Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Optimal Ensemble Deep Learning Framework for Non-Destructive Egg Quality Analysis Supporting Food Packaging Applications

Thisakya Ransarani
Sri Lanka Institute of Information Technology
Nushara Wedasingha
Sri Lanka Institute of Information Technology
Ilya Kavalchuk
ikavalchuk@swin.edu.au

Swinburne University of Technology

Research Article

Keywords: automated sorting, egg crack detection, ensemble convolutional neural network

Posted Date: September 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10517617/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Optimal Ensemble Deep Learning Framework for Non-Destructive Egg Quality Analysis Supporting Food Packaging Applications

Thisakya Ransarani [a], Nushara Wedasingha [a,c], Ilya Kavalchuk [b*]
[a] Faculty of Engineering, Sri Lanka Institute of Information Technology, Sri Lanka
[b] School of Engineering, Swinburne University of Technology, Australia
[c] Centre for Excellence in Intelligent Electronics & Telematics (CIET), Sri Lanka

*Corresponding author's email: ikavalchuk@swin.edu.au , Contributing Authors emails: thisakya2k@gmail.com, nushara.w@sliit.lk

ABSTRACT

Eggshell cracks during handling and distribution present significant challenges for the egg processing and packaging industry, causing economic losses and contamination risks. Manual inspection methods are labour-intensive, subjective, and unsuitable for high-throughput production environments. Existing automated approaches commonly depend on large architectures, controlled imaging conditions, or complex preprocessing pipelines, limiting practical deployment. This study proposes a lightweight, non-destructive ensemble convolutional neural network (CNN) framework for automated egg crack detection and quality analysis. The architecture integrates three parallel CNN branches with kernel sizes of 3×3, 5×5, and 7×7 to extract complementary fine, medium, and broad eggshell features from 32×32 grayscale images. The ensemble CNN achieved an accuracy of 97.50%, an F1-score of 0.97, and an AUC of 0.93, outperforming EfficientNetB0, MobileNetV2, GoogLeNet, ResNet50, VGG16, and VGG19 in overall classification performance. Five-fold cross-validation confirmed model stability under limited dataset conditions. The model requires only 4.9 million FLOPs, with a mean inference time of 25.5 ms per image and 1.66 GB RAM utilisation. Robustness evaluation under variable illumination conditions demonstrated reliable performance, and productivity analysis estimated a nearly 100-fold reduction in inspection time relative to manual grading. The proposed framework provides an accurate, computationally efficient, and scalable solution for non-destructive egg quality inspection in automated sorting and food packaging applications.

Keywords: automated sorting, egg crack detection, ensemble convolutional neural network

1 INTRODUCTION

Accurate, non-destructive quality inspection is a fundamental requirement in modern food processing, where eggshell cracks present persistent challenges during handling, distribution, and packaging [1], [2]. With global egg production increasing by approximately 40% over the past two decades [3], the scale of this challenge has grown significantly, causing economic losses and contamination risks that affect food safety and supply chain integrity [4]. Ensuring reliable crack detection at industrial throughput therefore demands intelligent, automated solutions beyond the capacity of manual grading.

Conventional inspection relies on manual grading methods including visual assessment, olfaction, candling, and flotation tests [5-13], which achieve batch-level accuracies of approximately 94.2% [14] but remain subjective, labour-intensive, and unsuitable for high-throughput environments [11], [15]. Non-destructive approaches such as candling require years of operator training [12], limiting scalability, while sensor-based systems using weight, light intensity, and vibration signals [9], [12], [16] are constrained by single-parameter reliance, complex calibration, and reduced sensitivity to subtle defects. These limitations

establish a clear need for intelligent, automated inspection solutions capable of reliable performance under practical industrial conditions.

Image-processing-based approaches have also been explored. Fuzzy logic with colour variation achieved 95% accuracy [17], deterministic pipelines combining edge detection and template matching reported approximately 90% accuracy [18], and grayscale conversion with contour analysis reached 98% accuracy [19]. However, these methods often rely on predefined thresholds, resulting in static learning behaviour, limited adaptability, and reduced robustness under variable industrial conditions. Their maintenance requires domain expertise, further limiting cost-effectiveness and scalability.

Machine learning (ML) techniques have been applied to overcome these limitations. Algorithms such as k-NN, SVM, Random Forest, Decision Trees, and LDA [22-30] have shown promising results. For example, Random Forest achieved >99% accuracy for micro-crack detection [23], while SVM with image processing reached 98.75% [22]. Nevertheless, ML methods rely heavily on handcrafted features, limiting generalisation and robustness in complex industrial settings.

Deep learning (DL) approaches address these challenges by learning hierarchical features directly from raw data, eliminating the need for manual feature engineering [25]. CNNs [10], [18], [32-34], LSTM and GRU models [23], [24], and hybrid frameworks [29], [30] have been explored. CNNs dominate crack detection [31], achieving accuracies between 93.8% and 98% across custom, hybrid GLCM-CNN, and VGG-based architectures [4], [13], [26], [28]. Despite their effectiveness, many DL models require extensive data augmentation, transfer learning, or complex preprocessing [26], [28], [32], [33], increasing computational cost, memory demand, and overfitting risk [34]. Large architectures like VGG-16, VGG-19, and ResNet-50 are difficult to deploy in resource-constrained environments [5], [13], and reliance on specialised hardware or hybrid pipelines limits scalability [5], [13], [26], [32], [35].

Although existing DL-based egg inspection methods have demonstrated strong classification performance, a clear gap remains in developing a system that is simultaneously accurate, computationally efficient, low-cost, and robust under practical industrial conditions. Many reported approaches depend on large pretrained networks, extensive augmentation, controlled lighting, specialised image acquisition setups, or multi-stage preprocessing pipelines, which increase implementation complexity and limit real-time deployment in high-throughput egg sorting environments. Furthermore, most studies focus on improving classification accuracy alone, with less emphasis on model simplicity, latency, scalability, and reliability when trained on limited or unconstrained image datasets. Therefore, there is a need for a lightweight, non-destructive, and adaptable deep learning framework capable of extracting complementary visual features while maintaining low computational requirements for practical food supply chain applications. To address these gaps, this study proposes a non-destructive ensemble CNN framework for classifying cracked and intact eggs. The architecture integrates three independent CNN models with distinct kernel sizes, capturing both fine-grained and global visual features to enhance robustness and accuracy. To the authors' knowledge, this multi-kernel ensemble has not been applied previously to egg quality assessment or food supply chain applications. Beyond classification performance, the development of lightweight, deployable AI systems for food quality assurance directly supports SDG 9 by fostering innovation and building resilient, sustainable industrial infrastructure. The key contributions of this study are as follows:

• Propose a multi-kernel ensemble CNN architecture that extracts complementary spatial features at three scales from limited training data *(Section 3.1)*.

• Design a computationally efficient CNN achieving 97.50% accuracy at only 4.9 million FLOPs, suitable for resource-constrained deployment *(Section 3.2)*.

• Develop a low-latency architecture that enables high-throughput egg sorting to enhance packaging efficiency *(Section 3.3)*.

• Leverage a diverse, unconstrained image dataset to remove the need for specialised lighting, reducing system complexity and cost and enabling robust deployment in industrial egg processing and packaging lines. *(Section 3.4)*.

The remainder of this paper is organised as follows. In Section 2, the procedures used for dataset preparation, model training, architecture development, and performance evaluation are presented. The evaluation results are reported in Section 3. In Section 4, an overall model summary is provided, and the extent to which the stated contributions were achieved is discussed. Finally, the conclusions and recommendations for future work are presented in Section 5.

2 METHODOLOGY

This section describes the development and evaluation process of the proposed ensemble CNN model for automated egg crack detection. The methodology includes dataset preparation, preprocessing, feature extraction using multiple convolutional kernel sizes, feature fusion, classification, and result derivation. The proposed approach was designed to achieve accurate egg crack classification while maintaining low computational complexity for practical deployment. In addition to classification performance, robustness under varying imaging conditions, inference speed, hardware resource utilisation, and productivity benefits were evaluated to assess the model's suitability for real-world egg sorting applications.

2.1 TRAINING AND DATASET

For this study, a comprehensive dataset was created by aggregating images from various online sources, including publicly available datasets [36-41]. The dataset was curated into two classes: cracked and uncracked eggs, comprising chicken eggs with both white-shelled and brown-shelled specimens, and each image was validated by three independent professionals to ensure accuracy. To enhance robustness, the images included varying numbers of eggs, diverse backgrounds, different camera distances, and real-world conditions. A subset of images containing other cracked surfaces was also included to support the learning of generalised crack visual patterns [38], comprising approximately 80 images and representing approximately 5.7% of the total dataset. In addition, uncracked egg images with natural surface imperfections, such as dirt marks, stains, and shell irregularities, were included to reduce the likelihood of false crack detection caused by non-crack surface features.

The final dataset consisted of 1,400 images, which were divided into training and testing subsets using an 80:20 split. Separate training and test directories were created, with cracked and uncracked images organised into their respective labelled class folders. Table 1 provides a detailed dataset breakdown.

