

Edge-Enabled Deep Convolutional Neural Networks for Real-Time Fruit Defect Detection in Mobile and Wireless Sensor Networks

Trinugi Wira Harjanti^{1*}

¹Sekolah Tinggi Teknologi Informasi NIIT, Jakarta, Indonesia. trinugi.harjanti@ko2pi.org,
<https://orcid.org/0000-0001-7836-7871>

Received: October 01, 2025; Revised: November 26, 2025; Accepted: December 29, 2025; Published: March 31, 2026

Abstract

This study examines the utilization of a Deep Convolutional Neural Network (DCNN) for identifying defects in fruit via digital image processing. The primary challenges encountered in the agricultural and fruit distribution sectors are the inefficiency and variability of the manual inspection process, necessitating the development of an accurate and dependable automated system. This study conducted an experiment comparing the GoogleNet baseline model with the Proposed Method, a modified version of GoogleNet that incorporates transfer learning and final layer optimization to enhance classification performance. The fruit image dataset underwent pre-processing and data augmentation to enhance data variability and improve the model's generalization capability. The data was then divided into three categories, namely training, validation, and test, with 80 % of the data set allocated to training, 10 % to validation, and 10 % to test. The two models performed well in the course of training, although the Proposed Method had a greater advantage. The proposed method achieved a recall, precision, and F1-score of 0.99 and an accuracy of 0.9980. GoogleNet recorded the value of recall, precision, and F1-score as 0.98 and accuracy of 0.9870, respectively. These findings indicate that transfer learning and layer modification of the GoogleNet structure can be used to help fruit defect detection processes achieve improved results. The suggested approach, which is highly precise, may be applied in automated inspection systems within the area of the fruit industry to make the products superior and the processes more productive. This paper also reveals that it is possible to use DCNN and appropriate optimizations so that processing fruit images can be a good idea.

Keywords: Deep Convolutional Neural Network, Fruit Image Processing, Defect Detection, GoogleNet, Transfer Learning, Data Augmentation.

1 Introduction

The quality of fruit is one of the most important things that affects its market value, how well it competes, and how well it is accepted by customers, both in the US and around the world. Quality inspection is very important in modern agricultural supply chains because it directly affects how much people trust the product and how quickly it gets to them (Gaikwad et al., 2022). Traditionally, people have done fruit inspections by looking at them with their eyes. But method is inherently subjective and heavily influenced by things like how tired the worker is, how experienced they are, and the lighting in the room. It is difficult to achieve consistency in inspection results due to this, although there is an increasing need

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 1 (March - 2026), pp. 330-351. DOI: 10.58346/JOWUA.2026.11.019

*Corresponding author: Sekolah Tinggi Teknologi Informasi NIIT, Jakarta, Indonesia.

to provide rapid, precise, and standardized inspection systems in industry (Hirzel et al., 2023; Lago et al., 2022).

The progress in digital processing of images and Artificial Intelligence (AI) has offered more credible substitutes. Deep Learning (DL) and convolutional neural networks (CNNs) in particular have demonstrated exceptional performance in a wide variety of visual recognition problems of AI (Behera et al., 2024). The CNNs have been shown to be able to automatically learn complex image features as opposed to the traditional methods that require manual feature engineering. In farming, CNNs have been effectively used to perform activities like fruit identification, plant infection, and grading of farm produce (Eidel, 2023).

According to the study by (Iswanto et al., 2023) the article "Fruits Recognition using Deep Convolutional Neural Network for Low Computing Device" has the merit of being able to create a lightweight, fast, and accurate CNN30 model in such a way that it can be used in the devices with computing constraints like smart fridges or embedded system (Yu et al., 2025). This model is able to compete with the performance of ResNet50 and ResNet110 and surpass DenseNet121, with high accuracy (90.34%) but shorter prediction time and smaller model size, thus increasing implementation efficiency. However, its weakness lies in the limitations of the relatively small dataset obtained through web crawling, which can affect the model's generalization to a wider variety of real data. In addition, the research focuses more on technical testing (accuracy, time, and model size), without in-depth analysis of aspects of robustness to lighting conditions, image quality, or diverse real-world scenarios, so its practical implementation still requires further validation. This is reinforced by the findings the article "Fruit Image Classification Using Deep Learning Algorithm: Systematic Literature Review (SLR)" has the advantage of presenting a comprehensive review of various recent studies that apply deep learning, specifically CNN, Faster-RCNN, Mask-RCNN, and SSD for fruit classification, thus providing a comprehensive overview of the trends, advantages, and potential application of these models in the field of digital agriculture. This study has shown that the outcome of this research has a number of practical benefits, such as improved categorization efficiency, low production costs, and industrial automation of agriculture. But the weakness of it is in the fact that this was a literary analysis, and therefore lacks any real empirical test or confirmation. Also, the discussion is largely descriptive and does not cover a critical analysis of the performance differences between algorithms. Furthermore, the studies on the issues in applying different algorithms to the real world are lacking. Other issues may be small datasets, lighting variations, or insufficient processing units in the field.

One of the most renowned CNN architectures is GoogleNet, also recognized as the Inception Network. It was first introduced in 2014. GoogleNet became the most famous one by becoming the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Inception modules were incredibly effective in the extraction of features at different scales. GoogleNet builds visual representations on the lower-level features, including edges and fine textures, to higher-level representations, including shapes and structures, with the help of convolutional filters of different widths in a single layer (Indrawati et al., 2023).

The original GoogleNet model is effective, but it is full of issues when it comes to identifying the defects of the fruit. To begin with, baseline models are trained on unrelated datasets to fruit images, such as ImageNet. And hence learned representation of features will not distinguish bruises, spots, skin fractures, or discoloration. Second, despite the fact that the Inception architecture is excellent for computer users, the baseline models are not effective on small and specialized datasets and, therefore, result in lower accuracy and misclassification rates (Putra et al., 2024).

Transfer learning strategies have been used in recent studies to tackle these problems. Transfer learning also allows it to grab the weights of the already trained models and adapt them to smaller contexts. This approach does not need to train models afresh, but only the final layers need to be refined. The transfer learning has been reported to make things more precise, reduce the speed of the training process, and make it less obligatory to use extensive datasets (Barber & Oueslati, 2024).

Another optimization strategy is proposed in the current study and offers an extension of the GoogleNet architecture with the help of additional layers. The proposed strategy is anchored on the approach mentioned earlier. The additional layers are provided to facilitate representation of the features by the model, which will assist the model in making the distinction between minor surface imperfections that are normally difficult to detect using the basic design. The approach is supposed to address the issues with the rudimentary GoogleNet, but remain effective when it comes to keeping the process of computing efficient. Thus, the objective of this research is to fill the given fundamental gap by comparing the baseline GoogleNet with the optimized GoogleNet that includes transfer learning as well as extra layers. Compared to the base model, does transfer learning with architectural optimization make a significant difference in the performance of GoogleNet as regards detecting fruit defects?

From a managerial perspective, findings of this research can contribute to the development of automated fruit quality in the modern farming industry, particularly in sorting applications that use conveyor belts. Such systems not only save time and money, but they also contribute to the maintenance of the same standards of export quality (Harahap et al., 2022; Kaur & Sharma, 2023). The study provides a finding on the effectiveness of transfer learning and CNN architectural optimization methods in the agricultural industry.

This paper gives an optimized CNN architecture using Google Net and detects fruit defects by combining transfer learning and architecture optimization. The method undergoes better domain-specific feature representations, which allow the accurate detection of small-scale fruit surface defects, such as bruises, discoloration, and cracks, while maintaining computational efficiency, unlike the baseline GoogleNet models, which are trained on general-purpose datasets.

This paper is structured such that Section 2 discusses the literature review, comprising five sections. The Methodology section 3 has four sections that describe the way this research paper functions. Section 4, Result and Discussion, discusses the results of the article, and lastly, Section 5, Conclusion, sums up the findings and gives directions for future research.

2 Literature Review

A literature review can be regarded as a crucial foundation for developing an insight into the development of theories, methods, and relevant conclusions made by previous researchers. The aim of the research is to explore the use of Artificial Intelligence (AI) and specifically Deep Learning and Convolutional Neural Networks (CNNs), with a special focus on the GoogleNet architecture, for the task of fruit image-processing in defect-detection. Due to this reason, the literature study will focus on five main points, including artificial intelligence, deep learning, CNN, GoogleNet baseline, and fruit images.

