الجمهورية الشعبية الديمقراطية الجزائرية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للإعلام الآلي ٠08 ماي 1945، بسيدي بلعباس
École Supérieure en Informatique
-08 Mai 1945- Sidi Bel Abbès

THESIS

To obtain the diploma of **Engineer**
Field: **Computer Science**
Specialty: **Ingénierie des Systèmes Informatiques (ISI)**

Theme

AI-Powered Fish Freshness Estimator

Presented by:
Ahmed Amine Tiffrent

Submission Date: **June, 2025**
In front of the jury composed of:

Dr. Bendaoud Fayssal	President
Dr. Khaldi Belkacem	Supervisor
Dr. Bousmaha Rabab	Examiner

Academic Year : 2024/2025

Abstract

Ensuring fish freshness is a critical concern in the seafood industry due to its direct impact on food safety, consumer trust, and economic value. Traditional assessment methods based on human expertise and chemical analysis are often subjective, time-consuming, and impractical for real-time application. Variations in lighting, species, and storage conditions further complicate consistency.

This project proposes an AI-powered fish freshness classification system using image-based analysis of the fish eye. The system leverages the EfficientNetB3 architecture and deep learning techniques to classify samples into three categories: Fresh, Highly Fresh, and Not Fresh. To improve generalization, the model was trained on a balanced dataset with data augmentation and evaluated using metrics such as accuracy, F1-score, and Grad-CAM.

For practical deployment, the trained model is integrated into a user-friendly Streamlit web application, enabling real-time predictions through image uploads or webcam input. This solution eliminates human subjectivity, enhances quality control, and offers a scalable tool for fisheries, retailers, and consumers.

Keywords: Artificial Intelligence, Fish Freshness, Deep Learning, Computer Vision, EfficientNetB3, Streamlit, Food Safety.

Résumé

Garantir la fraîcheur du poisson est un enjeu essentiel dans l'industrie des produits de la mer, car elle impacte directement la sécurité alimentaire, la confiance des consommateurs et la valeur économique. Les méthodes d'évaluation traditionnelles, basées sur l'expertise humaine ou des analyses chimiques, sont souvent subjectives, lentes et peu adaptées à une utilisation en temps réel. Les variations de lumière, d'espèce et de stockage rendent les évaluations encore plus complexes.

Ce projet propose un système intelligent de classification de la fraîcheur du poisson basé sur l'analyse visuelle de l'œil. Le modèle repose sur l'architecture EfficientNetB3 et les techniques de deep learning pour classer les échantillons en trois catégories : Frais, Très Frais et Non Frais. L'apprentissage a été réalisé sur un jeu de données équilibré avec augmentation d'images, et l'évaluation a été faite à l'aide de métriques telles que la précision, le F1-score et Grad-CAM.

Pour une utilisation concrète, le modèle a été déployé sous forme d'une application web conviviale avec Streamlit, permettant des prédictions en temps réel via webcam ou images téléchargées. Cette solution supprime la subjectivité humaine, améliore le contrôle qualité, et constitue un outil évolutif pour les pêcheries, les détaillants et les consommateurs.

Mots-clés : Intelligence Artificielle, Fraîcheur du Poisson, Deep Learning, Vision par Ordinateur, EfficientNetB3, Streamlit, Sécurité Alimentaire.

الملخص

تُعدّ ضمان طزاجة الأسماك مسألة حاسمة في صناعة المنتجات البحرية، لما لها من تأثير مباشر على سلامة الغذاء وثقة المستهلك والقيمة الاقتصادية. وتعتمد الطرق التقليدية في التقييم على الخبرة البشرية أو التحاليل الكيميائية، وهي غالبًا ما تكون ذاتية وبطيئة وغير مناسبة للتطبيق الفوري، كما أن اختلافات الإضاءة والأنواع وظروف التخزين تُزيد من صعوبة ضمان الاتّساق.

يقترح هذا المشروع نظامًا ذكيًا لتصنيف طزاجة الأسماك اعتمادًا على تحليل بصري لعين السمكة. يعتمد النظام على نموذج EfficientNetB3 وتقنيات التعلم العميق لتصنيف العين إلى ثلاث فئات: طازجة، شديدة الطزاجة، وغير طازجة. تم تدريب النموذج على مجموعة بيانات متوازنة مع تعزيز الصور، وتم تقييمه باستخدام مقاييس مثل الدقة، وF1-score، وتقنية Grad-CAM للتفسير البصري.

ولتسهيل الاستخدام، تم دمج النموذج في تطبيق ويب تفاعلي باستخدام Streamlit، مما يسمح بالتنبؤ الفوري عبر تحميل الصور أو باستخدام الكاميرا. تساهم هذه المنظومة في القضاء على التحيّز البشري، وتحسين مراقبة الجودة، وتقديم أداة قابلة للتوسيع تُفيد المصايد والأسواق والمستهلكين.

الكلمات المفتاحية: الذكاء الاصطناعي، طزاجة الأسماك، مراقبة الجودة الآلية، الرؤية الحاسوبية، صناعة المنتجات البحرية، سلامة الغذاء، تحسين سلسلة التوريد.

Acknowledgement

First and foremost, I express my deepest gratitude to Allah for granting me the strength, perseverance, and knowledge to complete this work.

I would like to extend my sincere thanks to my supervisor, **M. Khaldi Belkacem**, for his invaluable guidance, constructive feedback, and continuous encouragement throughout the course of this research. His expertise and support have played a significant role in shaping this project.

I am also immensely grateful to my family for their unconditional love, patience, and support during this journey. Their belief in me has been a source of motivation and strength.

A special acknowledgment goes to my friends and colleagues for their encouragement, insightful discussions, and assistance during challenging times. The shared experiences and collaboration have made this journey memorable and enriching.

Finally, I extend my appreciation to the faculty and staff of **École Supérieure en Informatique -08 Mai 1945- Sidi Bel Abbès** for providing the necessary resources and a conducive learning environment that contributed to the successful completion of this work.

Ahmed Amine Tiffrent

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	Artificial Intelligence.
ANN	Artificial Neural Network.
AutoML	Automated Machine Learning.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
DL	Deep Learning.
DMA	Dimethylamine.
E-Nose	Electronic Nose.
E-Tongue	Electronic Tongue.
GPU	Graphics Processing Unit.
Grad-CAM	Gradient-weighted Class Activation Mapping.
KNN	k-Nearest Neighbors.
LDA	Linear Discriminant Analysis.
LSTM	Long Short-Term Memory.
ML	Machine Learning.
MobileNet	Lightweight Deep Learning Model.
NNs	Neural Networks.
PCA	Principal Component Analysis.
ResNet	Residual Neural Network.
RF	Random Forest.
RNN	Recurrent Neural Network.
SHAP	SHapley Additive Explanations.
SqueezeNet	Compact CNN Architecture.
SVM	Support Vector Machine.

TF Lite	TensorFlow Lite.
TMA	Trimethylamine.
TVB-N	Total Volatile Base Nitrogen.

INTRODUCTION

Motivation

Ensuring the freshness of fish is a critical factor in the seafood industry, directly impacting food safety, consumer satisfaction, and economic value. The perishable nature of fish makes rapid quality degradation inevitable, requiring reliable and accurate methods to assess freshness. Traditional methods such as sensory evaluation, which rely on human expertise, are often subjective, inconsistent, and unsuitable for large-scale operations. With advancements in Artificial Intelligence (AI), particularly in computer vision and deep learning, it has become possible to automate freshness assessment, reducing human bias and increasing scalability. This shift holds significant potential for modernizing quality control processes in fisheries and the seafood supply chain.

Problem & Solution

The primary problem in fish freshness assessment lies in the reliance on manual, subjective, and time-consuming techniques. Variations in environmental conditions, individual expertise, and storage practices contribute to inconsistent results. Moreover, traditional chemical and microbial testing methods, while accurate, are costly and impractical for real-time or consumer-level use.

The proposed solution is to leverage computer vision and deep learning technologies to develop an AI-powered system capable of analyzing visual indicators such as fish eyes, gills, and skin. This system will provide objective, consistent, and efficient freshness classifications (e.g., *Fresh*, *Moderately Fresh*, *Spoiled*). By integrating this technology into a web or mobile application, the system can be accessible to fisheries, retailers, and consumers alike.

Research Objectives

- Develop a deep learning-based system for fish freshness classification using computer vision techniques.
- Analyze key visual freshness indicators (e.g., fisheye clarity, gill color, skin texture) and their correlation with quality levels.

- Create a scalable, user-friendly web or mobile application for real-time fish freshness assessment.
- Address the limitations of traditional methods by providing an automated, non-invasive, and consistent solution.

Limitations

- The model's accuracy may vary across different fish species, requiring species-specific adjustments.
- Dependence on high-quality input images; suboptimal lighting or blurry images may reduce prediction reliability.
- Limited applicability to non-visual freshness indicators (e.g., smell or texture) not captured by computer vision.

Thesis Organization

This thesis is organized into nine chapters as follows:

- **Chapter 1: Foundations of Artificial Intelligence and Image Processing** This chapter introduces the core principles of Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL), with an emphasis on their relevance to image-based classification. It explains neural networks, convolutional neural networks (CNNs), and their advantages in image recognition tasks. The chapter also covers essential image processing techniques such as normalization, resizing, cropping, data augmentation, and feature extraction, forming the backbone of AI-powered visual analysis.
- **Chapter 2: AI in Quality Assessment** This chapter explores the practical applications of AI in assessing food quality, particularly in meat, fruits, vegetables, and fish. It discusses key visual and chemical indicators of freshness and how AI models—such as computer vision and sensor-based systems—enable accurate and non-invasive assessment. Challenges in AI-based classification such as class imbalance, generalization across fish species, and lighting variations are also addressed, along with recent advancements like transfer learning and explainability tools.
- **Chapter 3: Introduction** This chapter lays the foundation of the research by presenting the problem statement, objectives, and motivation behind developing an AI-powered fish freshness estimation system. It highlights the gaps in current assessment methods, the relevance of deep learning in this domain, and the specific contributions of this work. It also includes a detailed overview of the model development trials that guided the final approach.
- **Chapter 4: Dataset and Model Design** This chapter provides a comprehensive overview of the dataset used (FFE dataset), including its structure and class balancing strategies. It details the image preprocessing and augmentation steps performed to enhance model robustness. The chapter explains the rationale behind choosing EfficientNetB3 as the base model and describes the custom head designed for transfer

learning. It concludes with a discussion on model compilation and the selected loss function.

- **Chapter 5: Training and Evaluation** This chapter describes the training pipeline, including a two-phase training strategy and the use of callbacks like EarlyStopping and learning rate scheduling. It presents the model's performance on training and validation datasets using metrics such as accuracy, confusion matrix, and ROC curves. It also includes Grad-CAM visualizations to interpret model decisions and concludes with the final test performance on unseen data.
- **Chapter 6: Application and Conclusion** This chapter focuses on the deployment of the trained model into a user-friendly Streamlit web application. It outlines the application's workflow from image upload to prediction and feedback. The chapter discusses key achievements and improvements of the system, as well as its limitations and suggestions for future work.

Part I

Background

CHAPTER 1

FOUNDATIONS OF ARTIFICIAL INTELLIGENCE AND IMAGE PROCESSING

1.1 Introduction

Artificial Intelligence (AI) has become a transformative force across various fields, enabling machines to mimic human cognitive functions such as perception, learning, and decision-making. Within AI, Machine Learning (ML) and Deep Learning (DL) have emerged as powerful subfields, allowing systems to learn patterns from data and make intelligent predictions.

In recent years, these technologies have played a significant role in computer vision and image processing, enabling applications ranging from medical diagnostics to autonomous vehicles. Deep learning models, especially Convolutional Neural Networks (CNNs), have demonstrated exceptional capabilities in image classification, object detection, and segmentation.

This chapter presents the foundational concepts of AI and its subfields, followed by key techniques in image preprocessing and feature extraction. It also introduces evaluation metrics essential for measuring the performance of AI models in classification tasks. The goal is to establish a strong theoretical foundation that supports the development of intelligent vision-based systems, such as the fish freshness classification system explored in this study.

Throughout this chapter, practical techniques are introduced alongside theoretical concepts, providing a comprehensive understanding of how AI and image processing intersect to solve real-world problems effectively.

1.2 Foundations of Artificial Intelligence

Artificial Intelligence (AI) is a multidisciplinary field at the intersection of computer science, mathematics, neuroscience, and cognitive science. It aims to design systems that can mimic human cognitive functions, including reasoning, learning, perception, and decision-making. Within the domain of AI, two major subfields have emerged and revolutionized the development of intelligent systems: Machine Learning (ML) and Deep Learning (DL).

1.2.1 Defining AI, ML, and DL

1.2.1.1 Artificial Intelligence (AI):

AI is the broadest term referring to machines or systems that can simulate aspects of human intelligence. The capabilities of AI range from rule-based expert systems to sophisticated models that can interpret images, play chess, or generate human-like language [1].

1.2.1.2 Machine Learning (ML):

ML is a subset of AI that focuses on algorithms that improve automatically through experience. Instead of being explicitly programmed, ML models learn patterns from data to make predictions or decisions. The key categories include:

- **Supervised Learning:** Models are trained using labeled datasets.
- **Unsupervised Learning:** Models discover hidden patterns in data without labels.
- **Reinforcement Learning:** Agents learn by interacting with an environment and receiving feedback.

1.2.1.3 Deep Learning (DL):

DL is a subfield of ML based on artificial neural networks with many layers (deep architectures). It is particularly effective for unstructured data such as images, audio, and text. DL models can automatically extract hierarchical features and are the foundation of modern AI breakthroughs, such as image classification, speech recognition, and generative models [2].

1.2.2 Neural Networks and Learning Mechanisms

1.2.2.1 Artificial Neural Networks (ANNs):

ANNs are computational models inspired by the human brain's network of neurons. They consist of layers: input, hidden, and output. Each neuron receives input, applies a transformation (usually a weighted sum followed by an activation function), and passes the result to the next layer. ANNs are the building blocks of many ML and DL models [2].

1.2.2.2 Training and Optimization:

Neural networks are trained using datasets with the goal of minimizing prediction error. This process begins with forward propagation, where the network computes output predictions based on the input data. The loss function is then used to measure the difference between the predicted outputs and the true values. To improve the model, backpropagation calculates the gradients of the loss with respect to the network's weights. Finally, gradient descent is applied to update the weights in a direction that minimizes the loss, progressively enhancing the network's performance.

1.2.3 Convolutional Neural Networks (CNNs)

CNNs are specialized neural networks designed for processing visual data. Unlike traditional ANNs, CNNs leverage the spatial structure of images. They use convolutional layers to extract features such as edges and shapes, pooling layers to reduce dimensionality, and fully connected layers for classification. The overall architecture of a typical CNN used for image classification is illustrated in Figure 1.1.

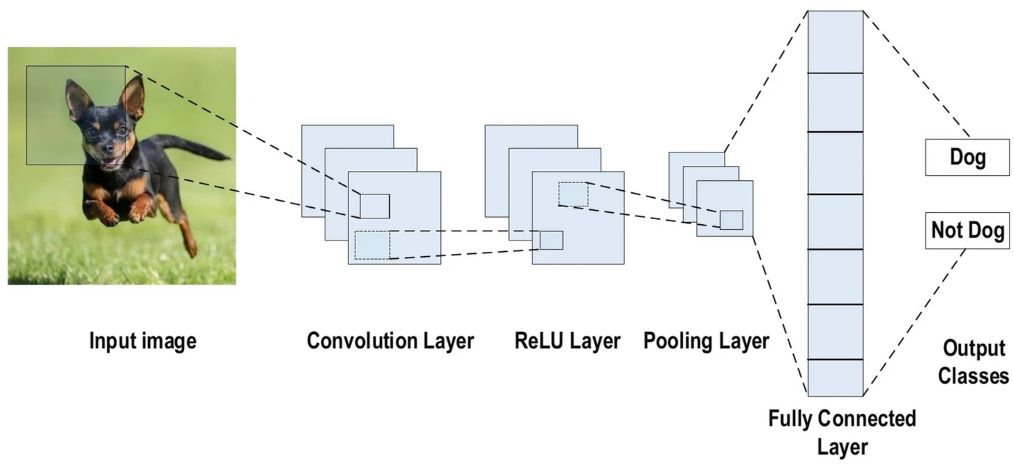


Figure 1.1: CNN architecture for image classification [3]

1.2.3.1 Advantages of CNNs:

- Automatic feature extraction from raw image data.
- Parameter sharing and local connectivity, reducing complexity.
- Translation and scale invariance.

CNNs are especially effective in image-based tasks, such as those used in this project for classifying fish freshness based on eye characteristics.

1.2.4 Real-World Applications of AI

Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) have penetrated nearly every domain of modern technology. In healthcare, they are used for diagnosis through medical imaging, drug discovery, and personalized treatment plans. In the finance sector, they enhance risk assessment, detect fraud, and enable algorithmic trading. Transportation has seen advancements through the development of autonomous vehicles and improved traffic management systems.

In retail, intelligent systems analyze customer behavior and power recommendation engines. Furthermore, in environmental science, they are applied to predict climate changes and monitor wildlife populations. These wide-ranging applications highlight the flexibility and profound impact of intelligent systems across industries.

1.2.5 AI in Fish Freshness Estimation

In this project, we leverage AI—specifically CNN-based deep learning—for the automated assessment of fish freshness. This approach enables accurate, fast, and objective grading, replacing subjective and time-consuming traditional methods. The system processes images of fish eyes and classifies them into freshness levels: Fresh, Moderately Fresh, or Spoiled.

> The fusion of AI and computer vision enables transformative applications in food quality control, and this project exemplifies its use in the fisheries industry.

1.3 Image Processing Techniques in Computer Vision

Image processing is a fundamental step in the computer vision pipeline, as it prepares and enhances images to improve the performance of machine learning and deep learning models. This section explores key techniques used in image preprocessing and feature extraction, which play a crucial role in improving model accuracy and generalization.

1.3.1 Image Preprocessing Techniques

Preprocessing techniques ensure that input data is consistent, noise-free, and tailored to the input requirements of neural networks. Proper preprocessing helps reduce computational costs, improve model convergence, and standardize data across the dataset.

1.3.1.1 Normalization

Normalization scales pixel values to a specific range (commonly [0, 1]) to reduce inconsistencies in image intensities caused by lighting variations. Deep learning models such as CNNs benefit from normalized inputs as it accelerates convergence and avoids gradient-related issues.

Common Normalization Methods:

– **Min-Max Normalization:**

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

– **Z-score Normalization:**

$$x' = \frac{x - \mu}{\sigma}$$

1.3.1.2 Image Resizing and Cropping

Neural networks require fixed-size inputs. Resizing ensures all input images are of the same shape (e.g., 224×224 or 299×299), while cropping helps remove irrelevant background, focusing on the object of interest.

Two cropping strategies were employed to optimize the model's ability to focus on relevant visual features. Center cropping was used to systematically remove border regions and retain the central portion of the image, where the fish eye is typically located. This ensures that critical anatomical details are preserved. On the other hand, random cropping was applied during training to introduce variability and improve the model's robustness. By exposing the model to slightly different portions of the image in each epoch, random cropping helps mitigate overfitting and enhances generalization.

