An advanced Bangladeshi local fish classification system based on the combination of deep learning and the internet of things (IoT)

Article *in* Journal of Agriculture and Food Research · June 2023

DOI: 10.1016/j.jafr.2023.100663

5 authors, including:

Wahidur Rahman
Uttara University

Abdul Hasib Uddin
Khwaja Yunus Ali University

Md Tarequl Islam
Khwaja Yunus Ali University

Journal Pre-proof

An advanced Bangladeshi local fish classification system based on the combination of deep learning and the internet of things (IoT)

Md. Asif Ahmed, Md. Shakil Hossain, Wahidur Rahman, Abdul Hasib Uddin, Md. Tarequl Islam

PII: S2666-1543(23)00170-9

DOI: https://doi.org/10.1016/j.jafr.2023.100663

Reference: JAFR 100663

To appear in: *Journal of Agriculture and Food Research*

Received Date: 20 February 2023

Revised Date: 8 April 2023

Accepted Date: 1 June 2023

Please cite this article as: M.A. Ahmed, M.S. Hossain, W. Rahman, A.H. Uddin, M.T. Islam, An advanced Bangladeshi local fish classification system based on the combination of deep learning and the internet of things (IoT), *Journal of Agriculture and Food Research* (2023), doi: https://doi.org/10.1016/j.jafr.2023.100663.

Reference ID: JAFR_100663

An advanced Bangladeshi local fish classification system based on the combination of Deep Learning and the Internet of Things (IoT)

Md. Asif Ahmed[1], Md. Shakil Hossain[1], Wahidur Rahman[2, 3*], Abdul Hasib Uddin[1], Md. Tarequl Islam[1]

[1]Department of Computer Science and Engineering, Khwaja Yunus Ali University, Enayetpur, Sirajganj-6751, Bangladesh

[2]Department of Computer Science and Engineering, Uttara University, Dhaka.

[3]Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Santosh, Tangail-1902, Bangladesh

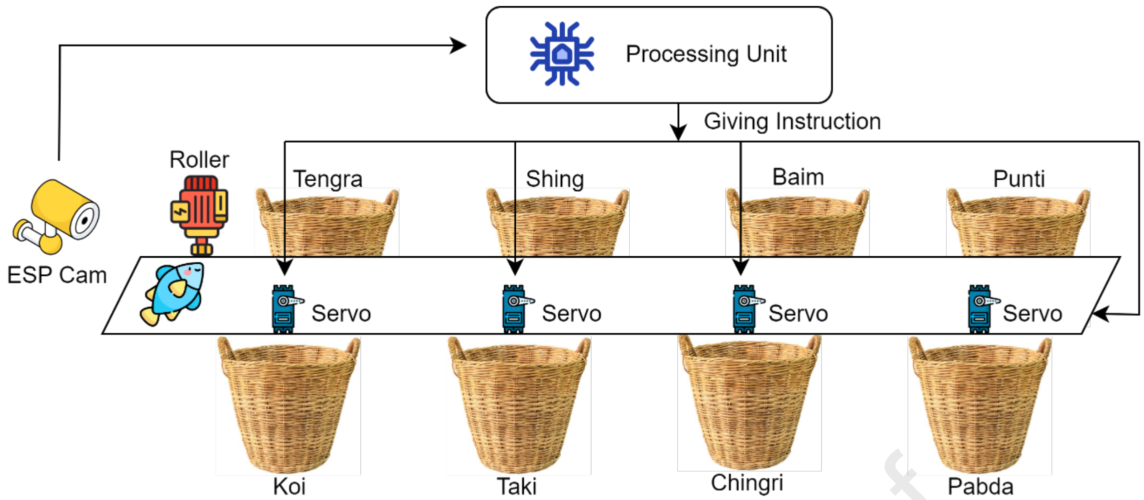
Email: asifahmedhridoy.cse@gmail.com, sakill.cse@gmail.com, wahidtuhin0@gmail.com, abdulhasibuddin@gmail.com, tareq.cse@gmail.com

***The corresponding Author:**

Name: Wahidur Rahman

Address: Department of Computer Science and Engineering, Uttara University, Dhaka Bangladesh.