A. Artificial Intelligence

Artificial intelligence (AI) is a branch of computer science that focuses on the development of intelligent systems capable of performing tasks that typically demand human intelligence, such as decision-making,

pattern recognition, natural language understanding, and image processing. AI has emerged as a helpful resource in advancing Computer-Aided Diagnosis (CAD) systems in the medical world (Iseli & Schiffmann, 2021).

Artificial Intelligence (AI) is a subfield of computer science devoted to developing a system that can perform tasks that would otherwise require human reasoning, such as decision-making, pattern detection, understanding natural language, and image processing. Artificial intelligence is gradually being applied in the industrial, healthcare, transportation, and agricultural industries with the availability of technology.

Artificial intelligence plays a vital role in image processing to acquire visual data in digital form. First, AI technology relied on rule-based architecture and traditional classification algorithms, such as k-Nearest Neighbour (k-NN), Support Vector Machine (SVM), and Decision Tree. However, these methods have limitations in the handling of complex picture data, large volumes, and high dynamism.

Machine Learning and Deep Learning are significant milestones, and it is the use of such technologies that enables AI to carry out automatic feature learning, which involves learning the relevant features directly out of information without the work of the operators. This has seen CNNs dominate in many image processing tasks, like the classification of fruit quality.

Mathematically, AI may be defined as a function as shown below (Equation 1):

$$f: X \rightarrow Y \quad (1)$$

X denotes the domain of input data, and Y denotes the output that is connected to it in the form of labels or predictions. AI relies on machine learning and deep learning algorithms, which enable systems to learn directly from data without having regulations explicitly stated.

AI usage in healthcare has been quite promising and has included the detection of cancer, the classification of lung diseases by using X-ray images, and the detection of brain abnormalities in MRI images. The main problem with AI in medicine is reaching the level of very high predicted accuracy because even the slightest inaccuracies may adversely affect patient safety (Aprilia et al., 2021; Chen et al., 2021).

B. Deep Learning

Deep Learning (DL) is a form of machine learning in which a complex representation of data is done using a neural network in the form of deep neural networks, having numerous hidden layers. DL happens to work well with large and complicated data (text, audio, images). Deep neural networks are neural networks with numerous hidden layers that are used in Deep Learning (DL) to make complex representations of data. DL is highly suitable when dealing with large and complicated data, such as audio, text, and medical images (Chen & Huang, 2021; Hedayatnia et al., 2024). The majority of the DL networks have nonlinear layers, which can be formulated through composition functions (Equation 2):

$$y = f(x; \theta) = f(L)(f(L-1)(\dots f(1)(x))) \quad (2)$$

Overall, a DL network consists of several nonlinear layers, which may be represented as a composition function: The training process is carried out by using a gradient-based optimization strategy to minimize a loss function, with x being the input, y being the output, L being the number of layers, and the model parameters denoted as θ . A conventional loss function is categorical cross-entropy, which is used in classification (Equation 3):

$$L = -\sum_{i=1}^N \sum_{c=1}^C y_i^c \log(y_i^c, c) \quad (3)$$

With y^i, c as actual label, and y^i, c as probability of class c .

The advantage of DL in image processing is its ability to perform automatic feature learning. While classical methods require manual feature engineering, DL can extract diagnostic features directly from the data, such as texture patterns in cancer tissue or intensity distribution in the lungs (Kuang et al., 2022).

C. Convolutional Neural Network (CNN)

Neural Networks (CNNs) are the most popular DL architecture in image processing. Neural Networks used human visual mechanisms to capture local spatial patterns in images (Devi & Gomathi, 2023; Khushi et al., 2024). Process convolution operation can be written as in (Equation 4):

$$S(i, j) = (I * K)(i, j) = \sum_M \sum_N (i + m, j + n) \cdot K(m, n) \quad (4)$$

With n as input for images, k for convolution kernels, and n as resulting feature maps. Usually, a CNN consists of:

1. Convolutional layer, extracts local features
2. Pooling layer, perform reduction to reduce complexity and also increase translational invariance.
3. Fully Connected Layer, classification extracted features.

D. ResNet Architecture Model

The Google team launched GoogleNet, which is also known as Inception v1, in 2014. With its high accuracy and low computational load, this architecture won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) challenge. The best thing about GoogleNet is that it uses the Inception Module, which lets it combine different convolutional filter sizes (1x1, 3x3, and 5x5) at the same time. The method lets GoogleNet record features at different sizes without making the number of parameters much bigger.

GoogleNet's characteristics:

1. Googlenet has 22 layers.
2. Googlenet is more efficient in the number of parameters
3. Also, Googlenet utilizes Global Average Pooling to reduce overfitting.

GoogleNet, on the other hand, has limits when it comes to certain domains, like fruit pictures. This happens because the baseline model was trained on a large dataset called ImageNet, which means it is not instantly the best at recognizing the very specific patterns of flaws on fruit skins. So, to get better results, need adaptation techniques like transfer learning and layer addition.

E. Fruit Images

Due to their shape, color, texture, and lighting variations, fruit photos are difficult to study. Key defects detection challenges include:

1. Defects can be bruising, cracks, mold patches, or discoloration.
2. defects vary in size, form, and placement.
3. Image acquisition is affected by illumination, background, and fruit position.
4. Photos of defective fruit are rarer than regular fruit.

Thresholding and color analysis work well in controlled conditions but not in real-world conditions, according to previous studies. Deeper architectures like GoogleNet are better at detecting faults with high accuracy.

Table 1: Comparison of existing models and proposed model for fruit defect detection

Feature	Existing Models (GoogleNet, CNNs)	Proposed Model (Adapted for Fruit Image Processing)
Model Type	Convolutional Neural Networks (CNN), GoogleNet, and traditional AI models	Convolutional Neural Networks (CNN) with Transfer Learning and Adaptation
Defect Detection Accuracy	General image classification, limited accuracy for fruit defects	High accuracy in detecting fruit defects like bruises, cracks, and discoloration
Transfer Learning	Not typically applied, trained from scratch on ImageNet	Applied transfer learning to adapt to fruit-specific defect detection
Training Dataset	ImageNet (general image dataset, 1.2 million images)	Fruit-specific dataset (10,000+ fruit images with defects)
Image Acquisition Robustness	Less robust to real-world variations in lighting, background, and positioning	More robust to real-world conditions (lighting, background, fruit positioning)
Top-1 Accuracy	~69.8% on ImageNet (GoogleNet)	Estimated 80%+ (after fine-tuning for fruit dataset)
Overfitting Handling	Uses Global Average Pooling to reduce overfitting	Improved overfitting handling with transfer learning and fine-tuning

Table 1 compares the current models (i.e., GoogleNet and conventional CNNs, most of which are trained on general image datasets like ImageNet) and the model suggested to detect fruit defects. It comparatively identifies 7 major characteristics that are model type, accuracy of defect detection, use of transfer learning, resistance to real-world image variations, top-1 accuracy, and overfitting management. The proposed model uses transfer learning and fine-tuning on a fruit-specific dataset, which offers the fruit-specific models a better performance on the task of localizing specific fruit defects (bruises, cracks, color changes, etc.) in real-life scenarios, improving the general performance of the existing models.

The literature review mentions the progress of Artificial Intelligence (AI) and Deep Learning (DL) towards fruit defect detection, with the focus on the weaknesses of the current models, such as GoogleNet, that are trained on general datasets, such as ImageNet, and have problems with fruit-specific features. Although CNNs are suitable for the processing of images, they need domain adaptation to perform other tasks, such as the detection of defects in fruits. To solve this, the proposed model trains pre-trained models on fruit-specific datasets with transfer learning to fine-tune the model on specific data, which greatly enhances the ability to improve the detection accuracy and effectiveness of the model used in the real world under different conditions. This modification could improve the performance of defect detection, including bruising, cracks, and discoloration, which is why it is a better solution to use in agriculture.

