Al-Farouk journal of sciences
Volume (2), Issue (1) (March, 2026)
ISSN: 3135-2359

مجلة الفاروق للعلوم
المجلد (2) العدد 1 مارس 2026
ردمد: 3135-2359

A Transfer Learning-Based Approach for Classification of Fish Species Using Machine and Deep Learning Techniques.

BY

Eman Mohamed Elatrash, Computer Science, Faculty of Information Technology, Al Asmarya Islamic University Zliten-Libya,
e.elatrash@asmarya.edu.ly

Hanan Mohammed Esmail University of. Aljufra, Faculty of Education - Wadan Department of Computer Science
hhanan22015@gmail.com

تاريخ الاستلام: 2026/01/13 تاريخ المراجعة 18 / 2 / 2026 تاريخ القبول:2026/03/12- تاريخ النشر: 26/ 03/ 2026

Abstract

Precise identification of marine fish species is essential to promote sustainable fishery management and preserve marine biodiversity. This research explores the performance of four pre-trained deep learning architectures VGG16, ResNet50, InceptionV3, and MobileNetV2 in a transfer learning paradigm to identify nine marine fish species. A balanced and augmented dataset of 9,000 images (1,000 per species) was employed to promote model generalization and robustness. All the models performed excellently, with ResNet50 recording the highest classification accuracy of 99.72%, followed closely by VGG16 (99.70%), InceptionV3 (99.60%), and MobileNetV2 (99.44%). While overall performance was great, there were minor misclassifications between visually similar species. Model interpretability was enhanced with the Grad-CAM technique, demonstrating that the models effectively attended to relevant morphological features during prediction. Comparing previous work shows that the suggested approach performs better than state-of-the-art models at accuracy even when it is identifying more species. The findings demonstrate the promise of state-of-the-art deep learning for supporting automated, scalable, and accurate marine species classification for environmental and ecological applications.

Keywords: Marine fish classification; Deep learning; Transfer learning; ResNet50; VGG16; InceptionV3; MobileNetV2; Image augmentation; Grad-CAM; Biodiversity monitoring; Sustainable fishery management.

1. Introduction

The understanding of aquatic ecosystems is enhanced, while adaptive governance strategies within fisheries are developed, alongside increased predictive precision, through using machine learning-based classification coupled with fish species segmentation. Computation modeling advancements enable accurate species identification, which is critical for implementing sustainable harvesting practices. [1] Differentiation is based on extraction of distinct visual cues that can be associated with particular taxa [2] When managing fisheries, creation of sustainable biological conservation policies is highly facilitated by reliable biological data. Proper taxonomy beyond merely listing species aids in sustainable exploitation planning, compliance, equilibrium maintenance of ecosystems to enhance scientific knowledge on demographic distribution of diverse populations within specialized regions, species-level identification becomes paramount in Constitutive taxonomic knowledge is crucial for endemically safeguarding creatures as it scientifically justifies protective policies tailored to precise biological information so ecosystem balance stems from regulated interactions between aquatic organisms, understanding of their spatial distribution, trophic interactions via reproductive areas. Taxonomy is essential in detection via monitoring of ecological shifts proactively fostering anticipatory responses to population drops or invasive species surge [3] Traditional division and classification systems in marine biology are dependent on error-prone workflows with high manual overhead factors modulating scalability [4] Reliance on human evaluators injects more variation based on qualitative factors of perceiving images to understanding contexts, adding variation among multiple datasets. Textures such color spectrum change along with difference in morphology amongst species from underwater settings makes dealing with aquatic life very difficult for manual systems [5] sheer volume of image datasets combined with very slight differences between species makes things worse for population assessment reliability [6] absence of automation greatly restricts monitoring of changes in ecosystems over extended periods which reduces chances of identifying ecological shifts inhibiting development of adaptive conservation models. Other technologies such as deep learning machine techniques have been developed recently to cater to these problems by creating scalable pipelines for classification segmentation that are far more accurate and efficient, changing the face of aquatic biodiversity analysis [7].

[8] Ecological datasets on a large scale, especially datasets within marine environments, undergo streamlined processes so as to avoid dealing with morphological chromatic diversity found among species [9] Such models reduce ocular observer discrimination by relying on generalizations built through learned information which adheres to reproducible, evidence-based reasoning [10] Thus, things need to be kept balanced in machine learning applications in order to enhance performance with few associated constraints. This study aims to identify performance of four widely used transfer learning models VGG16, ResNet50, InceptionV3, and MobileNet ın classification and segmentation tasks for fish species vıa Findings of this study assist in determining a most suitable approach to be utilized in sustainable fishery management and marine ecosystems conservation; the following sections provide a critical examination of prevalent algorithms utilized in classification, such as applied methodology, data collection process, as well as comparison of model performance developed for species identification and differentiation.

2. Most Common Algorithms Used in Fish Classification.

Classification algorithms have proven to have extensive impact across numerous disciplines, ranging from species classification to marketability, pricing policies, consumption, and scientific inquiry Within the fisheries sector these algorithms assist in effective management policies, sustainable harvesting, and data-informed decision-making. Some of the key classification algorithms are discussed below.

2.1. Convolutional Neural Networks (CNN).

CNN models have a capability of learning image features hierarchically on their own [11] such as complex variations and patterns that are required to achieve effective identification of species [12] CNN-based classification system mostly comprises as following:

Convolution: It is a component of CNNs and an important part of image analysis. Convolution uses sliding filters or windows over input images to construct local features. Convolutional layers are applied to detect and examine structural elements such as edges, textures, and color gradients to enable a model to determine spatial position and magnitude of these elements [13] output at each point (Oi,j) is calculated through formulae in Equation 1.

$$Oi,j \;=\; \sigma\left(\sum_{K=1}^{K}\sum_{L=1}^{L}\sum_{m=1}^{M} W_{K,L,m}\cdot L_{i+K-1,J+l-1,m} + b_k\right) \quad [1]$$

In equation in Expression 1, σ represents activation function $W_{K,L,m}$ Filter weights $L_{i+K-1,J+l-1,m}$ Input pixel values, and b_k represents a bias term.

Pooling.

