designed explicitly for binary classification. These custom layers include a GlobalAveragePooling2D layer, which reduces the spatial dimensions of the feature maps, followed by a fully connected dense layer with ReLU activation for further processing of the extracted features. The final layer is a fully connected dense layer with sigmoid activation. The binary cross-entropy loss function and Adam optimizer are used to compile the final model. Total parameter count of a model is 22,065,185 (or 84.17 MB), but only 262,401 of these parameters are trainable, and they belong to the custom layers. The remaining 21,802,784 parameters corresponding to the weights of the frozen layers are kept non-trainable. This modification ensures that the model leverages the pre-existing knowledge embedded in the InceptionV3 architecture while adjusting it for the binary classification of papadam images.

Table 2. Inception V3 model architecture summary

Components	Details
Dense Layer 1	Units: Variable, Activation: ReLU
Dense Layer 2	Units: 1, Activation: Sigmoid (for binary classification)
Trainable Parameters	262,401 (in custom layers)
Non-Trainable Parameters	21,802,784 (in frozen InceptionV3 layers)
Total Parameters	22,065,185 (84.17 MB)

By freezing the base layers of InceptionV3 and fine-tuning only the added layers, the model benefits from the robust feature extraction capabilities of the pre-trained network, resulting in improved performance and efficiency in detecting hair strands in papadam images. Table 2 summarizes the architecture of the Inception V3 model.

3. Results

The proposed system is implemented using Python 3.9.13. An 11th Gen Intel Core i5-1135G7 CPU running at 2.40 GHz and 8GB of RAM make up the execution environment. OpenCV (cv2) version 4.8.0, NumPy version 1.23.5, Pillow (PIL) version 9.2.0, TensorFlow version 2.12.0 with Keras (tensorflow.keras) version 2.12.0, and Matplotlib version 3.5.2 are the libraries used. The papadam image pre-processing significantly enhanced the image classification model’s ability to distinguish between pictures with and without microscopic hair particles. Several critical steps in the pre-processing pipeline enhanced the accuracy of the model.

These pre-processing procedures produced increasingly sharper and more focused images at each stage, providing the model’s categorization tasks with a strong basis. The original RGB image, which functions as the initial input image comprising possible hair contamination, is displayed in Figure 4(a). This picture was taken without any preprocessing and in its original colour format. The brightness of the image is changed in Figure 4(b) to increase the visibility of hair particles and make them more distinct from the backdrop. The greyscale conversion, which reduces computing complexity while maintaining crucial properties required for hair detection, is shown in Figure 4(c). By eliminating colour information while keeping crucial details, this phase streamlines the visual data. The Wiener filter's effect on noise reduction is seen in Figure 4(d). This stage produces a crisper representation of the image and increases the visibility of hair particles by reducing noise and improving image clarity. The masked image obtained by using Canny edge detection, contour detection, and a bitwise AND operation with the original image is displayed in Figure 4(e). By eliminating unnecessary information and highlighting the borders, this stage isolates particular areas of interest and highlights the hair particles. Finally, Figure 4(f) presents the smoothed masked image depicting hair. This final step further refines the masked image to produce a smoother representation of the hair particles, enhancing their visibility and making them easier to detect.