Table 1
Dataset composition for egg crack classification

Dataset split	Cracked	Non-cracked	Total
Training set	560	560	1,120
Test set	140	140	280

To ensure a reliable evaluation, the test set was kept completely independent from the training set. Prior to splitting, the dataset was reviewed to minimise the possibility of data leakage, including duplicate, near-duplicate, or highly similar images appearing across both subsets. Images with strong visual similarity, such as repeated captures of the same egg or images taken from nearly identical viewpoints, were prevented from being shared between the training and test sets. This ensured that the model was evaluated on unseen samples rather than memorised visual patterns from the training data, providing a more realistic assessment of generalisation capability.

2.2 MODEL ARCHITECTURE AND CLASSIFICATION WORKFLOW

This section presents the methodology for constructing and implementing the proposed CNN-based model for egg crack detection. It details the architectural design, including multiple convolutional layers with varying kernel sizes to extract diverse and representative features, and describes hyperparameter optimisation to enhance model performance. Input images are processed through the convolutional and hidden layers to extract features for classifying eggs into two categories. The overall architecture and processing workflow are illustrated in Fig. 1.

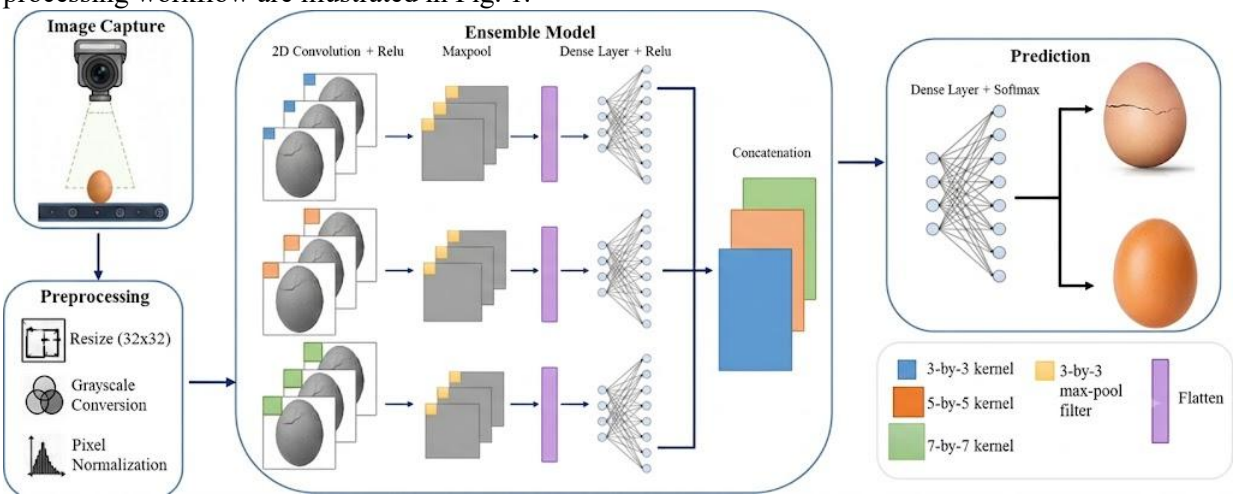


Fig. 1. Ensemble model's overall architecture

2.2.1 Feature extraction

A key challenge in training CNN-based classification models is the availability of a limited dataset, which can restrict the model's ability to learn diverse and representative features. To address this limitation, the proposed architecture uses a multi-branch ensemble CNN design that extracts complementary features from the same input image at different spatial scales. This approach is intended to strengthen feature learning by allowing the model to capture fine, medium, and broader visual patterns associated with eggshell cracks.

The feature extraction stage was designed to identify discriminative visual patterns associated with cracked and uncracked eggshell surfaces. The input to the proposed ensemble model is a digital image of eggs, resized to 32×32 pixels and converted to grayscale. As eggshell crack patterns are primarily defined by structural texture rather than colour, grayscale conversion preserves the discriminative features relevant to classification while reducing input dimensionality.

The ensemble architecture consists of three independent CNN models: Model 1, Model 2, and Model 3, each receiving the same input but employing different kernel sizes: 3×3, 5×5, and 7×7, respectively. Three branches were selected to provide sufficient feature scale diversity for capturing fine, medium, and broad spatial patterns, while maintaining a compact and lightweight architecture suitable for resource-constrained deployment. Using varying kernel sizes allows the extraction of both fine-grained and global features [42].

Each model applies 32 distinct kernels to learn a broad range of feature representations, improving classification performance even on a limited dataset. Each branch produces 32 feature maps that represent different learned visual characteristics of the input image. The convolutional outputs were passed through a ReLU activation function [42] to introduce non-linearity and retain the most relevant positive feature responses. The convolution and ReLU activation operation used in each CNN branch is expressed in Eq. (1).

$$A_{f,k}(i,j) = \max(0, \sum_{m=0}^{f-1} \sum_{n=0}^{f-1} X(i+m, j+n) K_{f,k}(m,n) + b_{f,k}, \qquad f \in \{3,5,7\}) \tag{1}$$

Where (X(i,j)) represents the input grayscale image, (f) denotes the kernel size used in the branch, and (k) represents the convolutional filter index. ($K_{f,k}(m,n)$) is the (k^{th}) convolution kernel of size (f x f), while ($b_{f,k}$) is the corresponding bias term. The output ($A_{f,k}(i,j)$) represents the activated feature response at spatial location ((i,j)) after applying the ReLU function.

Following convolution, a 3×3 max-pooling layer reduces spatial dimensions by selecting the maximum value from each region of the feature maps while preserving the strongest responses. This reduces computational complexity and helps to retain the most significant crack-related features. The pooled feature maps from each branch were flattened into a one-dimensional feature vector and passed through a dense layer containing 55 neurons with ReLU activation. The dense layer width of 55 neurons was selected to maintain a 1:1 mapping between each branch's feature vector and its compressed representation, producing a balanced 165-dimensional fused vector. This allowed each CNN branch to generate a compact feature representation based on its specific kernel scale. Dropout with a rate of 0.2 was applied after the dense layer to reduce overfitting and improve generalisation. The dropout rate of 0.2 was selected following empirical evaluation of alternative values, as it produced the best balance between regularisation and classification performance. The dense-layer transformation with ReLU activation and dropout regularisation is expressed in Eq. (2).

$$\widetilde{h_f} = d_f \odot \max(0, W_{fzf} + \beta_f) \tag{2}$$

Where (z_f) is the flattened feature vector from branch (f), (W_f) and (β_f) are the dense-layer weight matrix and bias vector, respectively, (d_f) is the dropout mask, and $\widetilde{h_f}$ is the dropout-regularised feature representation generated by branch (f). Since a dropout rate of 0.2 was used, 80% of the dense-layer activations were retained during training.

All models were implemented in TensorFlow/Keras and trained on Google Colab using an NVIDIA Tesla T4 GPU. Training was conducted over 30 epochs with a batch size of 32 and a learning rate of 0.001, consistent with the Adam optimiser default. Kernel weights were initialised using the Glorot uniform initialiser and biases were initialised to zero. Total training time for the proposed ensemble model was approximately six minutes.

2.2.2 Decision making

The decision-making stage was designed to support accurate and rapid classification while remaining simple, lightweight, and computationally efficient. Complementary features extracted by the three CNN branches were integrated to support high-throughput egg sorting. Since each branch captures visual features at a different spatial scale, the outputs of the dense ReLU layers were concatenated to form a single fused feature vector. The concatenated feature vector is expressed in Eq. (3).

$$H = [\widetilde{h_3}; \widetilde{h_5}; \widetilde{h_7}] \tag{3}$$

Where ($\widetilde{h_3}$), ($\widetilde{h_5}$), and ($\widetilde{h_7}$) represent the dropout-regularised feature vectors generated by the (3 x 3), (5 x 5), and (7 x 7) CNN branches, respectively. Since each branch produces a 55-dimensional feature vector, their concatenation forms a fused feature vector (H) with 165 features.

The concatenated feature vector was passed into the final dense classification layer with two output neurons, representing the two target classes: cracked and uncracked eggs. The final dense layer produced class scores, which were converted into class probabilities using the SoftMax activation function. The class with the highest probability was then selected as the final prediction, as expressed in Eq. (4).

$$\hat{C} = argmax_{c\in\{1,2\}}\left(\frac{e^{(W_cH+\beta_c)_c}}{\sum_{r=1}^{2} e^{(W_cH+\beta_c)_r}}\right) \quad (4)$$

Where (W_c) and (β_c) are the weight matrix and bias vector of the final dense classification layer, respectively, and (c) represents the class index. The two class indices correspond to cracked and uncracked eggs. The SoftMax function calculates the probability of each class, while ($\hat{C}$) represents the final predicted class with the highest probability.

Therefore, the model did not rely on a single feature scale for decision-making; instead, it used a combined representation from all three kernel branches to improve classification reliability. The model was compiled using the Sparse Categorical Crossentropy loss function and the Adam optimiser. To ensure a fair comparison, all benchmark architectures were trained from scratch on the same dataset, without the use of pretrained ImageNet weights.

2.2.3 Result Derivation

First, the model's ability to extract meaningful features from a limited dataset was evaluated. Five-fold cross-validation [43] was applied to the training set, while the independent test set was excluded. Performance across the folds was reported using the mean and standard deviation to assess stability and generalisation. Training and validation loss curves were not retained as primary evaluation outputs. Instead, model generalisation was assessed through five-fold cross-validation on the training set and evaluation on a fully independent held-out test set, which collectively provide a more robust estimate of generalisation performance under limited dataset conditions [43].