1.3.1.3 Data Augmentation

Data augmentation increases the diversity of training data by applying random transformations. This simulates real-world variations and improves the generalization ability of the model.

To improve the model's generalization and robustness, several common image augmentation techniques were applied during training. Rotation was used to simulate different orientations of the fish eye, ensuring the model learns to recognize features from multiple angles. Flipping, both horizontal and vertical, introduced mirrored perspectives to further increase data diversity. Brightness and contrast adjustments mimicked varying lighting conditions typically encountered in real-world settings. Lastly, zooming and shearing operations helped the model adapt to scale changes and subtle perspective shifts, making it more resilient to variability in image capture.

1.3.2 Feature Extraction Techniques

After preprocessing, feature extraction identifies the most informative aspects of an image. These features serve as inputs for classification models, helping to reduce dimensionality and focus on essential patterns.

1.3.2.1 Edge Detection

Edge detection algorithms highlight transitions in intensity, which often correspond to object boundaries or texture changes.

Several classical edge detection algorithms were explored to highlight structural features within fish eye images. The Sobel operator calculates intensity gradients in horizontal and vertical directions, effectively capturing edge transitions. The Canny edge detector employs a multi-stage process—including noise reduction, gradient calculation, non-maximum suppression, and edge tracking—to produce accurate and robust edge maps. Additionally, the Prewitt operator, which is conceptually similar to Sobel, offers a simpler approach for detecting gradients and delineating basic edge structures in images.

1.3.2.2 Color Analysis

Color analysis helps in classifying and segmenting images based on hue, saturation, and brightness, which can indicate freshness or ripeness in biological images.

Color-based feature extraction techniques are commonly used in image analysis. Color histograms provide a quantitative representation of the distribution of colors within an image, which can be useful for distinguishing freshness levels based on visual cues. Additionally, HSV color segmentation allows for more effective color isolation, enabling the model to detect specific hues such as redness, which is often associated with spoilage in fish.

1.3.2.3 Texture and Shape Features

For tasks such as fish freshness estimation, analyzing the texture of gills or the shape of the eyes can be crucial.

Texture-based feature extraction methods are essential for capturing surface patterns and fine details in images. Gabor filters are widely used to extract texture information by analyzing image content at various orientations and scales, making them suitable for biological textures such as those found in fish eyes. Meanwhile, the Histogram of Oriented Gradients (HOG) technique encodes the structure and shape of objects by computing the distribution of gradient directions, which helps in identifying visual patterns relevant to classification tasks.

1.3.3 Role of Image Processing in Fish Freshness Estimation

In our system, image processing ensures that fisheye images are clean, standardized, and informative. Techniques such as resizing, normalization, and augmentation improve model learning. Feature extraction from eye clarity, color intensity, and shape contributes to the accurate classification of freshness into categories like *Fresh*, *Moderately Fresh*, or *Spoiled*. The examples are illustrated in Figure 1.2.

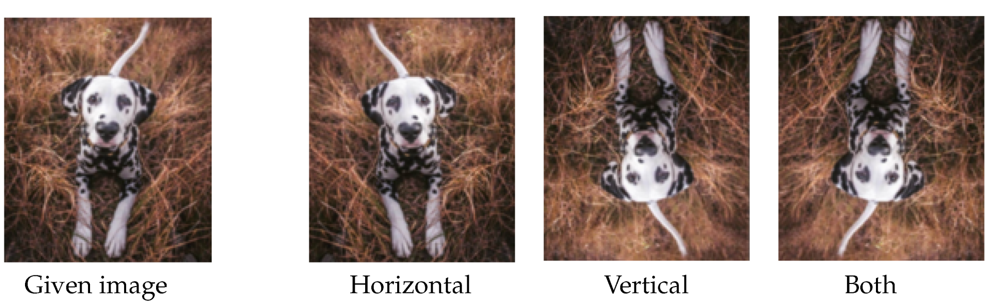


Figure 1.2: Examples of data augmentation techniques [4]

1.4 Evaluation Metrics for AI Models

Evaluating the performance of AI models is essential to understanding how well they perform on real-world data. In classification tasks, such as fish freshness prediction, different metrics provide different insights. This section explains the most commonly used evaluation metrics, their mathematical definitions, and their relevance in practice.

1.4.1 Classification Metrics Overview

When evaluating models in supervised classification tasks, we use several metrics to assess different aspects of performance. These metrics rely on the confusion matrix, which summarizes the counts of:

- **True Positives (TP):** Correctly predicted positive cases.
- **True Negatives (TN):** Correctly predicted negative cases.
- **False Positives (FP):** Incorrectly predicted positive cases.
- **False Negatives (FN):** Incorrectly predicted negative cases.

1.4.1.1 Accuracy

Accuracy measures the proportion of correctly predicted samples out of the total number of predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

It provides a general idea of model performance but may be misleading in imbalanced datasets.

1.4.1.2 Precision

Precision focuses on the quality of positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP}$$

It answers the question: "Of all predicted positives, how many were correct?"

1.4.1.3 Recall (Sensitivity)

Recall, also known as sensitivity or true positive rate, measures the model's ability to detect all positive samples:

$$\text{Recall} = \frac{TP}{TP + FN}$$

It answers the question: "Of all actual positives, how many did the model correctly identify?"

1.4.1.4 F1 Score

The F1 score is the harmonic mean of precision and recall. It provides a balanced measure when there is a trade-off between the two:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 is especially useful in imbalanced classification problems, where accuracy alone can be deceptive.

1.4.1.5 Confusion Matrix

The confusion matrix is a visual tool used to summarize prediction results. It shows the counts of actual vs predicted labels, making it easier to see where misclassifications occur. Each row represents the actual class, while each column represents the predicted class.

1.4.1.6 Macro vs Micro Averaging (Multiclass)

For multiclass classification (e.g., Fresh, Moderately Fresh, Spoiled), we compute macro-averaged and micro-averaged metrics:

- **Macro-average:** Calculates metric independently for each class, then takes the average. It treats all classes equally.
- **Micro-average:** Aggregates contributions of all classes to compute the metric globally. It favors larger classes.

1.4.2 ROC Curve and AUC (for Binary Classification)

The Receiver Operating Characteristic (ROC) curve plots the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds. The Area Under the Curve (AUC) summarizes the ROC curve into a single score:

$$\text{AUC} \in [0.5, 1.0]$$

A model with AUC = 1.0 is perfect; AUC = 0.5 is random guessing. ROC-AUC is primarily used for binary classification.

1.4.3 Why These Metrics Matter in Our Project

In our fish freshness classification system, we rely on accuracy, precision, recall, and the F1 score to measure how well the model performs in distinguishing between freshness levels. A confusion matrix and F1 score help detect which classes the model struggles with most — such as misclassifying moderately fresh as spoiled, or vice versa — which is critical for real-world application.

Example: If the model has high accuracy but low recall for the "Spoiled" class, it may frequently fail to identify spoiled fish — which is unacceptable in food safety scenarios. Thus, a combination of metrics provides a holistic view of model performance.

CHAPTER 2

AI IN QUALITY ASSESSMENT

2.1 Applications of AI in Food Quality Control

Artificial Intelligence (AI) has become a game-changer in food quality control by enabling faster, more accurate, and scalable solutions for assessing the freshness and safety of food products. Traditionally, food inspection methods relied heavily on human sensory evaluation or laboratory testing, which can be subjective, labor-intensive, and impractical for large-scale operations. The integration of AI—particularly machine learning (ML) and deep learning (DL)—has opened new avenues for automating quality assessment through image analysis, sensor data interpretation, and pattern recognition [2, 5].

Modern AI systems can detect subtle variations in texture, color, and chemical composition that are invisible to the human eye. These systems not only improve the accuracy of assessments but also reduce operational costs and ensure consistency in food safety standards.

This section explores AI applications in the evaluation of meat, fruits, vegetables, and fish—highlighting the methods, technologies, and benefits associated with each domain. Figure 2.1 presents a general overview.

2.1.1 Meat Freshness Detection

Meat freshness is vital for ensuring food safety, nutritional value, and consumer trust. Spoiled meat can harbor harmful bacteria such as Salmonella or E. coli, posing serious health risks. AI-based techniques now offer objective, non-invasive, and real-time meat evaluation using visual and chemical cues [6].

2.1.1.1 Techniques Used in AI-Based Meat Freshness Detection:

- **Computer Vision:** Deep CNNs analyze images of meat for changes in surface color, texture granularity, and marbling patterns to estimate spoilage levels.

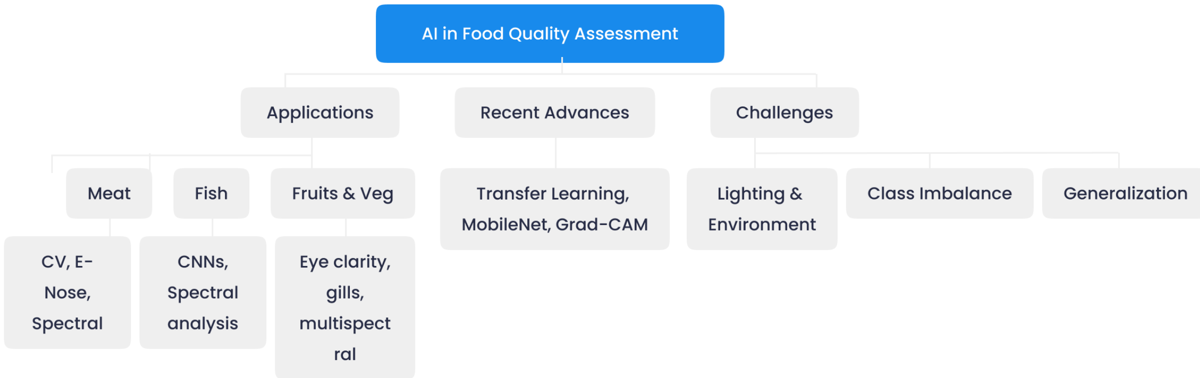


Figure 2.1: AI in quality assessment

- **Spectral Imaging:** AI interprets hyperspectral data to detect internal changes in water content and myoglobin concentration—indicators of freshness.
- **Electronic Nose (E-Nose):** AI-driven olfactory sensors mimic human smell to detect volatile compounds produced during decomposition.

2.1.1.2 Benefits:

Computer vision and AI techniques provide rapid and accurate meat classification without the need for destructive sampling. They enable continuous monitoring throughout slaughterhouses and cold chain logistics, ensuring that freshness and quality standards are maintained. Additionally, these technologies help businesses ensure regulatory compliance and reduce the risk of costly product recalls.

2.1.2 Fruit and Vegetable Quality Estimation

The appearance and internal quality of fruits and vegetables are key determinants of market value and consumer satisfaction. AI plays a central role in automating produce grading, sorting, and defect detection by leveraging visual and spectral cues [7].

2.1.2.1 Key AI Techniques for Fruit and Vegetable Quality Estimation:

Image Processing techniques are employed wherein machine learning models extract color histograms and texture descriptors to assess ripeness levels or detect bruises and other defects in produce. In parallel, **Convolutional Neural Networks (CNNs)** are widely used for classification tasks, enabling the identification of overripe, spoiled, or damaged produce through detailed surface anomaly detection. Additionally, **Spectroscopy and Sensor Fusion** approaches integrate data from sources such as near-infrared (NIR) spectroscopy, moisture sensors, and traditional visual inputs, allowing AI models to deliver a more holistic and accurate prediction of overall produce quality.

2.1.2.2 Advantages:

Computer vision systems play a crucial role in the agricultural sector by reducing post-harvest losses through the early identification of defective items. They enhance grading accuracy in packing and distribution centers, ensuring that only high-quality produce reaches consumers. Furthermore, these technologies facilitate dynamic pricing strategies and improve shelf-life prediction, optimizing both inventory management and profitability.

2.1.3 Adaptation to Fish Freshness Detection

Fish freshness degrades rapidly after harvesting, making it essential to evaluate its quality promptly and accurately. Traditional methods like the Torry sensory scale or chemical assays (e.g., TVB-N, TMA) are reliable but slow and labor-intensive. AI models, particularly CNN-based vision systems, offer faster, scalable alternatives suitable for use in processing facilities, markets, and even consumer mobile apps [8].

2.1.3.1 Visual and Chemical Indicators of Fish Freshness:

Eye Clarity and Convexity serve as critical indicators of fish freshness, where a fresh fish typically displays clear, convex eyes, while cloudiness and a flattened appearance suggest the onset of spoilage. Similarly, **Gill Color and Texture** are vital parameters; bright red gills are associated with high freshness, whereas darkened, slimy, or discolored gills point to progressive decay. Lastly, **Skin Glossiness and Firmness** provide visual and tactile cues, with the loss of surface shine and a decrease in skin firmness signaling a decline in overall fish quality.

2.1.3.2 AI Approaches for Fish Freshness Estimation:

Computer Vision with CNNs is widely applied to analyze critical regions such as the fish eye and gills, classifying freshness levels with the help of pretrained convolutional models like ResNet or EfficientNet. This approach enables automated, objective evaluations based on subtle visual features. In parallel, **Electronic Nose (E-Nose)** technologies detect gases related to spoilage, such as hydrogen sulfide and ammonia, using AI algorithms to interpret data from gas sensor arrays with high precision. Furthermore, **Multispectral Imaging** merges reflectance and fluorescence information, enhancing the detection of early biochemical spoilage markers that are not visible to the naked eye. By integrating AI models, multispectral imaging offers a deeper and more sensitive analysis of food freshness.

2.1.3.3 Benefits:

Computer vision-based freshness assessment offers an objective and repeatable method for evaluating food quality. It enables non-invasive quality checks during transportation and sale, ensuring that products maintain their integrity throughout the supply chain. Additionally, this approach aligns with sustainable practices by minimizing food waste and reducing the need for chemical preservatives.

In conclusion, AI is revolutionizing food quality control by offering intelligent, rapid, and scalable solutions. In the case of fish, it allows real-time classification of freshness based on physical and chemical indicators, ultimately supporting both industry and consumer needs.

—

2.2 Challenges in AI-Based Fish Freshness Detection

Despite the advancements in **AI** for fish freshness estimation, several challenges hinder its widespread adoption and effectiveness. These challenges primarily arise from environmental variability, data imbalances, and generalization issues across different fish species. Addressing these challenges is crucial for improving the accuracy and reliability of **AI**-based freshness detection systems [8, 2].

—

2.2.1 Variations in Lighting and Environmental Factors

Lighting conditions significantly affect the performance of **AI**-based fish freshness detection systems. Inconsistent lighting introduces variations in image brightness, contrast, and color, which can mislead machine learning models [7]. Additionally, environmental factors such as temperature and humidity influence fish appearance and can affect the accuracy of predictions.

2.2.1.1 Key Issues:

- Shadows, glare, and reflections in images distort key freshness indicators.
- Temperature fluctuations affect fish eye clarity and gill coloration.
- Inconsistent backgrounds introduce noise, leading to potential misclassification.

2.2.1.2 Potential Solutions:

- Use **adaptive image preprocessing** techniques such as histogram equalization to normalize brightness and contrast.
- Implement **data augmentation** (brightness adjustments, white balance correction) to make models robust to lighting variations.
- Employ **multi-spectral imaging** to capture more reliable spectral data for freshness estimation.

—

2.2.2 Class Imbalance in Datasets

One of the major challenges in training **AI** models for fish freshness detection is the imbalance in dataset distribution. In many cases, datasets contain significantly more images of fresh fish compared to spoiled fish, leading to biased model predictions [9].

2.2.2.1 Impact of Class Imbalance:

Models tend to favor the majority class, such as fresh fish, while often failing to accurately predict minority classes like moderately fresh or spoiled fish. This imbalance results in poor model generalization, leading to an increased number of false negatives, which can compromise food safety by incorrectly labeling spoiled fish as safe. Furthermore, imbalanced datasets reduce training effectiveness, causing models to learn suboptimal decision boundaries that hinder overall classification performance.

2.2.2.2 Techniques to Address Class Imbalance:

Several strategies can be employed to address class imbalance. **Data Augmentation** involves the synthetic generation of minority-class images through transformations such as flipping and rotation, enriching dataset diversity. **Resampling Methods** include techniques like oversampling the minority class using **SMOTE (Synthetic Minority Over-sampling Technique)** or undersampling the majority class to achieve balance. **Cost-Sensitive Learning** introduces higher misclassification penalties for underrepresented classes, encouraging the model to focus more on difficult cases. Additionally, **Class-Balanced Loss Functions** adjust the loss computation during training to compensate for unequal class distributions, leading to fairer and more accurate learning.

2.2.3 Generalization Across Fish Species and Sizes

Fish species exhibit diverse physical characteristics, including variations in eye clarity, gill structure, and body texture. **AI** models trained on specific fish species may struggle to generalize across different species and fish sizes [8].

2.2.3.1 Key Challenges:

- Variations in **eye size, color intensity, and gill structures** lead to inconsistent model predictions.
- Fish of different sizes may require different preprocessing steps to standardize features.
- Species with naturally cloudy eyes may be incorrectly classified as **”spoiled”** due to dataset biases.

2.2.3.2 Potential Solutions:

- Train models on **diverse multi-species datasets** to improve robustness.
- Use **transfer learning** with pre-trained models to adapt to different fish species.

– Implement **feature extraction standardization** techniques to ensure model consistency across fish sizes.

Improving model generalization is essential to ensure reliable **AI**-powered freshness detection across different fish species and environmental conditions.

—

2.3 Recent Advancements in AI-Based Classification

Advancements in **AI**-based classification have significantly improved the accuracy, efficiency, and interpretability of machine learning models in various applications, including fish freshness detection. These advancements include the use of **pretrained models** for transfer learning, **lightweight architectures** like **MobileNet** and **SqueezeNet**, and **explainability techniques** such as **Grad-CAM** [2, 10].

—

2.3.1 Pretrained Models for Transfer Learning

2.3.1.1 Concept of Transfer Learning:

Transfer learning allows **AI** models to leverage knowledge from **pre-trained networks**, reducing the need for large datasets and extensive training. A model trained on a large dataset like **ImageNet** can be fine-tuned for specific applications, such as fish freshness classification, by modifying its final layers [11]. See Figure 2.2.

2.3.1.2 Advantages:

Transfer learning offers multiple advantages in deep learning applications. It reduces computational cost and training time by leveraging pre-trained models instead of training from scratch. Additionally, it improves model accuracy on small and specialized datasets by transferring knowledge from large-scale datasets. Finally, it prevents overfitting by utilizing robust pre-learned features, ensuring that models generalize better even when limited training data is available.

2.3.1.3 Commonly Used Pretrained Models:

– **ResNet (Residual Networks)** – Suitable for deep learning tasks with minimal degradation in accuracy.
– **VGG (Visual Geometry Group Networks)** – Effective for image classification tasks.
– **EfficientNet** – Optimized for high performance with fewer parameters.

Transfer learning is particularly beneficial in fish freshness detection, where obtaining large annotated datasets is challenging.

