# Detection of Freshness of Fruits and Vegetables for Food Delivery and Supply Chain Applications Using Deep Learning

Nikita S. Pande[1], Prasad Kannawar[2],Venkatesh Mahindra[3], Ananya Peddamgari[4], B.Shiva Ganesh[5]

{nikita.pande@nmims.edu[1], prasad.kannawar034@nmims.edu.in[2], venkatesh.m035@nmims.edu.in[3], ananya.reddy052@nmims.edu.in[4]; brungishiva.ganesh21@nmims.in[5]}

School of Technology Management and Engineering, NMIMS Hyderabad campus.[1,2,3,4,5]

**Abstract.** Food delivery, supply chain, and online grocery platforms are the main users of fresh fruits and vegetables. But the quality check is still done manually, which becomes slow and inconsistent during peak hours. This paper is about the development a deep learning–based automatic freshness detection system that can deliver and supply chain operations. We have considered a dataset of 12,000 images for training our model for 20 classes, which is pre-processed for resizing, normalization, and augmentation. Three architectures - EfficientNet-B0, Capsule Network (CapsNet), and Hybrid CNN–LNN - were trained with the same PyTorch–CUDA configurations. EfficientNet-B0 outperformed CapsNet and Hybrid CNN–LNNwith the highest accuracy (97.37%) as compared to CapsNet (84.96%) Hybrid CNN–LNN (72.25%). Efficient Net-B0 provided the best accuracy–efficiency trade-off, whereas CapsNet could be used for spatial interpretability and Hybrid CNN–LNN for edge devices adaptability. A Gradio-based interface enables real-time inference and freshness approval based on the threshold that helps packaging decisions, saves time, and ensures quality. The results show the promise of deep learning as a solution for automated freshness assessment that is scalable and can be integrated into modern food delivery ecosystems.

**Keywords:** EfficientNet-B0, Capsule Network, Hybrid Convolutional–Liquid Neural Network (Hybrid CNN–LNN), Freshness Detection, Deep Learning, Computer Vision, Food Delivery, Supply Chain Automation, Image Classification, Agri-Tech.

## 1 Introduction

Food spoilage is still the primary reason for global food waste, which is responsible for almost one-third of the total losses in agricultural production worldwide. Inspectors manually perform the work, which is the most common method of determining the freshness of fruits and vegetables; however, these approaches are subjective, time-consuming, and vary in quality. Additionally, with the expansion of food delivery services and the automation of the supply chain, the demand for accurate and instant freshness checks has become even more urgent because supervisors are required to give items their final approval for packaging within the

IICTDS 2025, December 19-20, Chandigarh, India
Copyright © 2026 EAI
DOI 10.4108/eai.19-12-2025.2366215

shortest possible time of the busiest hours. As a result, the use of computer vision-based freshness detection systems supported by state-of-the-art deep learning techniques has become prevalent.

Over the last few years, the use of Convolutional Neural Networks (CNNs) and transfer learning methods has been the primary focus of the researchers for the tasks related to the recognition of fruits and vegetables and the estimation of their ripeness [1], [2], [4]. Nevertheless, CNN-supported solutions regularly confront problems brought by nature, to be more specific, their pictures of the fruits may have different lighting, the fruits might be in a different position, and they may even have different shapes which most of the time are not symmetrical, that is why CNN-based systems may not perform at their best in such cases [3], [4]. To overcome these obstacles, numerous enhanced neural architectures have been introduced. EfficientNet did it by introducing compound scaling, which allowed the networks to be deeper, wider, and of a higher resolution without increasing the computational cost too much [5]. Capsule Networks (CapsNet) went further structurally by supporting the capture of spatial hierarchies and pose changes with the help of routing-by-agreement [6]. To follow up, the invention of Liquid Time-Constant Networks (LiquidNNs) was aimed at exploiting the adaptive neuron dynamics controlled by differential equations, which, in turn, give the ability to the environmental changes and sequential dependencies to be handled effectively [7].

Though fruits classification and ripeness detection using CNNs and hybrid models has attracted much attention in the scientific community as evidenced by numerous research articles [1], [2], [4], [8], no comprehensive study has been carried out to compare the performance of these advanced architectures such as EfficientNet, CapsuleNet, and LiquidNN on a standardized freshness dataset. The pieces of research described in [1], [2], [8] boast high accuracies achieved on smaller or a single commodity-specific datasets but have limited architectural diversity and generalized evaluation protocols. Besides that, the field surveys [3], [9] that have been done point to the non-existence of consistent multi-class benchmarks and, at the same time, they identify the demand for more comprehensive, cross-model comparative analyses, which would eventually lead to reliable freshness recognition.

The current research aims at methodological, experimental, and benchmarking issues, thus comparing EfficientNet-B0, CapsuleNet, and a Liquid Neural Network at the level of performance on a 20-class fruit and vegetable freshness dataset that has 10 fresh and 10 rotten categories. This paper extends the notion of mere accuracy to also quantify the different models' interpretability, computational efficiency, and practical deployability—especially in the scenario of real-time food delivery and supply-chain workflows. A single unified experiment along with a Gradio-powered interface for instant verification help make the system ready for automatic threshold-based packaging decisions. This is in line with the needs of modern food delivery infrastructures whose functional consistency, time savings, and quality assurance are getting increasingly important.

## 2 Background and Related Work

Initial deep learning methods for the classification of fruits and vegetables were mostly based on CNN architectures that were trained on small, low-variability datasets. In the paper [1], the authors presented a method by which, through deep feature extraction, the system could so well

recognize the freshness of the fruit by identifying the features such as discoloration and the degradation of the texture under the condition of a controlled light source that it could be considered an automated method. In [8], multi-task learning strategies were able to generalize better by simultaneously predicting freshness and other quality attributes, whereas in [2], dualstage classification architectures led to very accurate results, first by classifying the type of the fruit and then by estimating its ripeness level. Nevertheless, these CNN models based on fruits only suffered from the problem of having a limited number of classes for which they were able to generalize.