The feature extraction capability of the ensemble model was then evaluated by comparing its multi-kernel architecture with individual CNN models using single kernel sizes. Classification performance on the independent test set was assessed using a confusion matrix. The proposed ensemble CNN was also compared with selected benchmark architectures, including established deep CNNs and lightweight deployment-oriented models. Accuracy, F1-score [44], and area under the ROC curve (AUC) [44], [45] were used as evaluation metrics. Direct quantitative comparison with previously published egg crack detection studies was not performed, as differences in dataset composition, image acquisition setup, egg type, and input resolution across studies preclude meaningful like-for-like comparison. Confidence-level testing was then conducted to assess prediction consistency over time, complementing static image classification evaluation. Minimum, maximum, and average confidence values were derived from frame-by-frame predictions to examine prediction stability during continuous visual input.

After feature extraction and classification performance had been evaluated, the computational requirements of the model were assessed. First, input resolution was systematically analysed across three scales to determine the optimal trade-off between classification performance and computational efficiency, as resolution directly influences feature richness and model complexity in CNN-based systems. For each resolution, accuracy and F1-score were used as performance metrics, while floating-point operations (FLOPs) [46], inference time, and RAM utilisation were recorded as computational measures. The computational efficiency of the proposed model was then evaluated against the benchmark architectures in terms of FLOPs and RAM utilisation.

To examine the latency and throughput of the model, the inference time required to classify a single egg image was measured and compared with those of the benchmark architectures. A productivity analysis was

then performed by comparing the estimated automated sorting time with manual inspection time, to quantify the throughput benefit of automated deployment.

Given that real-world egg processing environments are subject to variable illumination, the model was evaluated under controlled brightness and simulated colour-based lighting variations to assess deployment robustness beyond controlled laboratory conditions.

3 RESULTS

This section presents the evaluation results of the proposed ensemble CNN across four dimensions: feature extraction and classification performance, computational efficiency and resource utilisation, inference latency and productivity, and robustness under variable imaging conditions.

3.1 FEATURE EXTRACTION AND CLASSIFICATION PERFORMANCE EVALUATION

3.1.1 5-Fold Cross-Validation Analysis

Five performance metrics were evaluated across the validation folds: accuracy, precision, recall, F1-score, and AUC. The mean and standard deviation of each metric are illustrated in Fig. 2.

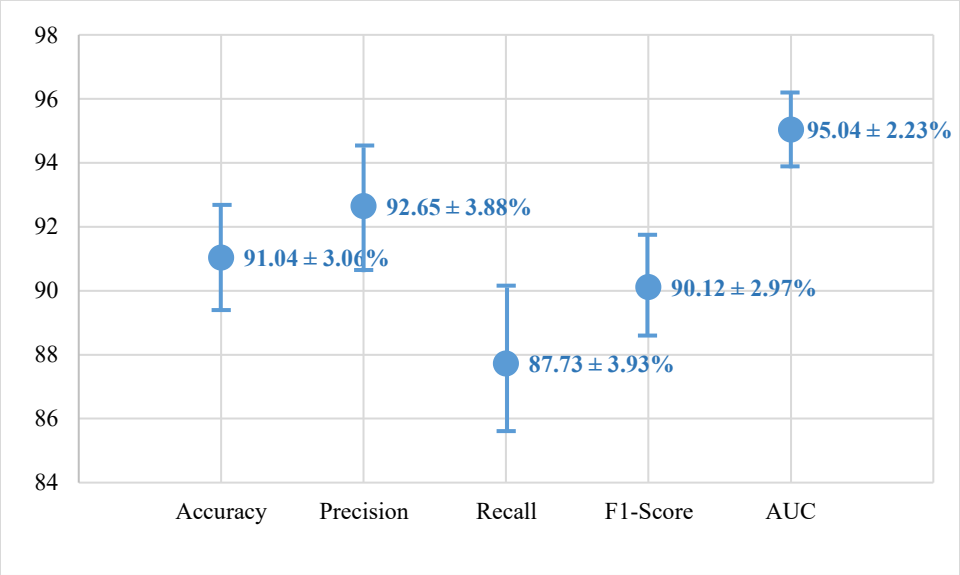


Fig. 2. Five-fold cross-validation performance of the proposed ensemble CNN

The standard deviations across all metrics ranged from 2.23% to 3.93%, indicating relatively low variation and consistent performance across the five validation folds. An average accuracy of 91.04% was achieved, with a balanced relationship between precision and recall, while an average AUC of 0.95 across the validation folds indicated good class separability. The cross-validation results may differ from the final independent test performance because in five-fold cross-validation, each model instance is trained on approximately 896 images, representing 80% of the 1,120 training samples, whereas the final model is trained on the full training set. Additionally, some validation folds may contain more challenging images, such as subtle cracks or variable backgrounds, contributing to the more conservative cross-validation estimate. Consequently, a conservative estimate of stability is provided, while performance on the unseen

independent test set is represented by the final test result. The cross-validation results therefore confirm that the proposed architecture extracts stable and generalisable features under limited dataset conditions.

3.1.2 Comparative Evaluation of Multi-Kernel Feature Extraction

The ensemble model was compared with individual CNNs using 3×3, 5×5, and 7×7 kernels on the independent test set of 280 images. The confusion matrices obtained for each model are presented in Fig. 3.

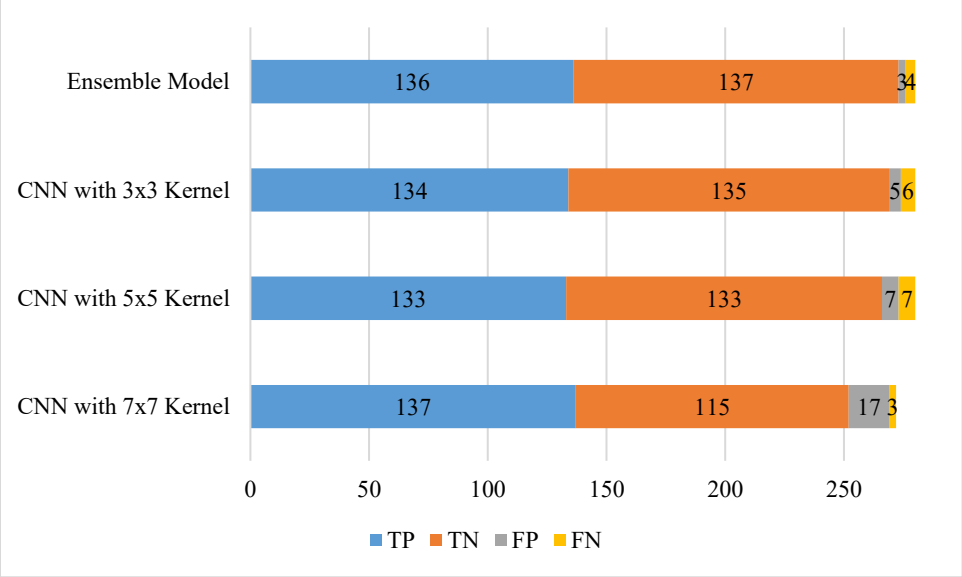


Fig. 3. The analysis of TP, TN, FP, and FN against different CNN model architectures

The highest accuracy of 97.50% was achieved by the ensemble model, compared with 96.07%, 95.00%, and 92.86% for the single-kernel models. Although a high true-positive rate was obtained by the 7 × 7 CNN, lower overall accuracy was recorded, suggesting class imbalance in predictions and susceptibility to overfitting on the training data. These results demonstrate that combining complementary features across three spatial scales produces more balanced and reliable classification than any single-kernel architecture alone.

3.1.3 Comparative Performance Evaluation against Benchmark CNN Architectures

The proposed ensemble CNN's performance was compared against four established CNN architectures that have been widely used in image-classification and egg-defect detection studies: GoogLeNet [47-49], ResNet50 [32], [47], [51], VGG16, and VGG19 [5], [13], [47], [49]. In addition, two lightweight CNN architectures, MobileNetV2 [52] and EfficientNetB0 [46], were included to provide a fairer assessment of the proposed model against deployment-oriented architectures designed to reduce computational cost while maintaining competitive classification performance. All models were trained from scratch on the same dataset to ensure a consistent and unbiased comparison. Accuracy, F1-score, and AUC were used as the primary performance metrics, as presented in Table 2 and Fig. 4.