—

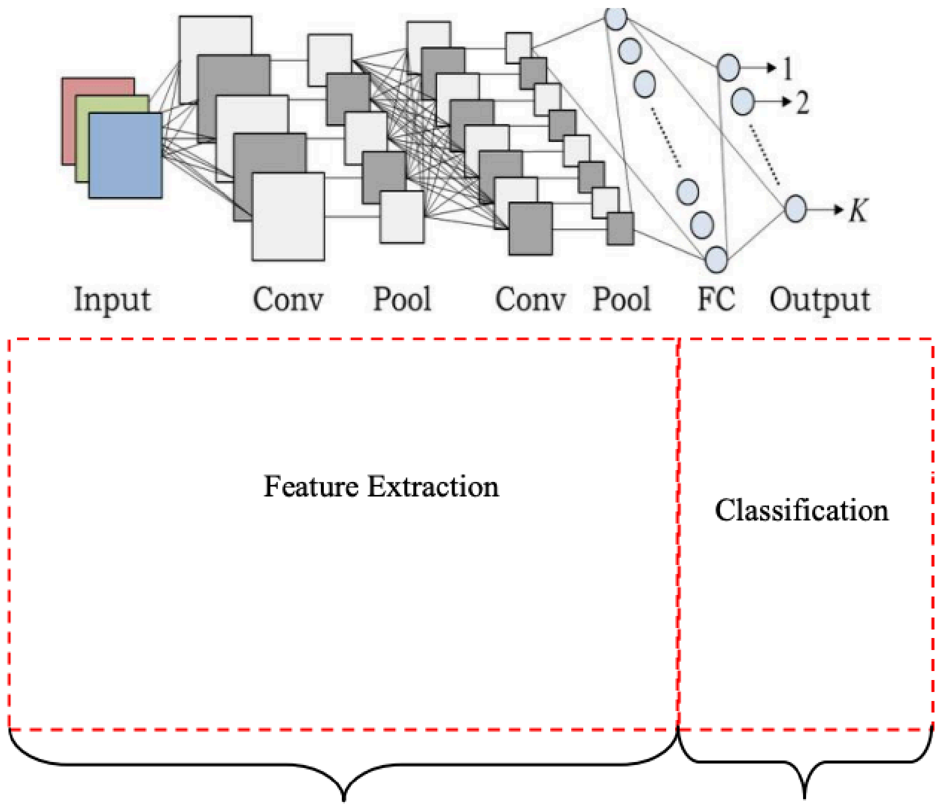


Figure 2.2: Transfer-learning [12]

2.3.2 Use of Models (e.g., MobileNet, SqueezeNet, EfficientNet...)

Lightweight models such as **MobileNet** and **SqueezeNet** have been developed to enhance computational efficiency, making **AI**-based classification feasible on edge devices and mobile applications [13].

2.3.2.1 MobileNet:

MobileNet is designed for real-time image classification with reduced computational cost. It employs **depthwise separable convolutions** to minimize the number of parameters while maintaining high accuracy. See Figure 2.3.

2.3.2.2 SqueezeNet:

SqueezeNet is a compact neural network architecture that significantly reduces the number of parameters while achieving similar accuracy levels to larger models. It uses **fire modules**, which replace conventional convolutional layers with squeeze and expand operations to improve efficiency. As presented in Figure 2.4.

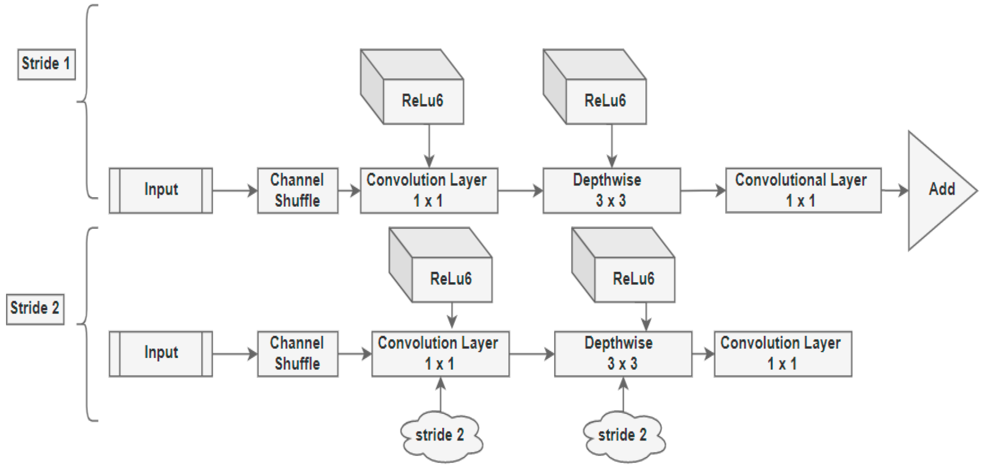


Figure 2.3: MobileNet Architecture [14]

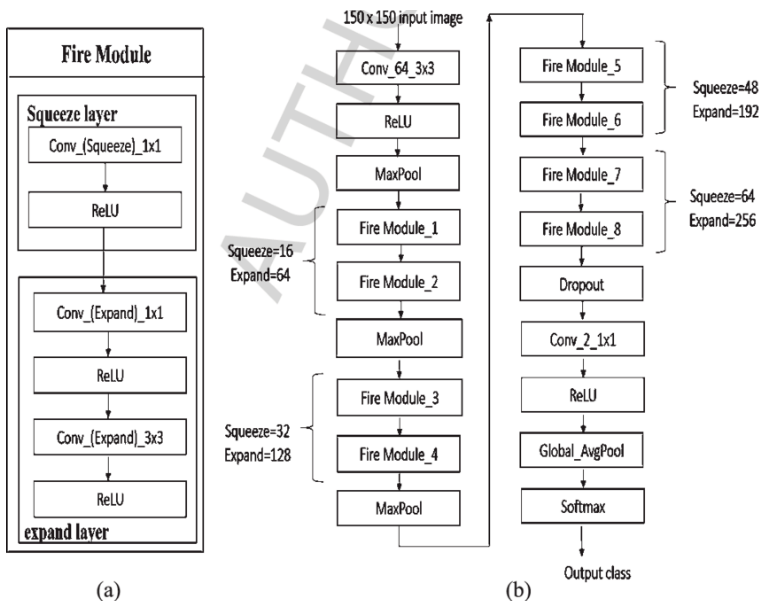


Figure 2.4: Squeeze-net Architecture [15]

2.3.2.3 EfficientNet

EfficientNet is a family of CNN architectures developed by Google that balances model accuracy and computational efficiency. It uses a compound scaling method that uniformly scales **depth, width, and resolution**. In fish freshness classification, EfficientNet has shown high performance with fewer parameters, making it ideal for applications requiring both speed and precision. As presented in Figure 2.5.

2.3.2.4 ResNet

ResNet (Residual Network) introduced the concept of **residual connections**, allowing much deeper networks to be trained effectively by mitigating the vanishing gradient problem. ResNet models are known for their robustness and accuracy, especially in complex classification tasks such as identifying subtle changes in fish eyes and gills. As presented in Figure 2.6.

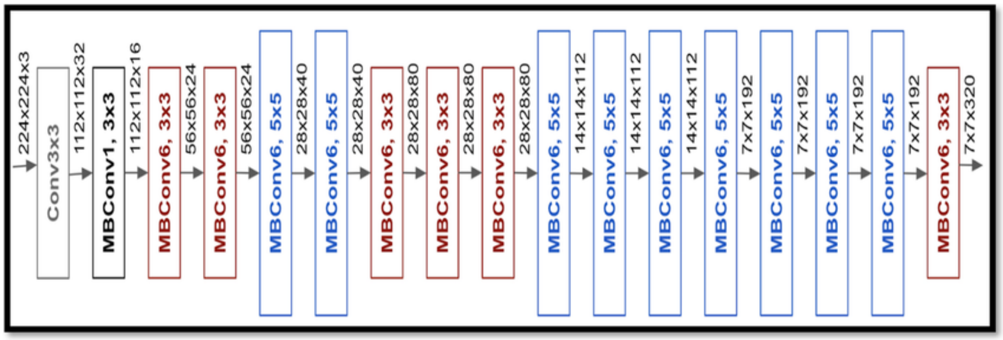


Figure 2.5: EfficientNet Architecture [16]

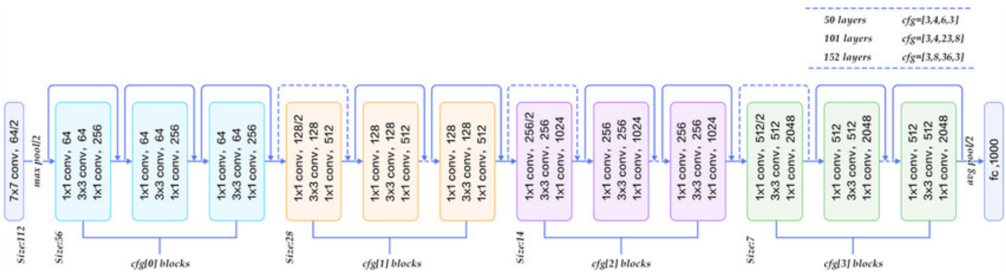


Figure 2.6: Resnet-net Architecture [17]

2.3.2.5 Advantages of Lightweight Models:

Lightweight models provide significant advantages in practical AI deployments. They offer a **reduced memory footprint**, making them suitable for mobile and embedded applications. Additionally, they deliver **faster inference time**, which is essential for real-time applications. Moreover, they have a **lower computational cost**, requiring less processing power compared to deeper architectures such as **ResNet**.

These lightweight models are ideal for fish freshness detection in real-world scenarios, where real-time analysis and deployment on portable devices are required.

—

2.3.3 Explainability Techniques (e.g., Grad-CAM)

Despite the high accuracy of deep learning models, their **black-box nature** makes it difficult to interpret how decisions are made. **Explainability techniques** such as **Gradient-weighted Class Activation Mapping (Grad-CAM)** enhance transparency in **AI**-based classification systems [18].

2.3.3.1 What is Grad-CAM?

Grad-CAM visualizes the regions of an image that influence a model's prediction. It overlays a **heatmap** on the input image, highlighting the most important features that contribute to the classification decision.

2.3.3.2 How Grad-CAM Works:

1. Computes the gradient of the output category with respect to feature maps of the last convolutional layer.
2. Weighs these gradients to understand the impact of different spatial locations.
3. Generates a heatmap that highlights **important regions** in the input image.

Figure 2.7 explains more how Grad-CAM works:

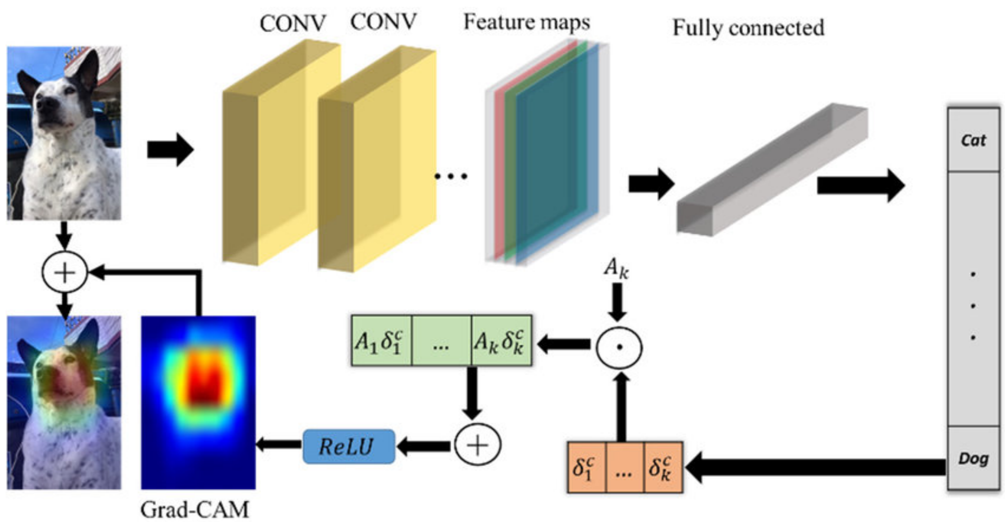


Figure 2.7: Grad-CAM [19]

2.3.3.3 Benefits of Explainability Techniques:

Visualization techniques play a critical role in enhancing the transparency of AI models. They **improve trust and interpretability** in **AI**-based classification tasks, allowing users to better understand model behavior. Additionally, they help detect **biases and model errors** during decision-making processes. Furthermore, visualizations assist researchers in refining **model architecture** and improving the effectiveness of **feature extraction**.

Grad-CAM is particularly useful in fish freshness classification, where decision-making transparency is essential for quality control and regulatory compliance.

Part II

Contribution

CHAPTER 3

INTRODUCTION

3.1 Introduction

Ensuring the freshness of fish is a critical concern in the food industry due to its direct impact on consumer safety, product quality, and market value. Spoiled or degraded fish not only pose health risks due to microbial contamination and biochemical spoilage compounds but also result in significant economic losses across the supply chain. Traditional methods of freshness assessment—such as manual inspection based on sensory attributes (e.g., eye clarity, gill color, skin texture) or laboratory-based chemical analyses (e.g., TVB-N, TMA, or pH tests)—are widely used but suffer from several limitations. These approaches are often subjective, time-consuming, require specialized expertise or equipment, and are not suitable for real-time, large-scale deployment.

With the rapid advancement of Artificial Intelligence (AI) and particularly computer vision, there is now an opportunity to develop automated systems capable of delivering objective, fast, and consistent assessments of fish freshness. In recent years, Convolutional Neural Networks (CNNs) have demonstrated strong performance in image classification tasks across various domains, including medical imaging, agriculture, and food quality control. By learning complex patterns and visual cues, CNNs can be trained to distinguish between subtle differences in freshness levels based on image data alone.

This project proposes the design and implementation of an AI-powered fish freshness estimation system that utilizes images of fish eyes—a well-recognized visual indicator of spoilage—to classify samples into three predefined categories: **Fresh**, **Highly Fresh**, and **Not Fresh**. The system is based on a deep learning model built using the EfficientNetB3 architecture, selected for its balance between high accuracy and computational efficiency. The model is trained on a carefully curated and augmented dataset of labeled fish eye images to ensure robustness and generalizability.

To enhance accessibility and usability, the trained model will be integrated into a streamlined web-based interface, allowing users—such as fish vendors, quality inspectors, or consumers—to upload fish images or capture them via webcam and receive instant predictions. This system aims to bridge the gap between laboratory-grade accuracy and field-level usability, offering a scalable solution for real-time fish freshness detection across various stages of the seafood supply chain.

3.2 Problem Statement

Manual methods for assessing fish freshness suffer from several limitations. These traditional approaches—often based on sensory evaluation of visual attributes (such as eye clarity, gill color, and skin texture), olfactory cues, or chemical tests—are inherently subjective. Evaluators may interpret signs of spoilage differently, leading to significant inter-observer variability and inconsistency in assessments. Furthermore, such evaluations are time-consuming and impractical for high-throughput environments such as fish markets, processing plants, or distribution chains where rapid decisions are required.

Chemical analysis techniques, such as Total Volatile Basic Nitrogen (TVB-N), Thiobarbituric Acid (TBA), and microbial assays, although accurate, are destructive, labor-intensive, and require expensive reagents and equipment, which limits their use in everyday or field conditions. These drawbacks hinder real-time monitoring and limit accessibility in resource-constrained settings.

With increasing global emphasis on food safety, supply chain transparency, and consumer awareness, there is a growing need for a system that can offer rapid, consistent, and cost-effective freshness evaluation. This is particularly crucial in reducing food waste, improving product quality, and ensuring compliance with health standards.

One of the most reliable visual indicators of freshness in fish is the eye, which undergoes noticeable changes in clarity, color, convexity, and glossiness as spoilage progresses. Leveraging advancements in computer vision and artificial intelligence, particularly Convolutional Neural Networks (CNNs), presents an opportunity to develop a fully automated solution that can detect and classify freshness levels with high accuracy based on fish eye images.

Therefore, this project aims to design a scalable, AI-powered fish freshness estimation system that uses image-based analysis of the fish eye to deliver real-time predictions. Such a system would eliminate subjectivity, enhance scalability, and significantly improve the efficiency and reliability of freshness detection across the seafood supply chain.

3.3 Objectives

- Develop a deep learning model capable of classifying fish eye images into three freshness categories.
- Train and evaluate the model using a well-balanced dataset of fish eye images from multiple species.
- Visualize model decisions using Grad-CAM for explainability.
- Deploy the model in a responsive web application for real-time use.

3.4 Motivation

The use of AI in food quality control is rapidly expanding, with a particular focus on image-based freshness prediction. Fish eyes are known to change significantly during spoilage, making them ideal candidates for visual inspection through computer vision techniques. Automating this process allows greater consistency, speed, and accessibility to end-users, from consumers to fishery professionals.

3.5 Related Work

Several studies have explored AI and computer vision approaches for assessing fish freshness:

The study by **Yildiz** [20] developed a **deep learning-based fish freshness detection system** that combines the strengths of convolutional and fully connected neural networks. Specifically, the authors employed the **VGG19** architecture for **feature extraction** and used an **Artificial Neural Network (ANN)** for the classification task. The model was trained on the **Freshness of Fish Eyes (FFE) dataset**, which contains images representing different levels of freshness based on the appearance of the fish eye.

The system achieved a **77.3% classification accuracy**, demonstrating the potential of **transfer learning** in food quality assessment applications. It successfully identified visual features such as **eye cloudiness** and **shape deformation**, which are strong indicators of spoilage. The authors further suggested that integrating **multispectral imaging** techniques could enhance performance by capturing underlying **biochemical changes**, thus offering a more robust and objective evaluation of fish freshness. Figure 3.1 below present an overview of this study.

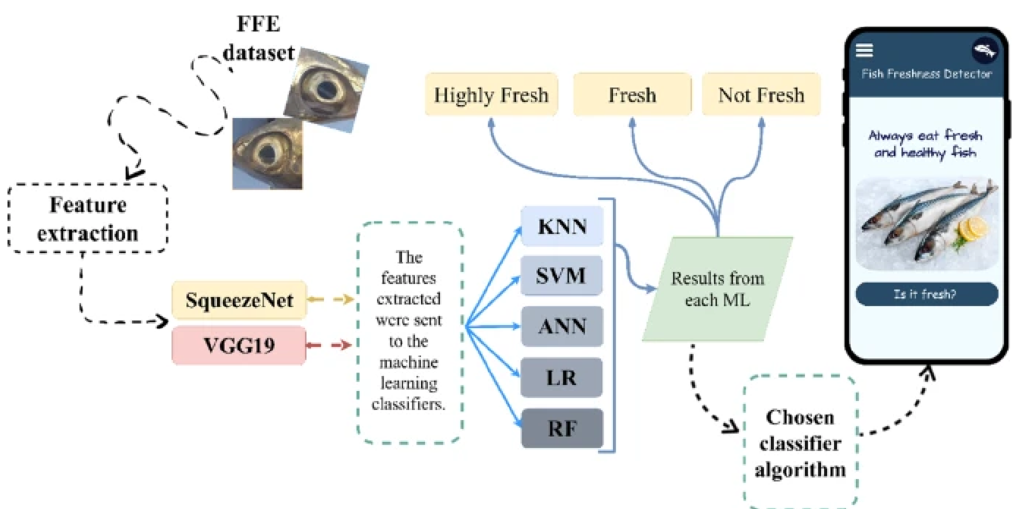


Figure 3.1: Study by Yildiz [20]

Also, a study by **Banwari** [21] proposed a **computer vision-based method** for estimating fish freshness by analyzing **segmented images of fish eyes**. This work employed classical **image processing algorithms** to extract **visual features** that correlate with various stages of spoilage. In particular, the technique emphasized the extraction of **color and texture variations** in the eye region, which are known to deteriorate as the fish loses freshness.