Email: wahidtuhin0@gmail.com



Processing Unit
Giving Instruction
ESP Cam
Roller
Tengra
Shing
Baim
Punti
Servo
Servo
Servo
Servo
Koi
Taki
Chingri
Pabda

An Advanced Bangladeshi Local Fish Classification System based on the Combination of Deep Learning and the Internet of Things (IoT)

Abstract

Fish classification leads to the automated machine-based fish separation system. In terms of classification and real-time data monitoring, deep learning and the Internet of Things (IoT) each provide an efficient solution. This paper focuses on the development an embedded system based on the principles of Deep Learning and IoT. The proposed methodology is classified into interconnected parts. The first part describes the working principles of DL with along the dataset building, model analysis and overall system architecture. A new dataset from eight different Bangladeshi fish species. In the process of DL, First, two sets of datasets have been created namely, setup-1(S1) containing original images and setup-2(S2) containing Unsharp masked photos. Then, seven conventional ImageNet pertained state-of-the-art deep learning models on both benchmarking setups: InceptionV3, Xception, DenseNet121, DenseNet169, DenseNet201, InceptionResNetV2, and ResNet152V2. In the process of IoT, the architectural design of a smart contained has been deployed with the aid of several kinds of sensors and microcontroller. This research has found satisfactory results with the DL models and IoT based components. The best benchmark accuracy for setup-1 was 96% for all of the DenseNet121, DenseNet169, and DenseNet201 architecture, and for setup-2, it was 96% for the Xception model. Finally, we have constructed a hybrid (CNN+ Convolutional LSTM) model, for which the accuracy was 97%, outperforming all of the abovementioned state-of-the-art methods. Besides, the research has performed some experiment with the IoT based Solution. Though the proposed solution has exhibited some drawbacks, it can be practicable in real-time solution.

Keywords: *Deep Learning (DL); Convolutional Neural Network (CNN); Internet of Things (IoT); Fish Classification*

1. Introduction

Fish classification is the process of separating distinct species of fish from one another from their mixing situation. Day by day we are diving into such era that is mostly automated and its called industrial revolution. Intelligent machine-based separation system or industrial automation is one of them. Industry 4.0, the name given to the subsequent industrial revolution, denotes a new stage that places a strong emphasis on connection, machine learning, automation, and real-time data [1]. Day by day AI is getting more popular. The artificial neural network computation utilizes the kind of extracting solvents and photo indexes as inputs [2]. The processes in the recognition of various species are carried out via dependable classification tasks using machine learning along with deep learning techniques including convolutional neural networks, artificial neural networks, and support vector machines [3]. A clean, profitable, and long-term energy-efficient industry is in high demand [4]. Maximum area of this highly demandable industry can be built by combining AI and IoT. Artificial Neural Networks (ANN), Fuzzy Logic algorithms (FL), Random Forest (RF), and Long Short-Term Memory (LSTM) are the mostly used and highly convenient approaches [5].

In this time around the globe, there are almost 32,000 varieties of fish, and roughly 40% of them inhabit freshwater. In Bangladesh, there are 401 species of marine fish. There are 251 species of inland fish (both coastal and freshwater) [6]. From the freshwater fish species, we have selected 8 of them for our research work, which are mostly available in the local fish market of Bangladesh. Bangladesh has a plentiful supply of water as well as other aquatic goods, such as seafood, because of its near proximity to the Bay of Bengal. In Bangladesh, fish is considered one of the most widely consumed foods. Fish consumption in Bangladesh is estimated to be nearly every other day for around 60% of the population [7]. In terms of freshwater fish production, Bangladesh has kept the third rank. Bangladesh also moved up two positions to third place in the world for fish farming. For six years, Bangladesh ranked sixth in the world for fish farming. In 1980, Bangladesh produced 440,000 tons of freshwater fish, according to government statistics. By 2020, which had expanded to 1.25 million tonnes, or 11% of the world's total production of fish. On a list of the top 25 marine fish producing nations worldwide, Bangladesh comes in at number 25. With 670,000 tonnes right now, the country's entire marine fish production has climbed by 100,000 tonnes. The country's fish farming and trading sector employs around 20 million people. Around 7.5 kilograms of fish were consumed annually per person in 1990's, and it's now 30 kilograms [8].

