FOREX Publication
Open Access | Rapid and quality publishing

International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

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

Sarika Panwar[1*], Shravani Doke[2], Prathamesh Mankar[3], Yash Sakat[4] and Shruti Sancheti[5]
[1]*Assistant Professor in E&TC Department; vrushalimendre@gmail.com*
[2]*Student in Department of Artificial Intelligence and Data Science; shravanidoke29@gmail.com*
[3]*Student in Department of Artificial Intelligence and Data Science; prathamesh.0550@gmail.com*
[4]*Student in Department of Artificial Intelligence and Data Science; yashsakat076@gmail.com*
[5]*Student in Department of Artificial Intelligence and Data Science; shrutisancheti539@gmail.com*

***Correspondence:** *Sarika Panwar; vrushalimendre@gmail.com*

ABSTRACT- Hair contamination in food products is a serious challenge that impacts quality, safety, and consumer trust. Conventional hair detection techniques often fall short in effectively identifying hair particles, demanding more advanced solutions. This research aims to design and develop a robust framework to correctly detect hair in papadam using image processing and deep learning techniques. The approach starts with an extensive preprocessing pipeline, including grayscale conversion, Wiener filtering for noise reduction, Canny edge detection to highlight edges, and contour detection to extract borderline details, followed by image masking to isolate areas of interest. Pre-trained InceptionV3 transfer learning model is applied for classification, where the initial layers are frozen to preserve learned features, and custom layers are fine-tuned specifically for hair detection. The model incorporates GlobalAveragePooling2D and dense layers with ReLU activation. The proposed approach attained an accuracy of 97.18% in detecting hair contaminants in papadam, validated through real-world datasets. This research highlights the effectiveness of combining comprehensive preprocessing with deep learning to enhance quality, particularly for automatic hair detection in papadam.

Keywords: Hair detection; Inception-V3; CNN, transfer learning.

ARTICLE INFORMATION

Author(s): Sarika Panwar, Shravani Doke, Prathamesh Mankar, Yash Sakat and Shruti Sancheti;

Received: 28/07/2024; **Accepted**: 20/08/2024; **Published**: 15/09/2024;
e-ISSN: XXXX-XXXX;
Paper Id: IJCSR-030304;
Citation: 10.37391/IJCSR.030304

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

The food industry encompasses the entire food product lifecycle, from production to consumption. The presence of foreign particles in food, such as hair, glass, plastic, metal, and other pollutants, is a significant problem. These pollutants cause substantial health hazards, expensive recalls, legal liabilities, and harm to a brand's reputation [1-2]. Hair particles pose a hazard to quality and safety, presenting health risks and the potential for allergen cross-contamination and pathogen transmission. While current quality analysis techniques can detect contaminants like metal, plastic, and stone, they often fail to identify hair particles.

The proposed research primarily focuses on hair strand prediction in papadam. Papadam is a snack and it is 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 specially in Gujarat and Rajasthan. Hair strand in papadam is considered as a contamination. It occurs before packaging during the production process. To improve quality control and preserve hygienic standards in food manufacturing, it is important to predict the presence of hair in papadam using the techniques such as Convolutional Neural Networks (CNN) and computer vision [3].

Many researchers 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. in 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]. Chen et al. developed the colorimetric sensors and smartphone-based platforms for monitoring food quality and detecting spoilage, emphasizing their potential to enhance food safety and reduce waste.[5].

Einarsdottir et al. focused on advanced X-ray imaging techniques, particularly grating-based multimodal approaches, to enhance the detection of foreign objects in food products, offering improved accuracy over conventional X-ray methods.[6]. Subhi et al. explored the CNN to increase the accuracy of classification of food image, with varying textures and colors. [7].

International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

Modern tools in image processing technology have provided improved food quality control. A contour detection method using image saliency was proposed by Pan et al. It helped in delineating irregular shapes and contours of contaminants. It enhanced detection capabilities in food quality check [8]. 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 novel edge detector based on transformer. It helped to extract clear and crisp boundaries and the accuracy of object boundary detection [10]. These developments in image processing technologies ensured its 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 the SVM based foreign body recognition method for food image [11]. Goel et al. have proclaimed the role of artificial intelligence in detecting food adultration [12].