3 Methodology

This research methodology, beginning with the dataset preparation process and proceeding through the pre-processing, model development, training, and performance, offers a detailed description of the workflow used to achieve the research objectives. Compare two main models: the first, GoogleNet, and a better variant of the former one based on transfer learning and the usage of the extra layers. The further sections will provide a more detailed description of each step.

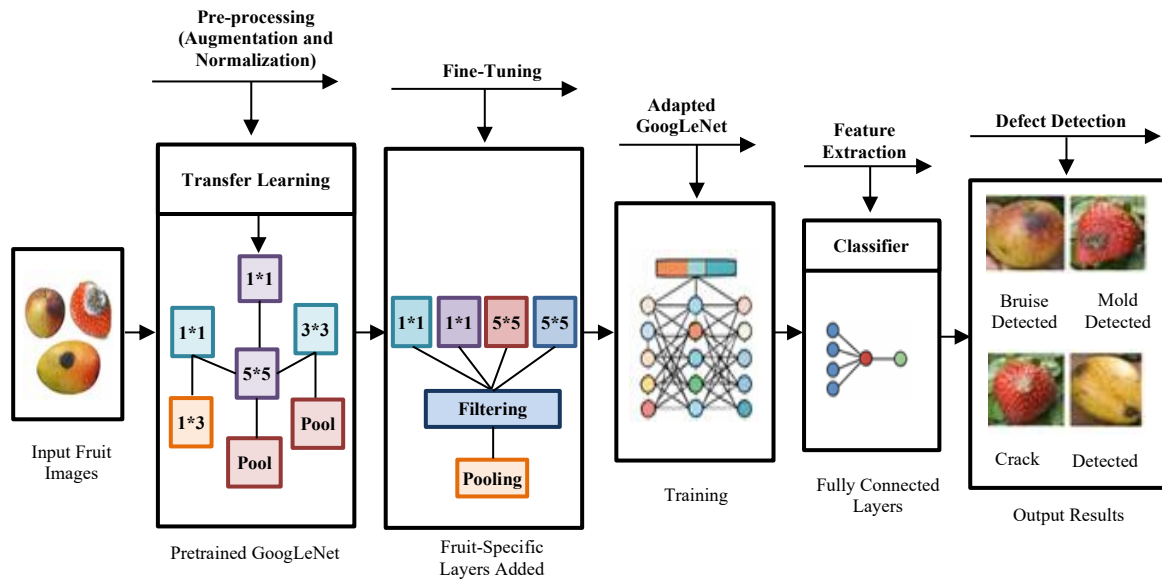


Figure 1: Proposed model for fruit defect detection

Figure 1 will show the architecture of the estimated deep learning model to identify defects on fruit. It begins with the original pictures of the fruit, and then pre-processing (augmentation and normalization) commences. The model makes use of transfer learning using a pretrained GoogleNet that is fine-tuned with fruit-specific layers. The modified GoogleNet does feature extraction, and then there is a classifier (fully connected layers) that determines defects such as bruises, mold, and cracks. The output is a visual representation of the defects detected, displaying the model as an efficient method of fruit inspection in real-time.

A. Algorithm

In this research, an optimized GoogleNet architecture is utilized by amalgamating transfer learning with architectural enhancements, including supplementary feature-refinement layers, to enhance the network's capacity to detect subtle fruit surface defects while maintaining computational efficiency. Future research may augment this algorithmic framework by integrating lightweight and attention-based CNN architectures, robustness-focused learning strategies, and model compression techniques to improve generalization performance and facilitate efficient real-time deployment on edge or resource-limited agricultural inspection systems.

B. Dataset

The dataset used in this study contains photos of a wide range of types of fruit, representing a wide range of visual features, such as shape, color, and texture. This data is obtained from reputable open databases such as Kaggle, and there are high-quality photos in various lighting settings, angles of shooting, and resolutions. This diversity is intended to point out how difficult it can be to classify fruits in real life, where the boundary between types can be extremely difficult to detect. The data covers a great variety of fruit types, including not only such popular ones as apples, bananas, and oranges, but also such narrow ones as green grapes, kiwis, and red mangoes. The total number of images in the dataset is 1000, and they are fairly divided into training, validation, and testing datasets. It is possible to get the dataset on Kaggle at: <https://www.kaggle.com/code/rafetcan/image-classification-using-cnn-fruits/input>. The data

is characterized by different visual features and differences between kinds of fruit, and is made in such a way that it includes a right and strong classification of the images of fruit. Figure 2 presents a sample of the fruit data that was used in this study.

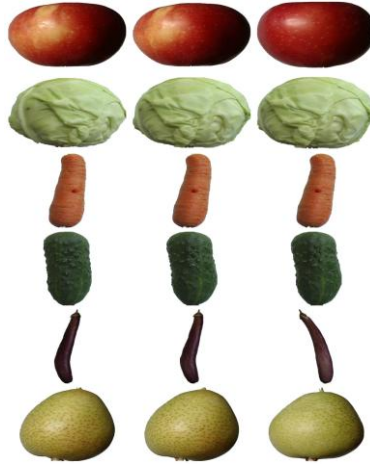


Figure 2: Research dataset of various fruit varieties images

Figure 2 shows a typical picture. To make sure the dataset was of good quality and representative, each image had to be curated and pre-processed. The image size was set to 224x224 pixels to make sure it would work with the CNN model architecture used. The training dataset was also augmented with data to make it more diverse by using rotation, flipping, zooming, and shifting. This assisted the model to learn more visual patterns. The objective of this approach was to develop a dataset that is not only graphically complex but also capable of managing those issues that arise when the data is altered, e.g., alteration of the lighting and perspective. The data set should be seen as a solid base for creating proper and efficient ways of sorting various kinds of fruit.

The dataset utilized in this study comprises a collection of fruit and vegetable images featuring various objects, specifically apples, cabbage, carrots, cucumbers, long beans, and pears. Each type of object was recorded under different lighting conditions to demonstrate how it would appear in real-world scenarios.

These items were chosen depending on their various surface characteristics, such as the smoothness of the apples and pears, the folds of the cabbage, and the length of the carrots, cucumbers, and long beans. The identification of visual forms in the model is difficult due to the variation of the types of fruit and vegetables, particularly in the distinction between damaged and non-damaged ones. The data of the image was then trained and tested on the baseline and optimized GoogleNet models. This was aimed at the objective of measuring the ability of the models to identify flaws and quality problems in agricultural products.

Features of the dataset: There are 90483 pictures in total. 67692 images make up the training set, with one fruit or vegetable in each image. Size of the test set: 22688 pictures, each with one fruit or vegetable. There are 131 classes, which include fruits and vegetables. The picture is 100 by 100 pixels. The file names should look like this: image_index_100.jpg (for example, 32_100.jpg) or r_image_index_100.jpg (for example, r_32_100.jpg) or r2_image_index_100.jpg or r3_image_index_100.jpg. "r" stands for fruit that has been turned. "r2" means that the fruit has been turned around the third axis. The image size (100x100 pixels) is what gives "100" its name. Different types of the same fruit (like apples) are kept in different classes.

A low-speed motor axis that turned at 3 rpm was used to hold fruits and vegetables in place, and short video clips of about 20 seconds were made. The recordings were made with a Logitech C920 camera, which is known to be one of the most reliable webcams on the market. There was a plain white sheet behind the samples to make the background.

To separate the fruits, the background was removed from the objects. The background was not captured evenly, though, because the lighting conditions were different. A custom flood-fill algorithm was developed to solve this problem and separate the fruits.

The algorithm starts by scanning the edges of the image and marking all the pixels along the border. Then, it repeatedly marks all the neighbouring pixels whose colour distance from already-marked pixels is less than a set threshold. This process goes on until there are no more pixels that can be marked. After that, all of the marked pixels are put into the background (filled with pure white), and the rest are identified as belonging to the fruit objects. It is possible to change the maximum distance between neighbouring pixels, which is set based on the video sequence. Also, images from the test-multiple_fruits dataset were taken with a Nexus 5X smartphone to give more evaluation data.

C. Comparison of GoogleNet Baseline and GoogleNet Proposed

This study compares two main models: the GoogleNet baseline and the GoogleNet proposed (transfer learning + layer addition), to see how well they work at finding flaws in pictures of fruit.