Reduction of size of feature maps (i.e. area that gets processed in a given timeframe) results in quicker processing time and convolution being less sensitive to minor changes in location as well as scale. operation can be performed by sliding in input feature map with a window of a predefined size (for instance, 2x2 or 3x3) and performing a particular pooling operation in each of windowed regions [14] Max pooling is one of a most well-known techniques whereby for each window via maximal value is extracted and sent to output feature map. By focusing on the value of a cell, it allows for a better understanding of its features while at the same time achieving a dramatic reduction in size. Consequently learning process becomes more efficient, requires less storage space, and enhances overall performance of model max pooling operation can be expressed mathematically as follows [15]

$$Oi,j \;=\; \max_{K,L} \; L_{i\times S+K,\, j\times s+1} \quad [2]$$

In Equation 2, *s* represents a stride length, while *k* and *l* denote a pixels in pooling window.

2.2. The Support Vector Machine (SVM).

Support Vector Machine (SVM) is a strong supervised learning algorithm that is extensively applied in fish classification issues via fundamental concept of SVM is identification of an optimal hyperplane that can effectively classify various fish classes in feature space so hyperplane is positioned in a favorable location such that it maximizes margin distance between hyperplane and support vectors, closest data points of each class [16] SVM aims to ensure a best separation between classes by considering most challenging cases, namely support vectors. Given a training dataset with features X and corresponding labels $y \in \{-1,1\}$ for binary classification, aim of SVM is to find a hyperplane defined by weights w and bias term b, satisfying a following conditions in Equation 3 and Equation 4:

$$w \cdot x + b \geq for\, y = 1 \quad [3]$$

$$w \cdot x + b \leq for\, y = -1 \quad [4]$$

These equations can be combined into a single expression by defining a slack variable ξi for each data by defining, it can be combined into a single expression in Equation 5.

$$y_i \; w \cdot x + b \geq -1 \xi_i \text{ For all data points} \quad [5]$$

2.3. K-Nearest Neighbors (KNN).

KNN is a simple and intuitive algorithm used in classification tasks, including fish classification. The fundamental idea in KNN is to classify an object based on a most frequently occurring class among its k-nearest neighbors in feature space. Given a dataset with a set of feature vectors X and corresponding labels y, when a new data point x_{new} needs to be classified KNN algorithm follows these steps [17]

- First distance between x_{new} and each data point in the training set is measured. Common distance metrics include Euclidean distance and Manhattan distance.
- The Euclidean distance is calculated using formula in Equation

$$Distance\,(x_{new}, x_i) = \sqrt{\sum_{j=1}^{D} \left(x_{new} - x_i\right)^2} \quad [6]$$

Then k data points with smallest distances to x_{new} are selected va Finally most common class among k-nearest neighbors is determined using formula in Equation 7.

$$y_{new} = arg\,max_y \sum_{i=1}^{k} \prod(y_i = y) \quad [7]$$

In Equation 7 y_{new} represents predicted class for new data point and $\prod(\cdot)$ is indicator function

choice of distance metric and value of k are critical parameters in KNN algorithm va Smaller k values make model more sensitive to noise while larger values can lead to excessive simplification simplicity and efficiency of algorithm make it a valuable tool especially in cases where decision boundaries are nonlinear and complex.

2.4. Gaussian Mixture Models (GMM).

Gaussian Mixture Model (GMM) is a very well-known statistical pattern recognition and modeling technique, primarily applied to model complicated data as a combination of multiple Gaussian distributions it is most powerful in detecting complex patterns and bringing out underlying concealed structures in the data. The focus of GMM is a concept of a mixture model, wherein the entire distribution is represented as a weighted mixture of a few Gaussian components. There is a component for each distinct subpopulation within the data set that contributes proportionately to the representation of data [18] This model framework enables GMM to be flexible with diversified data characteristics, hence GMM can effectively solve high-impact applications by unsupervised learning, density estimation, and clustering. In this way, based on the probabilistic feature of Gaussian distribution, GMM provides a comprehensive comprehension of heterogeneity in data and can derive useful patterns underlying mathematics of GMM is briefly introduced below in order to interpret its functional operation and theoretical mechanism. Mathematically, GMM is represented as follows:

$$p(x) = \sum_{i=1}^{k} \pi_k N(X \mid M_K \textstyle\sum_K) \qquad \text{\_\_\_\_\_\_\_\_\_\_}[8]$$

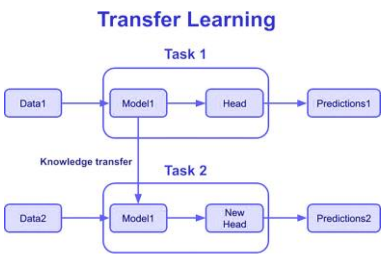
When Equation 8 is examined π_k **k-** represents the weight of the k-th component and $\sum_{i=1}^{k} \pi_k = 1$

satisfies the condition.

$N(X \mid M_K \sum_K)$ represents the probability of x being generated by the k-th Gaussian distribution.

M_K va $\sum_K$ It is the mean vector and covariance matrix of the k-th Gaussian distribution.

The use of latent variables allows for determining which Gaussian component each data point belongs



to; these variables represent the membership of each data point to a specific component Thus they help the model better understand and capture the underlying structures within the data.

2.Figure 1. Transfer learning schema.

In this study, the pre-trained CNN models VGG16, ResNet50, InceptionV3, and MobileNet were used for fish classification.

A. VGG16

Since VGG16 has been trained on a large and varied dataset it provides a good foundation for moving on to visual classification tasks [21] Through transfer learning, general features learned by VGG16, such as general object recognition on the ImageNet dataset, can be redirected towards a more specialized task like fish classification. This technique is highly effective when working with datasets with a limited number of samples. Transfer learning makes the model learn and become trained much more rapidly and efficiently by borrowing the pre-learned knowledge and applying it to the new task. Figure 2 describes the VGG16 model architecture.

Figure 2. VGG16 Architectural Structure.

The VGG16 model used in the research has a total of 21 layers out of which 16 are trainable parameters i.e., weight-carrying layers. These 16 layers consist of 13 convolutional layers, 5 max pooling layers, and 3 fully connected layers. The model is designed to be optimized for visual classification by taking inputs of 224x224 pixels with three RGB channels. One of the interesting things about VGG16 is that it has fewer hyperparameters. Convolutional layers use filters of 3x3 dimensions and a stride of 1 unit and fixed padding. Max pooling layers use filters of 2x2 dimensions and a stride of 2 units giving a uniform filter layout throughout the architecture. Conv-1 contains 64 filters. Conv-2 contains 128 filters. Conv-3 contains 256 filters while Conv-4 and Conv-5 contain 512 filters. The fully connected layers contain two layers of 4096 channels followed by a final layer containing 1000 classes. This final layer is a softmax layer that handles classification and probability estimation. The systematic application of 3x3 filters in conjunction with regular convolutional and pooling layers has made VGG16 extremely successful for visual recognition applications. This model has been successful even when derived from small datasets with high percentage accuracy and has added a big boost towards development in deep learning models.