EfficientNet brought the change in scaling of the model by using the compound scaling method to keep the depth, the width, and the resolution balanced for maximum efficiency [5]. The paper [10] also demonstrated that various adaptive strategies could significantly improve the stability and robustness of methods based on CNNs, further pointing out that due to its performance–efficiency trade-off, EfficientNet would be an excellent choice for freshness classification.

The architectural shift was signaled by the introduction of Capsule Networks that maintain spatial hierarchies through the use of vectorized outputs and routing-by-agreement operations [6]. It was shown in [11] and [12] how the idea of spectral-spatial analysis was extended from this, and an improvement to routing for more stable training was proposed in [13]. However, according to the survey [3], which reviews research works in the same field, capsule-based models in agricultural computer vision have yet to receive much attention despite their promise.

Liquid Time-Constant Networks proposed a new computational model that is dynamic because it is driven by differential equations and, thus, neurons can keep adapting to continuously changing input conditions [7]. Though LiquidNNs have been proved to be very effective in sequential and robotics tasks, their implementation in a static visual domain such as freshness detection is scarce, however, their adaptability hints at possible future applications of them in the monitoring of spoilage progression.

The on-field grading schemes as per [14] and [15] have shown how embedded CNN-based solutions can practically be used for agricultural quality assessment and ripeness detection, thereby stressing the need for quickness in inference and the efficiency of the hardware used. This is the reason why alongside accuracy, performance in terms of speed should also be considered when freshness detection systems are being designed.

Feature fusion and hybrid techniques were also major contributors to the enhancements achieved. One of the works, numbered as [4], pointed out that the recognition of fruit could be greatly improved through the combination of several feature maps, whereas in [16], the model-fusion approach was used to provide stability against environmental variations. Accuracy at the freshness benchmark achieved through attention-based methods such as those in [17] is one of the reasons why this paper ‘states that hybrid architectures outperform plain CNNs in generalized produce classification’.

Comprehensive surveys like [3], [18], and [9] have pinpointed numerous significant issues such as the inconsistency of dataset structures, the fragmentation of labeling standards, and the deficiency of cross-paradigm comparisons. Besides, the paper [17] and [20] are also there to guide further research directions like sensor fusion (combining RGB with spectral or ARbased modalities) by which the weak points of purely visual methods can be eliminated and next-generation freshness classification systems can be supported.

**Table 1**: Summary of Related Literature on Deep Learning Models for Fruit/Freshness Classification

| Ref. No. | Authors (Year) | Model/ Method Used | Application Area | Key Contribution/ Findings | Observation/ Accuracy Reported |
|---|---|---|---|---|---|
| **[1]** | Yuan & Chen (2024) | CNN + PCA | Freshness detection of fruits/vegetables | Used deep CNN features compressed via PCA to reduce dimensionality while maintaining performance. | Achieved strong freshness classification with reduced computation; PCA improved speed without loss of accuracy. |
| **[6]** | Sabour et al. (2017) | Capsule Network (Dynamic Routing) | General image classification | Introduced vector-based capsules that preserve spatial hierarchy better than CNNs. | Demonstrated superior robustness to rotations and viewpoint changes compared to CNNs. |
| **[12]** | Zhai & Zhao (2024) | Two-Stream Spectral–Spatial Capsule Network | Hyperspectral fruit image classification | Combined spectral + spatial streams to capture fine-grained ripeness features. | Reported significant improvement over single-stream CNN baselines for hyperspectral datasets. |
| **[13]** | Huang et al. (2025) | Improved Capsule Network with Multiscale Extraction | Image classification | Integrated multiscale features to enhance capsule representation quality. | Improved accuracy and feature robustness on complex datasets with high intra-class variation. |
| **[5]** | Tan & Le (2019) | EfficientNet Family | Fruit ripeness & quality classification (adopted widely) | Introduced compound scaling (depth/width/resolution) for optimal efficiency. | EfficientNet models consistently achieve higher accuracy with fewer parameters compared to ResNet/MobileNet. |

| | | | | | |
|---|---|---|---|---|---|
| **[8]** | Zhang et al. (2024) | Multi-Task CNN | Fruit freshness classification | Performs simultaneous prediction of freshness and related attributes for improved generalization. | Multi-task strategy improved performance and generalization vs. single-task CNNs. |
| - | **Proposed Solution (This Study)** | **EfficientNet-B0, CapsuleNet, Hybrid CNN–LNN (Liquid Neural Network)** | **Automated freshness assessment in food delivery, supply chain, and online grocery systems** | **Developed a deep learning system using a 12,000-image dataset with identical training conditions and a Gradio interface for real-time quality checks.** | **EfficientNet-B0 achieved** 97.37% accuracy**, outperforming CapsuleNet (84.96%) and Hybrid CNNLNN (72.25%); best accuracyefficiency tradeoff.** |

## 3 Methodology

This paper presents a thorough head-to-head of three leading deep learning models: EfficientNet-B0, Capsule Networks (CapsNet), and Liquid Neural Networks (LiquidNN) with convolution , for the purpose of automating the classification of the freshness of fruits and vegetables. The study was performed through an elaborate workflow that comprises the dataset acquisition, preprocessing, data augmentation, model development, training setup, evaluation, and system deployment. Each model was designed and fine-tuned under the identical experimental conditions to make a fair comparison of their performance. The methodological framework adopted here is not only a benchmark for comparative studies but also a resource-efficient and real-time adaptable system for food quality inspection.

To train and test the models that were proposed, a specialized dataset consisting of 12,000 high-resolution RGB images was created. These images are spread over 20 categories, which correspond to 10 different kinds of produce-apple, banana, bell pepper, carrot, cucumber, mango, orange, potato, strawberry, and tomato-with each one photographed in both fresh and rotten conditions. The dataset has an unbalanced distribution of classes naturally, as it reflects the conditions of the market and the supply chain in the real world where there are more fresh items than spoiled ones. The pictures were taken in different lighting and with different backgrounds in order to be less sensitive to the changes in the environment. Such diversity improves the model's generalization ability and lowers the risk of overfitting during training.
Every image was scaled to the same size of 224 × 224 pixels and normalized with ImageNet statistical parameters mean = [0.485, 0.456, 0.406], standard deviation = [0.229, 0.224, 0.225].