Table 2

Comparative performance of the proposed ensemble CNN and benchmark CNN architectures

Model	Ensemble	EfficientNetB0	MobileNetV2	GoogLeNet	ResNet50	VGG16	VGG19
Accuracy	97.50%	96.79%	51.02%	91.43%	82.50%	53.57%	52.86%
F1 score	0.97	0.95	0.38	0.90	0.91	0.68	0.61

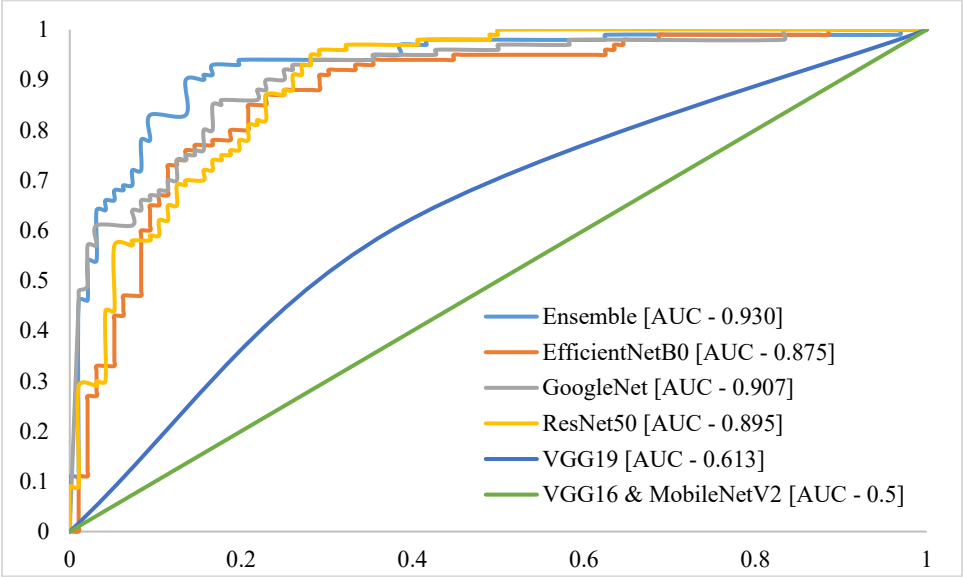


Fig. 4. ROC curve and AUC comparison of the proposed ensemble CNN model with benchmark CNN architectures

The highest overall performance was achieved by the ensemble model, with an accuracy of 97.50%, an F1-score of 0.97, and an AUC of 0.930, which, while lower than expected given the classification accuracy, reflects the presence of correct predictions made with lower confidence scores that reduce the ROC ranking without affecting binary classification performance. Among the benchmark models, the closest accuracy and F1-score were achieved by EfficientNetB0. Compared with EfficientNetB0, improvements of 0.71% in accuracy, 0.02 in F1-score, and 0.055 in AUC were obtained by the proposed ensemble CNN. Strong classification performance was also achieved by GoogLeNet, although lower accuracy and F1-score values were recorded. A higher AUC was nonetheless obtained, indicating better class separability across different decision thresholds. ResNet50 achieved an AUC of 0.895 and an F1-score of 0.91; however, its lower accuracy of 82.50% indicated less consistent classification performance on the test set.

Considerably weaker results were produced by VGG16 and VGG19. An AUC of 0.500 was obtained by VGG16, and performance close to random classification was thereby indicated. A slightly higher AUC of 0.613 was achieved by VGG19, although limited discrimination between the two classes was still observed. These larger and deeper architectures are likely less suited to the limited grayscale dataset, as their higher computational complexity and greater number of parameters increase susceptibility to overfitting under limited data conditions [13]. Although MobileNetV2 achieved high training accuracy, this performance was not retained on the test set, suggesting overfitting under the limited grayscale training conditions. Overall, the benchmark comparison demonstrates that the proposed ensemble CNN achieves the strongest classification performance among all evaluated architectures, attributable to its multi-scale feature extraction design.

3.1.4 Confidence Level Testing

Six 20-second video samples, each containing 490 frames, were used. Two uncracked (NC), two clearly cracked (C), and two slightly cracked (SC) egg videos were included. For each frame, the trained model generated a class prediction and corresponding confidence score. The results obtained for each video sample are presented in Fig. 5.

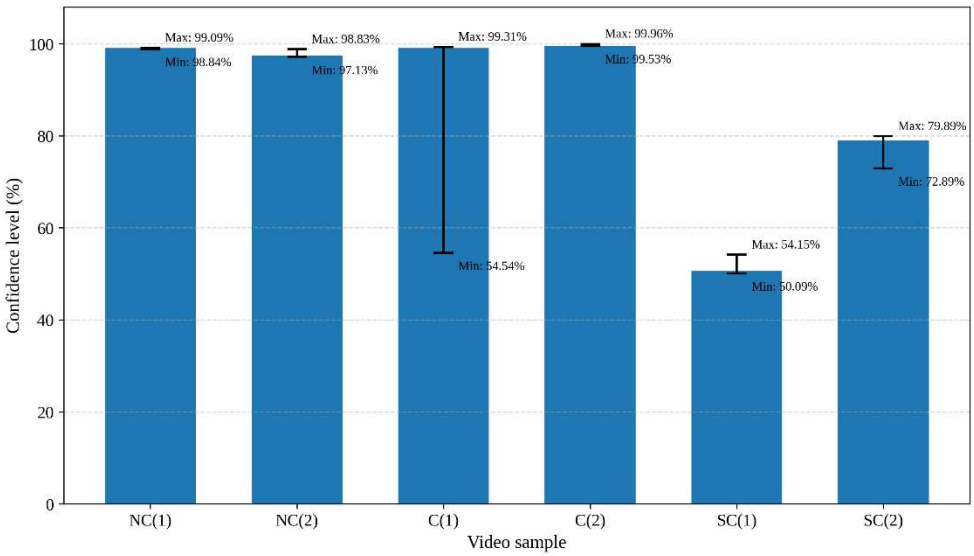


Fig. 5. Confidence level testing of uncracked, cracked, and slightly cracked egg video samples using minimum, average, and maximum confidence values.

Correct classifications were obtained for all six video samples. Average confidence levels above 90% were achieved for all uncracked and clearly cracked samples, indicating stable predictions when the visual features were clearly distinguishable. A minimum confidence of 54.54% was recorded for C(1) across individual frames, representing an isolated low-confidence prediction within an otherwise stable classification sequence. In contrast, lower confidence values were obtained for the slightly cracked samples. Average confidence values of approximately 51% and 78% were recorded for SC(1) and SC(2), respectively, reflecting variability in crack visibility between samples. The lower confidence observed for SC(1) indicates that the model operates near its decision threshold for subtle crack features, likely due to the visual similarity between hairline cracks and natural shell surface texture, while the higher confidence for SC(2) suggests that more visually prominent slight cracks are classified with greater certainty. It should be noted that the slightly cracked evaluation was based on two video samples, and further testing with a larger set of slightly cracked specimens is recommended to validate this finding. In a food safety context, the near-threshold confidence for slightly cracked eggs warrants careful consideration, as false negatives carry greater industrial risk than false positives, and threshold adjustment or post-processing may be required for deployment in safety-critical applications. These results demonstrate that the proposed model maintains reliable classification under continuous visual input, with prediction confidence reflecting the degree of visual distinction between crack classes.

3.2 Computational Efficiency and Resource Utilisation Analysis

3.2.1 Input Resolution Analysis

This analysis was conducted to identify the input resolution at which the best balance between classification performance and computational efficiency was achieved, as less visual information is retained when image resolution is reduced. The proposed ensemble CNN was evaluated at resolutions of 32 × 32, 64 × 64, and 128 × 128 pixels. Accuracy, F1-score, FLOPs, inference time, and RAM utilisation are presented in Table 3. The performance–efficiency trade-off is illustrated in Fig. 6, where the balance between precision and recall is represented by the F1-score, while theoretical computational demand is estimated using FLOPs.

Table 3
Effect of input image resolution on classification performance and computational efficiency

Input resolution	Accuracy (%)	F1-score (%)	FLOPs (Million)	Inference time (ms)	RAM utilisation (GB)
32 × 32	97.50	97.49	4.9	25.5	1.66
64 × 64	97.50	97.47	23.3	43.5	1.68
128 × 128	95.71	95.71	101.1	65.3	2.40

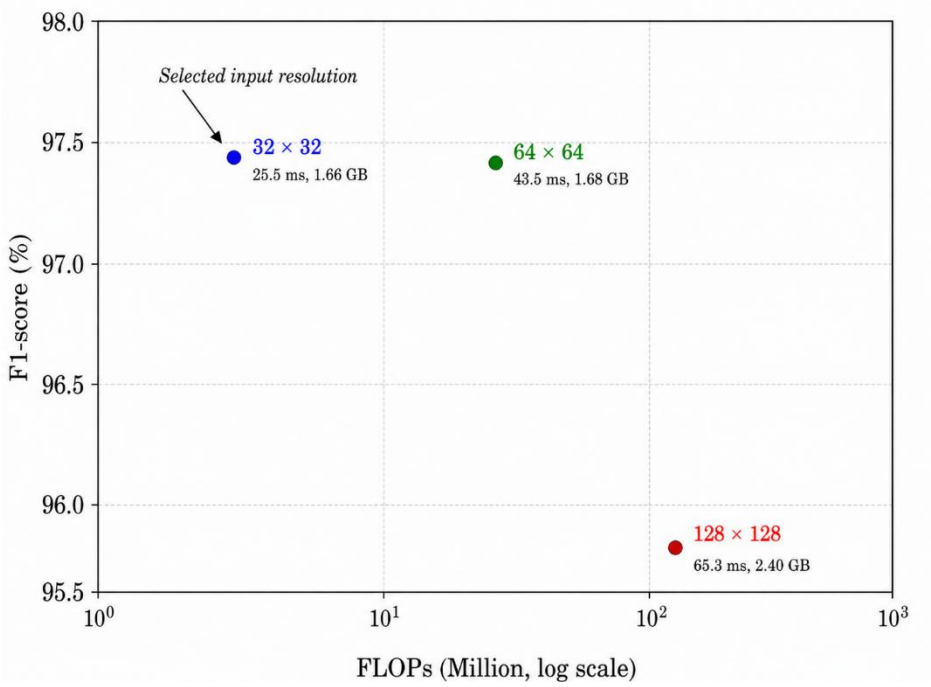


Fig. 6. Performance–efficiency trade-off of the proposed ensemble CNN under different input resolutions

The 32×32 and 64×64 input resolutions achieved the same accuracy of 97.50%; however, the 32×32 model produced a slightly higher F1-score, indicating a better balance between precision and recall. The slight reduction in F1-score at 64×64 was mainly due to an increase in false-negative predictions, where some cracked eggs were classified as uncracked. Although the 128×128 resolution preserved more image detail, it resulted in lower accuracy and F1-score, indicating that the additional visual information did not improve generalisation [53]. This may be due to the model learning non-discriminative details such as background texture, lighting variation, shell surface noise, and minor artefacts, particularly under limited dataset conditions. In terms of computational efficiency, the 32×32 resolution was the most suitable. It required only 4.9 million FLOPs, compared with 23.3 million for 64×64 and 101.1 million for 128×128. Inference time also increased from 25.5 ms to 43.5 ms and 65.3 ms as the resolution increased, while RAM utilisation increased notably at 128×128. Therefore, 32×32 was selected as the optimal input resolution because it maintained the highest overall classification performance while providing the lowest computational cost.