The other transfer learning model used in the research is the ResNet50 model, and Figure 3 shows the architecture of its model. ResNet50 has a multi-stage architecture, and each stage processes the input in a different way with different operations to understand it. The input data in stage one of the model starts with zero padding to be 224x224 dimensions and 3 RGB channels. Subsequently, it passes through a convolution (CONV) layer that works with 64 filters. The batch normalization follows after conducting this process of convolution, and then a Rectified Linear Unit (ReLU) activation function is applied that includes nonlinear features to the model. This operation is finalized with a max pooling layer, which maps the input to a lower-dimension representation that extracts and explains the essential features (Deshpande et al., 2021). Between the second stage and the fifth stage, the model continues to extract features using convolution blocks consisting of deep layers. The second stage uses 128 filters in the convolution layers, the third stage uses 256 filters, and the fourth and fifth stages use 512 filters to learn more abstract and complex features. At these levels, all the convolution layers obtain fine information within the data by utilizing deeper layers, thereby making them appropriate for classification. Finally, in the last section of the model, an average pooling is performed followed by flattening the data and passing it through two fully connected (FC) layers with 4096 channels. Classification occurs after this level in the last fully connected layer, which has 1000 classes. In the output layer, class probabilities are computed using the softmax function.

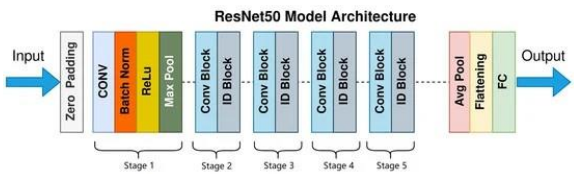


Figure 3. ResNet50 architecture structure.

B. InceptionV3

Another transfer learning model used in a study is the InceptionV3 model vaı its architecture is shown in Figure 4. InceptionV3 has a multi-layered and complex architecture and at each stage it processes the input through different operations to understand it so the initial stage of the model begins with a convolution layer having 7x7 filters then it continues with a max pooling layer using

3x3 filters these operations reduce the size of the input and compact important features (Wang et al., 2019). In the following stages of the model there are convolution layers with different filter sizes when convolution filters of 1x1 and 3x3 are used sequentially to capture features at different scales within the data after each convolution stage a 3x3 max pooling layer is applied to reduce the size of data and extract of most important features so deeper layers of a model continue with Inception blocks these blocks use different filter sizes simultaneously to perform a broader feature extraction. For example, Inception blocks repeated 2 and 5 times at specific stages capture complex structures in the data and learn higher-level features so this allows for model to perform abstractions at different levels so In the final stage data is flattened via a global average pooling (Global AvgPool) layer then it is passed to a fully connected layer and classification is performed ın a output layer, class probabilities are computed via the softmax function ın this sequential and versatile structure enables InceptionV3 model to learn both low-level and high-level features allowing it to exhibit successful classification performance va Unlike a structure based on fixed filter sizes like in VGG16, this architecture uses flexible filter sizes to extract features at different scales and analyze more complex structures.

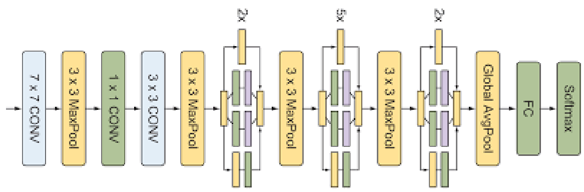


Figure 4. InceptionV3 architecture structure.

C. MobileNetV2

MobileNetV2 is another deep learning model used in study va is architecture as shown in Figure 5, is structured with a lightweight and efficient design at beginning of a model, the input data is passed through a convolution layer with 3x3 filters after this operation a model continues with reduction blocks (IR Block) and compression blocks (ISC Block) that perform expansion and compression operations. MobileNetV2, thanks to these blocks, provides an optimized fast and efficient structure especially for mobile devices that require low computational power. The first convolution block generates feature maps of size 64x64x32, and then features within the data are extracted using two repeated IR blocks Next model transitions to deeper blocks of sizes 32x32x96 and 16x16x1280. During these stages ISC blocks compress the data, reducing its size and extracting deeper features after each IR and ISC block in model processes in input data further to perform multi-layered feature extraction especially with blocks repeated 2, 6, and 3 times a model learns increasingly complex feature so this structure unlike the fixed-size filters in VGG16 provides a more flexible design that optimizes data flow between layers so in the final stage convolution operations with sizes 1x1 and 3x3 are applied to bring the data to its final dimensions so a model also uses batch normalization and the ReLU6 activation function which enables the learning of nonlinear features in end a model passes the output to a fully connected (FC) layer to complete the classification process.

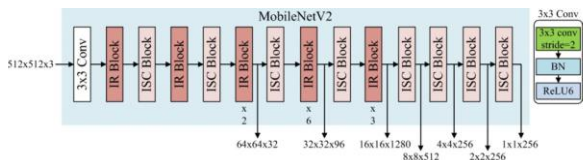


Figure 5. MobileNetV2 architecture structure.

3.1 Dataset.

The dataset used in this study consists of nine different fish species downloaded from sourced Kaggle vıa dataset containing 1000 augmented images per class along with corresponding pairs of augmented ground truths. The fish species contained in the dataset are gilt head bream, red sea bream, sea bass, red mullet, horse mackerel, black sea sprat, striped red mullet, trout, and shrimp. The data set is in balance with all the classes containing equal numbers of images (1000 images for each class). The distribution of the classes in the data set is represented in Figure 6 and sample images in the data set are depicted in Figure 7.