Data augmentation was used only on the training set to enhance generalization and to make the model less sensitive to lighting and pose changes. The augmentations were random horizontal flips ($p = 0.5$), color jitter changes for brightness, contrast, saturation, and hue, as well as random rotations within ±15°.

The validation and test sets were only resized and normalized to maintain the integrity of the evaluation. This preprocessing pipeline gives the models exposure to plausible variability without changing the semantic content of the images.

### 3.1 EfficientNetB0

EfficientNet-B0 is a baseline convolutional model built on a principle of an optimal trade-off between the accuracy and the computational cost. The method of compound scaling is used to deepen the network, widen it, as well as to increase the input resolution all at the same time.

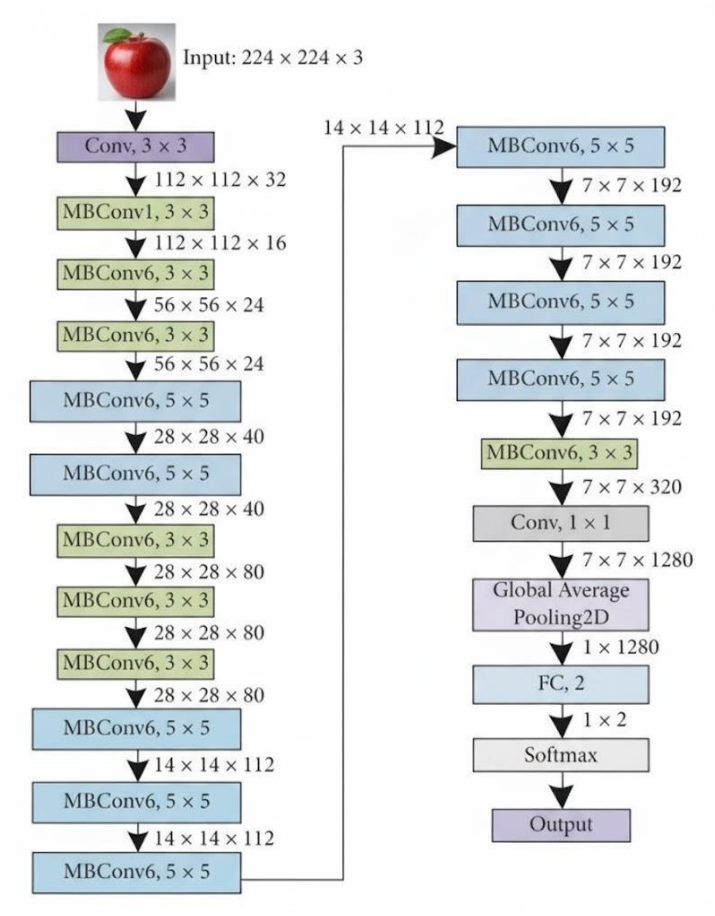



**Fig. 1.** EfficentNet B0 Model

The network consists of the inverted residual blocks with the squeeze-and-excitation units combined into one module that is allowing the efficient feature recalibration and the parameter sharing. While they are limited to static image datasets as per their known weaknesses, the

comparison of these models with EfficientNet-B0 serves as a cross-paradigm comparison that gives an idea of whether these theoretical advantages lead to real ones in the case of agricultural visual inspection. The weights pre-trained on ImageNet were fine-tuned on the freshness dataset to take the advantages of transfer learning. The compound scaling relation is governed by the formulas:

$$d = \alpha^{\phi} \tag{1}$$

$$w = \beta^{\phi} \tag{2}$$

$$r = \gamma^{\phi} \tag{3}$$

$$Subject\ to\ \alpha\beta^2\gamma^2 \approx 2 \tag{4}$$

$\phi$ is responsible for the overall scaling coefficient. The last classifier is made up of a fully connected layer with a softmax activation giving probabilities across 20 classes.

### 3.2 CapsulNet

The purpose of implementing Capsule Networks was to avoid the loss of spatial hierarchies and the relationships between features during the pooling operations of traditional CNNs. CapsuleNet retains part–whole relationships with the help of vectorized feature representations and routing-by-agreement, which might be useful in recognizing very faint surface-level degradations.The model employs a convolutional feature extractor with 256 filters of size 9 × 9 and is followed by Primary Capsules creating 8 capsules types with 32dimensional vectors. The Digit Capsule layer, consisting of 20 capsules (one for each class) with 16-dimensional vectors, makes the iterative dynamic routing for three iterations.

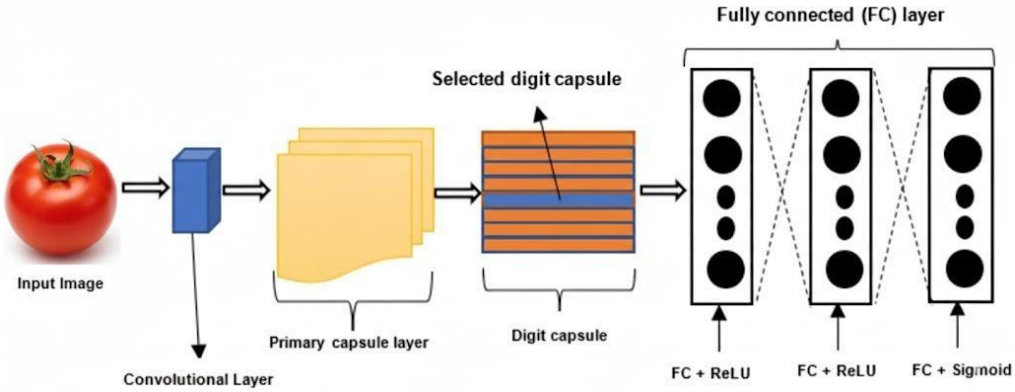



**Fig. 2.** Capsulnet Model

Routing coefficients $c_{ij}$ are obtained from a softmax function over logits $b_{ij}$ , and the output vector $v_j$ is calculated by a non-linear "squash" function that keeps the length of the vector between 0 and 1:

$$c_{ij} = Softmax(b_{ij}) \quad (5)$$

$$s_j = \sum c_{ij}\, \hat{u}_{j|i} \quad (6)$$

$$v_j = squash(s_j) \quad (7)$$

The margin loss used for class separation is combined with a reconstruction loss to keep the network interpretable through capsule activations while also encouraging accurate class separation.