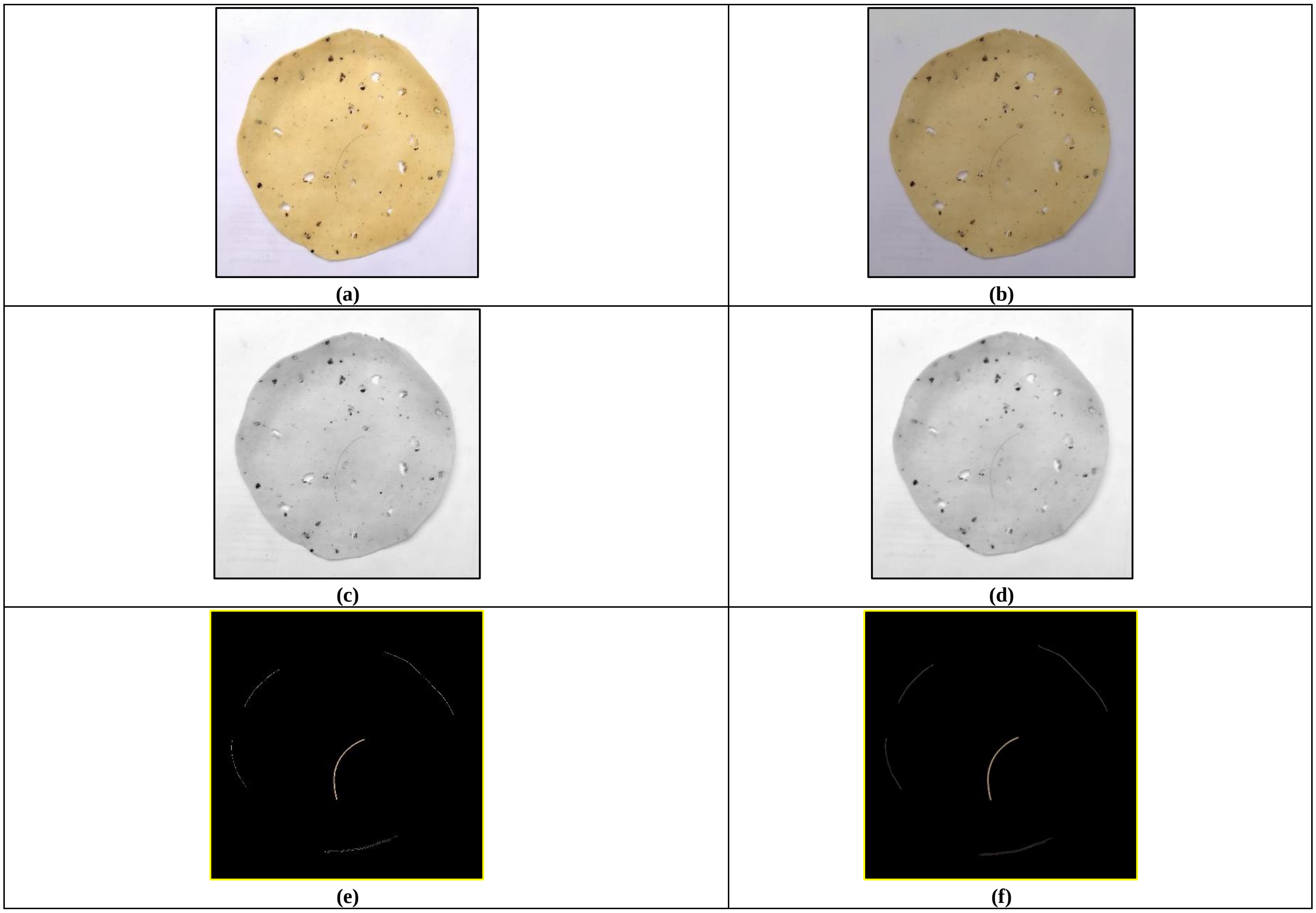
(a) (b) (c) (d) (e) (f)

Fig. 4 Inception V3 Architecture: (a) Original image, (b) Brightness adjusted image, (c)Grayscale converted image, (d) Noise reduction using weiner filter, (e) Masked image (after applying canny edge, contour detection, and bitwise and with original image), and (f) Smoothened masked image (depicting hair).

Figure 5(a) displays a pre-processed image that contains hair particles, and Figure 5(b) presents a pre-processed image that does not contain hair particles. All images have undergone the complete preprocessing pipeline.

Further, the model training was conducted using the pre-processed dataset. The dataset was split into 75% for training and 25% for testing, with 1,050 images used for training and the remaining images for testing.

The custom CNN achieved an accuracy of 87.71%, with notable precision and recall metrics. Strong detection performance was demonstrated by its excellent precision (0.81) and recall (0.99) for photos with hair particles ('with hair'). Nevertheless, it showed reduced recall (0.76) and precision (0.99) for images "without hair."

With an accuracy of 97.18%, the refined InceptionV3 model performed quite well. For photos containing hair particles, it showed excellent precision (0.95) and recall (0.99), and for images without hair, it showed similarly good precision (0.99) and recall (0.95). The evaluation results for the custom CNN and InceptionV3 models are shown in Table 3.