Hansen et al. explored the Principal Component Analysis (PCA) and supervised classification techniques to discriminate milk samples based on physicochemical properties. [13]. Sudipa et al. have demonstrated the effectiveness of the K-Nearest Neighbors (KNN) algorithm in agricultural applications, particularly for fruit classification and quality assessment, supporting its potential to improve quality control and inventory management in the sector [14].

Inglis et al. discussed the growing use of machine learning for detecting mycotoxins in food, offering a faster and more scalable alternative to traditional lab methods. While convolutional neural networks are the most employed architecture for mycotoxin detection, the paper highlights challenges in reproducibility due to a lack of detailed hyperparameter reporting and open-source code in many studies. [15]. Soltani et al. highlighted how the combination of machine vision and image processing enhances defect detection in agricultural products, addressing issues like insect damage and decay. By photographing products under optimal conditions and applying classification algorithms, this approach improves the efficiency of defect detection and quality inspection, benefiting the global agricultural industry [16]. Tripathi et al. conducted an extensive survey of 98 studies on the application of computer vision in agriculture, particularly focusing on fruits and vegetables. It highlights the gaps in existing research regarding mathematical frameworks and defect detection, proposes a generalized framework for quality grading, and concludes that Support Vector Machine (SVM) achieves superior classification accuracy across various datasets [17] . Elhariri et al. presented a classification system utilizing color features, Principal Component Analysis, and machine learning algorithms like Support Vector Machine and Random Forest to estimate ripeness stages of tomatoes and bell peppers. The experimental results, based on real farm images, showed that SVM with a linear kernel outperformed RF in terms of classification accuracy for ripeness detection.[18]. Zou et al. combined Fluorescence Hyperspectral Imaging (FHSI) with machine learning models like Random Forest and CatBoost for nondestructive classification and adulteration detection in teas. Preprocessing techniques such as Median Filtering (MF) and ensemble learning models have significantly improved the accuracy and performance of these methods. [19]

The literature survey highlights significant advancements using machine learning algorithms for detecting foreign particles in food, crucial for ensuring safety and quality. Convolutional Neural Networks (CNNs) and various image processing techniques have enhanced detection capabilities. Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) have been effective in classifying contaminants and adulterants in food products. Decision Trees and ensemble methods like Random Forests and Gradient Boosting algorithms, such as XGBoost and LightGBM, have proven effective in identifying contaminants in various food items, illustrating the potential of these techniques to improve food quality control.

The remainder of this paper is organized as follows. Section 2 explores materials and methods with in-depth description of the dataset collection, the detailed preprocessing steps undertaken to prepare the images, and the architecture of the models used. 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 training and testing set. CNN and Inception V3 (transfer learning model), both models are trained on the available dataset. The system output is an image classified into two categories: with hair and without hair.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