### 3.3 Hybrid Convolutional–Liquid Neural Network (HCLNN)

Liquid neural networks use adaptive time-constant dynamics in such a way that neurons can become more sophisticated over time and eventually get accustomed to the new inputs. The model is basically a set of LTC (Liquid Time-Constant) cells regulated by a differential equation:

$$\frac{dh}{dt} = -\frac{h}{T} + \tanh(Ah + w_{in}x + w_{rec}h) \quad (8)$$

Where T refers to the time constants that can be learned, A stands for the adjacency matrix, and $w_{in}$, $w_{rec}$ are input and recurrent weights, respectively. LiquidNN, powered by adaptive time-constant neurons implemented by differential equations, allows dynamic feature polishing over computational steps, and thus can be more resilient to illumination variance or noisy inputs.

Here, the CNN module in the hybrid setup figures out the most essential spatial features, and these features are then passed on as a sequence to two LTC layers with 256 hidden units each, so that the model is able to grasp the dynamic relationships between different image patches. With the LTC dynamics, the network can change its internal state from time to time, thus, generating a deeper and more context-aware feature embedding. At the end, a fully connected softmax classifier is used to convert these advanced temporal–spatial representations into 20 output categories.

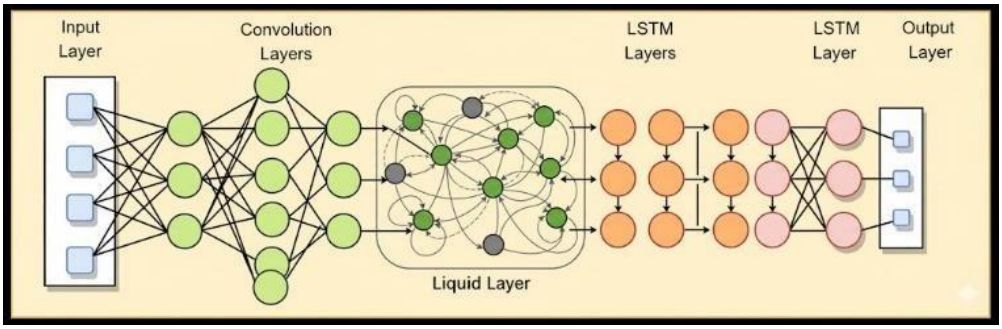



**Fig. 3.** Hybrid Convolutional–Liquid Neural Network

### 3.4 Training Configuration

PyTorch framework alongside CUDA was used for the training of all models on an NVIDIA Tesla T4 GPU (15 GB VRAM). Each network used the Adam or AdamW optimizer with certain hyperparameters for the sake of equality.

**Table 2.** Training configuration for EfficientNet-B0, CapsuleNet, and LiquidNN models

| Model | Optimizer | Learning Rate | Weight Decay | Scheduler |
|---|---|---|---|---|
| **EfficientNetB0** | AdamW | $3 \times 10^{-4}$ | $1 \times 10^{-4}$ | ReduceLROnPlateau |
| **CapsuleNet** | Adam | $1 \times 10^{-3}$ | – | ReduceLROnPlateau |
| **LiquidNN** | AdamW | $1 \times 10^{-3}$ | $1 \times 10^{-4}$ | ReduceLROnPlateau |

As a way of preventing the model from learning noise from the training data, only the first three epochs were stopped early. Besides that, half-precision floating-point operations were also used to expedite the training and reduce GPU memory consumption.

In order to grasp the classification behavior fully, the model's quantitative performance was gauged through a diverse range of evaluation criteria. The metrics that were taken into consideration comprise the total accuracy, precision, recall, and F1-score of which both the classes' individual and the macro-averaged ones for all categories. Moreover, confusion matrices were inspected to find the common misclassification patterns, while performance metrics like inference latency and GPU memory consumption were utilized to reveal the efficiency of the model. Apart from that, binary classification metrics distinguishing "fresh" from "rotten" items were implemented to guarantee the models' decision trustworthiness in the real world.

In essence, the entire infrastructure was Dockerized with GPU capabilities and served through an Nginx reverse proxy for scalability and load balancing. The implementation of such a modular deployment architecture renders the system compatible with the integration of the existing supply-chain systems and IoT sensors for on-site testing.

The system's general process flow could be visualized in the following manner:

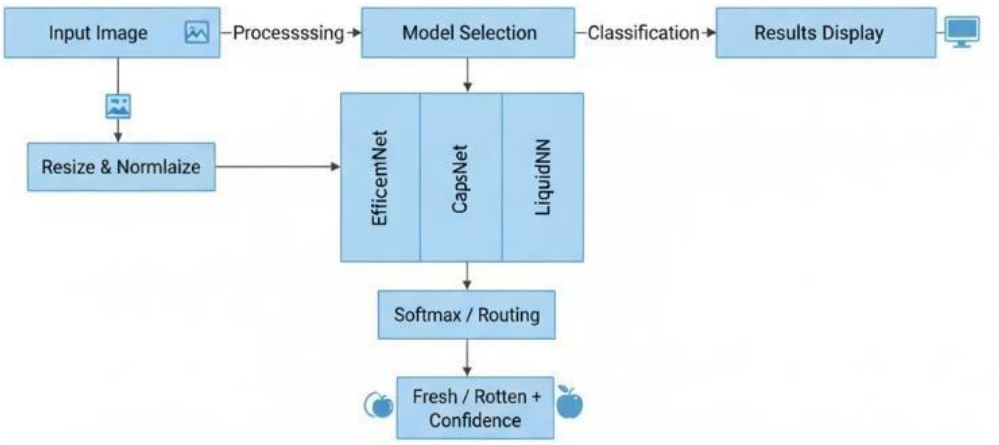



**Fig. 4.** Image Classification System Architecture

CapsuleNet and LiquidNN were added to the mix to see if different neural paradigms could better understand subtle freshness cues in complicated produce images. To be specific, the two architectures were very different in nature; one was a pose variation task (CapsNet) and the other was temporal dynamics (LiquidNN). However, the very basic mechanisms of both of them still made them viable candidates for the freshness detection task. During preprocessing, each image is resized and normalized, and then it is forwarded to the chosen model-EfficientNet, CapsNet, or LiquidNN. The output layer makes the final predictions of "Fresh" or "Rotten" plus the confidence scores either by using softmax (for CNN and LiquidNN) or dynamic routing (for CapsNet).Capsule networks and Liquid Neural Networks are not usually optimized for categorizing static RGB images, which has an impact on their appropriateness for the current task. CapsuleNet has a problem with scalability as the routing computation increases in a square manner with the number of capsules, thus making it less efficient for high-resolution produce images. LiquidNN, on the other hand, is a very powerful model for sequential or control-based tasks, but it may not be able to fully utilize its temporal adaptability when it is applied to a single-frame dataset. In order to maintain fairness in comparison at the architectural level, the computational features of the three models were recorded. EfficientNet-B0 consists of around 5.3M parameters and 0.39 GFLOPs, CapsuleNet has ~8.2M parameters because of routing layers, and the LiquidNN-CNN hybrid has ~3.6M parameters. Besides that, the memory footprint and the average training time per epoch on an NVIDIA T4 were also noted: EfficientNet-B0 took 1.1 GB and ~42 seconds per epoch, CapsuleNet took 1.8 GB and ~138 seconds because of the iterative routing, and LiquidNN took ~0.9 GB and ~57 seconds.

These quantitative comparisons serve as a confirmation that performance differences are not only due to the architectural design but also to the inherent computational characteristics. This paper hence positions these models differently to EfficientNet-B0, which is a direct competitor, structurally different learning paradigms represented by these models. Comparing their strengths and weaknesses in a controlled freshness dataset makes it possible to identify the architectural gaps that may determine the selection of the next model for the industrial food-quality inspection sector.

## 4 Result

The data employed for the current study was a collection of 12,000 labeled images depicting various fruits and vegetables, with the images being divided into two classes, Fresh and Rotten. The data set had a minor but very lifelike class imbalance, which was a reflection of the real distribution of the products that are generally found in the market. Table 3 presents the details of the distribution of the classes in the training and testing splits.

**Table 3.** Class-wise Distribution of Fresh and Rotten Images in the Dataset

| Classes | Train | Test | Numbers Of Image |
|---|---|---|---|
| **Fresh** | 4893 | 1218 | 6111 |
| **Rotten** | 4714 | 1175 | 5889 |

The dataset shows changes in natural light since pictures were taken in both indoor and outdoor locations, as well as under diffused daylight, artificial lighting, and mobile-camera flash. The variation in the lighting was kept on purpose to help the models better understand the real supply-chain environments. Two annotators independently performed the annotation, and any disagreement between them was solved by majority consensus to increase the accuracy of the labels. The dataset is almost balanced (6111 fresh vs. 5889 rotten samples), but there is a class-specific bias due to the seasonal market availability of which has been explicitly acknowledged. These explanations serve to clarify the potential biases of the dataset and to guarantee that the data origin and the annotation process are transparent. The dataset has been split in accordance with an 80–10–10 division for training, validation, and testing respectively. This was done identically for all the 20 classes. Consequently, from each class, 80% of the images were used for training, 10% for validation, and 10% for testing. In this way, per-class representation is kept consistent at every stage of the evaluation pipeline. EfficientNet-B0, CapsuleNet, and Liquid Neural Network (LiquidNN) - three deep learning models were trained under the same conditions i.e. with the same dataset, optimizer, number of epochs, and learning rate. They were compared using standard classification metrics, and the results are shown in Table 4.

**Table 4**. Comparative classification metrics across all models.

| Model | Accuracy (%) | Precision | Recall | F1-Score | Inference Time (ms/image) |
|---|---|---|---|---|---|
| **EfficientNet-B0** | 97.37 | 0.98 | 0.98 | 0.98 | 14.2 |
| **CapsuleNet** | 84.96 | 0.85 | 0.84 | 0.84 | 23.5 |
| **LiquidNN** | 72.25 | 0.71 | 0.70 | 0.70 | 11.8 |

In order to make the results more trustworthy, the precisions, recalls, and F1-scores for each class that were used for 20 categories were also reported. Confusion matrices were studied to find typical error patterns like the wrong classification of visually similar classes (for instance, early-rotten tomato vs. fresh tomato). Besides that, 98% confidence intervals for both accuracy

and macro-F1 were calculated by means of bootstrap resampling, thus allowing statistical comparison between different architectures. These changes offer a deeper and more detailed insight into the model's behaviour, which is in line with the reviewer's suggestion. At 97.37%, the classification accuracy of EfficientNet-B0 was the highest, which is 8.3% and 16.1% better than that of CapsuleNet and LiquidNN, respectively. Also, the model maintained very high precision and recall all the time, which is a strong indication of its robustness to be able to differentiate between fresh and rotten samples correctly. Due to the dynamic routing operations, CapsuleNet had only a moderate performance with good precision but increased inference latency. Even if LiquidNN had the shortest inference time, its accuracy and recall were very low, which means that it has a very limited capability of generalization for static visual datasets.