3.2.2 Model Complexity Analysis

Lower FLOP values are associated with lower theoretical computational complexity and demand, suggesting reduced computational cost, lower hardware demand, and improved suitability for real-time or resource-constrained applications. The FLOPs recorded for each model are presented in Table 4.

Table 4
Comparative FLOP analysis of the proposed ensemble CNN and benchmark CNN architectures

Model	Ensemble	EfficientNetB0	MobileNetV2	GoogLeNet	ResNet50	VGG16	VGG19
FLOPs in Millions	4.9	12.75	17.84	1758.8	7753.3	30811	35461

Only 4.9 million FLOPs were required by the proposed ensemble CNN, representing the lowest computational complexity among all evaluated models. Although EfficientNetB0 and MobileNetV2 were designed as lightweight architectures, they required approximately 2.6 and 3.6 times more FLOPs than the proposed model, respectively. Greater differences were observed for the larger benchmark architectures, with approximately 359, 1,582, 6,288, and 7,237 times more FLOPs required by GoogLeNet, ResNet50, VGG16, and VGG19, respectively. These results confirm that the proposed ensemble CNN achieves the lowest computational complexity among all evaluated architectures, including purpose-built lightweight models.

3.2.3 RAM Utilisation

RAM utilisation was evaluated as an important indicator of computational demand, particularly for resource-constrained and scalable industrial applications. The RAM utilisation of the proposed ensemble CNN was therefore compared with that of the benchmark models, as presented in Table 5.

Table 5
Comparative analysis of memory utilisation for the proposed ensemble CNN and benchmark architectures

Model	Ensemble	EfficientNet B0	MobileNet V2	GoogLeNet	ResNet50	VGG16	VGG19
RAM Utilisation	1.66GB	6.64GB	7.25GB	4.53GB	3.99GB	8.22GB	8.44GB

The lowest RAM utilisation of 1.66 GB was recorded for the proposed ensemble CNN during model operation. In comparison, approximately 4.0, 4.4, 2.7, 2.4, 4.95, and 5.08 times greater RAM utilisation was recorded by EfficientNetB0, MobileNetV2, GoogLeNet, ResNet50, VGG16, and VGG19, respectively. The proposed ensemble CNN therefore demonstrates substantially lower memory demand than both lightweight and larger benchmark architectures, confirming its suitability for resource-constrained deployment.

Although FLOPs and RAM utilisation are both used to evaluate computational efficiency, a direct relationship was not observed. This difference may be attributed to theoretical arithmetic complexity being represented by FLOPs, whereas operational memory demand is represented by RAM utilisation.

3.3 Inference Latency and Productivity Analysis

3.3.1 Model Inference Time

Execution speed was evaluated as a critical requirement for real-time, high-throughput automated egg-sorting systems, in which rapid decision-making is essential. The time required for single-image classification and the comparative benchmarking results are presented in Fig. 7.

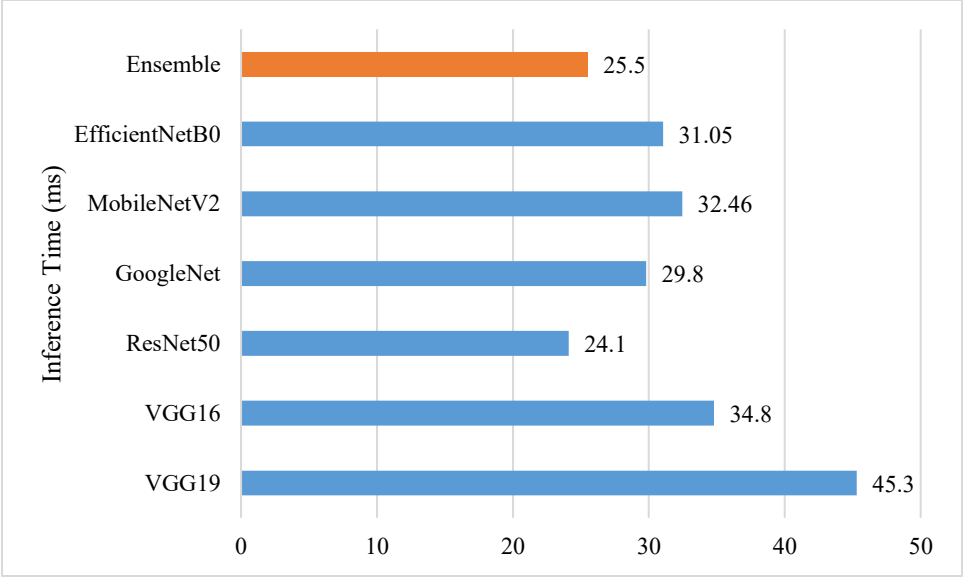


Fig. 7. Single-image inference time comparison of the proposed ensemble CNN and benchmark CNN architectures.

An inference time of 25.5 ms per image was achieved by the proposed ensemble CNN. Faster classification was achieved than with EfficientNetB0, MobileNetV2, GoogLeNet, VGG16, and VGG19, with inference times 5.55, 6.96, 4.3, 9.3, and 19.8 ms longer, respectively. The lowest inference time of 24.1 ms was recorded for ResNet50, representing a 1.4 ms reduction. This result indicates that inference time is not directly proportional to FLOPs at small input sizes, as execution is influenced by framework overhead, GPU kernel launch latency, and parallelism efficiency rather than arithmetic complexity alone. However, considerably lower classification accuracy and overall performance were obtained with ResNet50 than with the proposed model. The proposed model therefore achieves near-lowest inference latency among all evaluated architectures while maintaining substantially stronger classification performance.

3.3.2 Productivity Analysis

Automated tools and robotic systems operate with speed and precision that surpass manual labour, enabling faster production cycles, reduced processing times, and higher throughput. In the egg industry, this allows producers to handle larger volumes while consistently meeting market demands.

To evaluate time efficiency, the proposed system was compared with manual inspection. Human graders typically require over one hour to inspect 1,000 eggs, corresponding to a per-egg inspection time of more than 3,600 ms, including visual assessment and decision-making [54]. In contrast, the automated approach involves image acquisition (approximately 10 ms [55]) and model inference (25.5 ms), resulting in a total processing time of approximately 36 ms per egg. For a production throughput of 100,000 eggs per day [56], this represents a nearly 100-fold reduction in inspection time, as summarised in Table 6.

Table 6

Comparison of sorting time between human graders and the proposed model

Method	Time per egg (ms)	Time for 100,000 eggs (s)	Equivalent duration
Human	3,600	360,000	100 hours
Ensemble Model	36	3,600	1 hour

Beyond image acquisition and inference time, conveyor travel and actuator response delays were also considered. As the camera is positioned upstream of the sorting actuator, each egg can be classified before the sorting point is reached. The resulting latency affects mainly the first sorting action and is not accumulated for every egg when continuous flow, egg spacing, and actuator switching are properly controlled.

These results demonstrate that the proposed ensemble CNN supports the intended contribution of enabling high-throughput egg sorting, offering a computationally efficient and practically deployable solution for automated food packaging applications.

3.4 Evaluation under Variable Imaging Conditions

Evaluation was conducted using previously unseen images under controlled illumination variations. The tested conditions included changes in brightness intensity, represented by bright and dark images, and simulated colour-based illumination variations, represented by yellow, blue, and red lighting effects. For each condition, 20 images were used, consisting of 10 cracked and 10 uncracked egg images. It is acknowledged that this sample size is limited, and the results should be interpreted as indicative rather than statistically conclusive. Brightness adjustments were applied to simulate intensity fluctuations, while RGB channel modifications were used to approximate colour-based illumination changes that may occur in practical production-line environments.

Fig. 8 presents the number of correctly classified images under each altered illumination condition. Greater consistency was observed in the classification of uncracked eggs, while cracked-egg detection was found to be more sensitive to substantial brightness variations.