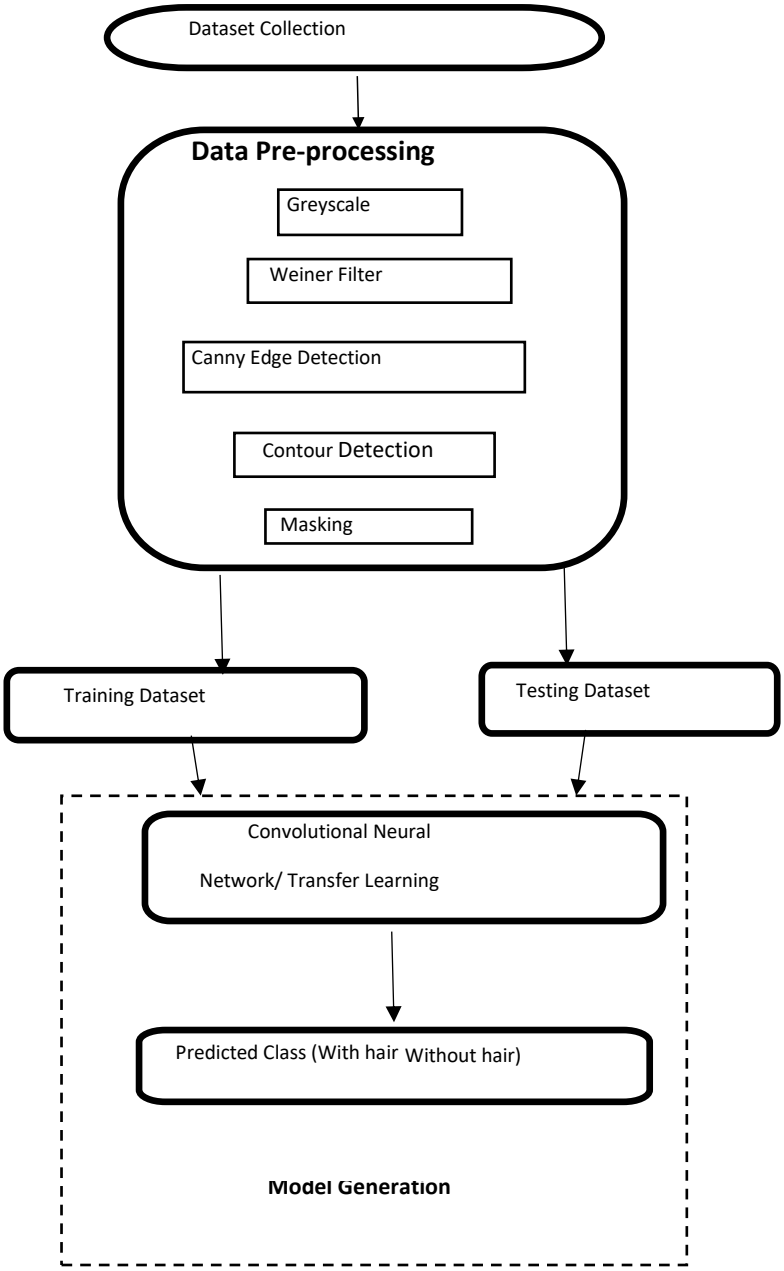

Figure 1. System Architecture

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.

2.2. Papadam Image Pre-processing

The papadam image pre-processing steps involves hair strand enhancement and isolation. The pre-processing pipeline includes grayscale conversion, contour detection, Canny edge recognition, Wiener filtering for noise reduction, and masking to isolate regions of interest. This aims to highlight the hair strand in papadam.

First, the original papadam image is converted to grayscale reduce it to a single channel, retaining essential structural and textural information using *equation 1*.

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

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

Noise reduction is done using 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*.

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

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 equation 4):*

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

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

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*.

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

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 in removing small, unwanted details so that hair strands, are 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).*

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

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 *equation 7* and *equation 8*.

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

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

The proposed preprocessing pipeline optimizes the clarity and definition of hair strands through careful adjustment of brightness and contrast, coupled with noise reduction techniques such as the Wiener filter. The pre-processing pipeline ensures that the subsequent analysis can focus specifically on identifying and analyzing hair particles with increased accuracy and reliability.

2.3. Model Training

CNN and Inception-V3 (transfer learning model), both are explored for the classification of papadam images to detect hair strands.

2.3.1. CNN Model

CNN has the capability of automatically extracting complex features from image data, making it highly effective for image classification tasks. CNN outperform traditional machine learning algorithms in capturing complex patterns and textures found in images due to their hierarchical structures and convolution layers. CNN excel by learning relevant features directly from image [24-25]. CNN is a deep learning model. It is designed to learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers automatically and adaptively [26]. *Figure 2* shows CNN architecture.

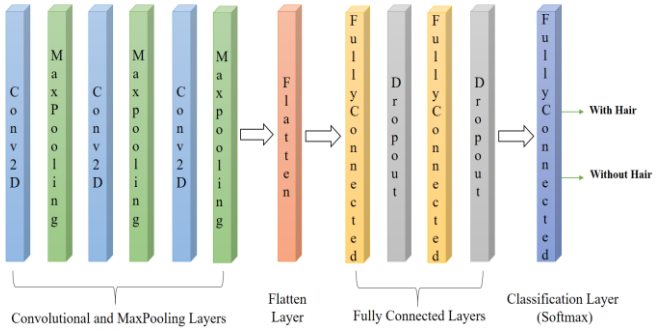

Figure 2. CNN Architecture

The proposed CNN architecture incorporates three convolutional layers, each using Rectified Linear Unit (ReLU) activation functions to enhance non-linearity. The convolutional layers utilize filters with sizes of (32, 3x3), (64, 3x3), and (128, 3x3) respectively. This configuration is designed to recognize complex spatial patterns in the input papadam images effectively. To reduce overfitting and efficiently down-sample the feature maps, max-pooling layers with a pool size of (2x2) are incorporated after each convolutional layer. The combination of convolutional and pooling layers ensures robust feature extraction from the input images. Following the convolutional and pooling layers, the architecture includes two dense (fully connected) layers with 512 and 256 units, respectively. These dense layers integrate the extracted features to facilitate accurate final predictions. Dropout layers with a dropout rate of 0.5 are strategically added after each dense layer to enhance model generalization and prevent overfitting. The final dense layer uses a softmax activation function, enabling the model to produce class probabilities for binary classification tasks effectively. *Table 1* summarizes the architecture and key details of the proposed CNN model for hair detection.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