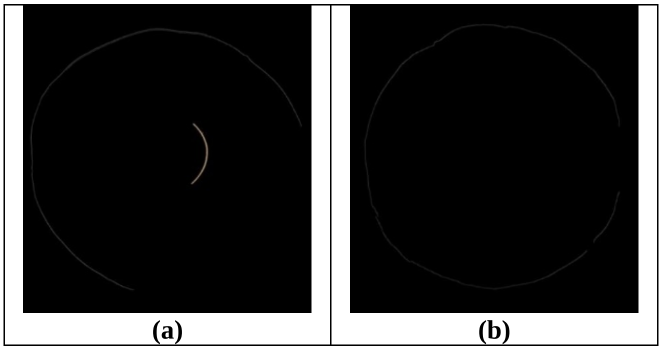

Fig. 5 Comparison of pre-processed Images with and without air Particles (a) Pre-processed image with hair particles, and (b) Pre-processed image without hair particles.

Table 3. Evaluation results for custom CNN and INCEPTIONV3 models

Model	Class	Precision	Recall	F1- Score
CNN	With hair	0.81	0.99	0.89
	Without hair	0.99	0.76	0.86
	Accuracy	87.71%		
InceptionV3	With hair	0.99	0.95	0.97
	Without hair	0.95	0.99	0.97
	Accuracy	97.187%		

Fig. 6 Confusion matrix the inception V3 model outperforms the CNN

Model	TP	TN	FP	FN
CNN	693	538	162	7
InceptionV3	665	693	7	35

The confusion matrix indicates that the Inception V3 model outperforms the CNN model, resulting in fewer false positives and false negatives. InceptionV3 achieves an overall accuracy of 97.18%, which means it reliably classifies both contaminated and uncontaminated papadam samples. On the other hand, the CNN model exhibits higher misclassification rates, particularly for clean samples, resulting in an overall accuracy of 87.71%. The improvement was steady across several runs. This suggests that the difference is statistically significant ($p < 0.05$). Misclassifications mainly happened in cases involving thin or semi-transparent hair strands. The findings from the Inception V3 model are examined to provide a better understanding of its design, following a comparison with the CNN and Inception V3 models. Basic convolutional layers are the first part of the model. To capture basic visual elements, such as edges and textures, it incorporates batch normalization and initial convolution procedures. These features are subsequently processed using a sequence of Inception modules, each of which has concurrent convolutions, configurable kernel sizes, and pooling operations to handle different scales and complexities. Deeper within the network, convolutional layers filter incoming input multiple times to capture complex patterns and abstract structures. These layers combine and process low-level data from earlier levels, allowing the model to identify intricate patterns and details. The high-level features at the conv_9 layer of the Inception V3 model are shown in Figure 7.

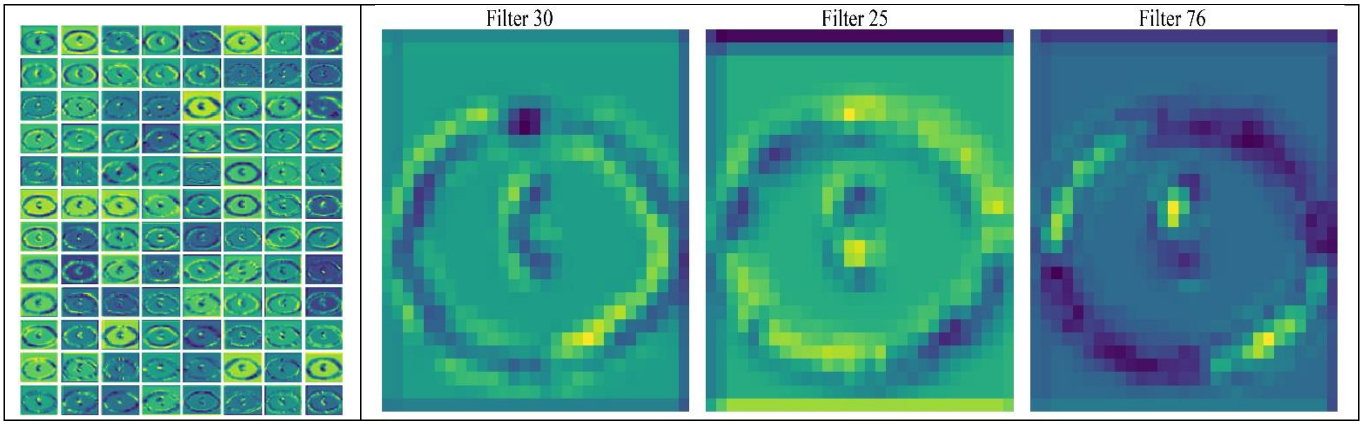

Fig. 7 Visuals of the feature map and high-level features learned at the conv_9 layer