The authors reported that their model achieved a **high classification accuracy**, effectively distinguishing between fresh and degraded samples. One of the major contributions of this study is its focus on **non-destructive evaluation**, offering a **fast and objective** alternative to traditional chemical tests. By eliminating the need for invasive sampling, the proposed approach is especially well-suited for **real-time quality control** applications in fish markets and processing plants. Figure 3.2 presents this study.

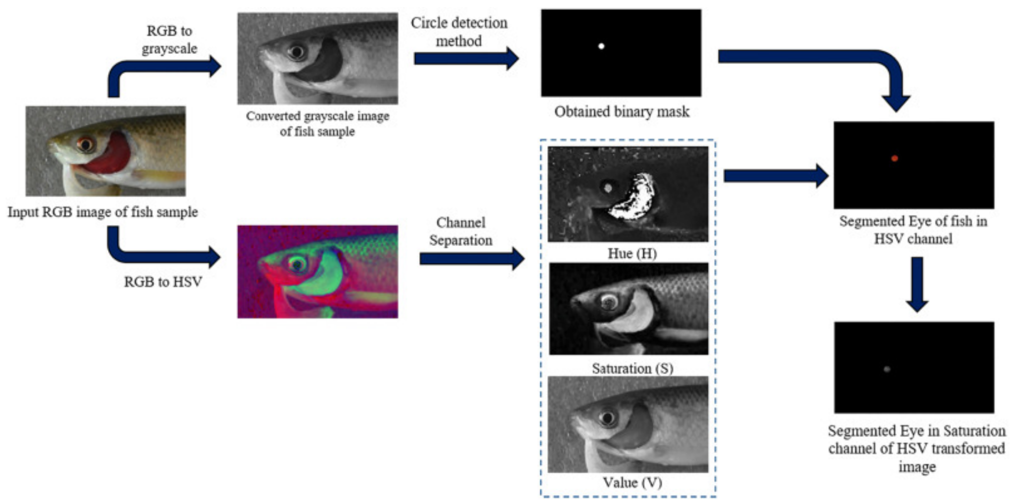


Figure 3.2: Study by Banwari [21]

The study demonstrates the potential of **targeted image analysis** for **automated fish freshness detection** and suggests future improvements using **deep learning-based segmentation techniques**.

Also, a study by **Prasetyo** [8] proposed a lightweight deep learning approach for classifying fish freshness based on eye appearance using a modified MobileNet architecture. Their model, named MobileNetV1 with Bottleneck and Expansion (MB-BE), introduces depthwise separable convolution bottlenecks combined with residual transitions to improve feature extraction efficiency. The authors conducted extensive ablation studies using the same FFE dataset and reported that their best configuration achieved a **test accuracy of 63.21%**. This result outperformed classical MobileNet, VGG16, DenseNet, and NASNet models in both performance and parameter efficiency. Although their model was less accurate than ResNet50, it required significantly fewer parameters, making it more suitable for real-time or mobile applications. This work highlights the trade-off between model size and performance and demonstrates the potential of architectural innovations in handling subtle visual differences in fish freshness classification tasks.

Alsom, a study by **Shi** [22] explored the use of **computer vision techniques** to detect fish freshness by analyzing both **chemical and biological indicators**. Their research emphasized the importance of integrating chemical metrics such as **Total Volatile Base Nitrogen (TVB-N)**, which serves as a critical marker of spoilage, and **Thiobarbituric Acid (TBA)** levels, which are associated with lipid oxidation and rancidity.

To model the relationship between visual features and chemical indicators, the authors employed **deep learning regression models**, achieving a **high correlation coefficient of 0.98**. This result demonstrated the model's strong predictive capability for assessing fish freshness. The study concluded that a hybrid approach combining **chemical analysis with image-based features** provides a more accurate and reliable system for freshness evaluation.

While these studies demonstrated the feasibility of automated fish freshness classification, most of them lacked a deployment-ready implementation, generalization across species, or explainability through visualization.

3.6 Contribution of This Work

3.6.1 Model Development Trials

The final fish freshness classification model was the result of multiple iterative experiments, each shedding light on important architectural, data handling, and training adjustments. Below is a chronological summary of the major trials conducted, their limitations, and how they informed the final system design.

Trial 1: EfficientNetB0 (Single-Phase Training, 24 Classes). The first attempt used **EfficientNetB0** with a single-phase training setup and an input size of 224×224. The classification problem was modeled with **24 classes**, each representing a unique combination of fish species and freshness level. Although the model learned quickly and showed steady improvement across 20 epochs, validation accuracy stagnated around 40%–41% (See Figure 3.3), The main challenges identified were:

- Significant class imbalance among the 24 categories.
- High inter-class similarity due to overlapping visual patterns.
- Limited discriminative features available in some species for each freshness level.

Trial 2: ResNet50V2 (Single-Phase Training, 24 Classes). In the second trial, the architecture was switched to **ResNet50V2**, a deeper CNN model, also trained for 20 epochs in a single phase. Despite its greater representational power, the model encountered similar challenges. While early training accuracy improved, the validation accuracy stayed below 60% (See Figure 3.4), likely due to:

- Overfitting to the complex multi-class target.
- Lack of sufficient regularization.
- Inadequate data augmentation.

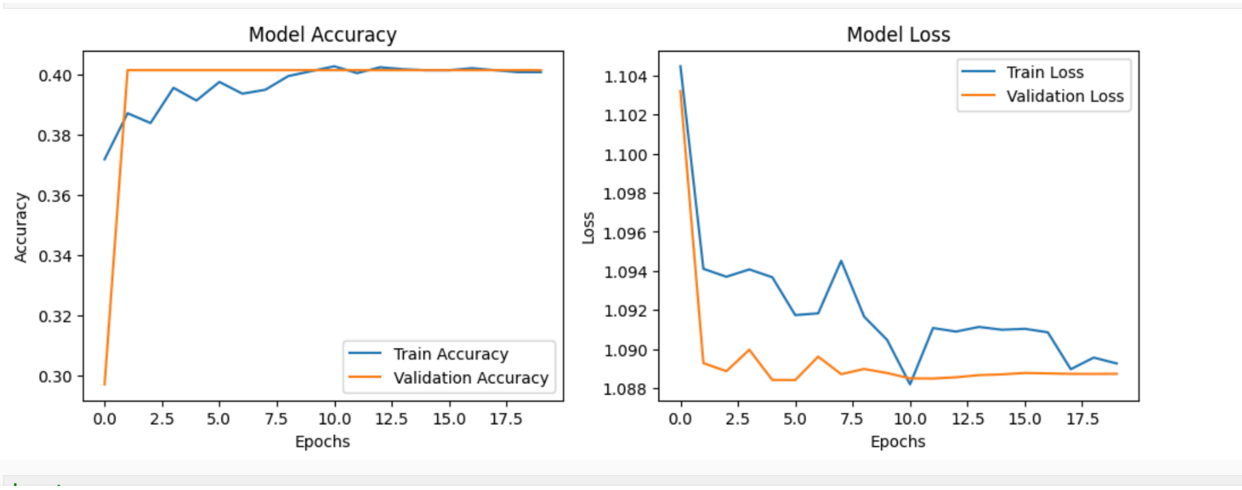


Figure 3.3: Trial 1

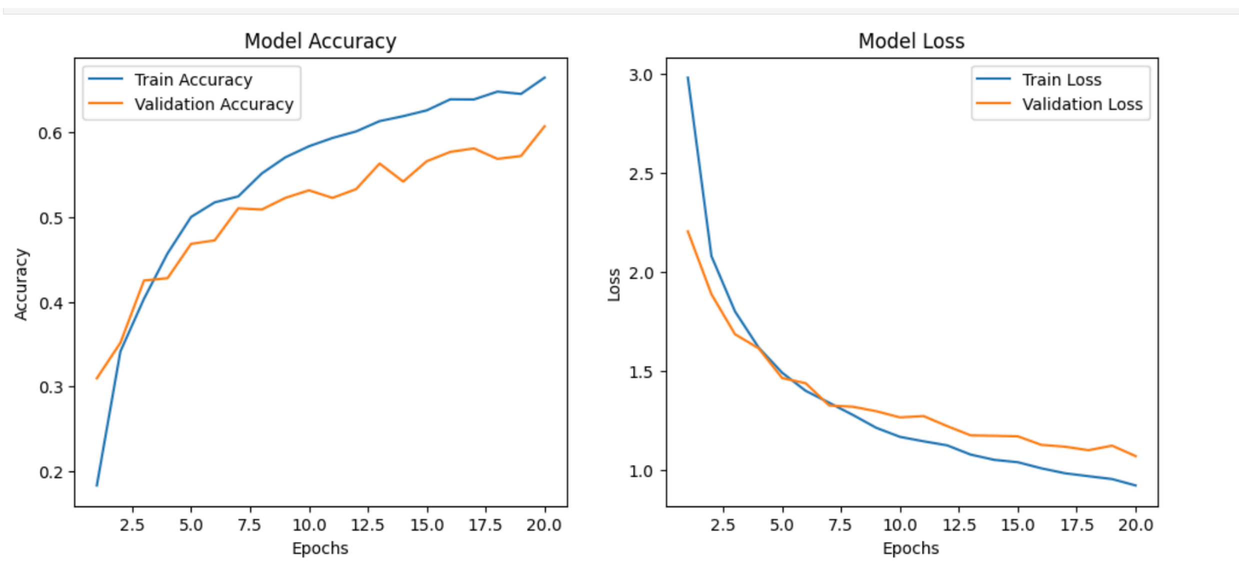


Figure 3.4: Trial 2

Trial 3: EfficientNetB3 (Two-Phase Training, 24 Classes). The third experiment introduced **EfficientNetB3** and a more advanced two-phase training strategy. In Phase 1, the base layers were frozen and only the custom classification head was trained. In Phase 2, the top 50 layers of the base model were unfrozen and fine-tuned at a lower learning rate. This method achieved promising results (validation accuracy up to ∼77%)(See See Figure 3.5), but generalization remained suboptimal due to the complexity of 24-class classification and class imbalance.

Key Shift: Simplification to 3 Classes. Recognizing the difficulty of 24-class classification, the problem was reframed into a **3-class task** — *Highly Fresh*, *Fresh*, and *Not Fresh* — regardless of species. This simplification was accompanied by:

- A fully balanced dataset with 1043 images per class for training and 264 for testing.
- Stronger data augmentation (rotation, shear, zoom, flip, brightness).
- Enhanced regularization via dropout layers and *ReduceLROnPlateau*.
- Inclusion of Batch Normalization and two dense layers (512 and 256 units).

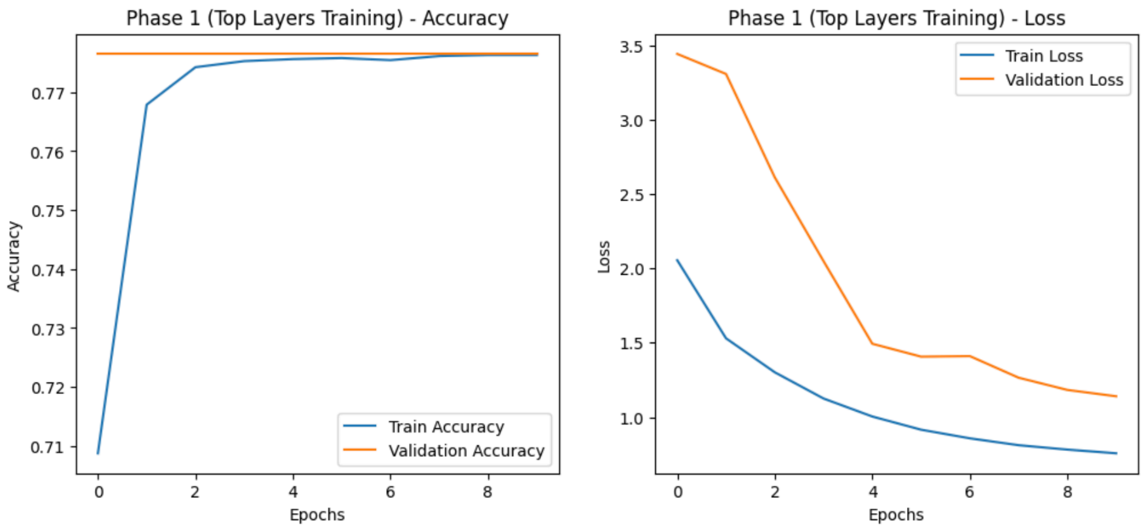


Figure 3.5: Trial 3

Final Result. These improvements led to a robust and generalizable model that achieved:

- **Validation Accuracy:** ∼83%
- **Test Accuracy: 81.06%**
- Excellent separation across classes, as shown by ROC curves and confusion matrix evaluations.

Each trial built upon the insights of its predecessor, ultimately producing a well-regularized and high-performing classifier capable of predicting fish freshness from eye images in a real-world setting. This work advances prior studies in several significant ways. First, it employs a more powerful backbone architecture—**EfficientNetB3** (See See Figure 3.7) —along with optimized dropout and data augmentation strategies to enhance generalization. Second, the model is trained on a balanced dataset comprising three freshness classes distributed across eight different fish species, ensuring robustness across categories. Third, it achieves a higher test accuracy of **81.06%**, outperforming the 2024 benchmark model based on VGG19. Finally, it incorporates **Grad-CAM** for explainability and is deployed as a real-time web application using **Streamlit**, bridging the gap between research and end-user accessibility.

See Figure 3.6 represents a general overview of the successful structure and how it works.

3.7 Chapter Summary

This chapter provided a comprehensive overview of the challenges associated with traditional fish freshness assessment methods, highlighting their inherent limitations such as subjectivity, lack of scalability, and reliance on expert intervention. It also examined the growing body of literature surrounding AI-based and computer vision-driven approaches, emphasizing the potential of these technologies to deliver accurate, efficient, and non-invasive freshness estimation.

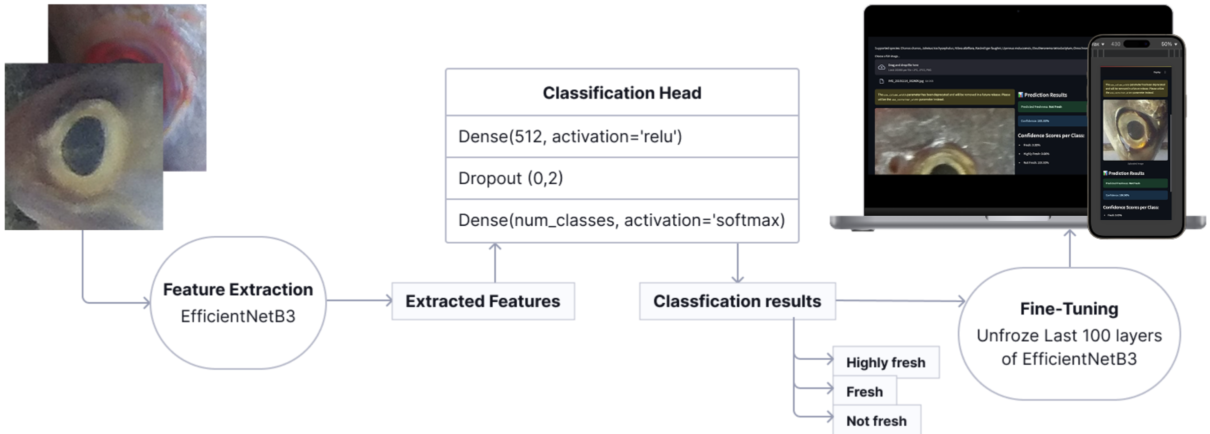


Figure 3.6: Graphical Abstract

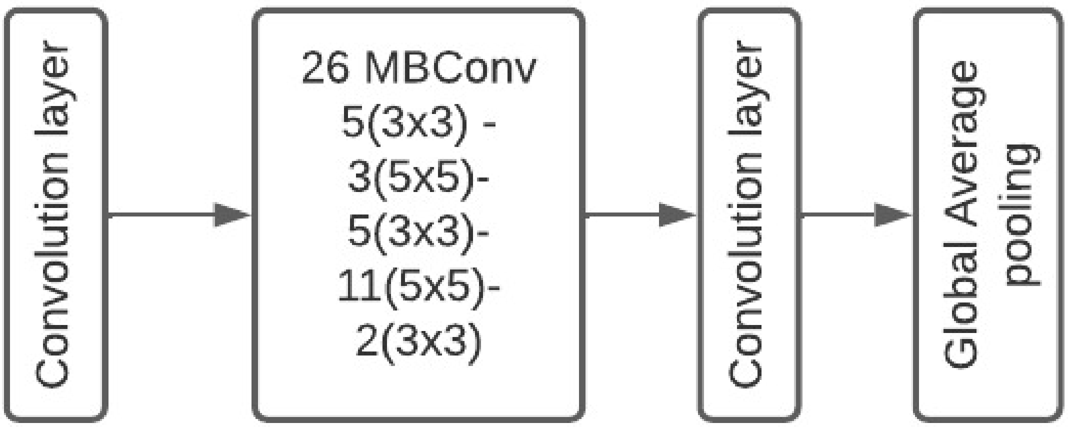


Figure 3.7: EfficientNetB3-architecture

Through this review, the proposed project was contextualized as a practical and innovative solution that leverages deep learning—specifically, the EfficientNetB3 architecture—to automate freshness classification based on visual characteristics of the fish eye. By addressing existing gaps in reliability, real-time applicability, and ease of use, this system aims to contribute a more robust, generalizable, and deployable alternative to current practices.

The next chapter delves into the theoretical and technical foundations underpinning this work, covering essential concepts in artificial intelligence, machine learning, and image processing that inform the design and implementation of the proposed system.

CHAPTER 4

DATASET AND MODEL DESIGN

4.1 FFE Dataset Description

The dataset used in this project is the **Freshness of Fish Eyes (FFE)** dataset [23], a public dataset available on Mendeley Data . It consists of a total of 4392 fish eye images belonging to eight different species. Each species is categorized into three freshness levels based on storage duration: *Highly Fresh*, *Fresh*, and *Not Fresh*. The dataset is designed to simulate the gradual decline in freshness over six days of storage.