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)

During model compilation and evaluation, meticulous attention is given to selecting the optimizer and loss function. The Adam optimizer is chosen for its efficiency in training deep neural networks. The categorical cross-entropy loss function is utilized as it reliably measures the discrepancy between actual and predicted class distributions, thus improving the model's performance. Accuracy is the primary evaluation metric, offering a comprehensive assessment of the model's performance. The total number of trainable parameters in this architecture is approximately 13,070,658, which corresponds to about 49.86 MB.

2.3.2. Inception V3 Model

The InceptionV3 model is designed by Google. 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 like ImageNet, and fine-tuning it for a specific task using a smaller dataset [28]. The already learned features of pre-trained model, significantly improve the performance and efficiency of training on new tasks. *Figure 3* shows the architecture of the Inception V3 model.

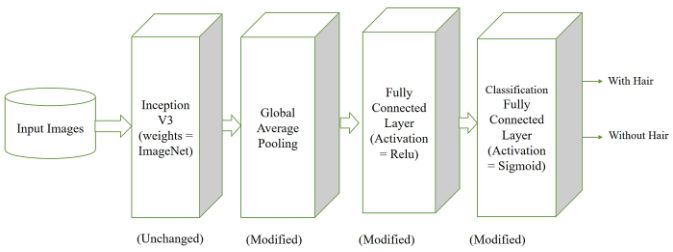

Figure 3. Inception V3 Architecture

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 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. 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 Inception V3 model.

Table 2. InceptionV3 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)

3. Results

The proposed system is implemented using Python 3.9.13. The execution environment includes 8GB of RAM and an 11th Gen Intel Core i5-1135G7 CPU @ 2.40 GHz. The libraries utilized are 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. The pre-processing of papadam images significantly enhanced the performance of the image classification model in discriminating between images containing tiny hair particles and those without. The pre-processing pipeline involved several crucial stages, each contributing to the model's overall accuracy. The progression of these pre-processing steps resulted in progressively more focused and sharper images at each stage, providing a robust foundation for the model's classification tasks. *Figure 4(a)* shows the original RGB image, which serves as the initial input image containing potential hair contaminants. This image is captured in its natural color format, with no preprocessing applied. In *Figure 4(b)*, the brightness of the image is adjusted to improve the visibility of hair particles, making them stand out more clearly against the background. *Figure 4(c)* demonstrates the grayscale conversion, where the RGB image is converted to grayscale, reducing computational complexity while preserving essential features needed for hair detection. This step simplifies the image data by removing color information while retaining important details. *Figure 4(d)* illustrates the effect of noise reduction using the Wiener filter.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

In this step, noise is reduced and image clarity is enhanced, resulting in a sharper representation of the image, making hair particles more visible. *Figure 4(e)* shows the masked image achieved after applying Canny edge detection, contour detection, and performing a bitwise AND operation with the original image. This step isolates specific regions of interest, highlighting the hair particles by removing extraneous information and emphasizing the boundaries. Finally, *Figure 4(f)* presents the smoothened 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.