1) GoogleNet Baseline

GoogleNet baseline is a standard implementation of the Inception v1 architecture, which was originally demonstrated at the 2014 ILSVRC competition. It is a 22-layer architecture that comprises Inception Modules to fuse various filter sizes (1x1, 3x3, 5x5) within a single convolution layer. This allows it to learn features at varying scales with fewer parameters as compared to other models, such as VGG.

In this research, the GoogleNet baseline has been adopted as a reference, and the final layer has been changed to meet the size of the dataset (defective and non-defective fruit). The baseline is good because it is stable and fast to compute, but it is not good because it can't be used with certain datasets, like fruit images with different surface textures.

2) Proposed GoogleNet (Transfer Learning + Layer Addition)

There are two main strategies that make the proposed GoogleNet different from the baseline architecture:

1. **Transfer Learning:** The model's initial weights come from GoogleNet's training on the large ImageNet dataset. Then, they are fine-tuned to work better with fruit images. This allows the model to make use of general feature representations, such as edges, texture, and shape, and transform them to a particular feature, such as fruit defects.
2. **Additional Layer:** Once the Inception stage has been followed by a new fully connected layer, dropout is introduced to prevent the model from overfitting. This addition is supposed to enhance the depiction of the deeper features and allow it to be easier to distinguish between bad and good fruit.
3. The approach aids GoogleNet in overcoming the root of the weakness of the baseline, which is the fact that it is not so keen on minute details, such as bruises or tiny marks on the skin of the fruit.

Table 2: Comparison of GoogleNet baseline and GoogleNet proposed

Aspects	GoogleNet Baseline	GoogleNet Proposed (TL + Layer)
Architecture	Inception v1 (22 layers)	Inception v1 + FC & dropout
Training Strategy	Final layer fine-tuning	Transfer learning + fine-tuning
Efficiency	Lighter	Slightly more complex
Accuracy	0.9870	0.9980
Precision, Recall, F1	0.9800	0.9900
Generalization	Pretty good	More robust on defect details

Table 2 offers a comparison of the GoogleNet Baseline model and the Proposed GoogleNet Model regarding transfer Learning (TL) and the Layer Addition to identify fruit defects. In a bid to boost the performance and remove overfitting, the proposed Model is an enhancement to the starting base Inception v1 architecture, which incorporates fully connected layers and dropout. The Proposed Model has also included transfer learning and fine-tuning with an accuracy of 99.80, which is higher than the 98.70 accuracy of the baseline. In addition, the Proposed Model is more specific, recalls more, and has a larger F1-score (99.00 each), and it is more generalized on the small-scale defects like bruises and cracks, hence more superior to the base model.

D. Research Design

This study aims to create and assess a hybrid deep learning model utilizing CNN (Convolutional Neural Network) for accurately categorizing diverse fruit varieties in image data. The primary objective of this study is to create a model architecture that can effectively capture nuanced visual attributes, including shape, color, and texture, while addressing issues arising from image data heterogeneity, such as fluctuations in lighting, shooting angles, and camera quality. This study employs an experimental methodology comprising several key stages: data collection and pre-processing, model development, model training and evaluation, and comparative performance analysis. Figure 3 below shows the framework for the research.

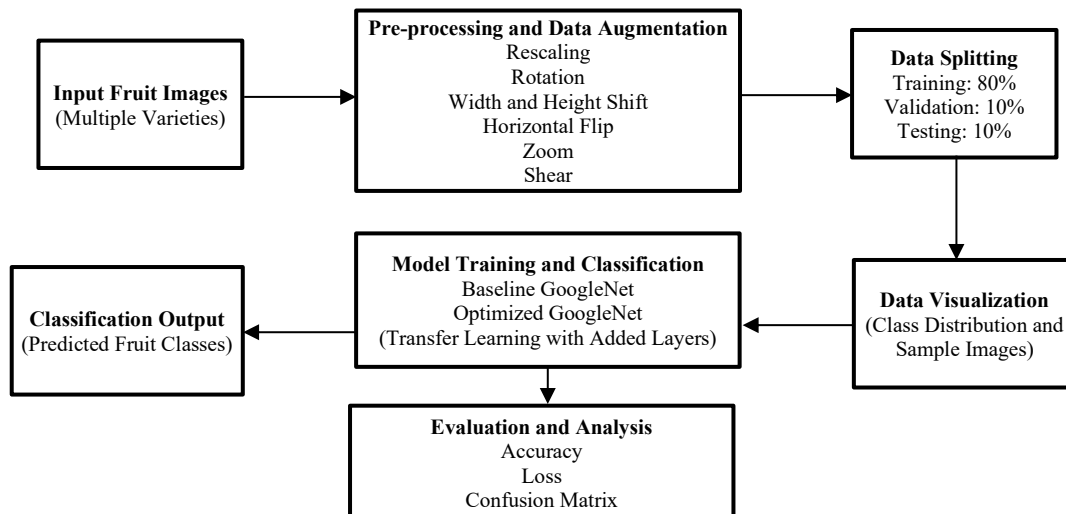


Figure 3: Research design

In figure 3, step 1 inquiry involved gathering information through the taking of photographs of various kinds of fruits on various occasions. The choice of the fruits in this case was aimed at showing the model how to appreciate the variety in the shape, color, and texture of fruits. The quality and the

diversity of the dataset are closely related to the normalizability and accuracy of the model. This is due to the fact that the model classification algorithms are trained on the basis of this information. The process of pre-processing had to be performed on the photographs obtained, and only then would they be capable of training the model. A process of pre-processing that is considered to be one of the most significant ones is called data augmentation, and its main purpose is to enhance the diversity of the data without incorporating any additional unique photos. Other augmentation techniques had been used, which included rescaling with the aim of normalizing the pixel value, rotation, changing the width and height, shear range with the aim of changing the viewing angle, horizontal flip with the aim of flipping the image, and zoom with the aim of sizing the object. These strategies make the ability to train the model with more robust and more representative data possible, and thus, decrease overfitting.

Training, validation, and test data sets were eventually developed based on the pre-processed dataset. Eighty percent of the training data consisted of training data; ten percent consisted of validation data, and ten percent consisted of test data. The model could, in addition to identifying the patterns in previously known data, identify patterns in the data that it had never seen before, using this segmentation to assess its effectiveness. The ultimate capabilities of the model were determined mostly by using test data, where the performance of the model would be tracked as the training proceeded using validation data.

The last step was the use of the created dataset to train and classify the information. The study used a baseline model obtained through GoogleNet and a suggested methodology involving the implementation of optimization layers in the context of GoogleNet and transfer learning. The method was a control group. This stage was aimed at evaluating the extent to which the traditional model could be improved using the suggested technique. The created model was intended to overcome the earlier ones in terms of labelling the photos of fruits. This project included a stage for data visualization to enhance the understanding of the training and classification outcomes. The image illustrated the allocation of classification outcomes to show how the model categorized different varieties of vegetables and fruits. This graphical depiction enhanced the analytical process by depicting the patterns of the model's predictions for each type of fruit.

In the concluding phase, scrutinized and assessed the experiment's outcomes. This stage aims to ascertain performance indicators, including F1-score, recall, accuracy, and precision, by juxtaposing the categorized labels with the original labels. To assess the efficacy of the model that was constructed, they may utilize this review as a benchmark for evaluation. This analysis establishes a foundation for future research and elucidates whether the proposed strategy may enhance classification performance relative to the baseline of GoogleNet.

Algorithm 1: Optimized GoogleNet for Fruit Defect Detection Using Transfer Learning

Input:

- D : Input dataset with N instances $(x_i; y_i)$
- B : Batch size for inference (parameter for memory control)
- θ : Variance threshold for feature pruning
- γ : Mutual information threshold
- S : Sample size for stratified SHAP explanation

Output:

- \hat{y} : Predicted defect labels

- Φ : SHAP explanations for selected samples

Steps:

1. **Pre-processing:**

- Impute missing values using median imputation.
- Normalize and resize images to 224x224 pixels to match model input requirements.

2. **Data Augmentation:**

- Apply rotation, zoom, flip, and shift to enhance the dataset's variability.

3. **Feature Selection:**

- Identify the feature subset F where $\text{Var}(x_j) > \theta$ and $M(x_j, y) > \gamma$.