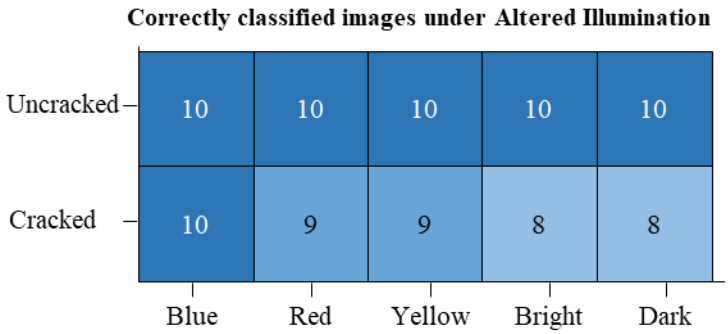


Fig. 8. Number of correctly classified cracked and uncracked egg images under altered illumination conditions.

An average classification accuracy of approximately 94% was achieved across the altered lighting conditions. Although reduced cracked-egg detection was observed under bright and dark conditions, stable overall performance was maintained, with the lowest average accuracy of approximately 90% recorded under these intensity variations. Further validation using a larger and more diverse set of lighting conditions is recommended to confirm robustness in practical egg processing environments where illumination cannot always be fully controlled.

4 Discussion

To assess the model's capabilities and determine whether the stated contributions were achieved, the evaluation results were examined. Feature extraction from the limited dataset was first evaluated through

five-fold cross-validation. Moderate variation across the folds was observed, indicating that model performance was not strongly dependent on a single training–validation split [43]. This consistency suggests that the multi-kernel ensemble design distributes feature learning across independent branches, reducing the risk of overfitting to any single data partition [42]. The ability of the ensemble model to extract more meaningful features than the single-kernel CNNs was then evaluated. More balanced and reliable classification was achieved by the proposed ensemble, with an accuracy of 97.50%. The advantage of the multi-kernel architecture was demonstrated through the integration of 3×3, 5×5, and 7×7 convolutional branches, by which complementary features were extracted at fine, medium, and broad spatial scales [42]. The 7×7 branch alone exhibited class bias despite a high true-positive rate, suggesting that larger kernels capture global structural patterns but lack the fine-grained sensitivity needed for balanced crack classification, a limitation that the ensemble fusion directly addresses.

The proposed ensemble CNN was then compared with established and lightweight benchmark architectures to assess its competitiveness following feature extraction and classification. The highest overall performance was achieved by the ensemble model, with an accuracy of 97.50%, an F1-score of 0.97, and an AUC of 0.93. Strong classification reliability and effective separation between cracked and uncracked eggs were demonstrated by these results. The superior accuracy, F1-score, and AUC can be attributed to the multi-scale feature extraction process. Additional evidence of model reliability under continuous visual input was provided by confidence-level testing. All tested cases were correctly classified, and stable predictions were observed when class-specific features were visually distinct. Lower confidence values were obtained for slightly cracked samples, which may be attributed to weaker visual features and similarities with natural shell texture, dirt, or minor surface imperfections. This finding suggests that slightly cracked eggs represent a fundamentally harder classification subproblem, where crack morphology overlaps with natural shell variation, and may benefit from targeted data augmentation or specialised threshold tuning in future implementations [34]. Further evaluation using a larger number of slightly cracked samples is recommended to validate this capability under more challenging defect conditions.

As deployment under low-computational-power conditions was targeted, computational efficiency was also evaluated. Input resolution was first analysed to determine whether reduced resolution caused excessive performance loss. Although less visual information was retained at lower resolutions, the 32×32 input size produced better classification performance while requiring substantially fewer FLOPs, lower inference time, and reduced RAM utilisation. This outcome may be attributed to the increased feature dimensionality of higher-resolution inputs, through which non-discriminative details such as background texture, lighting variation, shell noise, and minor artefacts may have been learned [27]. This finding implies that for texture-based crack detection on small, constrained objects such as eggs, aggressive resolution reduction acts as an implicit regulariser, forcing the model to focus on structurally discriminative features rather than high-frequency noise [33]. The computational requirements of the proposed model were then evaluated against the benchmark architectures using FLOPs and RAM utilisation. The best overall computational efficiency was achieved by the proposed model, for which only 4.9 million FLOPs and 1.66 GB of RAM were required. These were the lowest values recorded among all evaluated models, by which substantially lower arithmetic complexity and operational memory demand were demonstrated. Despite the multi-branch design, a computationally compact structure was retained because each branch was kept lightweight. This demonstrates that architectural parallelism does not necessarily increase computational cost when individual branches are designed with constrained depth and filter counts, a principle applicable to other resource-constrained visual inspection tasks [42].

Inference latency and throughput were then evaluated, as operational efficiency and economic productivity are strongly influenced by processing time during practical deployment. An inference time of 25.5 ms per

image was achieved by the proposed ensemble CNN, representing the second-lowest latency among the evaluated architectures and differing from the fastest model by only 1.4 ms. However, superior accuracy, F1-score, AUC, FLOP efficiency, and RAM utilisation were achieved by the proposed model. A stronger overall balance between execution speed, classification performance, and computational efficiency was thus demonstrated, with near-fastest inference being attained without the substantial performance limitations observed in the fastest benchmark model. For a production volume of 100,000 eggs, an approximately 100-fold improvement in throughput was estimated, with nearly one hour required by the automated system compared with approximately 100 hours for manual inspection. This throughput advantage is particularly significant in high-volume commercial operations where manual inspection creates production bottlenecks, and suggests that the proposed system could support continuous 24-hour sorting cycles without the fatigue-related accuracy degradation associated with human graders.

Finally, performance under variable imaging conditions was evaluated. An average classification accuracy of approximately 94% was maintained across the tested illumination conditions. This adaptability may have been supported by grayscale preprocessing and the inclusion of visually diverse training images. Grayscale preprocessing likely contributed by eliminating colour-channel sensitivity, ensuring that classification decisions are driven by structural texture rather than illumination-dependent colour information, which is particularly advantageous in environments with variable or inconsistent lighting. However, reduced performance under bright and dark conditions was also observed, indicating that illumination variation remains an important consideration for practical deployment. Greater robustness may be achieved through controlled lighting, improved image acquisition, or additional training samples representing a wider range of illumination conditions.

The proposed framework is considered well suited for integration into existing egg-processing and packaging lines [8], [9]. Through the use of a non-contact, camera-based inspection approach, the need for physical handling during quality assessment is eliminated, thereby reducing mechanical stress on the eggs and minimising the risk of contamination [6]. Only a single grayscale image resized to 32×32 pixels is required as the model input. Therefore, compatibility with standard industrial cameras and embedded vision systems can be achieved without the use of specialised imaging hardware. Classification can be performed upstream of the sorting actuator, allowing uninterrupted conveyor operation while sorting decisions are generated in advance [8]. Through this pipeline-compatible design, integration can be achieved without substantial modifications to existing conveyor infrastructure, thereby reducing implementation costs and deployment complexity [9].

The processing speed of the proposed system has direct implications for production line design. At approximately 36 ms per egg, the system supports a theoretical throughput of approximately 27 eggs per second, or approximately 97,200 eggs per hour. This is consistent with the throughput requirements of commercial egg grading operations, which typically process between 50,000 and 100,000 eggs per hour [8], [9], suggesting that the proposed model can operate within current industrial throughput demands without creating a processing bottleneck. Furthermore, unlike manual inspection, automated processing maintains consistent speed regardless of shift duration, eliminating the throughput variability associated with operator fatigue during extended production runs.

Beyond performance metrics, the proposed system offers sustainability advantages relevant to modern food supply chain operations. Automated crack detection reduces the volume of cracked eggs reaching downstream packaging and distribution stages, thereby minimising food waste at the source [1], [4]. The lightweight architecture requires only 1.66 GB of RAM and 4.9 million FLOPs per classification, making it deployable on low-power embedded hardware without the energy demands associated with large-scale GPU inference servers. This energy efficiency supports broader industry goals of reducing the

environmental impact of food processing operations. In addition, reduced manual labour requirements lower the occupational health risks associated with repetitive inspection tasks, supporting more sustainable working conditions in food manufacturing environments [2].

The impact of classification errors warrants careful consideration in the context of food safety and economic efficiency [7], [11]. False positive classifications, in which intact eggs are incorrectly identified as cracked, result in unnecessary product rejection and associated economic loss [2]. False negative classifications, in which cracked eggs are passed as intact, carry greater risk, as they may allow contaminated or structurally compromised eggs to enter the food supply, with potential implications for consumer safety and regulatory compliance [7], [11]. The ensemble model achieved only four false negatives across 280 test images, representing a false negative rate of 2.86%. While this rate is low, the near-threshold confidence observed for slightly cracked eggs suggests that a conservative decision threshold, set below the default classification boundary, may further reduce false negatives at the cost of a marginal increase in false positives. For safety-critical deployments, threshold calibration should be performed in accordance with food safety regulatory requirements applicable to the target market.

Several limitations of the present study should be acknowledged. The dataset comprised 1,400 images sourced from a single species, limiting the generalisability of the findings to other egg types, sizes, or shell textures [33]. The robustness evaluation under variable illumination was conducted on a limited sample of 20 images per condition, and the results should be interpreted as indicative rather than statistically conclusive. Training and validation loss curves were not retained, and model convergence was assessed through cross-validation and independent test performance alone [43]. Furthermore, the confidence-level evaluation for slightly cracked eggs was based on two video samples, which is insufficient to draw statistically robust conclusions about model behaviour for this defect category. Future work should address these limitations through expanded dataset collection across multiple egg species and production environments, real-world conveyor-based testing, and evaluation under a broader range of controlled and uncontrolled imaging conditions.