Further, the spatial dimensions of the feature maps are converted into a single vector using a Global Average Pooling layer. Dense layers refine this vector further, and the final dense layer places images into the designated categories using a sigmoid activation function. Effective automation of feature extraction and classification is possible with this architecture. Table 4 depicts the performance of the Inception V3 model for different epoch settings, with a batch size of 32. As the number of epochs grows, the model exhibits improved performance, as evidenced by higher test accuracy and lower test loss. The Inception V3 model effectively classifies papadam images into 'with' or 'without' hair particles. Figure 8 illustrates the confidence values associated with the Inception V3 model's classification for both categories. Figure 9 presents the model's training and testing loss values, along with its accuracy. A notable aspect of this graph is the close alignment between the training and testing accuracies throughout the training process. This alignment indicates that the Inception V3 model generalizes well to unseen data, and it is neither overfitting nor underfitting.

Table 4. Impact of epoch variation on inception V3 model performance for hair detection

Epochs	Batch size	Mini-Batch Accuracy	Mini-Batch Loss	Test Accuracy	Test Loss
100	32	94.30	0.1361	97.1875	0.0833245
50	32	93.46	1.527	96.8750	0.0902371
10	32	90.77	2.393	96.2499	0.1393666

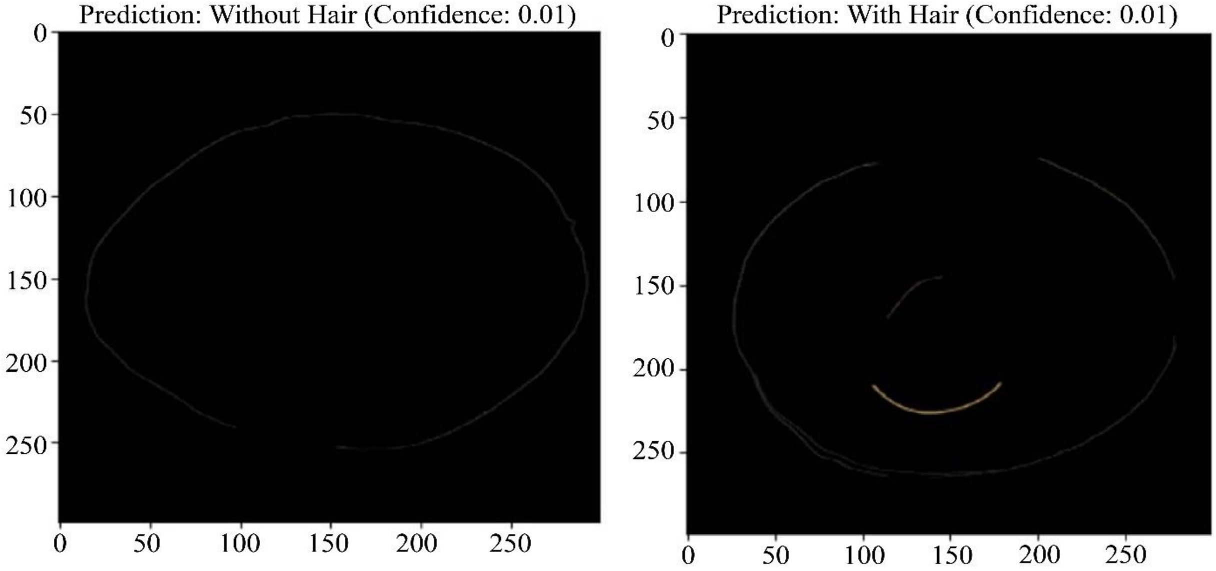
Prediction: Without Hair (Confidence: 0.01)
Prediction: With Hair (Confidence: 0.01)
0
50
100
150
200
250

Fig. 8 Images classification into two classes (without hair and with hair)