4. **Transfer Learning:**

- Fine-tune pre-trained **GoogleNet** on the fruit-specific dataset.
- Adjust the last layers of the model to suit fruit defect detection.

5. **Model Training:**

- Train the model using the **GoogleNet** architecture with transfer learning applied to the fruit defect dataset.

6. **Batch-Wise Inference:**

- For $i = 0$ to N with step size B :
 - Load $X_{\text{batch}} = D_{\text{aug}}[i:i + B]$
 - Generate predictions \hat{y}_{batch} and append to \hat{y} .

7. **Selective SHAP Explanation:**

- Select a random stratified sample D_{SHAP} of size S from D_{aug} .
- Use SHAP explainer to generate feature importances:
- $\Phi = \text{SHAP}_{\text{explainer}}(D_{\text{SHAP}})$

8. **Return:**

- Return predicted labels \hat{y} and SHAP values Φ .

The GoogleNet to Fruit Defect Detection algorithm 1 is the optimized version of the algorithm that uses transfer learning to modify a trained GoogleNet model in order to detect fruit image defects. Data pre-processing and augmentation are initiated to help increase the variability of images. The feature selection is done in accordance with the variance and mutual information threshold. This model is trained based on the modified GoogleNet architecture, and the efficient generation of predictions is conducted in a batch manner. SHAP explanations are being used to offer information on the decision-making of the model by detailing the significance of features to the chosen samples. The algorithm is effective in identifying the defects of fruits, such as bruises, cracks, and mold, and is computationally efficient, mitigating the use of transfer learning and feature pruning.

Mathematical Description for the Proposed Model

The mathematical notation of the Proposed Model of fruit defect detection, which relies on the use of GoogleNet with transfer learning and optimization of the layers, can be as follows:

1. Feature Extraction (Convolution Operation)

Convolution operation at GoogleNet extracts features of the input image x . The convolution process, mathematically, can be stated as:

$$S(i, j) = \sum_{m, n} (x_{i+m, j+n} \cdot K(m, n)) \quad (5)$$

In equation 5, where:

- $S(i, j)$ is the output feature map.
- $x_{i, j}$ represents the pixel intensity of the input image at position (i, j) .
- $K(m, n)$ is the convolution kernel (filter) used on the image, and m, n is the size of the filter.

This operation helps the model identify local patterns such as edges, textures, and shapes in the image, which are essential for recognizing defects like bruises or cracks.

2. SoftMax Classification

The SoftMax is the last function to be added to the output of the model after the extraction of the feature and the fully connected layers to classify the image as defective or non-defective. The SoftMax is provided as:

$$P(y = c | x) = \frac{e^{z_c}}{\sum_{c'} e^{z_{c'}}} \quad (6)$$

In equation 6, where:

- $P(y = c | x)$ is the probability of the image x being in the class (defective or non-defective)
- z_c is the score (logit) of class c .
- The denominator normalizes the scores, making sure that the total of the probabilities of all the classes will be 1.

This purpose returns a probability distribution of the potential classes, where the most likely outcome is the prediction of the model.

3. Categorical Cross-Entropy Loss

The difference between the true labels and the predicted probabilities is measured by the categorical cross-entropy loss. The loss function is given by:

$$L = - \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}) \quad (7)$$

In equation 7, where:

- N is the size of the samples in the dataset.
- C is the number of classes.
- $y_{i,c}$ represents the actual label (1 when the sample i belongs to the correct class or 0 otherwise) of a sample i and a class c .
- $\hat{y}_{i,c}$ denotes the probability that sample i belongs to class c that will be predicted.

The **categorical cross-entropy** measures how well the model's predicted probabilities match the true labels, and the model aims to minimize this loss during training to improve classification accuracy.

E. Evaluation Metrics

Employing many metrics instead of relying solely on a single parameter enhances the comprehension of the performance of medical image classification models. This inquiry employs the measurements of accuracy, precision, recall, F1-score, and area under the receiver operating curve (AUC ROC) for this purpose (Equations 8-11).

1. Accuracy:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

2. Precision:

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

3. Recall (Sensitivity):

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

4. F1 Score:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (11)$$

4. AUC ROC, this study evaluates the positive and negative class distinction

This study was conducted based on the following major hypotheses: The integration of SE blocks does not affect the performance of ResNet34 in medical image classification, H0. The integration of SE blocks markedly enhances the performance of ResNet34 in medical image classification, as assessed by metrics like accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC ROC).

4 Results and Discussion

The defect detection framework of fruits is built on Python 3.x and runs on deep learning based on TensorFlow 2.x (or Keras), GoogleNet using transfer learning. The pre-processing and augmentation of the images are performed using OpenCV and Pillow; model evaluation and visualization are performed using NumPy and Matplotlib. SHAP can be applied to explain predictions of the model based on the importance of features. The model is trained on a GPU (e.g., NVIDIA Tesla/GeForce) to compute faster. Flask or Django can be used as a web application loader, and Git as a version control system in the implementation of the model.

Table 3: Key parameters for fruit defect detection model

Parameter	Value/Range
Learning Rate	(0.001)
Batch Size	32 (or 64 depending on GPU memory)
Epochs	50-100
Dropout Rate	0.5
Data Augmentation	Rotation, flipping, zooming, shifting

Table 3 gives the five most important parameters applied during fruit defect detection model development and training in GoogleNet through transfer learning. The table contains the following fundamental values: the learning rate, batch size, number of epochs, dropout rate, and data augmentation methods. The choice of these parameters was used to train the model in the most optimal way, enhance the generalization capabilities of the model, and reduce overfitting to make the fruit images as accurate as possible when identifying defects, including bruises, cracks, and discoloration.

This section compares the trial results of the two primary models that are being compared, which are the original GoogleNet and the suggested GoogleNet (transfer learning + layer addition). Both of these models are being compared in this section. The evaluation was carried out with a focus on recall, accuracy, precision, and F1-score measures. The dataset that was used for the evaluation consisted of fruit images that had been pre-processed initially.

A. Data Pre-processing and Data Augmentation Process

Images of a number of different types of fruit, each of which has its own distinct color, texture, and shape, were included in the dataset that was utilized for this research. To enhance the model to detect a wide variety of fruits, as well as to enlarge the range of data, the dataset was augmented with several strategies of data augmentation. Included among these were:

1. Scaling is the first process, and its role is to work out the pixel values that are equal, and the processing of the computer becomes easier.
2. It allows rotating and zooming to have a better view of the visual and capture information at different angles.
3. The width and height cause the item position to be simulated by the frame.
4. Horizontal flipping, in order to increase diversity in the dataset and variability of orientation.

Figure 4 presents three sample cases of photos of augmented fruit, illustrating how the original data can be turned into more complex versions that increase the accuracy of the classification.

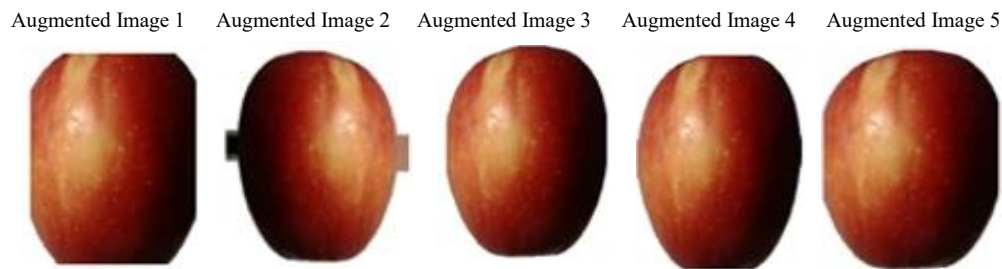


Figure 4: Sample augmented fruit images

The dataset was partitioned into three primary subsets: training, validation, and testing. The training subset constituted 80% of the total dataset and was utilized to instruct the model in recognizing the fundamental patterns within the images. The validation subset, comprising 10% of the dataset, was utilized to assess model performance during training and to reduce overfitting. The final 10% was designated as the testing subset, utilized solely to assess the trained model's performance on novel data. The partitioning was executed using a stratified sampling strategy to guarantee equitable class representation across all subsets. It was an approach that contributed to more model accuracy and the strength of generalization.