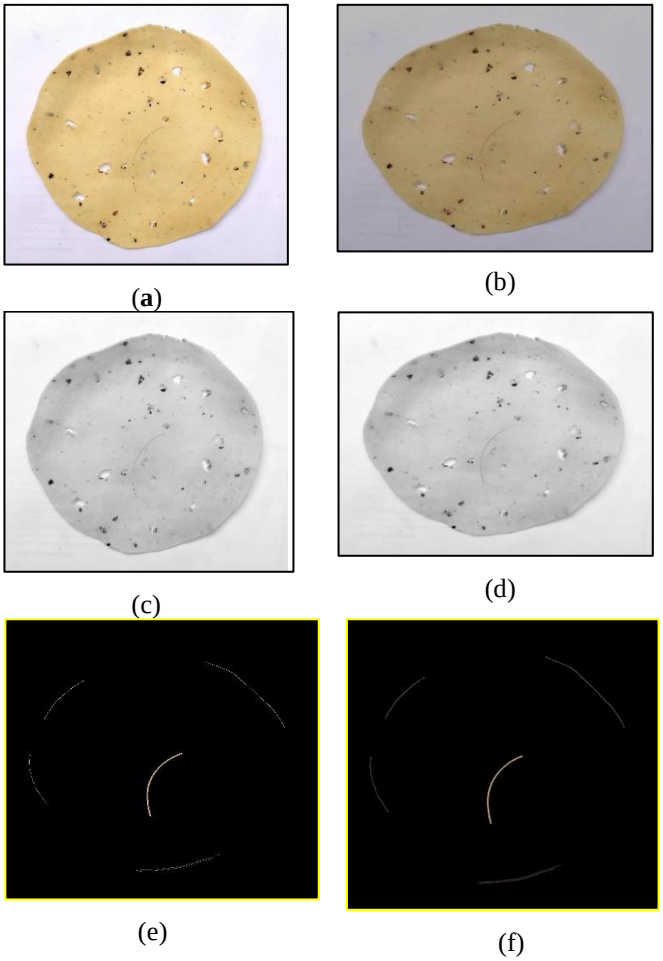

Figure 4. Inception V3 Architecture (a) Original image; (b) Brightness adjusted image; (c)Grayscale converted image; (d) Noise reduction using Weiner Filter; (e) Masked Image (After applying Canny Edge, contour detection and Bitwise AND with original image); (f) Smoothened Masked Image (depicting hair)

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.

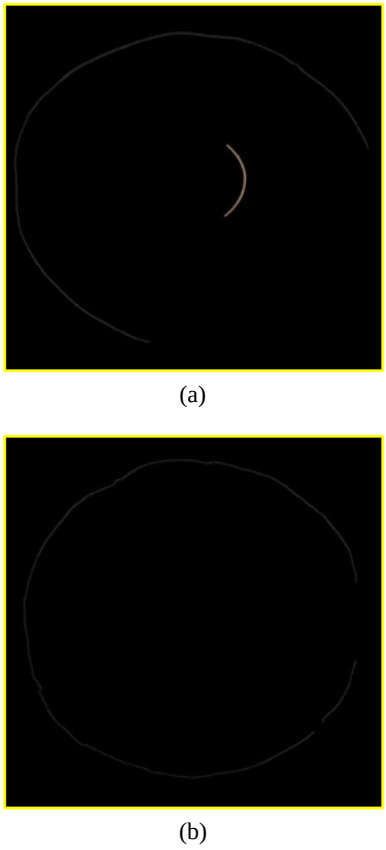

Figure 5. Comparison of pre-processed Images with and without air Particles (a) Pre-processed Image with Hair Particles; (b) Pre-processed Image without Hair Particles

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. It showed high precision (0.81) and recall (0.99) for images containing hair particles ('with hair'), indicating strong detection capability. However, it exhibited lower precision (0.99) and recall (0.76) for hair-free images ('without hair'). The fine-tuned InceptionV3 model excelled with an accuracy of 97.18%. It demonstrated superior precision (0.95) and recall (0.99) for images with hair particles, and similarly high precision (0.99) and recall (0.95) for without hair images. *Table 3* depicts the evaluation results for custom CNN and InceptionV3 Models.

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%		

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

After the CNN and Inception V3 models are compared, the comprehensive results of the Inception V3 model are explored to provide a deeper understanding of its architecture. The model begins with basic convolutional layers. It includes initial convolution operations and batch normalization to capture fundamental image features such as edges and textures. A series of Inception modules, each containing parallel convolutions and variable kernel sizes and pooling operations to address various scales and complexities, are used to further process these characteristics.

Convolutional layers, which reside deeper in the network, filter the incoming data several times to capture intricate 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 6*.