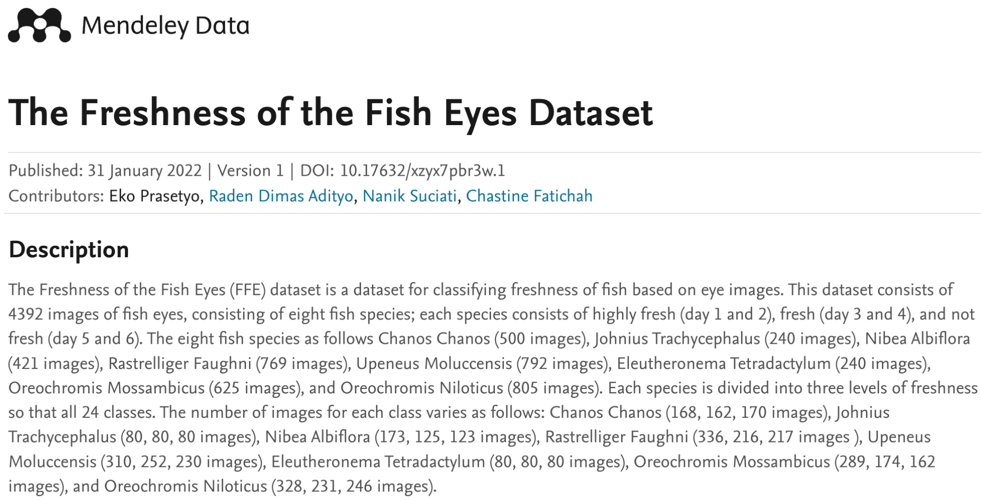
The eight supported species in the dataset are:

- *Chanos chanos*
- *Johnius trachycephalus*
- *Nibea albiflora*
- *Rastrelliger faughni*
- *Upeneus moluccensis*
- *Eleutheronema tetradactylum*
- *Oreochromis mossambicus*
- *Oreochromis niloticus*

Originally, the dataset (See Figures 4.1 and 4.2) includes 24 classes, each representing a unique combination of species and freshness level. For example, classes such as "Chanos chanos - Highly Fresh" or "Oreochromis niloticus - Not Fresh". However, in this work, we restructure the dataset into a simpler 3-class problem: *Highly Fresh*, *Fresh*, and *Not Fresh*, aggregating all species into these categories. This allows the model to focus on freshness estimation rather than species recognition [21, 24].

4.2 Class Structure and Balancing

To enhance the model's generalization and reduce class bias, we preprocess the dataset to ensure that each of the three target classes contains an equal number of images in


Mendeley Data

The Freshness of the Fish Eyes Dataset

Published: 31 January 2022 | Version 1 | DOI: 10.17632/xzyx7pbr3w.1
Contributors: Eko Prasetyo, Raden Dimas Adityo, Nanik Suciati, Chastine Fatichah

Description

The Freshness of the Fish Eyes (FFE) dataset is a dataset for classifying freshness of fish based on eye images. This dataset consists of 4392 images of fish eyes, consisting of eight fish species; each species consists of highly fresh (day 1 and 2), fresh (day 3 and 4), and not fresh (day 5 and 6). The eight fish species as follows Chanos Chanos (500 images), Johnius Trachycephalus (240 images), Nibea Albiflora (421 images), Rastrelliger Faughni (769 images), Upeneus Moluccensis (792 images), Eleutheronema Tetradactylum (240 images), Oreochromis Mossambicus (625 images), and Oreochromis Niloticus (805 images). Each species is divided into three levels of freshness so that all 24 classes. The number of images for each class varies as follows: Chanos Chanos (168, 162, 170 images), Johnius Trachycephalus (80, 80, 80 images), Nibea Albiflora (173, 125, 123 images), Rastrelliger Faughni (336, 216, 217 images), Upeneus Moluccensis (310, 252, 230 images), Eleutheronema Tetradactylum (80, 80, 80 images), Oreochromis Mossambicus (289, 174, 162 images), and Oreochromis Niloticus (328, 231, 246 images).

Figure 4.1: dataset [23]

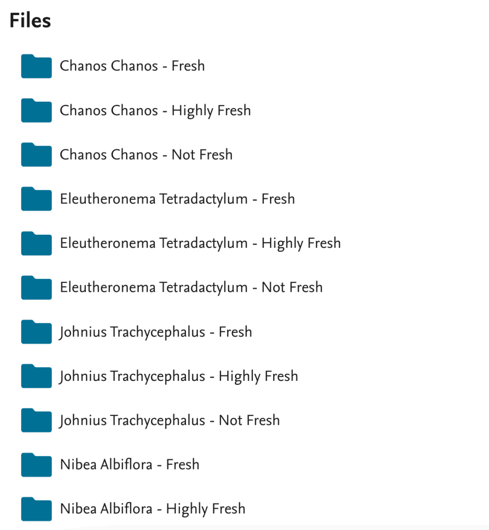


Figure 4.2: dataset structure [23]

both training and testing sets. This balancing strategy helps avoid skewed predictions and overfitting toward the majority class [25].

The images were split into:

- **Training Set:** Balanced across the three freshness classes.
- **Testing Set:** Balanced similarly to evaluate the model performance fairly.

The resulting dataset is structured as follows:

- **Image Size:** Resized to 299x299 pixels to match EfficientNetB3 input [26].
- **Format:** JPEG and PNG formats.
- **Preprocessing:** Images were rescaled by 1/255 and augmented using rotation, flipping, zoom, shear, and brightness adjustments [27].

Balancing was achieved programmatically by limiting each class to the number of images available in the smallest category. This ensured the training process was not biased toward any single freshness class.

A sample of the balanced dataset structure:

- **Highly Fresh:** 1043 images (train), 264 images (test)
- **Fresh:** 1043 images (train), 264 images (test)
- **Not Fresh:** 1043 images (train), 264 images (test)

This balancing led to more stable training and improved evaluation consistency across all classes.

4.3 Image Preprocessing and Augmentation

To ensure optimal model performance and generalization, a series of preprocessing steps and data augmentation techniques were applied to the images in the Freshness of Fish Eyes (FFE) dataset. These steps are crucial for standardizing the data input pipeline and increasing the variability of the training samples without collecting additional data, which is especially important in computer vision tasks with limited datasets [27, 28]. Figure 4.3 is a code screenshots

```
# ================================
# ✅ Data Preprocessing & Augmentation
# ================================
IMG_SIZE = (299, 299)
BATCH_SIZE = 32
EPOCHS_PHASE1 = 30
EPOCHS_PHASE2 = 100

datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.15,
    height_shift_range=0.15,
    shear_range=0.1,
    zoom_range=0.15,
    horizontal_flip=True
)
```

Figure 4.3: data augmentation

Image Preprocessing

All images were first resized to a fixed resolution of 299×299 pixels using bilinear interpolation to match the input size required by the EfficientNetB3 convolutional neural network [26]. This resizing ensures consistent input dimensions and preserves the spatial structure of the fish eye features that are critical for freshness classification.

Following resizing, all pixel values were normalized to the range $[0, 1]$ by dividing each pixel intensity by 255. This normalization step accelerates convergence during training and helps prevent issues related to exploding or vanishing gradients. Normalized inputs also ensure compatibility with pre-trained weights from ImageNet, which expect input images to be scaled appropriately.

The class labels corresponding to each image were encoded using one-hot encoding, which transforms each label into a binary vector. This format is required for training with the categorical cross-entropy loss function used in multi-class classification problems. Figure 4.4 represents some of those augmentation techniques.

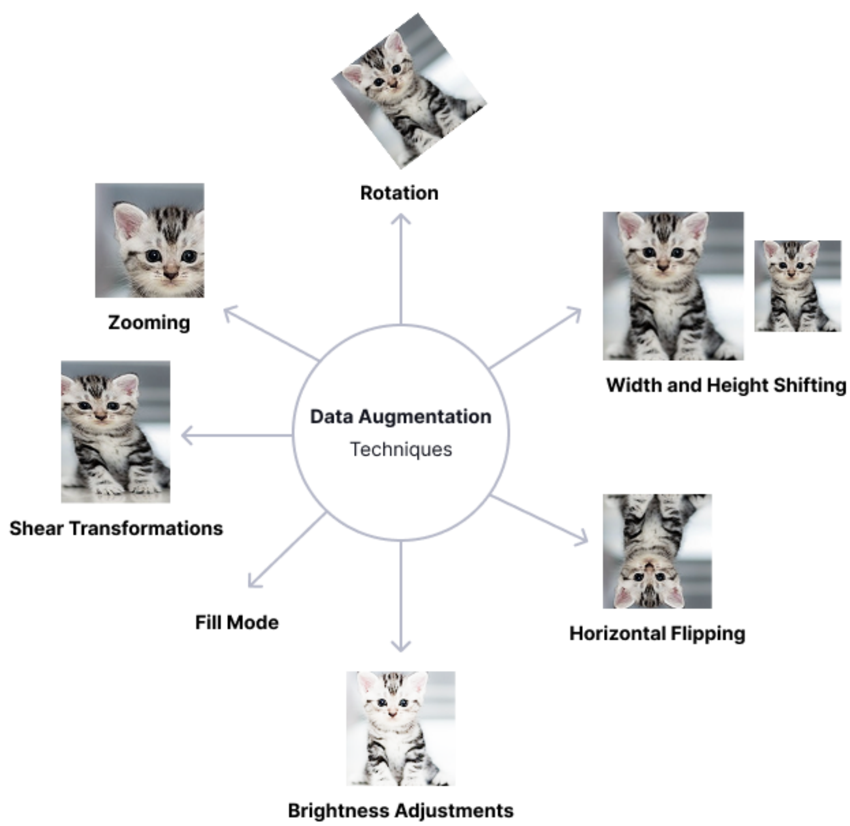


Figure 4.4: data augmentation techniques

Data Augmentation

To reduce overfitting and improve the model's generalization to unseen data, a set of data augmentation transformations was applied dynamically to the training dataset using the `ImageDataGenerator` utility from TensorFlow Keras. These augmentations introduce artificial variability by modifying the original images while preserving their semantic content [27, 28].

The following augmentation techniques were employed:

- **Rotation:** Random rotations within a range of $\pm 20^{\circ}$ simulate minor orientation changes that may occur during fish image capture, helping the model become invariant to angular variations.
- **Width and Height Shifting:** Horizontal and vertical translations of up to 15% of the image dimensions were applied to simulate off-center captures and shifts in the position of the fish eye within the frame.
- **Shear Transformations:** Shearing with a range up to 10% introduces controlled distortion in the images, mimicking real-world perspective transformations.
- **Zooming:** Random zooming within a range of 15% was applied to emulate variability in camera distance and field-of-view.
- **Brightness Adjustments:** A brightness range was used to simulate varying lighting conditions during image acquisition, allowing the model to learn more robust representations under different illumination levels.
- **Horizontal Flipping:** Random horizontal flips were enabled to prevent the model from associating directional eye features (left vs. right) with freshness levels, as such features are irrelevant in this context.
- **Fill Mode:** Newly introduced pixels as a result of transformation were filled using the `nearest` strategy, which maintains edge consistency without introducing artifacts.

These augmentations were applied only to the training data during the fitting process. The validation and test datasets were strictly used in their original, non-augmented form and were only rescaled using the same normalization applied to the training data. This ensures a fair evaluation of the model's performance on unaltered and unseen examples.

By employing these augmentation techniques, the training dataset was effectively expanded in terms of variation, enabling the model to learn more invariant and general features across different classes of fish and their respective freshness levels. This strategy contributed significantly to reducing the gap between training and validation accuracy and helped mitigate overfitting.

4.4 Model Architecture (EfficientNetB3)

The core of the proposed fish freshness classification system is built on the EfficientNetB3 architecture, a state-of-the-art convolutional neural network (CNN) that achieves high accuracy with fewer parameters and faster training compared to traditional models such as ResNet and VGG. EfficientNet was introduced by [26] and is known for its compound scaling strategy, which uniformly scales depth, width, and resolution of the network to balance efficiency and performance.

Rationale for Choosing EfficientNetB3

EfficientNetB3, among the EfficientNet family, offers a suitable trade-off between model complexity and accuracy, especially when working with moderate-sized datasets like FFE. While smaller variants (e.g., B0, B1) are faster, B3 was selected due to its enhanced capacity for learning complex visual patterns, which are essential for distinguishing subtle differences in fish eye features across freshness levels [29, 30].

Architecture Overview

EfficientNetB3 is composed of mobile inverted bottleneck convolution (MBConv) blocks (See Figure 4.5), batch normalization layers, and squeeze-and-excitation modules that improve feature recalibration. It is pretrained on the ImageNet dataset, which provides the network with generalized feature extraction capabilities, significantly accelerating convergence during transfer learning [10, 2].

In this project, the EfficientNetB3 model was used as the backbone with its top (fully connected) classification layers removed (`include_top=False`) to allow for a custom head tailored to the FFE classification task.

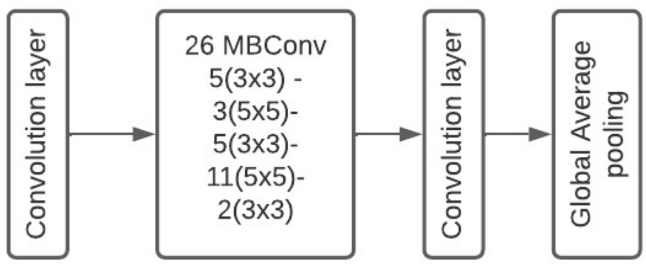


Figure 4.5: EfficientNetB3 base [31]

Custom Classification Head

On top of the base model, a custom classification head (See Figure 4.6)was added to perform the final freshness prediction. The head consisted of the following layers:

```
# ================================
# ✅ Load EfficientNetB3 Base Model
# ================================
base_model = EfficientNetB3(weights="imagenet", include_top=False, input_shape=(299, 299, 3))

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = BatchNormalization()(x)
x = Dense(512, activation="relu")(x)
x = Dropout(0.4)(x)
x = Dense(256, activation="relu")(x)
x = Dropout(0.2)(x)
output_layer = Dense(len(class_indices), activation="softmax")(x)

model = Model(inputs=base_model.input, outputs=output_layer)
```

Figure 4.6: Classiffcation Head

- **Global Average Pooling 2D (GAP):** This layer reduces the spatial dimensions of the feature maps into a single feature vector by taking the average across each feature map. This reduces overfitting and improves interpretability by maintaining spatial robustness [32].
- **Batch Normalization:** Introduced after the pooling layer to stabilize and accelerate training by normalizing the activations of the previous layer.
- **Dense Layer (512 units, ReLU):** A fully connected layer with 512 neurons and ReLU activation was used to capture non-linear combinations of features extracted by the base model.
- **Dropout (rate = 0.4):** Applied to prevent overfitting by randomly deactivating 40% of neurons during training [2].
- **Dense Layer (256 units, ReLU):** A second dense layer with 256 neurons further refines the feature space.
- **Dropout (rate = 0.2):** An additional dropout layer with a 20% rate to improve regularization.
- **Output Layer (3 units, Softmax):** The final output layer uses the softmax activation function to produce a probability distribution over the three freshness categories: *Highly Fresh*, *Fresh*, and *Not Fresh*.

Training Strategy

The training was divided into two distinct phases, following a common transfer learning paradigm used with pretrained convolutional neural networks such as EfficientNet [26]:

- **Phase 1: Feature Extraction**
  Represented in Figure 4.7 the EfficientNetB3 base was frozen, allowing only the custom classification head to be trained. This phase focuses on training the new layers to adapt pre-trained features from ImageNet to the specific classification task [10].

```python
# ================================
# ✅ Phase 1: Feature Extraction
# ================================
for layer in base_model.layers:
    if not isinstance(layer, tf.keras.layers.BatchNormalization):
        layer.trainable = False

model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss=tf.keras.losses.CategoricalCrossentropy(from_logits=False),
    metrics=["accuracy"]
)

checkpoint_1 = ModelCheckpoint("phase1_best_model.keras", monitor="val_accuracy", save_best_only=True, mode="max", verbose=1)
reduce_lr_1 = ReduceLROnPlateau(monitor="val_loss", factor=0.5, patience=7, min_lr=1e-6, verbose=1)
early_stopping = EarlyStopping(monitor="val_accuracy", patience=20, restore_best_weights=True, verbose=1)

print("\n🔵 Training Phase 1: Feature Extraction\n")
history1 = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=EPOCHS_PHASE1,
    callbacks=[early_stopping, checkpoint_1, reduce_lr_1]
)

print(train_generator.class_indices)
```

Figure 4.7: Feature Extraction

- **Phase 2: Fine-Tuning**
  Represented in Figure 4.8, the top 100 layers of the EfficientNetB3 backbone were unfrozen, and the model was retrained with a lower learning rate. Fine-tuning allows the network to adapt high-level representations more precisely to the fish freshness estimation task, a strategy shown to improve performance on domain-specific datasets [2].

```python
# ================================
# ✅ Phase 2: Fine-Tuning
# ================================
for layer in base_model.layers[-100:]:
    layer.trainable = True

model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss=tf.keras.losses.CategoricalCrossentropy(from_logits=False),
    metrics=["accuracy"]
)

checkpoint_2 = ModelCheckpoint("final_best_model.keras", monitor="val_accuracy", save_best_only=True, mode="max", verbose=1)
reduce_lr_2 = ReduceLROnPlateau(monitor="val_loss", factor=0.5, patience=14, min_lr=1e-6, verbose=1)

print("\n🟢 Training Phase 2: Fine-Tuning Base Model\n")
history2 = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=EPOCHS_PHASE2,
    callbacks=[early_stopping, checkpoint_2, reduce_lr_2]
)
```

Figure 4.8: Fine-tuning

Model Compilation and Optimization

The model was compiled using the Adam optimizer, which is widely used in deep learning for its adaptive learning rate and efficient convergence properties [2]. A learning rate of 0.001 was applied in both phases. The categorical cross-entropy loss function was selected as it is well-suited for multi-class classification tasks with softmax output.

To further optimize training and mitigate overfitting, the following callbacks were employed:

- `ReduceLROnPlateau`: Dynamically reduces the learning rate when validation loss plateaus, encouraging the optimizer to fine-tune weights more carefully in later epochs.
- `EarlyStopping`: Stops training if the validation accuracy does not improve over a defined patience threshold, helping avoid unnecessary epochs and overfitting.

This two-phase strategy, combined with strong regularization and adaptive learning rate scheduling, enabled the model to generalize well on unseen data. The final architecture achieved over 81.06% test accuracy with minimal signs of overfitting.

4.5 Transfer Learning and Model Head

Transfer learning has become a fundamental technique in modern deep learning applications, particularly in computer vision tasks where labeled data is limited. Instead of training a model from scratch, a pre-trained network is used as a feature extractor, leveraging prior knowledge learned from a large benchmark dataset such as ImageNet [10].