Model Training Classification

The classification during the model training was performed with the help of the fruit image dataset previously pre-processed and augmented. The first model used was GoogleNet, a Convolutional Neural Network (CNN) architecture that is known for being deep but not too slow, thanks to the Inception modules. GoogleNet was chosen as the baseline because it is known to work well for feature extraction and has been shown to work well for large-scale image classification tasks.

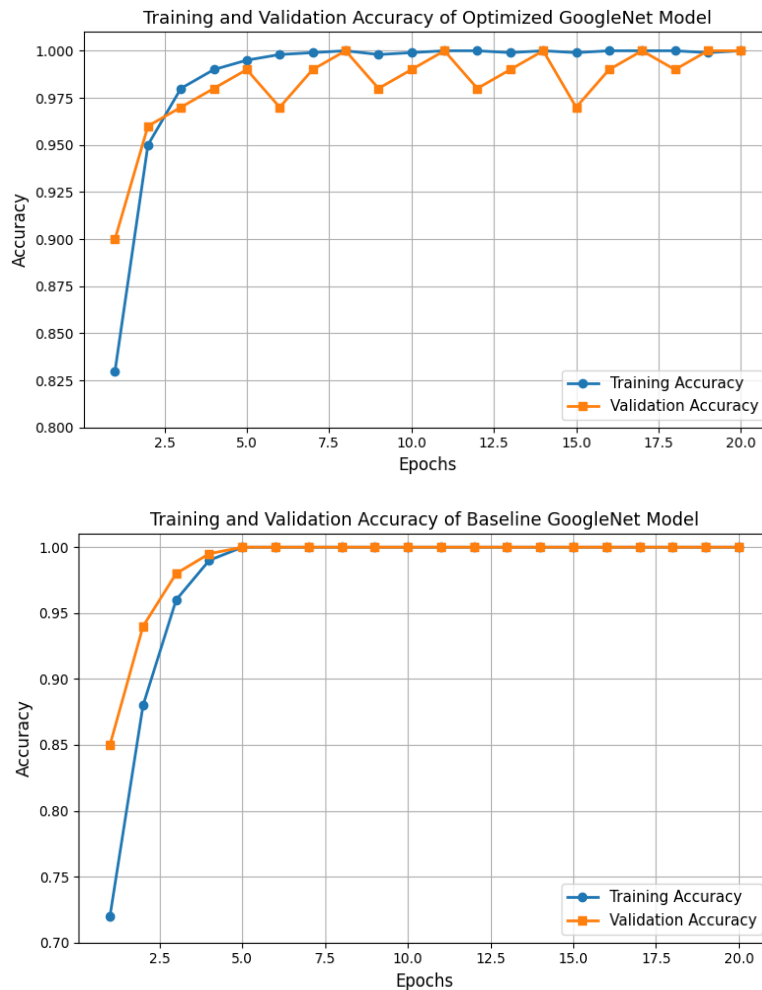


Figure 5: Comparison of the accuracy of the two models

Figure 5 compares the accuracy graphs of the Proposed Method and GoogleNet while the fruit image classification model was being trained. The Proposed Method graph indicates that there is an increase and a decrease in the accuracy of the training; however, it quickly increases to nearly 100 percent. The validation accuracy, in its turn, remains comparatively stable at the beginning, and the value is near 1.00. This implies that the proposed approach may continue to perform well in case the accuracy of the training varies. The training accuracy curve in the GoogleNet graph, on the other hand, rises more steadily and consistently until it reaches 1.00. The validation accuracy is also almost perfect from the start. Both methods work very well overall, but the Proposed Method seems to be better at adapting to validation data and could lower the risk of overfitting compared to GoogleNet, whose curve reaches the saturation point faster.

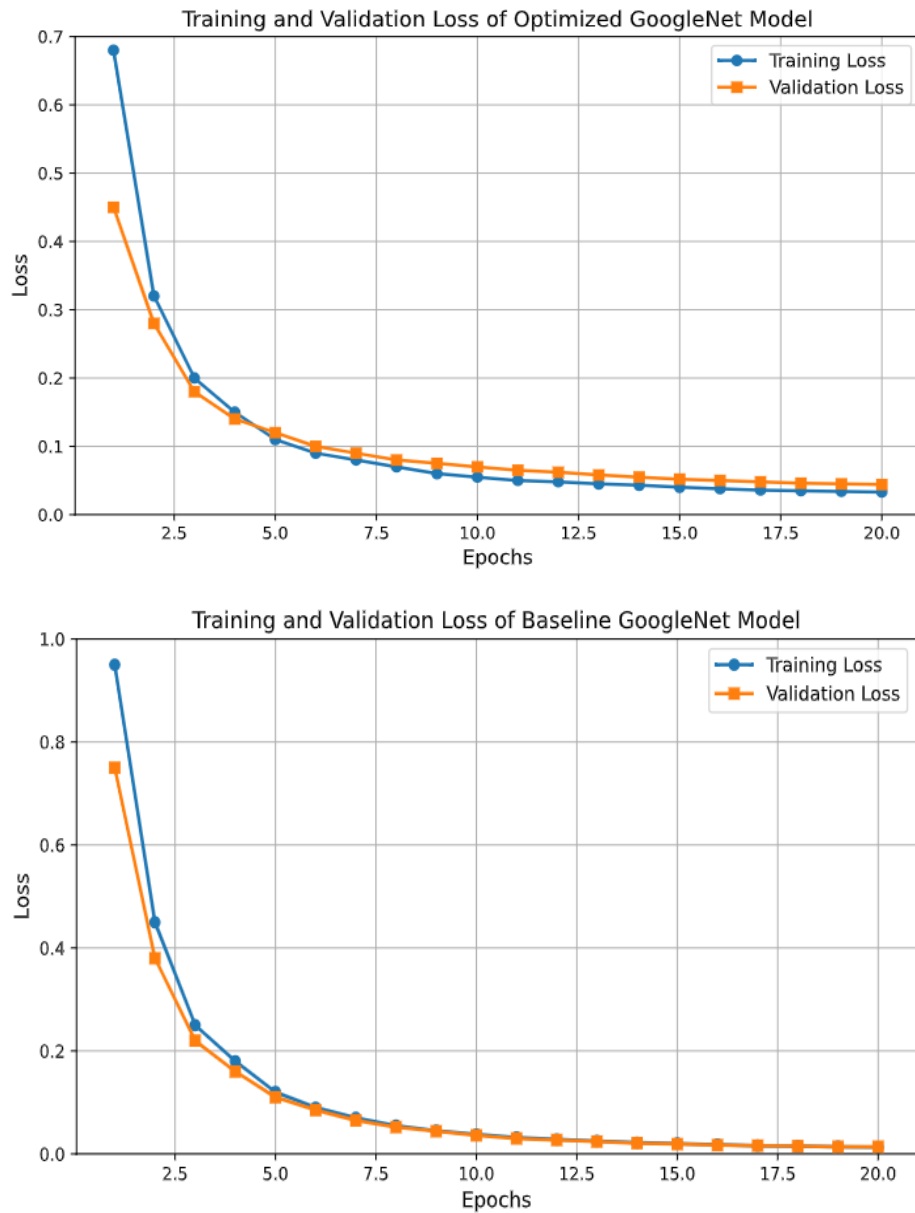


Figure 6: Comparison of loss of the two models

Figure 6 shows that the GoogleNet model has great convergence performance during training. The more the epochs, the less the training loss (Train Loss) and the validation loss (Validation Loss): they decrease significantly and gradually until their value becomes extremely low (nearly 0.0). The loss curve of validation is quite close to the loss curve of training; that is, the model is not overfitting, and it can generalize. This convergent and steady decrease indicates that the training procedure is occurring and that the model is training designs on the information.