5 CONCLUSION AND FUTURE WORK

This study proposed a lightweight, non-destructive ensemble CNN framework for automated egg crack detection, addressing the limitations of manual inspection methods and the computational demands of existing deep learning approaches. The proposed architecture integrates three parallel CNN branches with convolutional kernels of 3×3, 5×5, and 7×7 to extract complementary fine, medium, and broad eggshell features from 32×32 grayscale images. The model achieved an accuracy of 97.50%, an F1-score of 0.97, and an AUC of 0.93, demonstrating strong classification performance on a balanced independent test set. Five-fold cross-validation confirmed model stability under limited dataset conditions, and input resolution analysis established that 32×32 pixels provided the optimal balance between classification performance and computational efficiency.

The proposed ensemble CNN was benchmarked against six architectures, including purpose-built lightweight models MobileNetV2 and EfficientNetB0, as well as established deep CNNs GoogLeNet, ResNet50, VGG16, and VGG19, all trained from scratch under identical conditions. The proposed model achieved the best overall balance between classification accuracy, computational efficiency, and deployment suitability, requiring only 4.9 million FLOPs, 1.66 GB of RAM, and 25.5 ms per image for inference. Robustness evaluations under variable illumination conditions demonstrated reliable performance, with an average classification accuracy of approximately 94% maintained across tested conditions. Productivity analysis estimated a nearly 100-fold reduction in inspection time relative to manual

grading, supporting high-throughput deployment at industrial scale. These results confirm that the proposed framework provides an accurate, computationally efficient, and practically deployable solution for non-destructive egg quality inspection in automated sorting and food packaging applications.

Future work should focus on expanding the dataset to include more diverse egg species, shell textures, lighting conditions, camera angles, and subtle defect patterns to improve generalisation across production environments. Architectural improvements may include attention-based feature fusion, self-supervised pre-training, and systematic optimisation of kernel combinations to further enhance classification reliability for challenging defect categories such as slightly cracked eggs. Extension of the framework to multi-class egg defect detection, encompassing cracks, stains, and shell deformities, would broaden its practical applicability. Real-time validation on conveyor-based production lines should also be conducted to assess the scalability, reliability, and operational performance of the proposed system under true industrial conditions.

FUNDING DECLARATION

No financial support was received for the conduct of this research or the preparation of this manuscript.

AUTHOR CONTRIBUTION STATEMENT

T.R. conducted the dataset preparation, model development, experimental evaluation, prepared the figures and tables, and wrote the original manuscript. N.W. supervised the technical implementation, contributed to the methodology design, and provided critical review of the manuscript. I.K. conceptualised the study, defined the research framework and evaluation criteria, led manuscript writing and revision, and provided overall supervision. All authors reviewed and approved the final manuscript.

DATA AVAILABILITY

The processed dataset, model outputs, and experimental results generated during the study are available from the corresponding author upon request.

The custom model implementation code developed for this study is available from the corresponding author upon reasonable request. The model was implemented using TensorFlow/Keras and trained on Google Colab using an NVIDIA Tesla T4 GPU.

CONFLICT OF INTEREST

The authors declare no competing interests.

CLINICAL TRIAL NUMBER

Clinical trial number: Not applicable.

CONSENT TO PUBLISH

Consent to Publish declaration: Not applicable.

CONSENT TO PARTICIPATE

Consent to Participate declaration: Not applicable.

ETHICS DECLARATION

Ethics declaration: not applicable. This study did not involve human participants, animal subjects, or identifiable personal data. All data used comprised publicly available egg image datasets.

6 REFERENCES

[1] M. Afrrin, B. Rajasekaran, R. S. Nandhini, and P. Karthik, "Valorization of Eggshell Waste For Sustainable Food Packaging: A Review," *Food Bioprocess Technol.*, vol. 18, no. 12, pp. 10209–10228, Dec. 2025, doi: 10.1007/s11947-025-04042-x.

[2] A. Seydim and P. Dawson, "Packaging effects on shell egg breakage rates during simulated transportation," *Poult. Sci.*, vol. 78, no. 1, pp. 148–151, Jan. 1999, doi: 10.1093/ps/78.1.148.

[3] R. Wang *et al.*, "Dietary survey of poultry egg intake among residents in Kunming city, southwest China," *Front. Nutr.*, vol. 11, p. 1314924, Mar. 2024, doi: 10.3389/fnut.2024.1314924.

[4] K. Gunjan, R. P. Pandey, Himanshu, R. Mukherjee, and C.-M. Chang, "Navigating the global egg shortage: a comprehensive study of interconnected challenges," Mar. 16, 2024, *Health Economics*. doi: 10.1101/2024.03.13.24304250.

[5] A. Nasiri, M. Omid, and A. Taheri-Garavand, "An automatic sorting system for unwashed eggs using deep learning," *J. Food Eng.*, vol. 283, p. 110036, Oct. 2020, doi: 10.1016/j.jfoodeng.2020.110036.

[6] X. Yang, R. B. Bist, S. Subedi, and L. Chai, "A Computer Vision-Based Automatic System for Egg Grading and Defect Detection," *Animals*, vol. 13, no. 14, p. 2354, Jul. 2023, doi: 10.3390/ani13142354.

[7] R. S. Vargas Cruz, L. C. Ruiz Salvador, and M. C. Navas Lema, "Merging Manual and Automated Egg Candling: A Safety and Social Solution," *Enfoque UTE*, vol. 9, no. 2, pp. 70–76, Jun. 2018, doi: 10.29019/enfoqueute.v9n2.292.

[8] A. N. A. Thohari, T. R. Yudantoro, S. Handoko, A. K. Nuzli, A. Yahya, and P. Raharjo, "Design and Implement a Smart Conveyor System for Sorting Eggs Based On Quality and Weight," in *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (iCAST-ES 2023)*, vol. 230, M. U. H. Al Rasyid and M. R. Mufid, Eds., in Advances in Engineering Research, vol. 230. , Dordrecht: Atlantis Press International BV, 2024, pp. 1078–1089. doi: 10.2991/978-94-6463-364-1_98.

[9] H. K. Hernando, J. C. Rabanera, J. G. Bravo, M. C. Echanique, A. G. Laureano, and V. B. Solaina, "Design and Evaluation of an Egg Sorting Machine," Nov. 08, 2024, *In Review*. doi: 10.21203/rs.3.rs-5410876/v1.

[10] R. Karoui, B. Kemps, F. Bamelis, B. De Ketelaere, E. Decuypere, and J. De Baerdemaeker, "Methods to evaluate egg freshness in research and industry: A review," *Eur. Food Res. Technol.*, vol. 222, no. 5–6, pp. 727–732, Mar. 2006, doi: 10.1007/s00217-005-0145-4.

[11] E. M. Atwa *et al.*, “Advances in Emerging Non-Destructive Technologies for Detecting Raw Egg Freshness: A Comprehensive Review,” *Foods*, vol. 13, no. 22, p. 3563, Nov. 2024, doi: 10.3390/foods13223563.
[12] W. E. Hauver, *Egg Grading Manual*. U.S. Department of Agriculture, Agricultural Marketing Service, 1961.
[13] Y. S. Taspinar and I. Cinar, “Detection of Broken Eggs with Deep CNNs and Performance Analysis of Models,” in *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Dortmund, Germany: IEEE, Sep. 2023, pp. 724–728. doi: 10.1109/IDAACS58523.2023.10348756.
[14] Kurt C Lawrence, Deana R Jones, Gerald W Heitschmidt, and Bosoon Park, “Modified Pressure System for Imaging Egg Cracks,” in *2008 Providence, Rhode Island, June 29 - July 2, 2008*, American Society of Agricultural and Biological Engineers, 2008. doi: 10.13031/2013.24818.
[15] Q. Wang, Z. Yang, C. Liu, R. Sun, and S. Yue, “Research Progress on Non-Destructive Testing Technology and Equipment for Poultry Eggshell Quality,” *Foods*, vol. 14, no. 13, p. 2223, Jun. 2025, doi: 10.3390/foods14132223.
[16] E. H. Rachmawanto *et al.*, “Eggs Classification based on Egg Shell Image using K-Nearest Neighbors Classifier,” in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia: IEEE, Sep. 2020, pp. 50–54. doi: 10.1109/iSemantic50169.2020.9234305.
[17] I. Mahesa, A. G. Putrada, and M. Abdurohman, “Egg Quality Detection System Using Fuzzy Logic Method,” *Kinet. Game Technol. Inf. Syst. Comput. Netw. Comput. Electron. Control*, pp. 207–216, Jul. 2019, doi: 10.22219/kinetik.v4i3.839.
[18] B. Narin, S. Buntan, N. Chumuang, and M. Ketcham, “Crack on Eggshell Detection System Based on Image Processing Technique,” in *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, Bangkok: IEEE, Sep. 2018, pp. 1–6. doi: 10.1109/ISCIT.2018.8587980.
[19] B. Purahong, V. Chaowalittawin, W. Krungseanmuang, P. Sathaporn, T. Anuwongpinit, and A. Lasakul, “Crack Detection of Eggshell using Image Processing and Computer Vision,” *J. Phys. Conf. Ser.*, vol. 2261, no. 1, p. 012021, Jun. 2022, doi: 10.1088/1742-6596/2261/1/012021.
[20] İ. Yabanova, Z. Balcı, M. Yumurtacı, and T. Ünler, “Autoencoder-Based Eggshell Crack Detection Using Acoustic Signal,” *J. Food Process Eng.*, vol. 47, no. 11, p. e14780, Nov. 2024, doi: 10.1111/jfpe.14780.
[21] B. Yildiz, K. Eskioğlu, D. Özdemir, and M. Akşit, “Predicting Quail Egg Quality Using Machine Learning Algorithms,” *Braz. J. Poult. Sci.*, 2025, doi: https://doi.org/10.1590/1806-9061-2024-2037.
[22] C. Haoran, H. Chuchu, J. Minlan, and L. Xiaoxiao, “Egg Crack Detection Based on Support Vector Machine,” in *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, Sanya, China: IEEE, Dec. 2020, pp. 80–83. doi: 10.1109/ICHCI51889.2020.00025.
[23] C. Shi *et al.*, “Nondestructive Detection of Microcracks in Poultry Eggs Based on the Electrical Characteristics Model,” *Agriculture*, vol. 12, no. 8, p. 1137, Jul. 2022, doi: 10.3390/agriculture12081137.
[24] S. Saifullah, “Comparative Analysis of Long Short-Term Memory and Gated Recurrent Unit Models for Chicken Egg Fertility Classification Using Deep Learning,” in *ASEC 2024*, MDPI, Feb. 2025, p. 7. doi: 10.3390/engproc2025087007.
[25] H. Alaskar and T. Saba, “Machine Learning and Deep Learning: A Comparative Review,” in *Proceedings of Integrated Intelligence Enable Networks and Computing*, K. K. Singh Mer, V. B. Semwal, V. Bijalwan, and R. G. Crespo, Eds., in Algorithms for Intelligent Systems. , Singapore: Springer Singapore, 2021, pp. 143–150. doi: 10.1007/978-981-33-6307-6_15.