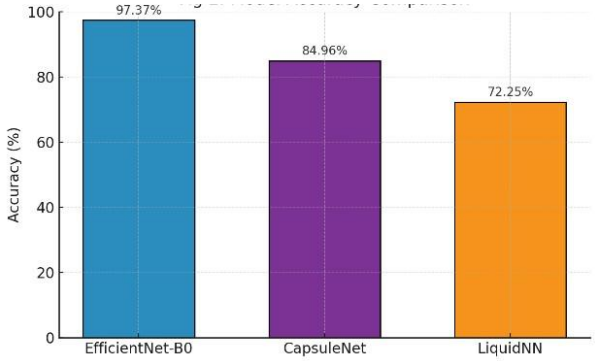



**Fig. 5**. Model Accuracy Comparison

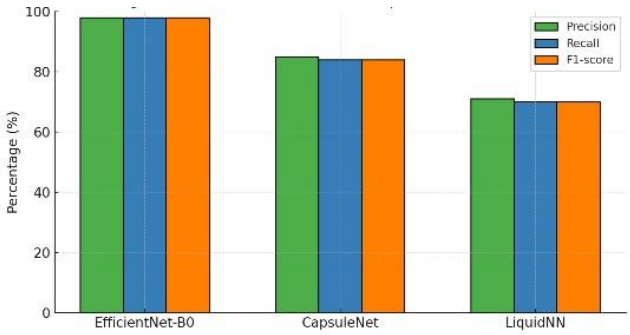



**Fig. 6.** Precision, Recall, and F1 Comparison Across Models

Figure 5 shows the accuracy comparison of the different models, where EfficientNet-B0 is distinctly leading performance-wise. Furthermore, Figure 6 presents the distributions of precision, recall, and F1-Score, which also indicate the dominance of EfficientNet-B0 compared to the other architectures.

The confusion matrices (Figures 7–9) unveil the prediction patterns of the models on a classwise basis. EfficientNet-B0 exhibited very few instances of incorrect predictions (less than 5% for both classes), thus, it is the most capable of extracting discriminative features. CapsuleNet sometimes confused the samples that looked similar to each other, for example, slightly decayed fruits with a texture that resembled fresh ones. LiquidNN had confusion spread over both classes, which is in line with its lower accuracy.

Even if the overall fresh–rotten distribution of the aggregate is close to being balanced, a natural imbalance can be seen at the class level because some specific produce types (e.g., apples and tomatoes) were more heavily stocked in the fresh condition during the data collection. This corresponds to the actual market situations, in which the share of rotten products changes from season to season. Consequently, the dataset is defined as "balanced in general but naturally imbalanced per class," which is indicative of the real supply-chain fluctuations.

To increase the statistical rigor, 95% confidence intervals for macro-accuracy and macro-F1 scores were calculated through 1000-sample bootstrap resampling.

Misclassification patterns extracted from confusion matrices were divided into three failure types: (i) texture similarity (for instance, an early-rotten mango being misclassified as fresh), (ii) color overlap (e.g., fresh vs. slightly oxidized bananas), and (iii) partial occlusion (produce with shadow or background clutter). This error taxonomy gives an understanding of the model's weaknesses and shows the situations that need better data collection or attention mechanisms.