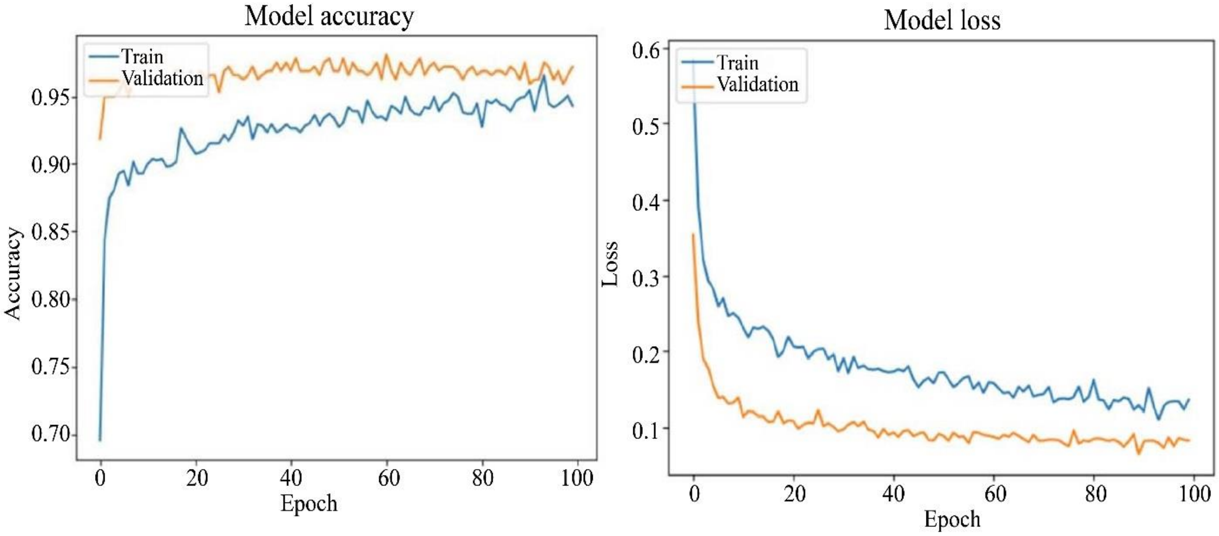

Fig. 9 Training and testing loss values and accuracy for inception V3 model

4. Discussion

Hair contamination in Papadam during production is a significant challenge. It impacts customer trust and the brand name. The proposed approach combines robust pre-processing with the Inception V3 model for classification. This integrated approach offers a streamlined solution to the issue of hair contamination in Papadam. The pre-processing pipeline includes gray scale conversion of an image, the Wiener filter, the Canny edge detector, and image masking.

This pre-processing pipeline is designed to enhance the papadam image and facilitate accurate classification (with and without hair). Inception V3 model parameters are fine-tuned for the task of papadam image classification. The model demonstrated an accuracy of 97.18% on the test data set. This highlights the effectiveness and efficiency of the papadam hair detection model in accurately detecting hair contamination despite the challenges posed by various image conditions, such as changing lighting and background noise. These results also highlight the robustness of the transfer learning model and how reliable a pre-trained model is when used on a large dataset. The inception V3 model demonstrated balanced performance in both classes. Balanced performance is required for maintaining high standards in food safety and quality control in the papadam industry. Table 3 presents a direct comparison of the accuracies achieved by the CNN and Inception V3 models, demonstrating the superior performance of the Inception V3 model. The better performance of InceptionV3 comes from its deeper structure and effective feature extraction using inception modules. This allows it to distinguish fine edges and textures better. Its ability to generalize from limited data helped reduce overfitting compared to the CNN model, leading to improved classification accuracy.

5. Conclusion

The proposed system presented a significant advancement in the application of deep learning and image processing for hair contaminant detection in the papadam industry. It provides a robust, reliable, and efficient solution for maintaining high standards. Future research can examine wider implementation of this methodology in other food industries where detecting hair contaminant is crucial, such as bakery product Industries, to ensure product quality and prevent consumer complaints.

Acknowledgments

The authors would like to express their sincere gratitude to Sona Papad, a leading papadam manufacturing industry located at 8, 144, near Hari Hareshwar temple or TV's Shelar showroom, Laxmi Nagar, Parvati Paytha, Pune, Maharashtra 411009, for their invaluable support in providing real-time images of Papadam. This contribution was essential for the successful completion of this research. We are also thankful for their willingness to collaborate and assist in ensuring the accuracy and relevance of the study.