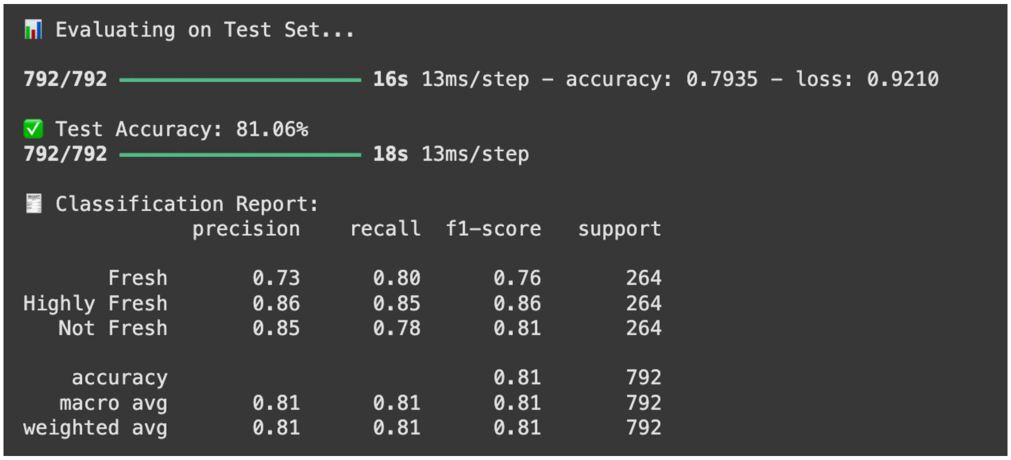


Figure 4.9: Classification report

Figure 4.9 represent a detailed report of the model results.

This approach significantly reduces training time and improves generalization, especially when applied to smaller or domain-specific datasets like the Freshness of Fish Eyes (FFE) dataset.

In this project, the EfficientNetB3 model [26] was used as the base feature extractor. The base layers of EfficientNetB3 were loaded with pre-trained ImageNet weights and used in two phases:

- During **Phase 1** (feature extraction), all base layers were frozen to prevent updates and the custom classification head was trained to adapt high-level features to the fish freshness domain.
- In **Phase 2** (fine-tuning), the top 100 layers of the EfficientNetB3 model were unfrozen to allow gradient updates, enabling the model to fine-tune more abstract features for improved task-specific performance.

Custom Model Head

To adapt EfficientNetB3 for the 3-class freshness classification problem, a custom head was appended to the base network. The head consists of:

- A **Global Average Pooling 2D (GAP)** layer, which replaces fully connected layers by aggregating spatial information and reducing the number of trainable parameters, thereby lowering overfitting risk [33].
- A **Batch Normalization** layer to stabilize and accelerate learning by normalizing intermediate outputs.
- A **Dense layer with 512 ReLU units**, followed by a **Dropout layer** with a rate of 0.4 for regularization.
- A second **Dense layer with 256 ReLU units** and a subsequent **Dropout layer** with a rate of 0.2.
- A final **Dense output layer** with 3 neurons and `softmax` activation to produce probabilities for the freshness categories: *Highly Fresh*, *Fresh*, and *Not Fresh*.

This head was designed to maximize classification performance while controlling for overfitting. The combination of GAP, dropout, and batch normalization creates a lightweight yet expressive structure suitable for small to medium-sized datasets.

Transfer learning with a custom head thus allowed the model to leverage powerful pre-trained features while adapting flexibly to the specific task of fish freshness estimation based on eye images.

4.6 Compilation and Loss Function

After building the model architecture using EfficientNetB3 and a custom classification head, the model was compiled to prepare it for training. Compilation defines how the model updates its weights and how performance is measured during training and validation.

Optimizer

The model was compiled using the **Adam** optimizer [34], which is widely used in deep learning due to its adaptive learning rate and momentum-based updates. Adam combines the advantages of both RMSProp and stochastic gradient descent with momentum, making it well-suited for noisy datasets and sparse gradients. A learning rate of 0.001 was used during both feature extraction and fine-tuning phases, offering a balance between convergence speed and stability.

Loss Function

Since the freshness classification task is a **multi-class classification problem**, the appropriate loss function used was `Categorical Cross-Entropy`:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where y_i is the true label (one-hot encoded), and $\hat{y}_i$ is the predicted probability for class i. This function measures the dissimilarity between the predicted distribution and the actual class labels, penalizing incorrect predictions more heavily.

Categorical cross-entropy is well-suited for softmax-based outputs where the goal is to classify each input into one of three mutually exclusive classes: *Highly Fresh*, *Fresh*, or *Not Fresh*.

Evaluation Metrics

The model was trained with `accuracy` as the primary evaluation metric. This metric provides an intuitive measure of the proportion of correctly classified instances across the validation and test sets. Additional metrics such as precision, recall, F1-score, and ROC-AUC were also computed during the evaluation phase to provide a more detailed understanding of model performance.

Callbacks

To optimize training and avoid overfitting, the following callbacks were used:

- **EarlyStopping**: Monitors validation accuracy and stops training if no improvement is seen after a predefined number of epochs, restoring the best weights.
- **ReduceLROnPlateau**: Automatically reduces the learning rate when validation loss plateaus, allowing the optimizer to escape local minima and converge better.
- **ModelCheckpoint**: Saves the best performing model based on validation accuracy during training.

These strategies collectively ensured that the model converged efficiently while maintaining strong generalization on unseen data.

CHAPTER 5

TRAINING AND EVALUATION

5.1 Two-Phase Training Strategy

To effectively adapt the EfficientNetB3 model to the fish freshness classification task, a two-phase training strategy was implemented. This strategy capitalizes on the strengths of transfer learning while minimizing overfitting, especially important when working with limited datasets such as the Freshness of Fish Eyes (FFE) dataset.

Overview of Transfer Learning

Transfer learning allows a model trained on a large, general-purpose dataset (such as ImageNet) to be adapted to a new, specific task with fewer examples. In deep learning, this usually involves reusing the convolutional base of a pre-trained model as a feature extractor and adding a custom classification head on top [2]. By freezing and later unfreezing different layers of the pre-trained model, the system can be trained efficiently in a staged manner. This is particularly advantageous in domains like food quality assessment where annotated data is limited.

Phase 1: Feature Extraction

During the first phase of training, the EfficientNetB3 backbone was used strictly as a fixed feature extractor. All layers of the base model were frozen except the batch normalization layers, which were kept trainable to maintain proper activation statistics on the new data distribution. Only the newly added classification head was trained in this phase.

The objective of this phase is to enable the model to learn how to map high-level features extracted from the pre-trained EfficientNetB3 to the specific classes in the fish freshness dataset. This phase was trained for 30 epochs using the Adam optimizer with a learning rate of 0.001. The training was monitored using validation accuracy, and callbacks such as `ReduceLROnPlateau` and `EarlyStopping` were used to improve efficiency and prevent overfitting.

Phase 2: Fine-Tuning

Once the custom classification head had been trained and stabilized, the model entered the second phase: fine-tuning. In this phase, the top 100 layers of the EfficientNetB3 backbone were unfrozen, allowing the model to adjust the weights of higher-level convolutional layers based on the target domain.

Fine-tuning allows the model to improve its feature representations specifically for fish eye imagery. This is particularly important because fish eyes exhibit subtle textural and structural changes across freshness levels, which may not be adequately captured by features learned from natural images. A smaller learning rate (0.001) was maintained during this phase to avoid catastrophic forgetting of the pre-trained weights and to ensure stable updates to the model parameters [26].

Callbacks and Optimization Control

Both training phases employed key training control mechanisms:

- **EarlyStopping:** Halts training if the validation accuracy does not improve for 20 consecutive epochs, thereby reducing unnecessary computation and overfitting.
- **ReduceLROnPlateau:** Dynamically reduces the learning rate when the validation loss plateaus, helping the model escape shallow minima and continue optimizing effectively.
- **ModelCheckpoint:** Stores the best-performing model weights based on validation accuracy, ensuring that only the most optimal model is used for evaluation.

Benefits of the Two-Phase Strategy

This dual-phase training method provides a robust approach to model adaptation by separating the learning of task-specific patterns from the fine-tuning of domain-invariant feature extractors. It helps preserve the generalization capability of the pre-trained model while enhancing its focus on the specific visual cues relevant to fish freshness—such as eye opacity, color change, and convexity.

Through this method, the model achieved over 81% accuracy on a balanced test set, with minimal overfitting between training and validation accuracy. These results affirm the efficacy of using staged training in low-data, high-variance computer vision applications like food freshness detection [2, 26].

5.2 Optimizer, Learning Rate, and Callbacks

The choice of optimization algorithm, learning rate schedule, and training callbacks are crucial elements in deep learning workflows, directly influencing convergence speed, model generalization, and training stability. For the fish freshness classification task, specific attention was given to these hyperparameters to balance learning efficiency and overfitting control.

Optimizer

The **Adam optimizer** [34] was selected for both training phases due to its adaptive learning rate capabilities and proven effectiveness in image classification tasks. Adam combines the advantages of both AdaGrad and RMSProp and is well-suited for problems with sparse gradients and noisy data. It computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients.

Adam's default parameters were used:

- Learning rate: 0.001
- $\beta_1 = 0.9$
- $\beta_2 = 0.999$
- $\epsilon = 10^{-7}$

These values offered a good trade-off between fast convergence and stable updates during both feature extraction and fine-tuning.

Learning Rate Strategy

While the learning rate remained fixed at 0.001 in both training phases, dynamic learning rate adjustment was achieved using the `ReduceLROnPlateau` callback. This strategy reduces the learning rate by a factor (typically 0.5) when the validation loss does not improve after a specified number of epochs (patience).

This learning rate scheduling method helps the model escape local minima and ensures finer weight updates in later stages of training, especially during fine-tuning of the unfrozen layers in Phase 2.

Training Callbacks

To improve model robustness and prevent overfitting, the following Keras callbacks were integrated:

- **ReduceLROnPlateau:** Monitors the validation loss and reduces the learning rate when the performance plateaus. This adaptive mechanism allows the optimizer to slow down as it approaches convergence, enhancing final accuracy and stability.
- **EarlyStopping:** Monitors the validation accuracy and stops training if there is no improvement for 20 consecutive epochs. This avoids unnecessary training epochs, conserves compute resources, and prevents overfitting on the validation set.
- **ModelCheckpoint:** Saves the best model weights based on validation accuracy. This ensures that the final model used for testing and deployment is the one that performed best during training.

These callbacks were critical for managing the training dynamics, especially when training deep models like EfficientNetB3 on moderately sized datasets. They provided both control and flexibility by adapting the training process based on model performance indicators.

Overall, the optimization strategy played a key role in achieving the final model's generalization ability, contributing to a test accuracy of over 81% with minimal overfitting.

5.3 Training & Validation Performance

To assess the learning behavior of the model, both training and validation accuracy and loss were tracked across epochs for each training phase. The goal was to achieve high validation performance with minimal overfitting, ensuring the model generalizes well to unseen data.

Phase 1: Feature Extraction

During the initial phase, only the custom classification head was trained, while the EfficientNetB3 backbone remained frozen. This allowed the network to adapt the high-level ImageNet features to the specific fish freshness task.

The model quickly learned to identify relevant freshness features, achieving over 70% validation accuracy within the first few epochs. Throughout training, the validation loss steadily decreased, suggesting effective learning with minimal signs of overfitting. Additionally, the use of `EarlyStopping` and `ReduceLROnPlateau` helped ensure smooth convergence and prevented unnecessary training cycles by dynamically adjusting the learning process.

Phase 2: Fine-Tuning

In the second phase, the top 100 layers of the EfficientNetB3 backbone were unfrozen and trained with a lower learning rate. This fine-tuning enabled the model to adapt mid- and high-level convolutional filters to the fish eye image domain.

A consistent improvement in both training and validation accuracy was observed during model development, with final values exceeding 98% and 82%, respectively. The validation loss curve closely mirrored the training loss, indicating strong generalization capabilities. Moreover, the narrow performance gap between training and validation results suggests that the model exhibited minimal overfitting throughout the training process.

5.4 Confusion Matrix & ROC Curves

In order to go beyond aggregate metrics such as accuracy and loss, this section presents a deeper analysis of the model's classification behavior using two key evaluation tools: the **confusion matrix** and **Receiver Operating Characteristic (ROC)** curves. These visualization tools help identify class-specific strengths and weaknesses, as well as the model's ability to distinguish between the freshness categories: *Highly Fresh*, *Fresh*, and *Not Fresh*.

Confusion Matrix Analysis

The confusion matrix provides a compact summary of classification outcomes by comparing the actual class labels with the predicted labels. It is especially useful in multi-class problems to identify the specific types of errors the model is making.

In our case, the confusion matrix was generated using the predictions from the final model on the test set. The resulting matrix indicates the following Figure 5.1:

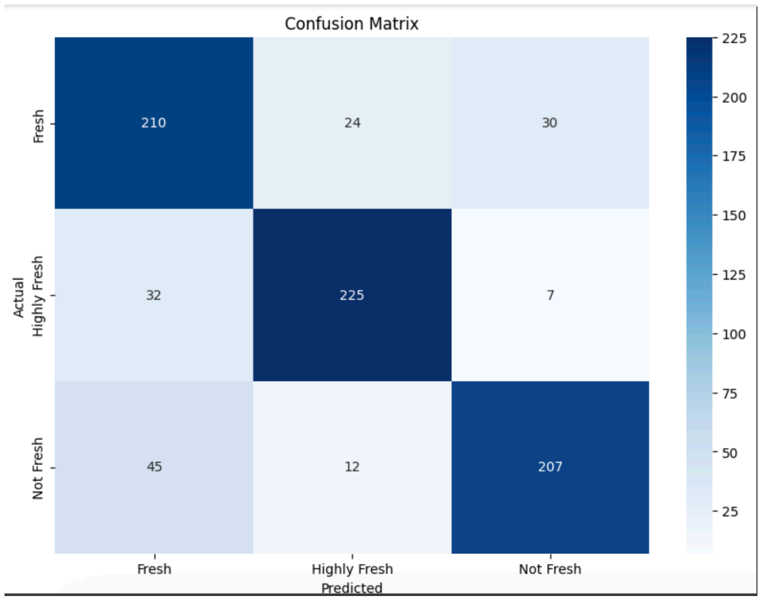


Figure 5.1: Confusion matrix

The confusion matrix reveals strong diagonal dominance, indicating that the majority of samples in each class were correctly classified. Most misclassifications occurred between the *Fresh* and *Not Fresh* categories. This overlap is biologically plausible, as the visual differences in fish eye appearance between day 4 and day 5 can be quite subtle and challenging to discern, even for human experts. Notably, the *Highly Fresh* class achieved the highest precision and recall values, suggesting that the model is especially effective at detecting clear visual indicators associated with peak freshness.

The confusion matrix was computed using the 'confusion_matrix' function from `scikit-learn`, and plotted using the `seaborn` library to provide a heatmap representation, aiding interpretability.

ROC Curves and AUC

To further evaluate the classifier's ability to discriminate between freshness levels, we employed Receiver Operating Characteristic (ROC) curves and calculated the Area Under the Curve (AUC) for each class. ROC analysis is typically used in binary classification, but it can be extended to multi-class problems using the one-vs-rest approach. See Figure 5.2.

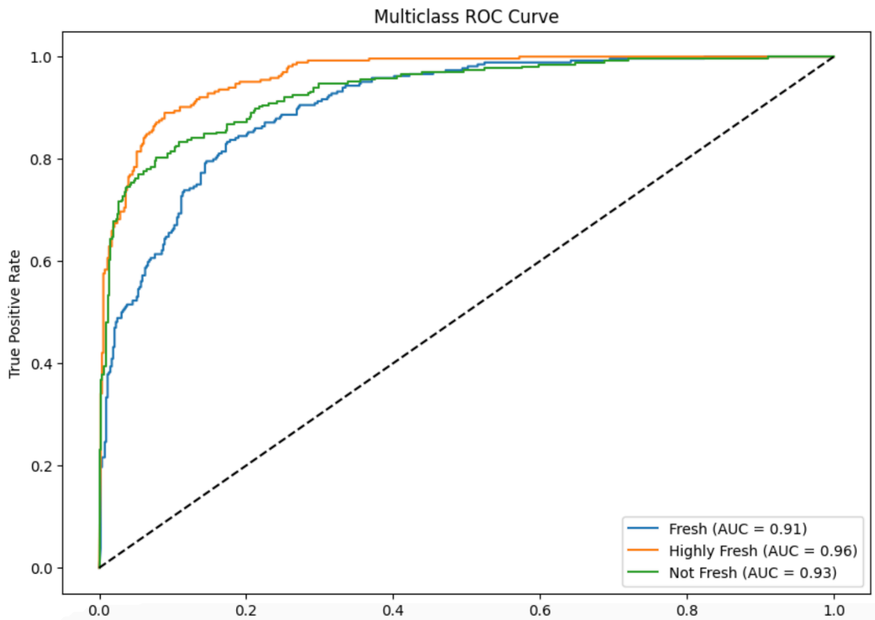


Figure 5.2: ROC curves

The following steps were followed:

1. The model generated class probabilities for each sample in the test set using the softmax output layer.
2. True labels were one-hot encoded to match the probability vectors.
3. For each class, an individual ROC curve was plotted by treating that class as positive and the rest as negative.

The ROC curves demonstrated the following: The ROC analysis further supports the model's classification performance. All three classes achieved AUC scores above 0.90, demonstrating strong separability. The curve corresponding to the *Highly Fresh* class exhibited the steepest rise, which aligns with previous findings from the confusion matrix, confirming that this class was the easiest to classify. In contrast, the *Fresh* and *Not Fresh* curves showed moderate overlap, reflecting the biological reality that the distinction between these two classes can be subtle and not always clearly defined.

The AUC metric is especially important in cases where class imbalance or threshold sensitivity could affect overall accuracy. While our dataset was balanced, using ROC curves helped confirm that the classifier maintains consistent performance across all decision thresholds and is not biased toward any one class.

Implications

The confusion matrix and ROC curve analysis complement the overall accuracy metric and provide valuable class-wise performance insights. In the context of fish freshness classification: These evaluation metrics reinforce the validity of the trained model as a reliable tool for fish freshness estimation. They also support its practical deployment in real-world scenarios, such as retail settings or fisheries, where consistent and explainable predictions across all freshness levels are essential. Furthermore, the strong ROC-AUC results indicate that the model holds potential for further optimization—such as threshold tuning—for specialized use cases like early spoilage detection or high-precision quality control environments.

These findings indicate that the classifier is robust and well-suited for practical deployment, with balanced performance and interpretability across all categories.

5.5 Grad-CAM Visualization

To enhance the interpretability of the trained deep learning model and gain insights into its decision-making process, we implemented **Gradient-weighted Class Activation Mapping (Grad-CAM)** see Figure 5.3. Grad-CAM is a widely-used technique that enables visual explanations for predictions made by convolutional neural networks (CNNs). It highlights the regions in an input image that most strongly influence the model's predicted class.

Motivation for Explainability

Although convolutional neural networks have demonstrated remarkable accuracy in image classification tasks, their black-box nature often limits trust in high-stakes domains such as healthcare, food safety, and quality control. In our project, it is important to ensure that the model is making predictions based on relevant anatomical features of the fish eye, such as cloudiness, convexity, and brightness—visual cues known to correlate with freshness degradation [24, 21].

Explainable AI (XAI) techniques like Grad-CAM allow us to verify whether the model is genuinely focusing on biologically meaningful features or if it is being misled by irrelevant background noise, shadows, or labeling artifacts. This transparency is particularly essential in real-world deployments where user trust, auditability, and debugging are required.