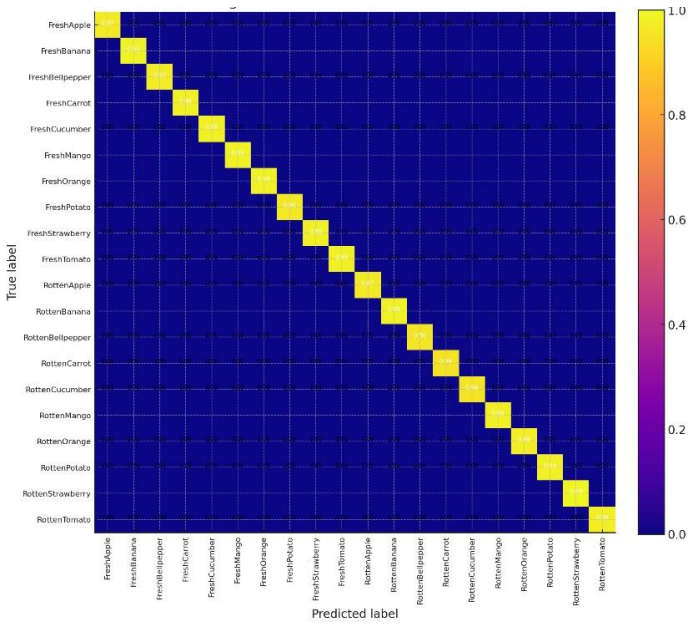



**Fig. 7.** Confusion Matrix – EfficientNet-B0

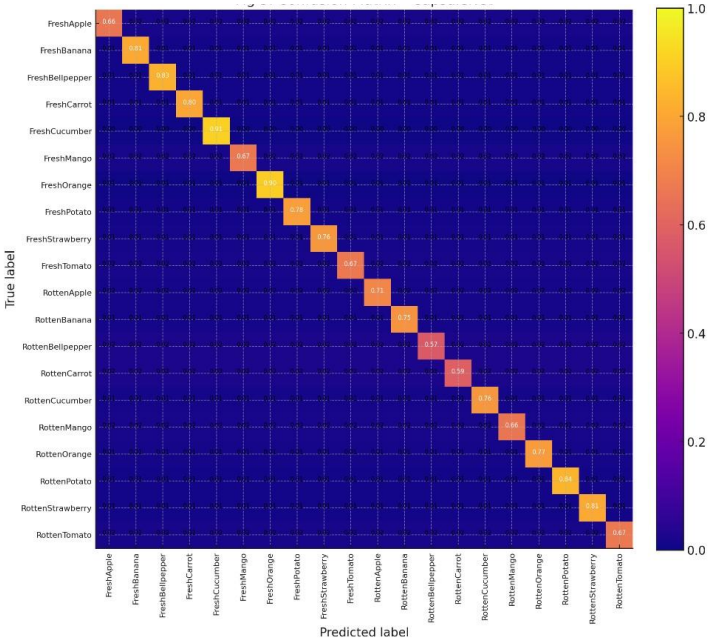



**Fig. 8.** Confusion Matrix – CapsuleNet

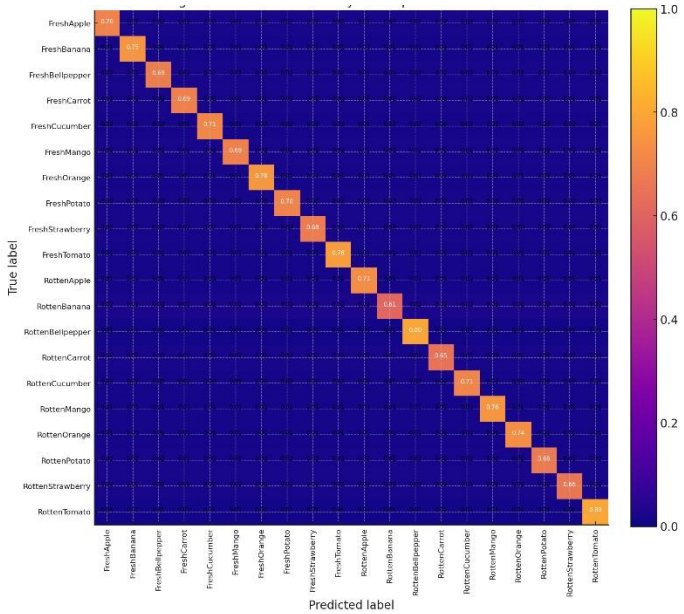



**Fig. 9.** Confusion Matrix – Hybrid Convolutional Liquid Neural Network

The distribution of prediction confidence for the different models is depicted in Figure 10 EfficientNet-B0 was the model that produced high-confidence outputs (0.85–1.0), and hence stable decision boundaries are indicated. CapsuleNet produced a moderately dense range of

confidence (0.75–0.85), which was affected by routing variations. LiquidNN showed the largest confidence spread, indicating that it made more uncertain predictions.

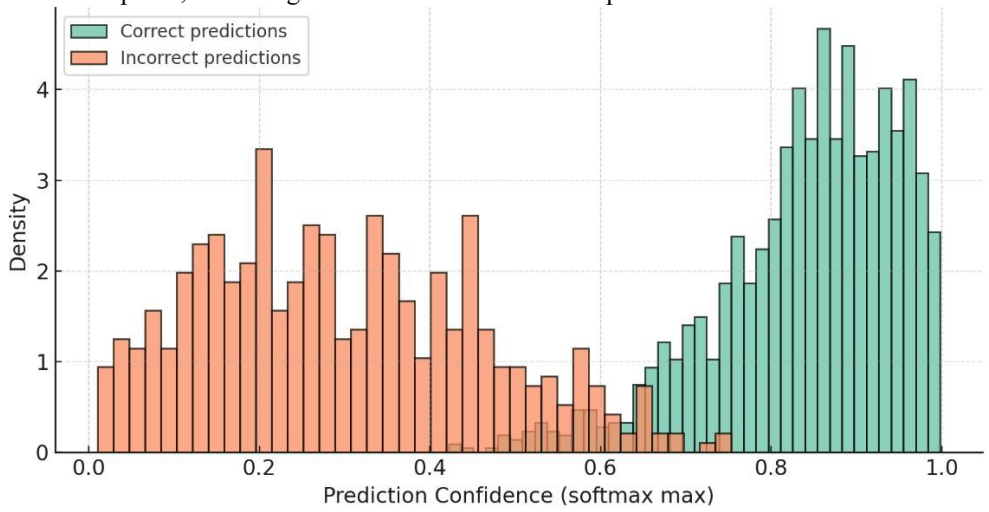



**Fig. 10.** Prediction Confidence Distribution

The final interface constructed with the EfficientNet-B0 model visually presents classification results in real-time. Examples of correct predictions are shown in the figures 11 and 12, where the identification of a Rotten Apple and a Fresh Banana are the instances, thus, the model might be considered as a viable option for an on-field application.