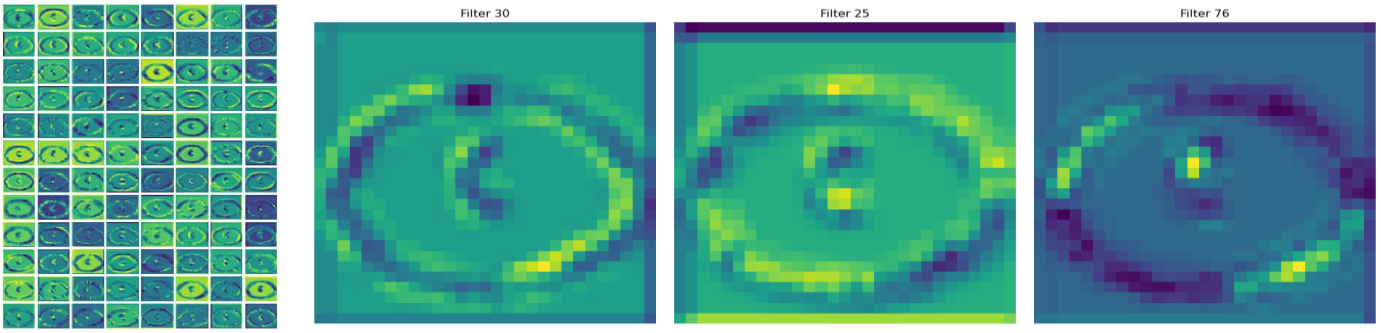

Figure 6. Visuals of feature map and high-level features learned at conv_9 layer

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 Inception V3 model for different epoch settings, with a batch size of 32. As the number of epochs grow, the model exhibits improved performance, as evidenced by higher test accuracy and lower test loss.

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

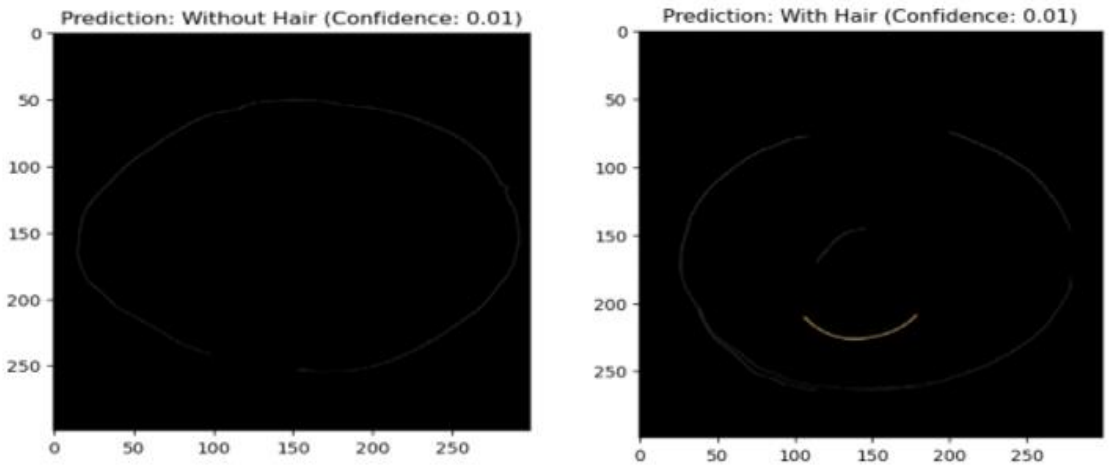

Figure 7. Images Classification into Two Classes (Without hair and with hair)

The Inception V3 model effectively classifies papadam images into 'with' or 'without' hair particles. *Figure 7* illustrates the confidence values associated with the Inception V3 model's classification for both categories. *Figure 8* 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 under-fitting.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Computer Science Research (IJCSR)
Survey Article | Volume 3, Issue 3 | Pages 109-117 | ISSN XXXX-XXXX