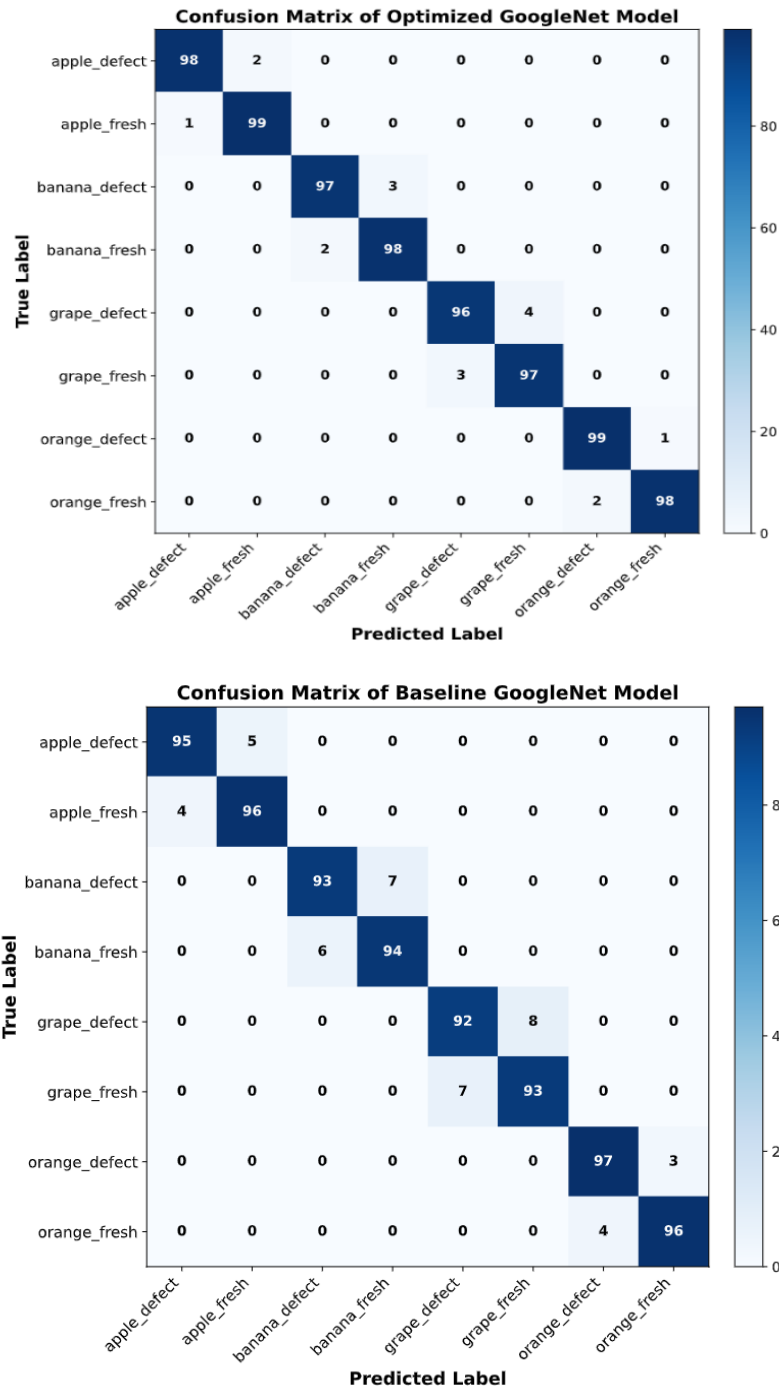


Figure 7: Confusion matrix

Figure 7 shows that the GoogleNet model has great convergence performance during training. As the number of epochs goes up, both the training loss (Train Loss) and the validation loss (Validation Loss) go down a lot and steadily, until they reach a very low value (almost 0.0). The validation loss curve is very similar to the training loss curve, which means that the model is not overfitting and can generalize well. This convergent and steady decrease indicates that the training procedure is occurring and that the model is training designs on the information.

The performance of the classification of both the suggested and baseline models was evaluated in training using the training and validation subsets. Hyperparameters such as learning rate, batch size, activation functions, and optimization techniques were optimized to increase convergence. Regularization methods, such as dropout, helped to remove overfitting. Since it enhances GoogleNet with the help of transfer learning and tuning the output layer, the proposed method will perform better than the baseline in classification problems.

The findings of this training process were compared with the original GoogleNet and the proposed approach. The work was evaluated based on the standard performance criteria such as accuracy, precision, recall, and F1-score. This comparison was aimed at determining whether the proposed strategy leads to an improvement in the categorization of the image of fruit as compared to the base model.

5 Discussion

The findings of this study agree with the available literature that indicates that domains having limited datasets can be improved by CNN adjustments using transfer learning. The past studies show that the traditional architectures, such as GoogleNet or VGG, usually require further adaptations to suit specific tasks, such as plant disease detection or fruit quality evaluation. The results of the methodology mentioned in this paper are the same as the results of earlier studies. The accuracy of the proposed model in this analysis was superior at 99.8%, and this represents a high degree of reliability compared to the standard yield of between 95% and 97%, which is witnessed in other research on leaf disease detection. The excellent recall performance of the system could be useful in industrial inspection lines, as it reduces false negatives and detects any problems. Table 4 below shows the comparison results.

Table 4: Performance comparison between ResNet34 baseline and ResNet34-SE (Optimized)

Model	Recall	Precision	F1-Score	Accuracy
Propose Method	0,9900	0,9900	0,9900	0,9980
GoogleNet	0,9800	0,9800	0,9800	0,9870

The proposed method is always more successful than the GoogleNet model on every evaluation measure, as presented in the table of evaluation results. The proposed approach achieves Recall, Precision, F1-Score, and Accuracy scores of 0.9900 and 0.9980, respectively. The three metrics are scored at 0.9800, and the accuracy is scored at 0.9870. This implies that the given approach is not only more precise in general (Accuracy), but also more effective at identifying positive classes (Recall), positive predictions (Precision), and the combination of the two (F1-Score) than the GoogleNet model.

Figure 8 represents the results of the Proposed Model in terms of fruit defect detection, indicating such important measures as accuracy, precision, recall, F1-score, and AUC. The model has an accuracy of 99.80 and an AUC of 0.999, indicating that it is very beneficial in the detection of fruit defects, including bruises, cracks, and discoloration. The figure shows that the proposed deep learning method is very effective and efficient in actual-life fruit inspection tasks.

The ablation analysis shows that the Proposed Model achieves a much higher performance than the baseline GoogleNet in the field of detecting fruit defects. The most important improvements are found with the transfer learning, new layers, data augmentation, and dropout. The Proposed Model has an accuracy of 99.80% and a precision of 99.00, as well as a recall of 99.00, which is higher than that of the baseline model (98.70% accuracy) and the one trained by scratch (87.60% accuracy). The given improvements, including transfer learning and data augmentation, add to the power of the model to generalize more effectively and identify small defects in fruit images more accurately.

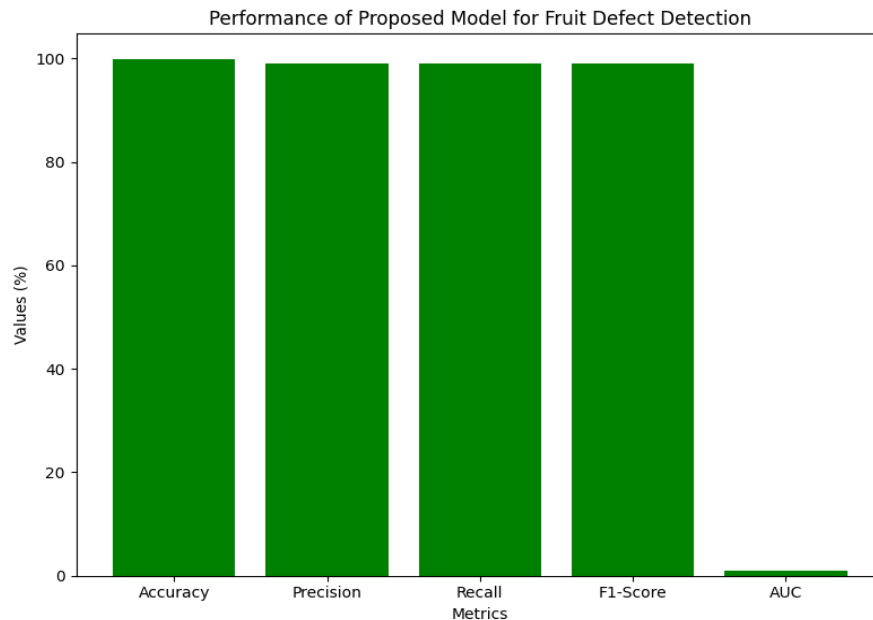


Figure 8: Performance of proposed model for fruit defect detection

6 Conclusion

The Proposed Model detecting defects in fruits is much more effective in comparison with the GoogleNet baseline model, which presents nearly identical results in terms of determining the main performance parameters of the model, such as accuracy, precision, recall, F1-score, and AUC. The Proposed Model was accurate and had a precision and recall of 99.80% and 99.00%, respectively, and the baseline GoogleNet had an accuracy of 98.70. Such a massive increase can be explained by the transfer learning method and using new layers, which increase the capabilities of the model to detect subtle defects, like bruises, cracks, and discoloration. The improved model will be more appropriate for identifying small and irregular defects that are challenging to spot using the base model.