[26] P. Setiaji, K. Adi, and B. Surarso, “Development of Classification Method for Determining Chicken Egg Quality Using GLCM-CNN Method,” *Ingénierie Systèmes Inf.*, vol. 29, no. 2, pp. 397–407, Apr. 2024, doi: 10.18280/isi.290201.
[27] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, “Review on Convolutional Neural Networks (CNN) in vegetation remote sensing,” *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 24–49, Mar. 2021, doi: 10.1016/j.isprsjprs.2020.12.010.
[28] B. Botta, S. S. R. Gattam, and A. K. Datta, “Eggshell crack detection using deep convolutional neural networks,” *J. Food Eng.*, vol. 315, p. 110798, Feb. 2022, doi: 10.1016/j.jfoodeng.2021.110798.
[29] A. P. J. Putra, F. Dewanta, H. M. Jumhur, and A. Arianto, “Evaluation of Hybrid Deep Learning Models for Forecasting Chicken Egg Prices in Indonesia,” in *2025 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, Surakarta, Indonesia: IEEE, Jun. 2025, pp. 1–6. doi: 10.1109/SIML65326.2025.11081101.
[30] T. H. Kim, J. H. Kim, J. Y. Kim, and S. E. Oh, “Egg Freshness Prediction Model Using Real-Time Cold Chain Storage Condition Based on Transfer Learning,” *Foods*, vol. 11, no. 19, p. 3082, Oct. 2022, doi: 10.3390/foods11193082.
[31] J.-H. So, S. Y. Joe, S. H. Hwang, S. J. Hong, and S. H. Lee, “Current advances in detection of abnormal egg: a review,” *J. Anim. Sci. Technol.*, vol. 64, no. 5, pp. 813–829, Sep. 2022, doi: 10.5187/jast.2022.e56.
[32] B. Botta and A. K. Datta, “Deep transfer learning-based approach for detection of cracks on eggs,” *J. Food Process Eng.*, vol. 46, no. 11, p. e14425, Nov. 2023, doi: 10.1111/jfpe.14425.
[33] Ms. A. Bansal, Dr. R. Sharma, and Dr. M. Kathuria, “A Systematic Review on Data Scarcity Problem in Deep Learning: Solution and Applications,” *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–29, Jan. 2022, doi: 10.1145/3502287.
[34] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
[35] B. Guanjun, J. Mimi, X. Yi, C. Shibo, and Y. Qinghua, “Cracked egg recognition based on machine vision,” *Comput. Electron. Agric.*, vol. 158, pp. 159–166, Mar. 2019, doi: 10.1016/j.compag.2019.01.005.
[36] GoSeeri, “Egg Dataset Computer Vision Project.” Accessed: Apr. 20, 2025. [Online]. Available: https://universe.roboflow.com/goseeri-dataset/egg-dataset-npkgq
[37] A. K. Kakar, “Egg Image Dataset.” Accessed: Apr. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/abdullahkhanuet22/eggs-images-classification-damaged-or-not
[38] L. Middha, “Crack Segmentation Dataset.” Accessed: Apr. 20, 2025. [Online]. Available: https://www.kaggle.com/datasets/lakshaymiddha/crack-segmentation-dataset
[39] T. Sirikham, “Egg crack detect Computer Vision Project.” Accessed: Apr. 20, 2025. [Online]. Available: https://universe.roboflow.com/thanapat-sirikham/egg-crack-detect
[40] “train egg Computer Vision Project.” Accessed: Apr. 20, 2025. [Online]. Available: https://universe.roboflow.com/egg-classification-kgyv8/train-egg
[41] “Egg Crack Detection Computer Vision Project.” Accessed: Apr. 20, 2025. [Online]. Available: https://universe.roboflow.com/innodata/egg-crack-detection-ajv0f
[42] Y. Guo, X. Jia, B. Zhao, H. Chai, and Y. Huang, “Multifeature extracting CNN with concatenation for image denoising,” *Signal Process. Image Commun.*, vol. 81, p. 115690, Feb. 2020, doi: 10.1016/j.image.2019.115690.
[43] J. D. Rodriguez, A. Perez, and J. A. Lozano, “Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 569–575, Mar. 2010, doi: 10.1109/TPAMI.2009.187.
[44] D. M. W. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” 2020, doi: 10.48550/ARXIV.2010.16061.

[45] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
[46] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” 2019, doi: 10.48550/ARXIV.1905.11946.
[47] M. Turkoglu, “Defective egg detection based on deep features and Bidirectional Long-Short-Term-Memory,” *Comput. Electron. Agric.*, vol. 185, p. 106152, Jun. 2021, doi: 10.1016/j.compag.2021.106152.
[48] T. A. Cengel, B. Gencturk, E. T. Yasin, M. B. Yildiz, I. Cinar, and M. Koklu, “Automating egg damage detection for improved quality control in the food industry using deep learning,” *J. Food Sci.*, vol. 90, no. 1, p. e17553, Jan. 2025, doi: 10.1111/1750-3841.17553.
[49] M. Jiang, P. Wu, and F. Li, “Detecting Dark Spot Eggs Based on CNN GoogLeNet Model,” in *Simulation Tools and Techniques*, vol. 370, H. Song and D. Jiang, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 370. , Cham: Springer International Publishing, 2021, pp. 116–126. doi: 10.1007/978-3-030-72795-6_10.
[50] L. Chen, K. Bo, F. Lee, and Q. Chen, “Advanced Feature Fusion Algorithm Based on Multiple Convolutional Neural Network for Scene Recognition,” *Comput. Model. Eng. Sci.*, vol. 122, no. 2, pp. 505–523, 2020, doi: 10.32604/cmes.2020.08425.
[51] K. Vamsi, B. Swathi, K. Vijay, and S. Kamepalli, “Deep Learning Approaches for Egg Crack Detection: A Comparative Study Using CNN, ResNet, and Xception Models,” in *2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC)*, Davangere, India: IEEE, Oct. 2024, pp. 1–9. doi: 10.1109/ICICEC62498.2024.10808692.
[52] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
[53] X. Du, Y. Sun, Y. Song, W. Chi, L. Dong, and X. Zhao, “Impact of Input Image Resolution on Deep Learning Performance for Side-Scan Sonar Classification: An Accuracy–Efficiency Analysis,” *Remote Sens.*, vol. 17, no. 14, p. 2431, Jul. 2025, doi: 10.3390/rs17142431.
[54] J. Alikhanov *et al.*, “Design and Performance of an Automatic Egg Sorting System Based on Computer Vision,” *TEM J.*, pp. 1319–1325, Nov. 2019, doi: 10.18421/TEM84-31.
[55] W. Tang, J. Hu, and Q. Wang, “High-Throughput Online Visual Detection Method of Cracked Preserved Eggs Based on Deep Learning,” *Appl. Sci.*, vol. 12, no. 3, p. 952, Jan. 2022, doi: 10.3390/app12030952.
[56] Y. Banjarat, W. Anucha, and W. Wiyaratn, “Plant Layout Design for Improvement and Eggs Grading Process,” in *Proceedings of the 2019 International Conference on Management Science and Industrial Engineering*, Phuket Thailand: ACM, May 2019, pp. 19–25. doi: 10.1145/3335550.3335572.