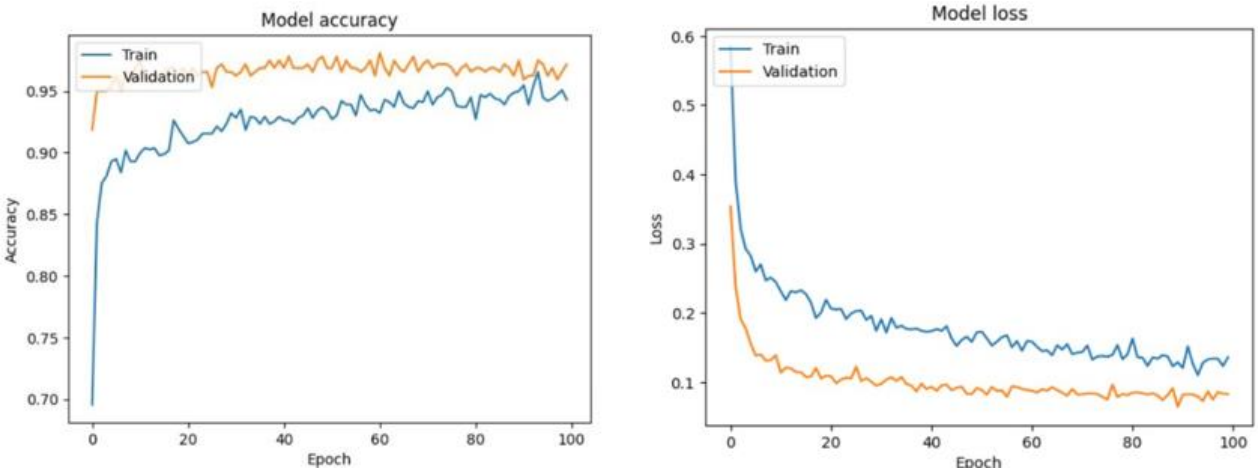

Figure 8. Training and Testing Loss Values and Accuracy for Inception V3 Model

4. DISCUSSION

Hair contamination in papadam during production is a significant challenge. It impacts customer trust and the brand name. The proposed approach combines the robust pre-processing with Inception V3 model for classification. This integrated approach provides an efficient solution to the issue of hair contamination in papadam. The pre-processing pipeline includes gray scale conversion of an image, Weiner filter, Canny edge detector and image masking. This pre-processing pipeline is designed to enhance the papadam image and facilitate classification (with hair and without hair) accurately. Inception V3 model parameters are fine tuned for the task of papadam image classification. Model demonstrated the accuracy of 97.18% on the test data set. This highlights the effectiveness and efficiency of the papadam hair detection model in detecting hair contamination accurately despite the challenges posed by various image conditions such as changing light and background noise. These results also highlight the robustness The results also highlight the robustness of transfer learning model and how reliable pre-trained model is when using on huge dataset. The inception V3 model demonstrated the balanced performance in both classes. Balanced perform is required for maintaining high standards in food safety and quality control in papadam industry. Table 3 shows the direct comparison of the accuracies achieved by CNN and Inception V3 model demonstrating the superior performance of Inception V3 model.

The proposed system presented a significant advancement in the application of deep learning and image processing for hair contaminant detection in 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.

Funding: This research received no external funding.

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.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1] K. Payne, C. A. O'Bryan, J. A. Marcy, and P. G. Crandall, "Detection and prevention of foreign material in food: A review," Heliyon, vol. 9, no. 9, Sep. 2023, doi: 10.1016/J.HELIYON.2023.E19574.

[2] D. G. Caldwell, "Automation in Food Manufacturing and Processing," Springer Handbooks, vol. Part F674, pp. 949–971, 2023, doi: 10.1007/978-3-030-96729-1_44.

[3] K. Patel, "Image Processing Tools and Techniques Used in Computer Vision for Quality Assessment of Food Products: A Review," International Journal of Food Quality and Safety | Year-2015 |, vol. 1, no. December, pp. 1–16, 2015, [Online]. Available: www.jakraya.com/journal/ijfqs

[4] B. Geueke, K. Groh, and J. Muncke, "Food packaging in the circular economy: Overview of chemical safety aspects for commonly used materials," J Clean Prod, vol. 193, pp. 491–505, Aug. 2018, doi: 10.1016/J.JCLEPRO.2018.05.005.

[5] Y. Chen et al., "Low cost smart phone diagnostics for food using paper-based colorimetric sensor arrays," Food Control, vol. 82, pp. 227–232, Dec. 2017, doi: 10.1016/J.FOODCONT.2017.07.003.

[6] H. Einarsdóttir et al., "Novelty detection of foreign objects in food using multi-modal X-ray imaging," Food Control, vol. 67, pp. 39–47, Sep. 2016, doi: 10.1016/J.FOODCONT.2016.02.023.

[7] M. A. Subhi and S. Md. Ali, "A Deep Convolutional Neural Network for Food Detection and Recognition," pp. 284–287, Jan. 2019, doi: 10.1109/IECBES.2018.8626720.

[8] Y. Pan, Y. Zhang, Q. Liu, Z. Wu, W. Hu, and Y. Wei, "A Contour Detection Method Based on Image Saliency," Proceedings of 2020 IEEE 9th Data Driven Control and Learning Systems Conference, DDCLS 2020, pp. 42–46, Nov. 2020, doi: 10.1109/DDCLS49620.2020.9275283.