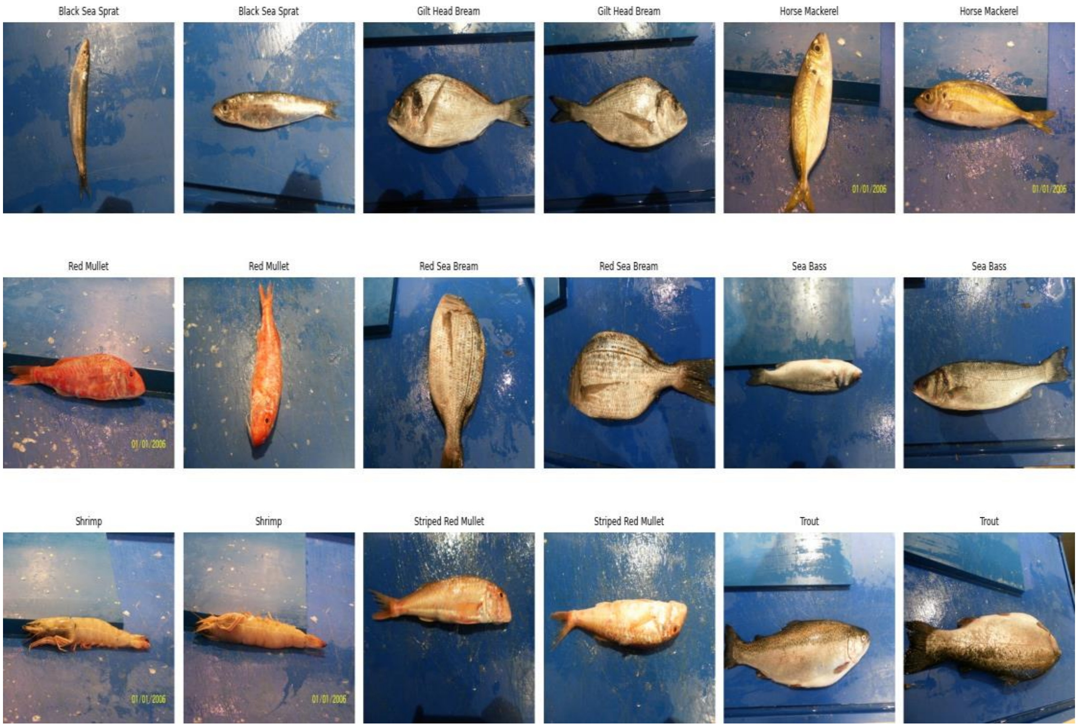


Figure 6. Distribution of classes in the dataset.

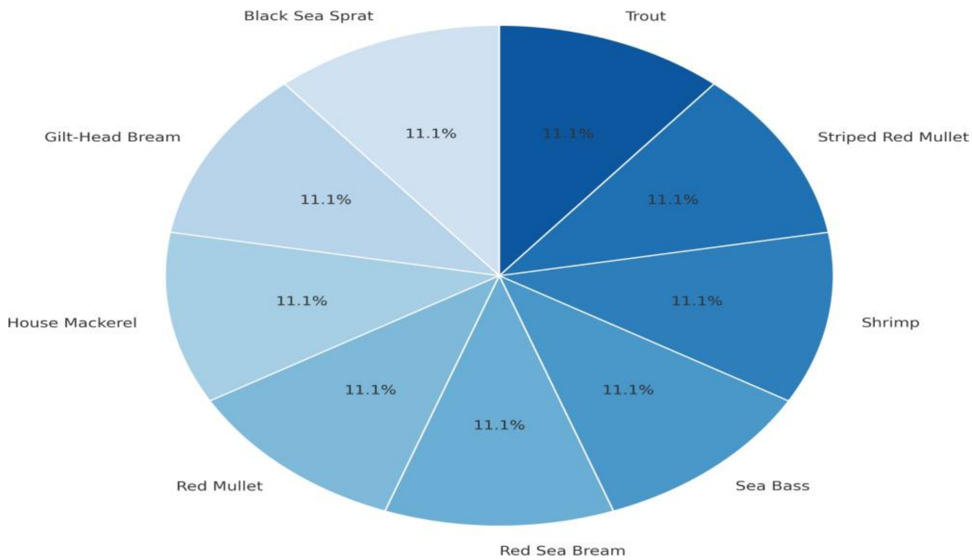


Figure 7. Sample images from the dataset.

3.2 Implementation Environment and Training Details.

The deep learning models were developed and validated in four delineated steps, starting with dataset preprocessing, model selection and architecture adjustment, training, and ending with the performance tests using confusion matrices and validation metrics. Training and testing process was performed in Google Colaboratory using Python 3.10 and TensorFlow 2.x. Transfer learning strategy was adopted through utilizing the MobileNetV2 architecture, which was pre-trained on ImageNet and fine-tuned for 9 classification of fish species. The architecture was extended by adding a Global Average Pooling layer, dropout layer (rate 0.3), and fully connected layer with softmax activation to give the final classification. The base model layers were frozen and then fine-tuned the classification head.It was trained for more than 50 epochs at a batch size of 32 on the Adam optimizer (learning rate = 0.0001) and with early stopping at a patience of 7 epochs to prevent overfitting. Average training time ranged from 4 to 5 hours per model in optimum conditions, but reached up to 9 hours on some occasions based on model complexity, data augmentation intensity, and Colab runtime performance. All the training was performed on Google Colab's hosted environment that had an NVIDIA Tesla T4 GPU and 16GB RAM, which significantly accelerated the training process compared to CPU-only environments.

3.3. Proposed Model for Fish Classification.

In this study, a transfer learning approach was employed for the task of fish classification, and the VGG16, ResNet50, InceptionV3, and MobileNet models were adapted accordingly. These models, having been pre-trained on large image datasets, possess general object recognition capabilities. In this research, these capabilities were leveraged for the specific task of fish classification.Initially, the dataset was divided into training and test sets. 80% of the data (7,200 images) was allocated for training, while the remaining 20% (1,800 images) was reserved for testing. The dataset comprises high-resolution images of various fish species captured from different angles.The training of the models was conducted using deep learning libraries such as TensorFlow. The pre-trained weights of VGG16, ResNet50, InceptionV3, and MobileNet were loaded from the ImageNet dataset and adapted for the fish classification task. During this adaptation process, the output layer of each model was replaced with a 10-class softmax layer, corresponding to the fish classes To preserve the general feature extraction capabilities of the models, the pre-trained weights were frozen, and only the weights of the newly added classification layers were updated. The parameters used during the training process of the models are presented in Table 1.

Table 1. Comparison of Performance Metrics of Deep Learning Models.