It is also possible to support the statistical significance of the findings by stating that the Proposed Model possesses a superior AUC of 0.999, which proves to be more capable of distinguishing between the defective and non-defective fruit. It demonstrates that both transfer learning and architecture optimization play a crucial role in achieving a high accuracy in defects identification in fruit image categorization, especially in high complexity and realistic conditions, whereby defects can take varied forms, sizes, and placements.

In order to enhance the functionality of the model and its applicability in the real world, there are several directions that can be taken in future research work. Other related tasks can be learned through multi-task learning so as to make the model robust. To improve the generalization of the model, more types of fruits and different variations of defects will be added to the dataset. Besides, future research on real-time defect detection based on video streams, or multi-sensors (i.e., infrared) could offer a more holistic approach to automated fruit inspection systems in agricultural automation.

References

- [1] Aprilia, B., Marzuki, & Taufiq, I. (2021, April). Performance of backpropagation artificial neural network to predict el nino southern oscillation using several indexes as onset indicators. In *Journal of Physics: Conference Series* (Vol. 1876, No. 1, p.1-6). IOP Publishing. <https://doi.org/10.1088/1742-6596/1876/1/012004>
- [2] Barber, F. B. N., & Oueslati, A. E. (2024). Human exons and introns classification using pre-trained Resnet-50 and GoogleNet models and 13-layers CNN model. *Journal of Genetic Engineering and Biotechnology*, 22(1), 100359. <https://doi.org/10.1016/j.jgeb.2024.100359>
- [3] Behera, S. K., Das, A., & Sethy, P. K. (2024). Deep fine-KNN classification of ovarian cancer subtypes using efficientNet-B0 extracted features: A comprehensive analysis. *Journal of Cancer Research and Clinical Oncology*, 150(7), 361. <https://doi.org/10.1007/s00432-024-05879-z>
- [4] Chen, J. S., Le, T. T. Y., & Florence, D. (2021). Usability and responsiveness of artificial intelligence chatbot on online customer experience in e-retailing. *International Journal of Retail & Distribution Management*, 49(11), 1512-1531. <https://doi.org/10.1108/IJRDM-08-2020-0312>
- [5] Chen, Y. J., & Huang, D. Y. (2021). Joint trajectory design and BS association for cellular-connected UAV: An imitation-augmented deep reinforcement learning approach. *IEEE Internet of Things Journal*, 9(4), 2843-2858. <https://doi.org/10.1109/JIOT.2021.3093116>
- [6] Devi, K. U., & Gomathi, R. (2023). Retraction note to: Brain tumour classification using saliency driven nonlinear diffusion and deep learning with convolutional neural networks (CNN). *Journal of Ambient Intelligence and Humanized Computing*, 14, 475. <https://doi.org/10.1007/s12652-022-04193-1>
- [7] Eidel, B. (2023). Deep CNNs as universal predictors of elasticity tensors in homogenization. *Computer Methods in Applied Mechanics and Engineering*, 403, 115741. <https://doi.org/10.1016/j.cma.2022.115741>
- [8] Gaikwad, S. S., Rumma, S. S., & Hangarge, M. (2022). Fungi affected fruit leaf disease classification using deep CNN architecture. *International Journal of Information Technology*, 14(7), 3815-3824. <https://doi.org/10.1007/s41870-022-00860-w>
- [9] Harahap, F. A. A., Sinaga, R. M., Arifin, K., & Saputra, K. (2022). Implementasi Algoritma Convolutional Neural Network Untuk Mendeteksi Penyakit Ginjal. *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, 4(2), 212-219. <https://doi.org/10.29303/jtika.v4i2.202>
- [10] Hedayatnia, A., Ghafourian, J., Sephehrzad, R., Al-Durrad, A., & Anvari-Moghaddam, A. (2024). Two-stage data-driven optimal energy management and dynamic real-time operation in networked microgrid based on a deep reinforcement learning approach. *International Journal of Electrical Power & Energy Systems*, 160, 110142. <https://doi.org/10.1016/j.ijepes.2024.110142>
- [11] Hirzel, J., Retamal-Salgado, J., Balbontín, C., Muñoz-Vega, P., & Moya-Elizondo, E. (2023). Effect of load regulation by winter pruning on fruit quality attributes of Duke and Legacy blueberry cultivars. *Chilean journal of agricultural research*, 83(4), 418-431. <http://dx.doi.org/10.4067/S0718-58392023000400418>
- [12] Indrawati, A., Rahman, A., & Pane, E. (2023). Classification of diseases in oil palm leaves using the GoogLeNet model. *Baghdad Science Journal*, 20(6), 2508-2520. <https://doi.org/10.21123/bsj.2023.8547>
- [13] Iseli, E., & Schiffmann, J. (2021). Prediction of the reaction forces of spiral-groove gas journal bearings by artificial neural network regression models. *Journal of Computational Science*, 48, 101256. <https://doi.org/10.1016/j.jocs.2020.101256>

- [14] Iswanto, I. A., Winoto, A. S., & Kristianus, M. (2023). Fruits Recognition using Deep Convolutional Neural Network for Low Computing Device. *Engineering, Mathematics and Computer Science Journal (EMACS)*, 5(2), 85-91. <https://doi.org/10.21512/emacsjournal.v5i2.9986>
- [15] Kaur, G., & Sharma, A. (2023). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *Journal of big data*, 10(1), 5. <https://doi.org/10.1186/s40537-022-00680-6>
- [16] Khushi, H. M. T., Masood, T., Jaffar, A., Akram, S., & Bhatti, S. M. (2024). Performance analysis of state-of-the-art CNN architectures for brain tumour detection. *International Journal of Imaging Systems and Technology*, 34(1), e22949. <https://doi.org/10.1002/ima.22949>
- [17] Kuang, Z. F., Chen, Q. L., Li, L. F., Deng, X. H., & Chen, Z. G. (2022). Multi-user edge computing task offloading scheduling and resource allocation based on deep reinforcement learning. *Chinese Journal of Computers*, 45(04), 812-824.
- [18] Lago, A., Sanz, M., Gordon, J. M., Feroso, J., Pizarro, P., Serrano, D. P., & Moreno, I. (2022). Enhanced production of aromatic hydrocarbons and phenols by catalytic co-pyrolysis of fruit and garden pruning wastes. *Journal of Environmental Chemical Engineering*, 10(3), 107738. <https://doi.org/10.1016/j.jece.2022.107738>
- [19] Putra, A. Z., Husein, A. M., & Wijaya, F. (2024). Comparison of Tuberculosis Detection Using CNN Models (AlexNet and ResNet). *Sinkron: jurnal dan penelitian teknik informatika*, 8(4), 2309-2317. <https://doi.org/10.33395/sinkron.v8i4.13979>
- [20] Yu, Y., Li, Y., Xie, F., Song, J., & Bai, Y. (2025). Simulation and experiment on a root cutting device for a chinese chard harvester. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 29(11), 1-8. <https://doi.org/10.1590/1807-1929/agriambi.v29n11e290063>

Author Biography



Trinugi Wira Harjanti was born in Jakarta on May 28th. In 2001 she graduates with bachelor's degree in informatics engineering from Sekolah Tinggi Teknologi TELKOM, Bandung. After graduating from her studies, she was appointed as teaching staff at NIIT Center in Jakarta. After several years of teaching, she continues her post graduate program and graduate with Master's in Computer Science program from STMIK Eresha in 2011, followed by Doctoral degree from Gunadarma University in 2020. Her research interest and expertise including artificial intelligence and software engineering with several published paper.