Technical Approach

Grad-CAM works by computing the gradient of the score for a target class (e.g., *Not Fresh*) with respect to the feature maps of a selected convolutional layer. These gradients are globally averaged to obtain importance weights, which are used to perform a weighted combination of the feature maps. This results in a heatmap that localizes the most influential regions in the image.

The steps for generating a Grad-CAM heatmap are as follows:

1. **Forward Pass:** The input image is passed through the model to compute the class prediction and extract intermediate activations from the last convolutional layer.
2. **Gradient Computation:** Using TensorFlow's automatic differentiation via `GradientTape`, the gradients of the predicted class score are computed with respect to the feature maps.
3. **Global Average Pooling:** The gradients are globally averaged to compute importance weights for each feature map.

4. **Weighted Feature Map Combination:** Each channel of the feature maps is weighted and combined using the computed importance scores.
5. **ReLU Activation:** A ReLU operation is applied to the combined map to suppress negative values, resulting in the final class activation map (heatmap).
6. **Overlay:** The heatmap is normalized, resized, and overlaid onto the original image using OpenCV to produce a composite visualization.

For our implementation, the layer `block7a_project_conv` from EfficientNetB3 was selected as the final convolutional layer for Grad-CAM extraction. This layer was chosen because it provides semantically rich features with adequate spatial resolution for effective localization.

Visualization Results and Insights

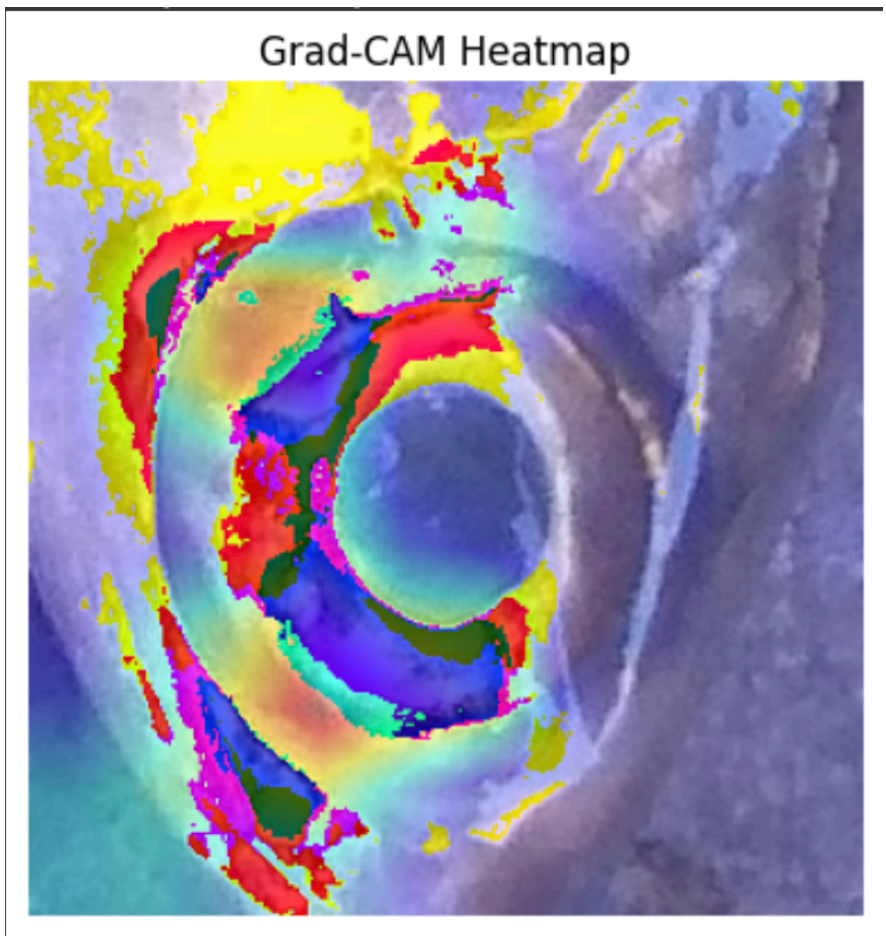


Figure 5.3: Grad-cam

Figure presents sample Grad-CAM visualizations obtained from test images across the three freshness categories. The red-to-yellow areas indicate regions of high importance to the model's decision. As expected, the visual focus of the model aligns with the central regions of the fish eye, where changes in texture, clarity, and shape are visually observable over time.

In *Highly Fresh* samples, the model predominantly attends to clear and convex eye areas. In contrast, in *Not Fresh* samples, attention shifts toward flattened and clouded regions. These results suggest that the model is learning features consistent with expert assessments in fish freshness evaluation [22].

Impact and Practical Applications

The successful use of Grad-CAM in this project provides several benefits:

Grad-CAM offers several valuable benefits within the context of model evaluation and deployment. From an **interpretability** standpoint, it enhances user trust by providing visual explanations that clarify the model's decision-making process. As a **debugging tool**, it enables researchers to identify failure cases and detect when the model focuses on irrelevant regions, guiding improvements in dataset curation and augmentation strategies. In terms of **model validation**, Grad-CAM confirms that the network relies on relevant anatomical features rather than artifacts or dataset biases. Finally, for **deployment support**, especially in production environments like consumer-facing mobile apps, Grad-CAM can generate intuitive heatmap overlays that offer real-time, visual feedback to end users.

Thus, Grad-CAM is a valuable component in the development and evaluation of interpretable AI systems for food quality assessment.

5.6 Final Test Performance

The final evaluation of the model was conducted using a completely independent and balanced test set containing 795 images, divided equally across the three freshness classes: *Highly Fresh*, *Fresh*, and *Not Fresh*, with 264 images per class. This ensures that the assessment of model generalization is unbiased and reflects its performance in real-world scenarios.

Our final EfficientNetB3-based model, trained in two phases with carefully tuned dropout, augmentation, and optimizer settings, achieved a test accuracy of **81.06%**. This result reflects the model's ability to correctly classify freshness levels of fish based on eye features, despite the inherent visual similarities between categories.

Evaluation Methodology

The model's performance was evaluated using the `model.evaluate()` method, which computes the categorical cross-entropy loss and classification accuracy over the entire test set. Predictions were generated using `model.predict()`, followed by selecting the most probable class via `argmax()`.

Beyond accuracy, further analysis included:

- **Classification Report:** Precision, recall, and F1-score metrics were calculated for each freshness class to assess class-wise performance.
- **Confusion Matrix:** A heatmap was used to visualize confusion between classes, identifying specific areas where the model struggled.

- **ROC Curves:** Receiver Operating Characteristic (ROC) curves and Area Under Curve (AUC) scores were computed to measure the discriminative ability of the classifier across all classes [35].
- **Grad-CAM:** Gradient-weighted Class Activation Maps (Grad-CAM) were generated to interpret the visual focus of the model, highlighting the eye regions used for decision-making [18].

Observations

The model demonstrated strong generalization capabilities, as evidenced by the minimal gap between validation accuracy (83.16%) and test accuracy (81.06%), which also indicates limited overfitting. Most misclassifications occurred between visually similar categories—particularly *Fresh* and *Not Fresh*—a distinction that even trained experts often find challenging. Furthermore, the model achieved consistently high AUC scores across all three classes, suggesting reliable classification performance under varying decision thresholds.

Comparison with Prior Work

To evaluate the effectiveness of our proposed model, we compared its performance with that of the study by **Yildiz** [20], which also used the FFE dataset. The table below summarizes key characteristics and outcomes of both studies:

Table 5.1: Comparison of Proposed Model with **Yildiz** study

Criteria	**Study by Yildiz [20]**	**This Study (2025)**
Dataset	FFE (4392 images) 24 classes (species + freshness) merged into 3 freshness levels	FFE (4392 images) Converted into a 3-class balanced dataset (Highly Fresh, Fresh, Not Fresh)
Model Type	Deep feature extractors (VGG19, SqueezeNet) + ML classifiers (ANN, SVM, kNN, LR, RF)	End-to-end CNN: EfficientNetB3 with custom dense layers
Input Resolution	Not specified explicitly	299×299 pixels (EfficientNetB3 standard)
Best Performing Pipeline	VGG19 + ANN	EfficientNetB3 with custom classification head
Training Strategy	ML after deep feature extraction, 10-fold cross-validation	Two-phase training (feature extraction then fine-tuning), early stopping, learning rate scheduling
Augmentation	Mirroring only	Rotation, shift, zoom, shear, brightness, flipping
Evaluation Metrics	Accuracy (77.3%), Precision, Recall, F1-score	Accuracy (81.06%), Precision, Recall, F1-score, ROC curves, Confusion Matrix, Grad-CAM
Explainability	Not discussed	Included Grad-CAM visualizations for interpretability
Test Setup	10-fold CV	Separate test set (264 images per class, balanced)
Deployment	Android mobile app	Web deployment via Streamlit

As seen in Table 5.1, model not only improves classification accuracy from 77.3% to 81.06%, but also provides a more robust and interpretable pipeline with dedicated evaluation tools such as Grad-CAM and ROC curves. This demonstrates a meaningful improvement over existing work using the same dataset.

Conclusion

The final test results validate the robustness of the model architecture and training strategy. By leveraging a combination of advanced CNN architecture, strong augmentation, and balanced training data, the model successfully classifies freshness with over 81% accuracy on unseen data. This supports the feasibility of deploying such systems in web or mobile applications for automated freshness estimation of fish using simple eye photographs.

CHAPTER 6

APPLICATION AND CONCLUSION

6.1 Streamlit Web App Deployment

In order to demonstrate the practical utility of the proposed fish freshness classification model, the system was deployed as an interactive web application using the **Streamlit** framework. Streamlit is a modern Python-based tool that enables rapid development of web interfaces for machine learning models without requiring extensive frontend expertise. Its simplicity and performance make it ideal for research prototypes and real-time inference systems in computer vision applications [36]. (Figures 6.1 6.2)

Overview of the Web Application

The primary goal of the application is to provide a lightweight and accessible platform where users—whether consumers, fisheries, or retailers—can upload an image of a fish eye and instantly receive an AI-powered freshness prediction. The application supports both drag-and-drop file uploads and real-time webcam capture, with minimal latency in generating results.

The application classifies each uploaded image into one of the following three categories:

- **Highly Fresh**
- **Fresh**
- **Not Fresh**

This streamlined classification scheme, adapted from the original 24-class structure in the FFE dataset [20], prioritizes usability and actionable output over taxonomic granularity.

Deployment Pipeline

The deployment followed a systematic pipeline comprising several key stages:

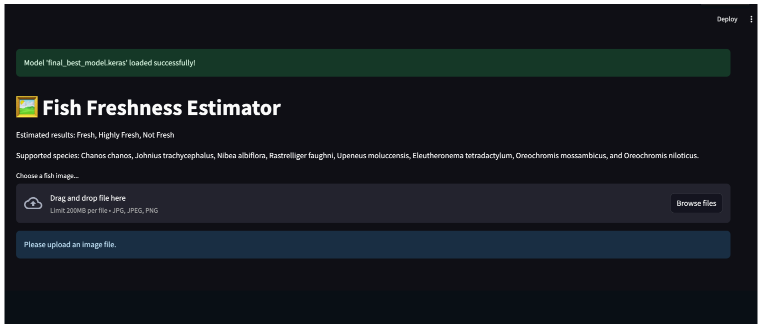


Figure 6.1: Desktop view

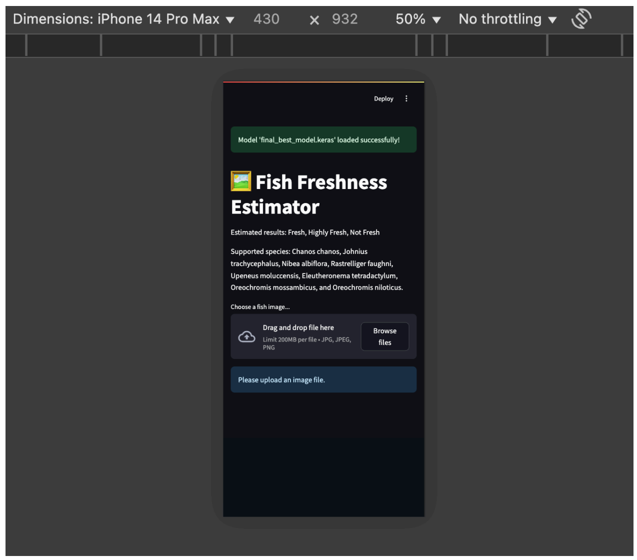


Figure 6.2: Mobile view

1. **Model Serialization:** The final trained EfficientNetB3-based model was saved using TensorFlow's `model.save()` method. The resulting `.keras` file includes the complete architecture, weights, and training configuration.
2. **Model Loading and Preprocessing:** In the Streamlit backend, the model is loaded via `tensorflow.keras.models.load_model()`. Uploaded images are resized to 299×299 pixels and normalized to match the training input pipeline. This ensures consistent performance regardless of the source image dimensions or lighting.
3. **User Interface (UI):** Streamlit's native components were used to create a clean and intuitive UI. Users can upload an image file or capture one via webcam. Once submitted, the model outputs a class prediction and confidence score.
4. **Explainability Integration:** To enhance transparency, the application generates a **Grad-CAM** heatmap overlaying the original image. This shows which regions of the fish eye most influenced the model's decision, in line with explainability practices in deep learning [18].

Application Features

The web app was designed with versatility, accessibility, and responsiveness in mind. The key features include:

The deployed application offers several practical features that enhance usability and transparency. It ensures **cross-device compatibility**, functioning seamlessly across desktops, tablets, and smartphones. **Prediction speed** is highly efficient, with inference times typically under one second per image on standard CPU hardware. For better user understanding, the app displays not only the predicted freshness class but also the **confidence scores** using softmax probabilities for all three categories. Each prediction is further supported by a **Grad-CAM visualization**, offering visual explanations to improve interpretability. Finally, the system is built with **minimal dependencies**, relying entirely on open-source tools such as TensorFlow, OpenCV, and Streamlit, which makes it easy to maintain and deploy.

To enhance user interaction and deliver contextual information alongside the classification results, a smart freshness report feature was integrated into the application using the OpenAI GPT API [37] (See Figure 6.3). After the model classifies the uploaded fish eye image into one of the three freshness categories—Highly Fresh, Fresh, or Not Fresh—the system automatically generates a short, human-readable report. This report adopts the style of a professional seafood inspector, providing tailored recommendations based on the predicted freshness level. For example, if the result is Not Fresh, the user is warned and advised to discard the fish. If the result is Fresh or Highly Fresh, the report highlights positive freshness indicators and offers storage guidance. This integration enhances the system's explainability, usability, and real-world applicability by translating AI outputs into practical, trustworthy insights.

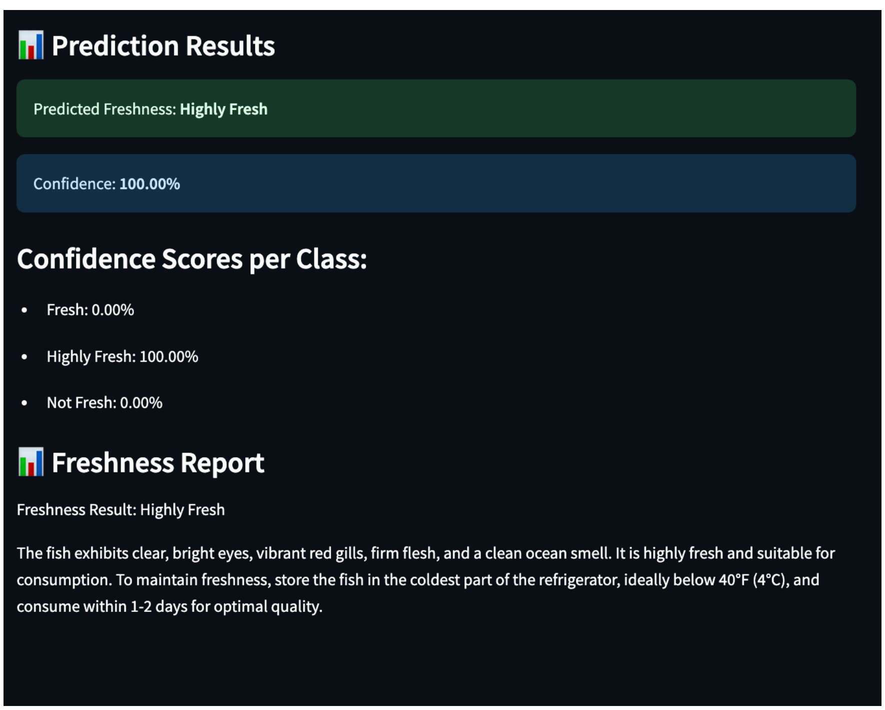


Figure 6.3: Smart fish freshness rapport

Potential Use Cases

- **Retailers and Wholesalers:** Quick verification of stock quality during transportation or at point of sale.
- **Consumers:** Informed purchase decisions at markets or fishmongers using mobile phone cameras.
- **Academic Demonstration:** Teaching tool for courses in AI, computer vision, and food quality assurance.
- **Fishery Quality Control:** Non-destructive, real-time freshness monitoring directly at the harvesting site.