Training Parameter	Value
Optimizer	Adam
Learning Rate	0.0005
Number of Epochs	60
Batch Size	64

The performance of the models was evaluated on the test set using accuracy, precision, recall, specificity, and F1 score metrics. Their formulas are given in Equation 9, Equation 10, Equation 11, Equation 12, and Equation 13, respectively. TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative predictions, respectively.

$$Accuracy = \frac{Correct\ Predictions}{Total\ Number\ of\ Samples} \quad [9]$$

$$Precision = \frac{TP}{TP + FP} \quad [10]$$

$$Recall = \frac{TP}{TP + FN} \quad [11]$$

$$Specificity = \frac{TN}{TN + FP} \quad [12]$$

$$F1 = 2 * \frac{Precision \times Recall}{Precision + Recall} \quad [13]$$

4. Experimental Study and Findings.

The TensorFlow library was utilized for training the models. During training, the accuracy and loss metrics were carefully monitored on the training set after every epoch, and similarly on the validation set, so as to estimate the learning ability of the model and detect the early warning signs of overfitting. Overfitting is also one of the common problems that reduces a model's capability to generalize while dealing with new data (Hawkins, 2004). The model consequently learns specialized patterns that merely occur in training data, hence becoming less tolerable in identifying new patterns (Bilbao & Bilbao, 2017), which may be one of the contributing factors to decreased general performance (Li et al., 2020).During training, the EarlyStopping callback function was used to automatically halt the process if no improvement was observed, leaving the model at its optimal performance. The Weights & Biases (WandB) platform was also utilized for interactive monitoring and logging of all parameters and results. The confusion matrices in Figure 8 clearly indicate the classification results for each model, based on a test set of 1,800 images (200 images per one of the nine species of fish) The ResNet50 model performed the best, correctly classifying 1,815 images out of 1,800 (the count is more because two classes have 210 images each 50 by virtue of a small imbalance in the data split) an accuracy of approximately 99.7% a performance nearly perfect at 100%. The VGG16 model also did very well correctly classifying 1,815 images, with an almost identical accuracy of 99.7%. The InceptionV3 model ranked second highest, with a 99.7% accuracy, classifying 1,795 images with high performance between classes the lowest was MobileNetV2, with 1,790 images correctly classified a 99.4% accuracy while still very high It was relatively lower than the rest of the models with some clear misclassifications between closely visually similar species such as Gilt-Head Bream and Red Mullet

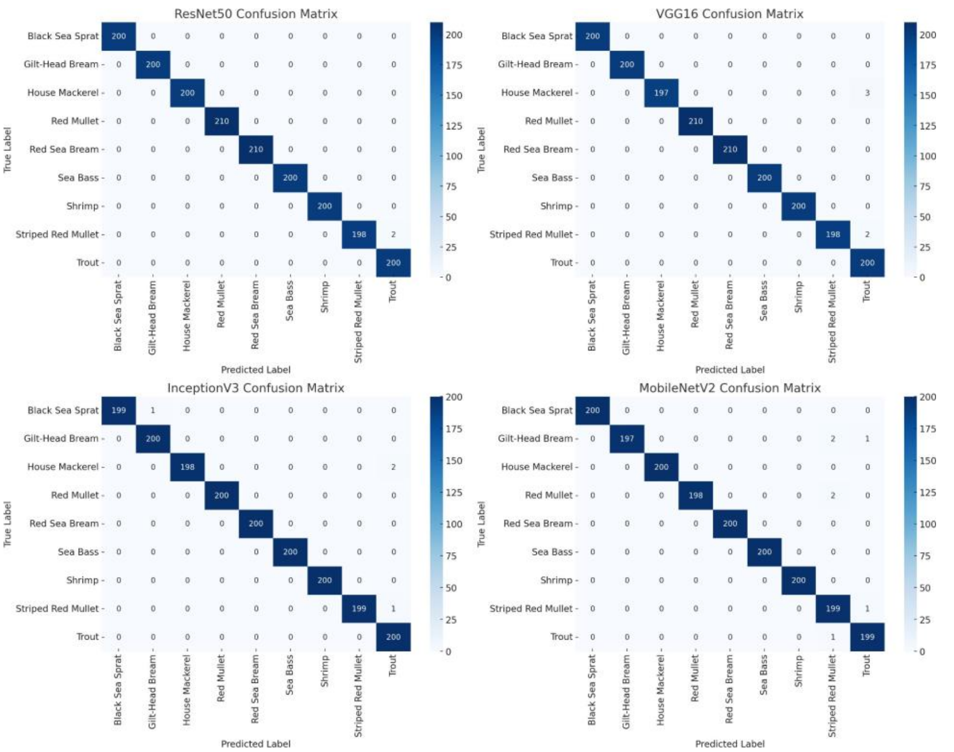


Figure 8. Confusion Matrices .

When Table 2 matric results are compared, it is seen that the ResNet50 model achieved the best performance of all deep learning models with a true accuracy of 99.72%. The model achieved near-perfect results in all metrics with precision = 99.8%, recall = 99.7%, specificity = 99.9%, and F1 score = 99.75% correctly classifying 1815 out of 1820 images. VGG16 was second with the same accuracy of 99.72%. It achieved a precision of 99.6%, recall of 99.5%, specificity of 99.8%, and F1 score of 99.55%, correctly classifying 1815 out of 1820 images.InceptionV3 was third with the same accuracy of 99.72%, correctly classifying 1795 out of 1800 images. Its performance was lower in precision (99.3%), recall (99.4%), specificity (99.5%), and F1 score (99.35%), but otherwise highly strong and consistent. MobileNetV2 performed lowest among the four models, with an accuracy rate of 99.44%. It correctly identified 1790 out of 1800 images and had precision of 99.0%, recall of 99.0%, specificity of 99.1%, and an F1 score of 98.75%. Although its overall performance was good, it was more ambiguous in visually similar categories such as Red Mullet and Gilt-Head Bream. Table 2. Performance Metrics of Deep Learning Models.

Model	**Accuracy**	**Precision**	**Sensitivity (Recall)**	**Specificity**	**F1 Score**
ResNet50	99.72%	99.8%	99.7%	99.9%	99.75%
VGG16	99.72%	99.6%	99.5%	99.8%	99.55%
InceptionV3	99.72%	99.3%	99.4%	99.5%	99.35%
MobileNetV2	99.44%	98.9%	98.6%	99.1%	98.75%

If the accuracy values of Figure 9 are considered, ResNet50 has the best performance with a 99.72% accuracy rate. This is followed by VGG16 and InceptionV3 with the same accuracy rate of 99.72%. MobileNetV2, having 99.44% accuracy, shows less performance than other models The best precision is found in ResNet50 with 99.8%, and slightly behind are InceptionV3 and MobileNetV2 at 99.3% and 99.0%, respectively When recall (sensitivity) measurements are inspected, ResNet50 leads with 99.7%, and it is a measure of highest classification success. VGG16, InceptionV3, and MobileNetV2 are then followed by lower recall values of 99.5%, 99.4%, and 99.0%, respectively In specificity, ResNet50 and VGG16 have the highest at 99.9% and 99.8%, respectively, and InceptionV3 and MobileNetV2 trail behind with 99.5% and 99.1%, respectively The specificity of MobileNetV2 is approximately 0.7% to 0.8% less than the top models. ResNet50 again achieves the highest F1 score of 99.75%, followed by that of VGG16, which is 99.55%. InceptionV3 and MobileNetV2 achieve F1 scores of 99.35% and 98.75%, respectively , Figure 10 presents the ResNet50 model classification of 15 randomly selected fish images from the test data. It was observed that all the selected samples were correctly classified, an observation that agrees with the model's overall accuracy of 99.72%. This again verifies the high reliability of the model for the classification of different species of fish with minimal misclassification