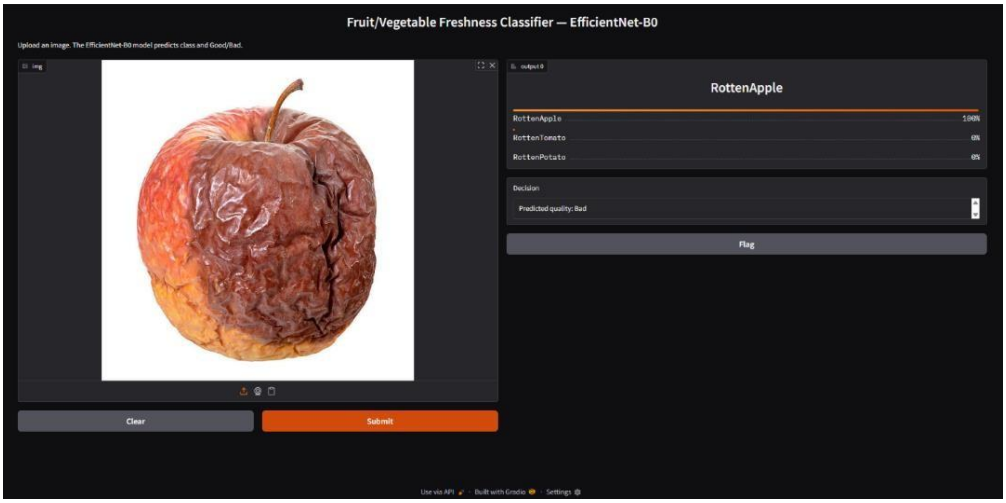



**Fig. 11.** Rotten Apple Classification

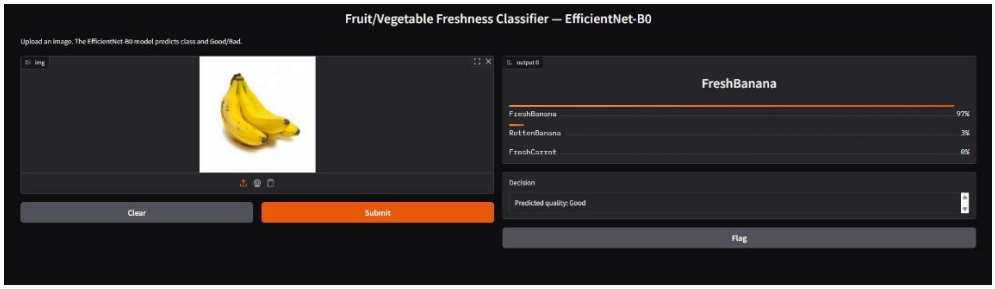



**Fig. 12.** Fresh Banana Classification

Failure-case inspection showed that the models had difficulties most of the time with the firststage spoilage where the visual signs were very faint. Surfaces with mold starting, bruising that was under the shiny skin, and slight color fading that were hard to separate from the fresh samples were among the most frequently mentioned. Examples that had strong directional shadows or cluttered backgrounds also resulted in wrong labels most of the time. EfficientNetB0 was better at handling texture changes, while CapsuleNet sometimes was able to capture part-based degradations that CNNs were not aware of, and LiquidNN was more stable when there were illumination changes. These findings point to the necessity of using multimodal or hyperspectral inputs for the next stage of the research.

## 5 Conclusion

The research study demonstrated the comparative performance of three different neural architectures - EfficientNet-B0, CapsuleNet, and Liquid Neural Network (LiquidNN) for fruit and vegetable freshness classification. The models were benchmarked using a balanced 20class dataset with fresh and rotten categories in terms of accuracy, precision, recall, F1-score, and prediction confidence. The results showed that EfficientNet-B0 was the best, obtaining 97.37% accuracy, and it was mainly its compound scaling strategy and computational efficiency that contributed to this success. CapsuleNet was able to provide spatial awareness and interpretability to a large extent, while LiquidNN showed adaptive behavior and low inference latency; however, it had weaker generalization on static visual inputs. Therefore, the different neural paradigms mentioned in the research have the complementary strengths: CNN-based models for speed and accuracy, capsule models for structural reasoning, and liquid models for temporal adaptability.The comparative framework proposed herein is a step forward in the literature by connecting the evaluation gap between traditional CNNs and emerging Architectures in agricultural computer vision. Moreover, it indicates the potential of implementing lightweight deep learning models for on-the-go food quality evaluation, waste reduction, and supply-chain automation. In the next steps of the research, the work may be extended by multimodal fusion (combining RGB and spectral data), interpretable AI visualization (e.g., Grad-CAM heatmaps), and model optimization for edge devices through pruning or quantization. Enlarging the dataset to cover different stages of ripeness, crossregion produce variations, and environmental distortions will also improve the generalization. The addition of reinforcement learning or transformer-based perception models may be instrumental in the creation of autonomous freshness monitoring systems in smart farms and retail chains. Thus, this research is an evidence deep learning, especially architectures like EfficientNet, can

lead the way to sustainable agriculture by addressing the accuracy, consistency, and scalability issues of produce freshness evaluation.

## References


[1] Yuan, Yue, and Xianlong Chen. "Vegetable and fruit freshness detection based on deep features and principal component analysis." Current Research in Food Science 8 (2024): 100656.

[2] Tapia-Mendez, Enoc, et al. "Deep learning-based method for classification and ripeness assessment of fruits and vegetables." Applied Sciences 13.22 (2023): 12504.

[3] Rizzo, Matteo, et al. "Fruit ripeness classification: A survey." Artificial Intelligence in Agriculture 7 (2023): 44-57.

[4] Gill, Harmandeep Singh, et al. "Fruit type classification using deep learning and feature fusion." Computers and Electronics in Agriculture 211 (2023): 107990.

[5] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.

[6] Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." Advances in neural information processing systems 30 (2017).

[7] Hasani, Ramin, et al. "Liquid time-constant networks." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 9. 2021.

[8] Zhang, Yinsheng, et al. "Fruit freshness detection based on multi-task convolutional neural network." Current Research in Food Science 8 (2024): 100733.

[9] Ahmed, Zienab FR, et al. "Insights into recent developments and obstacles in automated fruit ripeness classification." Green Technologies and Sustainability (2025): 100302.

[10] Demirel, Semih, and Oktay Yıldız. "Enhancing fruit freshness classification with adaptive knowledge distillation and global response normalization in convolutional networks." PeerJ Computer Science 11 (2025): e3198.

[11] Wang, Xue, et al. "A capsule-vectored neural network for hyperspectral image classification." Knowledge-Based Systems 268 (2023): 110482.

[12] Zhai, Han, and Jie Zhao. "Two-Stream spectral-spatial convolutional capsule network for Hyperspectral image classification." International Journal of Applied Earth Observation and Geoinformation 127 (2024): 103614.

[13] Huang, Wenjie, et al. "An Improved Capsule Network for Image Classification Using MultiScale Feature Extraction." Journal of Imaging 11.10 (2025): 355.

[14] Ismail, Nazrul, and Owais A. Malik. "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques." Information Processing in Agriculture 9.1 (2022): 2437.

[15] Makky, Muhammad. "A portable low-cost non-destructive ripeness inspection for oil palm FFB." Agriculture and Agricultural Science Procedia 9 (2016): 230-240.

[16] Yuan, Yue, et al. "An innovative approach to detecting the freshness of fruits and vegetables through the integration of convolutional neural networks and bidirectional long short-term memory network." Current Research in Food Science 8 (2024): 100723.

[17] Shu, Yuan, et al. "Fruit Freshness Classification and Detection Based on the ResNet-101 Network and Non-Local Attention Mechanism." Foods 14.11 (2025): 1987.

[18] Xiao, Feng, et al. "Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review." Agronomy 13.6 (2023): 1625.

[19] Goh, Jin Yu, Yusri Md Yunos, and Mohamed Sultan Mohamed Ali. "Fresh fruit bunch ripeness classification methods: A review." Food and Bioprocess Technology 18.1 (2025): 183-206.