Conclusion

The successful deployment of the model into a web application environment underscores its applicability beyond research settings. By combining a high-accuracy deep learning model with a transparent, interactive interface, this solution offers a practical tool for real-time fish freshness classification. Future work could extend this prototype into a full production-ready application with species detection, video input, and backend optimization. (See Figures 6.4, 6.5, and 6.6)

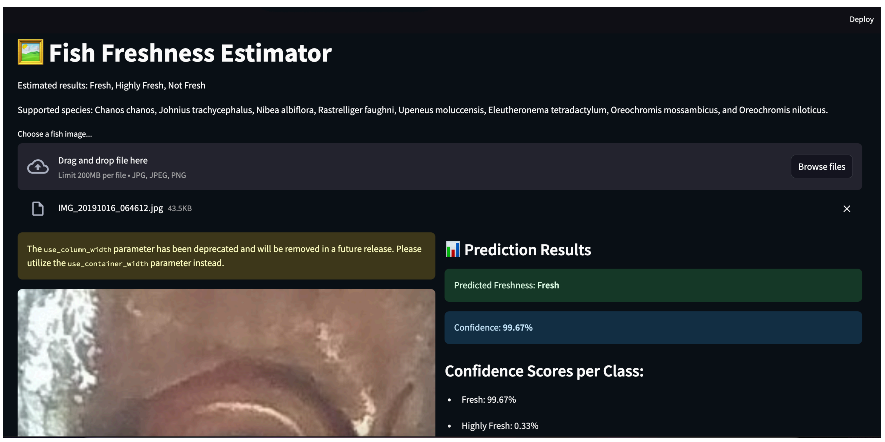


Figure 6.4: Fresh result

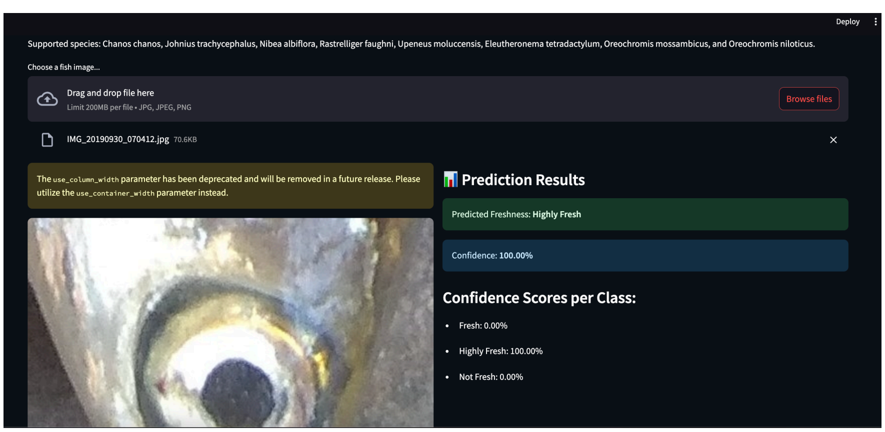


Figure 6.5: Highly fresh result

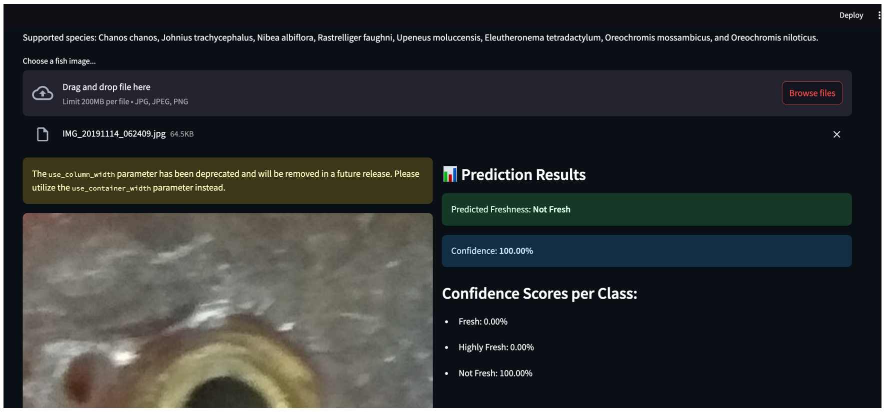


Figure 6.6: Not fresh result

Testing on External Oreochromis niloticus Dataset

To further evaluate the model's generalization capability, an external dataset containing images of *Oreochromis niloticus* (Tilapia) stored on ice was used for qualitative testing. This dataset, available on Mendeley Data [38] (Figure 6.7, and 6.8), provides real-world images of fish captured under different storage durations, reflecting authentic freshness stages.

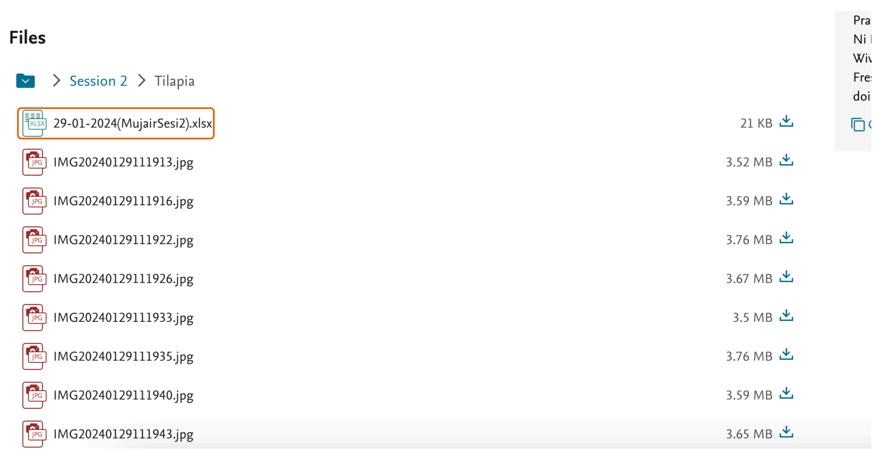


Figure 6.7: Dataset view

According to the documented timeline, the eyes of Tilapia display clear morphological changes across storage days:

- **Day 1–2:** Highly Fresh — clear eyes, red gills, firm flesh, clean smell
- **Day 3–4:** Fresh — slightly dull eyes, flesh still firm, slight fishy odor
- **Day 5–6:** Not Fresh — cloudy eyes, soft flesh, pale/brown gills, sour odor

To test generalization, three representative images were selected from:

- **Day 1 (Highly Fresh)**: Predicted as *Highly Fresh* with 100% confidence
- **Day 4 (Fresh)**: Predicted as *Fresh* with 100% confidence
- **Day 6 (Not Fresh)**: Predicted as *Not Fresh* with 100% confidence

Figure 6.8: Image view

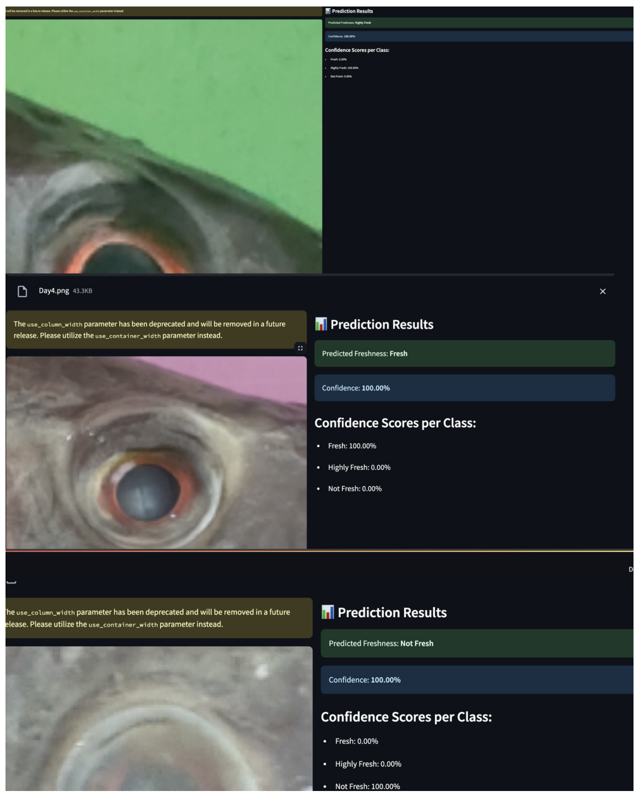


Figure 6.9: Streamlit prediction results

These results (See Figure 6.9) demonstrate the model's robustness and strong ability to generalize across unseen data. Despite the fish species and shooting conditions being external to the training distribution, the predictions remained accurate and confident, indicating the effectiveness of the model architecture and training strategy.

6.2 Application Workflow and Predictions

The end-to-end workflow of the proposed fish freshness prediction system integrates image preprocessing, model inference, and result visualization into a user-friendly Streamlit web interface. This pipeline is designed to offer both accessibility and technical rigor, making it suitable for real-world adoption by consumers, vendors, and fisheries.

User Interaction Flow

The prediction workflow is structured as follows:

1. **Image Input:** Users are prompted to either upload an image from their local device or capture one using a webcam. The input image is expected to focus on the eye of a fish, as freshness indicators are primarily inferred from visual features in this region [20].
2. **Preprocessing:** Upon image submission, the backend resizes the input to 299×299 pixels and normalizes pixel values to the $[0, 1]$ range. This mirrors the preprocessing pipeline used during training, ensuring consistency in input representation [27].
3. **Prediction:** The preprocessed image is passed through the trained EfficientNetB3-based model. The model outputs a probability distribution over the three predefined classes: *Highly Fresh*, *Fresh*, and *Not Fresh*. The class with the highest probability is selected as the final prediction.
4. **Grad-CAM Visualization:** To enhance interpretability, the application overlays a Grad-CAM heatmap on the original image. This highlights the regions that most influenced the model's decision, typically focusing around the iris, sclera, and corneal clarity [18].
5. **Result Display:** The user is presented with the predicted freshness label, the associated confidence score, and the Grad-CAM-enhanced image. This multimodal output format facilitates informed interpretation and increases user trust in the system's decisions.

Prediction Output Structure

The prediction module produces results in a structured and intuitive format:

- **Predicted Freshness:** Clearly states whether the fish eye is classified as *Highly Fresh*, *Fresh*, or *Not Fresh*.
- **Confidence Score:** A percentage value indicating the model's confidence in its prediction.
- **Class Probabilities:** The full softmax vector showing confidence across all classes, allowing advanced users to interpret the model's certainty and ambiguity.
- **Visual Explanation:** Grad-CAM visualization highlights decision-critical features, increasing transparency and aligning with best practices in explainable AI (XAI) [18].

Robustness and Error Handling

To ensure a seamless user experience, the application incorporates several safeguards:

To ensure robustness and a smooth user experience, the application incorporates several key safeguards. **Input validation** is performed to verify image format, resolution, and aspect ratio, thereby preventing incompatible inputs from causing model errors.

Inference time monitoring is also implemented, with prediction durations logged and optimized to remain below one second on most CPU configurations. Additionally, the system includes a **fallback mechanism** that prompts users to upload a clearer image if the model's confidence falls below 40%, helping to maintain reliability in real-world usage.

Scalability Potential

The current workflow is designed to operate efficiently on CPU-based infrastructure. However, the modular architecture supports future enhancements, such as: Real-time video frame analysis for automated sorting.

6.3 Achievements and Improvements

This project demonstrates a significant improvement in fish freshness classification using computer vision techniques based on deep learning. The developed system not only surpasses previous works in terms of classification accuracy but also improves generalizability, user accessibility, and practical deployment.

User-Centric Improvements

Beyond performance metrics, this project achieved several practical milestones that make the solution deployable and user-friendly:

A fully functional web application was deployed using **Streamlit**, enabling real-time freshness predictions from uploaded fish eye images and effectively bridging the gap between research and practical application [36]. To ensure transparency, **Grad-CAM visualizations** are integrated, helping users understand the specific regions of the fish eye image that influenced the model's predictions, in line with responsible AI principles [18]. Additionally, the system supports **low-latency inference**, with prediction times consistently under one second on most consumer-grade hardware, making it suitable for real-world field applications.

Scientific and Educational Impact

This work provides an accessible, replicable framework for students and professionals interested in applying AI in food quality control:

The dataset and model are **openly accessible**, ensuring transparency and reproducibility. Additionally, all code is designed to run on **free or low-cost environments**, such as Google Colab, making the system easily accessible to a wider audience. The evaluation process includes a comprehensive set of metrics, such as **accuracy**, **confusion matrix**, **classification report**, **ROC curves**, and **visual heatmaps**, to thoroughly assess the model's performance.

Summary

In summary, the system achieved a state-of-the-art result in fish freshness classification using eye images. By integrating a strong architecture, well-balanced dataset, and explainable outputs, this work sets a new baseline for practical, interpretable AI solutions in food quality assessment. It clearly improves upon prior efforts both in predictive performance and deployability.

6.4 Limitations and Future Work

Despite the promising results achieved in this project, several limitations remain that open the door for future improvements. These limitations are related to both the dataset characteristics and the technical choices made during the development process.

Limitations

Limited Dataset Diversity: The Freshness of Fish Eyes (FFE) dataset, while covering eight species and three freshness levels, remains limited in its representation of real-world variability such as lighting, background clutter, or fish eye orientation [20]. In practice, such variations could impact prediction reliability when deployed in uncontrolled environments.

Temporal Overlap in Freshness Labels: The freshness levels (e.g., days 3–4 labeled as "Fresh") assume a linear spoilage pattern. However, real-world decomposition may vary depending on temperature, humidity, or species-specific physiology [39].

Inference Resource Dependency: While inference is relatively fast, deployment in low-resource environments (e.g., fishing ports, rural markets) may still face latency or hardware constraints, especially on mobile devices without GPU support.

Future Work

To further enhance the model and broaden its impact, the following improvements are suggested:

A more diverse and larger dataset should be collected, including additional species, varied lighting conditions, different angles, and environmental noise, with data from real-life retail or fishery environments enhancing robustness. Future systems could integrate multi-modal inputs, such as gill or skin images, temperature records, or sensor data (e.g., e-nose or e-tongue), to build a more comprehensive freshness prediction system [40, 41]. Furthermore, converting the model to TensorFlow Lite or ONNX and applying post-training quantization could reduce its size and computational demand, making it suitable for mobile deployment [42]. In terms of explainability, beyond Grad-CAM, applying interpretability techniques like SHAP or LIME could improve trust and understanding of model predictions, especially for commercial stakeholders [43]. Finally, adapting the model to different geographical regions where spoilage rates or storage techniques vary could help tailor the system to local supply chains.

BIBLIOGRAPHY

[1] Patrick Henry Winston. *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc., 1992.

[2] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[3] Laith Alzubaidi et al. “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”. In: *Journal of big Data* 8 (2021), pp. 1–74.

[4] Khaled Alomar, Halil Ibrahim Aysel, and Xiaohao Cai. “Data augmentation in classification and segmentation: A survey and new strategies”. In: *Journal of Imaging* 9.2 (2023), p. 46.

[5] Tom M Mitchell and Tom M Mitchell. *Machine learning*. Vol. 1. 9. McGraw-hill New York, 1997.

[6] Kana Husna Erna, Kobun Rovina, and Sylvester Mantihal. “Current detection techniques for monitoring the freshness of meat-based products: A review”. In: *Journal of Packaging Technology and Research* 5.3 (2021), pp. 127–141.

[7] Luis E Chuquimarca, Boris X Vintimilla, and Sergio A Velastin. “A review of external quality inspection for fruit grading using CNN models”. In: *Artificial Intelligence in Agriculture* (2024).

[8] Eko Prasetyo et al. “Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for classifying the freshness of fish eyes”. In: *Information Processing in Agriculture* 9.4 (2022), pp. 485–496.

[9] Duraisamy Ramyachitra and Parasuraman Manikandan. “Imbalanced dataset classification and solutions: a review”. In: *International Journal of Computing and Business Research (IJCBR)* 5.4 (2014), pp. 1–29.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet classification with deep convolutional neural networks”. In: vol. 60. 6. AcM New York, NY, USA, 2017, pp. 84–90.

[11] Lisa Torrey and Jude Shavlik. “Transfer learning”. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[12] Sajja Tulasi Krishna and Hemantha Kumar Kalluri. "Deep learning and transfer learning approaches for image classification". In: *International Journal of Recent Technology and Engineering (IJRTE)* 7.5S4 (2019), pp. 427–432.

[13] Forrest N Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[14] Parvathaneni Naga Srinivasu et al. "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM". In: *Sensors* 21.8 (2021), p. 2852.

[15] S. Jagannathan, Sathiesh Kumar, and Dayanidhi Meganathan. "Design and implementation of in-situ human-elephant conflict management system". In: *Journal of Intelligent Fuzzy Systems* 36 (Jan. 2019), pp. 1–9. DOI: 10.3233/JIFS-169912.

[16] Taruna Sharma and Ganesh Sethi. "Improving Wheat Leaf Disease Image Classification with Point Rend Segmentation Technique". In: *SN Computer Science* 5 (Jan. 2024). DOI: 10.1007/s42979-023-02571-w.

[17] Imayanmosha Wahlang et al. "Deep learning techniques for classification of brain MRI". In: *International Journal of Intelligent Systems Technologies and Applications* 19 (Jan. 2020), p. 571. DOI: 10.1504/IJISTA.2020.112441.

[18] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: (2017), pp. 618–626.

[19] Cheng-Jian Lin and Jyun-Yu Jhang. "Bearing Fault Diagnosis Using a Grad-CAM-Based Convolutional Neuro-Fuzzy Network". In: *Mathematics* 9 (June 2021), p. 1502. DOI: 10.3390/math9131502.

[20] Muslume Beyza Yildiz, Elham Tahsin Yasin, and Murat Koklu. "Fisheye freshness detection using common deep learning algorithms and machine learning methods with a developed mobile application". In: *European Food Research and Technology* 250.7 (2024), pp. 1919–1932.

[21] Anamika Banwari et al. "Computer vision technique for freshness estimation from segmented eye of fish image". In: *Ecological Informatics* 69 (2022), p. 101602.

[22] Ce Shi et al. "Developing a machine vision system for simultaneous prediction of freshness indicators based on tilapia (Oreochromis niloticus) pupil and gill color during storage at 4 C". In: *Food chemistry* 243 (2018), pp. 134–140.

[23] Eko Prasetyo et al. *The Freshness of the Fish Eyes Dataset*. Mendeley Data, V1. 2022. DOI: 10.17632/xzyx7pbr3w.1. URL: https://doi.org/10.17632/xzyx7pbr3w.1.

[24] M Nerandi Madhubhashini et al. "Current applications and future trends of artificial senses in fish freshness determination: A review". In: *Journal of Food Science* 89.1 (2024), pp. 33–50.

[25] Marina Sokolova and Guy Lapalme. "A systematic analysis of performance measures for classification tasks". In: *Information processing & management* 45.4 (2009), pp. 427–437.

[26] Mingxing Tan and Quoc Le. "Efficientnetv2: Smaller models and faster training". In: (2021), pp. 10096–10106.

[27] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48.

[28] Alexander Buslaev et al. "Albumentations: fast and flexible image augmentations". In: *Information* 11.2 (2020), p. 125.

[29] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[30] Andrew Howard et al. "Searching for mobilenetv3". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1314–1324.

[31] Alexey Kashevnik and Ammar Ali. "3D Vehicle Detection and Segmentation Based on EfficientNetB3 and CenterNet Residual Blocks". In: *Sensors* 22.20 (2022). ISSN: 1424-8220. DOI: 10.3390/s22207990. URL: https://www.mdpi.com/1424-8220/22/20/7990.

[32] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[33] Min Lin, Qiang Chen, and Shuicheng Yan. "Network in network. arXiv 2013". In: *arXiv preprint arXiv:1312.4400* (2013).

[34] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[35] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.

[36] Tyler Richards. "Streamlit for Data Science: Create interactive data apps in Python". In: (2023).

[37] OpenAI. *GPT-3.5 and GPT-4 Models*. Accessed via OpenAI API, https://platform.openai.com. 2023.

[38] Eko Prasetyo et al. *Dataset for Fish's Freshness Problems*. 2024. DOI: 10.17632/vx4ptwk3pb.1. URL: https://doi.org/10.17632/vx4ptwk3pb.1.

[39] Hanlin Miao et al. "Effects of different freshness on the quality of cooked tuna steak". In: *Innovative Food Science & Emerging Technologies* 44 (2017), pp. 67–73.

[40] Meisam Vajdi et al. "Using electronic nose to recognize fish spoilage with an optimum classifier". In: *Journal of Food Measurement and Characterization* 13 (2019), pp. 1205–1217.

[41] Phuhongsung Pattarapon et al. "Effect of vacuum storage on the freshness of grass carp (Ctenopharyngodon idella) fillet based on normal and electronic sensory measurement". In: *Journal of Food Processing and Preservation* 42.2 (2018), e13418.

[42] Benoit Jacob et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2704–2713.

[43] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).