[9] T. G. Devi and N. Patil, "Analysis &evaluation of Image filtering Noise reduction technique for Microscopic Images," 2020 International

Conference on Innovative Trends in Information Technology, ICITIIT 2020, Feb. 2020, doi: 10.1109/ICITIIT49094.2020.9071556.

[10] M. Pu, Y. Huang, Y. Liu, Q. Guan, and H. Ling, "EDTER: Edge Detection with Transformer," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2022-June, pp. 1392–1402, Mar. 2022, doi: 10.1109/CVPR52688.2022.00146.

[11] Y. Zhang, H. Huang, Y. Ma, Q. Liu, and G. Ding, "Recognition of Foreign Objects in Food Images Using Support Vector Machine," Lecture Notes in Electrical Engineering, vol. 551 LNEE, pp. 667–674, 2020, doi: 10.1007/978-981-15-3250-4_84.

[12] K. Goyal, P. Kumar, and K. Verma, "Food Adulteration Detection using Artificial Intelligence: A Systematic Review," Archives of Computational Methods in Engineering, vol. 29, no. 1, pp. 397–426, Jan. 2022, doi: 10.1007/S11831-021-09600-Y.

[13] L. Hansen and M. F. Ferrão, "Identification of Possible Milk Adulteration Using Physicochemical Data and Multivariate Analysis," Food Anal Methods, vol. 11, no. 7, pp. 1994–2003, Jul. 2018, doi: 10.1007/S12161-018-1181-6/METRICS.

[14] I. G. I. Sudipa, R. A. Azdy, I. Arfiani, N. M. Setiohardjo, and Sumiyatun, "Leveraging K-Nearest Neighbors for Enhanced Fruit Classification and Quality Assessment," Indonesian Journal of Data and Science, vol. 5, no. 1, pp. 30–36, Mar. 2024, doi: 10.56705/IJODAS.V5I1.125.

[15] A. Inglis, A. C. Parnell, N. Subramani, and F. M. Doohan, "Machine Learning Applied to the Detection of Mycotoxin in Food: A Systematic Review," Toxins 2024, Vol. 16, Page 268, vol. 16, no. 6, p. 268, Jun. 2024, doi: 10.3390/TOXINS16060268.

[16] M. Soltani Firouz and H. Sardari, "Defect Detection in Fruit and Vegetables by Using Machine Vision Systems and Image Processing," Food Engineering Reviews 2022 14:3, vol. 14, no. 3, pp. 353–379, Mar. 2022, doi: 10.1007/S12393-022-09307-1.

[17] M. K. Tripathi and D. 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, Jun. 2020, doi: 10.1016/J.INPA.2019.07.003.

[18] E. Elhariri, N. El-Bendary, A. E. Hassanien, A. Badr, A. M. M. Hussein, and V. Snášel, "Random Forests Based Classification for Crops Ripeness Stages," Advances in Intelligent Systems and Computing, vol. 303, pp. 205–215, 2014, doi: 10.1007/978-3-319-08156-4_21.

[19] Z. 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, p. 105141, Apr. 2023, doi: 10.1016/J.JFCA.2023.105141.

[20] "Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications MIT Press eBooks | IEEE Xplore." Accessed: Aug. 22, 2024. [Online]. Available: https://ieeexplore.ieee.org/book/6267356

[21] J. Canny, "A Computational Approach to Edge Detection," IEEE Trans Pattern Anal Mach Intell, vol. PAMI-8, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.

[22] S. Suzuki and K. A. be, "Topological structural analysis of digitized binary images by border following," Comput Vis Graph Image Process, vol. 30, no. 1, pp. 32–46, Apr. 1985, doi: 10.1016/0734-189X(85)90016-7.

[23] L. Nucci, D. Narvaez, and T. Krettenauer, Digital Image Processing Second Edition, no. June. 2014.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[26] I. Goodfellow, Y. Bengio, A. Courville, and J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning," Genetic Programming and Evolvable Machines 2017 19:1, vol. 19, no. 1, pp. 305–307, Oct. 2017, doi: 10.1007/S10710-017-9314-Z.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-December, pp. 2818–2826, Dec. 2016, doi: 10.1109/CVPR.2016.308.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans Knowl Data Eng, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.