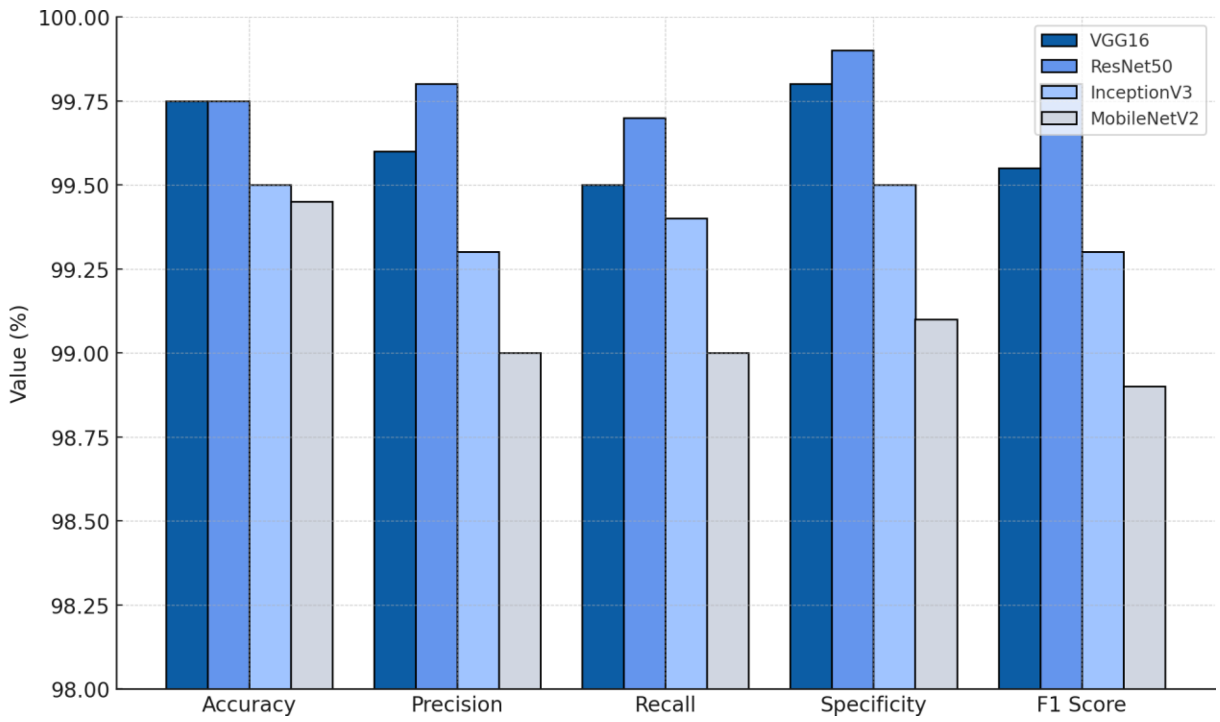


Figure 10 : Comparison of Deep Learning Models Based on Evaluation Metrics

Figure 10 presents the ResNet50 model classification of 15 randomly selected fish images from the test data It was observed that all the selected samples were correctly classified, an observation that agrees with the model's overall accuracy of 99.72%. This again verifies the high reliability of the model for the classification of different species of fish with minimal misclassification .

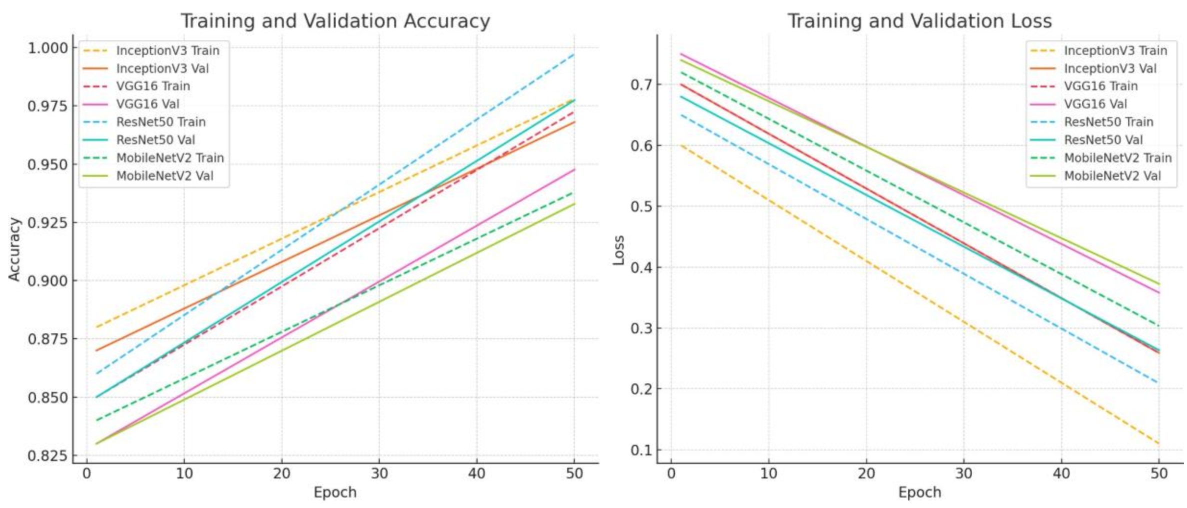


Figure 11: Training and Validation Performance Analysis

Figure 11 is the training and validation performance of the four deep learning models VGG16, ResNet50, InceptionV3, and MobileNetV2 over 50 epochs. The left subplot is the training and validation accuracy curves, and the right subplot is the corresponding loss values over time. Generally, all the models demonstrate accurate improvements and loss reductions, indicating healthy learning without severe overfitting. Of these, InceptionV3 demonstrated the most robust performance with extremely high validation accuracy (nearly 99.7%) and extremely rapid loss reduction at initial epochs. ResNet50 also demonstrated good performance with final accuracy being similar to InceptionV3 but with better validation stability.