References

[1] Keila Payne et al., "Detection and Prevention of Foreign Material in Food: A Review," *Heliyon*, vol. 9, no. 9, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Darwin G. Caldwell, *Automation in Food Manufacturing and Processing*, Springer Handbook of Automation, Springer, Cham, pp. 949-971, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] K.K. Patel et al., "Image Processing Tools and Techniques Used in Computer Vision for Quality Assessment of Food Products: A Review," *International Journal of Food Quality and Safety*, vol. 1, pp. 1-16, 2015. [Google Scholar]

[4] Birgit Geueke, Ksenia Groh, and Jane Muncke, "Food Packaging in the Circular Economy: Overview of Chemical Safety Aspects for Commonly Used Materials," *Journal of Cleaner Production*, vol. 193, pp. 491-505, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Yu Chen et al., "Low Cost Smart Phone Diagnostics for Food using Paper-Based Colorimetric Sensor Arrays," *Food Control*, vol. 82, pp. 227-232, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Hildur Einarsdóttir et al., "Novelty Detection of Foreign Objects in Food using Multi-Modal X-Ray Imaging," *Food Control*, vol. 67, pp. 39-47, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mohammed A. Subhi, and Sawal Md. Ali, "A Deep Convolutional Neural Network for Food Detection and Recognition," *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, Sarawak, Malaysia, pp. 284-287, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yongcai Pan et al., "A Contour Detection Method Based on Image Saliency," *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, Liuzhou, China, pp. 42-46, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Tulasi Gayatri Devi, and Nagamma Patil, "Analysis &Evaluation of Image Filtering Noise Reduction Technique for Microscopic Images," *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*, Kottayam, India, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Mengyang Pu et al., "EDTER: Edge Detection with Transformer," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 1392-1402, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yifan Zhang et al., "Recognition of Foreign Objects in Food Images using Support Vector Machine," *International Conference on Frontier Computing*, Singapore, Singapore, pp. 667-674, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Kashish Goyal, Parteek Kumar, and Karun Verma, “Food Adulteration Detection using Artificial Intelligence: A Systematic Review,” *Archives of Computational Methods in Engineering*, vol. 29, no. 1, pp. 397-426, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Lucas Hansen, and Marco Flôres Ferrão, “Identification of Possible Milk Adulteration Using Physicochemical Data and Multivariate Analysis,” *Food Analytical Methods*, vol. 11, no. 7, pp. 1994-2003, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] I. Gede Iwan Sudipa et al., “Leveraging K-Nearest Neighbors for Enhanced Fruit Classification and Quality Assessment,” *Indonesian Journal of Data and Science*, vol. 5, no. 1, pp. 30-36, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Alan Inglis et al., “Machine Learning Applied to the Detection of Mycotoxin in Food: A Systematic Review,” *Toxins*, vol. 16, no. 6, pp. 1-30, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mahmoud Soltani Firouz, and Hamed Sardari, “Defect Detection in Fruit and Vegetables by using Machine Vision Systems and Image Processing,” *Food Engineering Reviews*, vol. 14, no. 3, pp. 353-379, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mukesh Kumar Tripathi, and Dhananjay D. Maktedar, “A Role of Computer Vision in Fruits and Vegetables among Various Horticulture Products of Agriculture Fields: A Survey,” *Information Processing in Agriculture*, vol. 7, no. 2, pp. 183-203, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Esraa Elhariri et al., “Random Forests Based Classification for Crops Ripeness Stages,” *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*, pp. 205-215, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[19] Zhiyong Zou et al., “Classification and Adulteration of Mengding Mountain Green Tea Varieties based on Fluorescence Hyperspectral Image Method,” *Journal of Food Composition and Analysis*, vol. 117, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Norbert Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: with Engineering Applications*, MIT Press, 1949. [CrossRef] [Google Scholar] [Publisher Link]

[21] John Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, 1986. [CrossRef] [Google Scholar] [Publisher Link]

[22] Satoshi Suzuki, and KeiichiA be, “Topological Structural Analysis of Digitized Binary Images by Border Following,” *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32-46, 1985. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rafael C. Gonzalez, and Richard E. Woods, *Digital Image Processing*, 4th ed., Pearson Education, 2019. [Google Scholar] [Publisher Link]

[24] Yann LeCun et al., “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[26] Jeff Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning,” *Genetic Programming and Evolvable Machines*, vol. 19, no. 1, pp. 305-307, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[27] Christian Szegedy et al., “Rethinking the Inception Architecture for Computer Vision,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2818-2826, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[28] Sinno Jialin Pan, and Qiang Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010. [CrossRef] [Google Scholar] [Publisher Link]