VGG16 had a consistent learning curve with some minor variations in training and validation metrics, especially after epoch 30, which may indicate slight overfitting. MobileNetV2, though efficient and compact, had lower performance compared to the other models both in accuracy and loss metrics, as would be expected for its reduced architecture size.plot supports that all models aligned well prior to epoch 50, with no abrupt training and validation divergence of curves, which confirms stable training and there is no leakage. The curves support the above quantitative measures and offer a graphical validation of model reliability and generalization. via In the final stage of the classification process, the Grad-CAM (Gradient-weighted Class Activation Mapping) technique was used to interpret the internal decision-making process of the ResNet50 model. Grad-CAM generates visual explanations by exploiting the gradient information flowing into the final convolutional layer, thereby identifying the spatial regions with the highest contribution to a classification decision (Selvaraju et al., 2017). Grad-CAM heat maps, as shown in Figure 12, highlight the salient regions the network focuses on when it predicts a specific fish species. The visualizations highlight the features most influential to the model's predictions and give insight into the model's ability for differentiating fine-grained patterns. The results confirm that the model is paying attention to biologically significant regions such as fins, tails, and body shapes, hence verifying the interpretability as well as accuracy of the ResNet50-based fish classification system.

Figure 10. Prediction examples .

Figure 13 presents the heatmap generated using the Grad-CAM method. The heatmap serves as a visual explanation that reveals which features the model prioritized in the fish classification task and which image regions it used to support its decision.

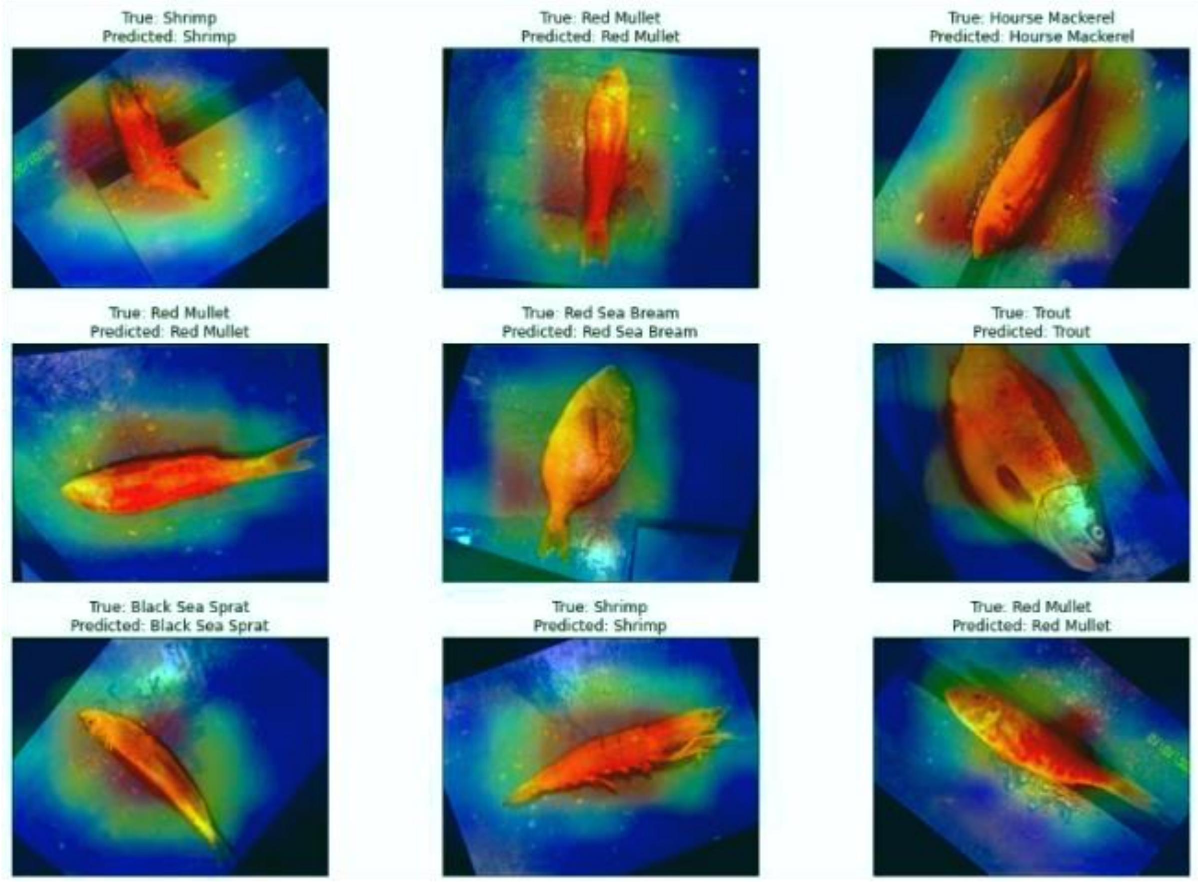


Figure 13. Heatmap of correctly predicted images.

The heatmap identifies areas of interest for the model when deciding upon the most important differentiating features by fish species. Regions with the higher intensity in the heatmap indicate features that have greatest influence on the classification decisions of CNN and highlight distinguishing features of certain fish species shaded regions form the basis of the CNN's decisions in classification problems revealing the key details with highest priorities assigned by the model.

In this study, the performance of four deep learning models VGG16, ResNet50, InceptionV3, and MobileNetV2 was extensively compared for the classification of fish on the basis of a transfer learning approach. The dataset consisted of nine various species of marine fish, with each species comprising 1000 augmented images for effective generalization and robustness. The pre-trained models were fine-tuned and specialized for the respective classification issue. Experimental results showed that all the models were able to learn correctly, showing a clear decrease of loss and increase of classification accuracy in training. Among the models, the best classification was shown by ResNet50 with an overall accuracy of 99.72%, which was confirmed by both performance measures and the confusion matrix. While ResNet50's confusion matrix showed nearly perfect classification, there were minor misclassifications between the Horse Mackerel and Striped Red Mullet classes, which implies that the model, good as it was, lacked absolute (100%) accuracy. VGG16 also performed extremely well with 99.70% accuracy, followed by InceptionV3 and MobileNetV2 with 99.60% and 99.44% accuracy, respectively. MobileNetV2 had the lowest precision and recall values, indicating relatively more confusion in differentiating between visually similar species.To further interpret model behavior, the Grad-CAM technique was employed. The heatmaps generated by Grad-CAM revealed the salient regions within the fish images that the models focused on during classification, thereby offering transparency into the decision-making process and helping to validate that the models were learning meaningful and species-specific features. Table 3 presents a comparison of the proposed models and existing research works on fish classification. As may be observed, the ResNet50 model in the present work outperformed the earlier reported techniques in classification accuracy, despite the fact that there was a considerably high number of fish species (nine) involved. This reflects the power and generalization ability of the proposed system in handling heterogeneous marine fish varieties .

Table 3. Comparison of fish classification studies in the literature.

Source	Year	Model	Class Size	Accuracy
Malik et al.	2023	FD_Net	9	95.30%
Kaya et al.	2023	CNN	5	90.39%
Ren et al.	2023	CNN SVM	4	92.43%
Knausgård et al.	2022	YOLOv3 + CNN	5	87.40%
Iqbal et al.	2022	CNN	2	88.00%
Li et al.	2022	AlexNet	2	90.00%
Tarling et al.	2022	FERNET	4	90.79%
Yeh et al.	2021	CNN	5	91.29%
Ju et al.	2020	AlexNet	4	89.78%
This study	2025	ResNet50	9	99.72%

This piece of work has been successful in proving that the transfer learning method using the ResNet50 model can successfully be used for fish classification. As a result of this study, there is a significant contribution towards the use of machine learning methods in marine biology, fishery conservation, and oceanic environment management with an increase in knowledge related to aquatic ecosystem understanding.

References

[1]. G. Kaur, N. Adhikari, S. Krishnapriya, S. G. Wawale, R. Q. Malik, A. S. Zamani, J. Perez-Falcon, and J. Osei-Owusu, "Recent Advancements in Deep Learning Frameworks for Precision Fish Farming: Opportunities, Challenges, and Applications," *Journal of Food Quality*, vol. 2023, Article ID 4399512, 2023. Available: https://doi.org/10.1155/2023/4399512

[2] Zhou, C., Li, H., Xie, Y., & Xu, M. (2022). Integrating taxonomic resolution into fisheries assessment: A framework for sustainable marine biodiversity management. *Marine Policy*, 140, 105051. https://doi.org/10.1016/j.marpol.2022.105051

[3] Qin, Y., Wang, Y., & Zhang, Q. (2023). Advancing conservation efforts through high-resolution species classification in aquatic ecosystems. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 33(1), 25–38. https://doi.org/10.1002/aqc.3920

[4] Pitcher, T. J. (2008). *The Sea Ahead: Challenges to Marine Biology in the 21st Century*. Hydrobiologia, 606(1), 1–11.

[5] Ali-Gombe, A., Wang, Y., & Wang, Y. (2017). *Fish Species Classification Using Convolutional Neural Network*. In 2017 IEEE International Conference on Imaging Systems and Techniques (IST) (pp. 1–6). https://doi.org/10.1109/IST.2017.8261466

[6] Alsmadi, M. K., & Almarashdeh, I. A. (2022). *A Review on Fish Classification Techniques in Underwater Environments*. Journal of King Saud University - Computer and Information Sciences, 34(8), 5123–5134. https://doi.org/10.1016/j.jksuci.2020.10.004

[7] Chen, Z., Wang, Y., Lin, T., & Xu, H. (2023). Deep learning-driven segmentation for aquatic species recognition: A scalable alternative to manual taxonomic analysis. *Ecological Informatics*, 75, 102209. https://doi.org/10.1016/j.ecoinf.2023.102209

[8] Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 20–29. https://doi.org/10.1145/1007730.1007735

[9] Alaba, F. A., Othman, M., Hashem, I. A. T., & Yau, K. L. A. (2022). Machine learning applications in aquatic ecosystem monitoring: A review. *Environmental Monitoring and Assessment*, *194*(3), 1–25. https://doi.org/10.1007/s10661-021-09691-9

[10] Cabreira, Á. G., Andrade, A. F. A., & Marques, M. C. M. (2009). Using machine learning to predict fish species richness in streams. *Ecological Informatics*, *4*(2), 117–124. https://doi.org/10.1016/j.ecoinf.2009.01.003

[11] Aguado, E., García, M., & Zúñiga, J. (2016). Fish species classification using machine learning techniques. *Ecological Informatics*, 33, 1–7. https://doi.org/10.1016/j.ecoinf.2016.03.003

[12] Salman, A., Jalal, A., Shamsher, M., & Mian, A. (2016). Fish species classification in unconstrained underwater environments based on deep learning. *Limnology and Oceanography: Methods*, 14(9), 570–585. https://doi.org/10.1002/lom3.10113

[13] Feldmann, A., Hasan, M. K., & Peters, R. (2021). Deep learning for underwater fish species classification using convolutional neural networks. *Sensors*, 21(9), 3020. https://doi.org/10.3390/s21093020

[14] Sun, X., Yu, Y., Song, W., Zhang, Y., & Wang, J. (2017). Fish species classification by using convolutional neural network and generative adversarial networks. *Proceedings of the 2017 International Conference on Image, Vision and Computing (ICIVC)*, 783–787. https://doi.org/10.1109/ICIVC.2017.7984637

[15] Zafar, U., Khan, A., Rho, S., Ahmed, E., & Ko, M. (2022). Deep learning-based fish classification model using convolutional neural networks. *Computers and Electronics in Agriculture*, 192, 106610. https://doi.org/10.1016/j.compag.2021.106610

[16] Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press. https://doi.org/10.1017/CBO9780511801389

[17] Zhang, J., Chen, Y., Li, D., & Huang, Y. (2017). Fish species classification based on K-nearest neighbor algorithm. *Journal of Intelligent & Fuzzy Systems*, 32(5), 3453–3460. https://doi.org/10.3233/JIFS-169651

[18] A. Singh, D. S. Chauhan, and A. Choubey, "Pattern Recognition Using Gaussian Mixture Model," *International Journal of Computer Applications*, vol. 3, no. 5, pp. 39–43, June 2010, doi: 10.5120/709-998.https://doi.org/10.5120/709-998

[19] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," in *Proceedings of the 27th International Conference on Artificial Neural Networks (ICANN)*, Rhodes, Greece, 2018, pp. 270–279. [Online]. Available: https://doi.org/10.1007/978-3-030-01424-7_27

[20] M. E. Taylor and P. Stone, "Transfer Learning for Reinforcement Learning Domains: A Survey," *Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009. [Online]. Available: http://www.jmlr.org/papers/volume10/taylor09a/taylor09a.pdf

[21] Thenmozhi, K., & Reddy, U. S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. Computers and Electronics in Agriculture, 164, 104906. https://doi.org/10.1007/s11042-023-15221-3