Small-scale fisheries (SSF) are essential for reducing poverty, promoting rural advancement, and ensuring the security of food and wellbeing in developing nations [9]. Fish is a vital source of nutrition, particularly in situations when all the other protein sources are limited or unaffordable [10]. Fish is a type of food that spoils quickly and has significant losses between harvest to consumption [11]. Inadequate infrastructure, which includes a lack of processing and cold chain capabilities, insufficient transportation, and fishing equipment that increases post-harvest wastage, place restrictions on small-scale fisheries [12]. Fish dealers travel to various fish ports to purchase fish and carry it to marketplaces in major cities [13]. According to reports, 10% of the commercial fishing worldwide is ruined because of inappropriate management, processing, preservation, and transportation [14].It takes a long time to separate the fish once the fisherman has sold their catch, during which time the fish may spoil. Every processing plant must have some laborers to separate different species of fish. The price of labor is high, so the production cost increases. When using an automated system, labor costs might be cut to minimize production costs. So, it necessitates a system that automatically recognizes different fish species. A smart fish classification system can accomplish automated fish monitoring and packing. The following aspects are included in such a system, especially for fish processing plants. For example, remote monitoring, local and remote alerting systems, AI-based recognition (Deep Learning), Sensor-based monitoring, and automated packaging system. The term "classification" in deep learning relates to a predictive analysis concern in which a class label is estimated for a particular instance of input data [15]. An automated fish classification system is essential to distinguish valuable fish for considerable financial advantage.

In previous studies, K.Kottursamy, developed a multi-scale CNN for underwater fish image classification [16]. X. Xu et al., introduced identify of imbalanced fish species on a small scale using the SE-ResNet152 model [17]. Self-supervised fish segmentation in underwater films utilizing transformers was exhibited by Saleh et al., based on transformer-based and (CNN)-based self-supervised techniques[18]. Badawi, introduced the classification of fish using appropriate feature extraction. With diverse species in each of the 24 fish families, using his suggested GSB classifier[19]. CNN-based fish species detection application in the Leyte Gulf, the article presented by Dialogo et al.[20]. Paraschiv et al., has categorized images and videos of underwater fish via Very Small CNN[21]. Jin et al., achieved that DNA barcode sequences are used to classify fish using a deep learning technique. Elastic Net-Stacked Autoencoder (EN-SAE) and Kernel Density Estimation (KDE) are used to create the revolutionary deep learning technique known as the ESK model [22]. An efficient system for fish detection using autonomous spatial pseudo-labeling was proposed by R. J. M. Veiga et al., In their paper, based on Transfer Learning [23]. Convolutional neural network topologies for fish species classification have been developed by Muslim and Zulkifli, with the use of data augmentation, and transfer learning was used to train CNN models to categorize fish photos [24]. Kholoud Elbatsh et al., included data augmentation, and transfer learning was used to train CNN models to categorize fish photos. Image categorization was performed using (DCNN) AlexNet. [25]. Fish categorization using a DCNN was presented by Smadi et al.[26]. Employs convolutional neural networks to categorize several koi fish varieties (CNN). The research done by Cueto et al., To extract the necessary textures from the image, the researchers examined RGB, HSV, and grayscale color spaces [27]. A content-based fish classification system that combines local and global aspects developed by Islam et al., Local Binary Pattern (LBP), Speeded-Up Robust Feature (SURF), and (SIFT) are employed for the local feature extraction from the fish photos [28]. Deep Learning Techniques Used for Automatic Fish Recognition and Segmentation in Taiwan's Fish Market developed by Chen et al., The fish recognition system presented in their work combines a cutting-edge instance segmentation technique using ResNet-based classification [29]. Jose et al., used 2D-LBP models based on region-based CNN groups to classify tuna. Once the models are optimized, the output from the last convolutional layer is sent to a grouped 2D-local binary pattern descriptor (G2DLBP) [30]. Rastrelliger kanagurta and Rastrelliger brachysoma were proposed species by Kurniawan et al., using CNN. Their work uses CNN, which has architectures with two to five convolutional layers, to classify those two species, utilizing CNN1, CNN2, CNN3, and CNN4 [31]. Jafari et al., suggested an accurate identification of common carp from both naturalistic and farmed sources. To examine the common carp species categorization in Iranian and farmed common carp, data mining techniques were applied to their study [32]. M. Hasson, aims to review several fish species identification methods, including pre-processing procedures, feature extraction approaches, recognition/classification methods, and databases in his article [33].

With the help of the intelligent fish classification approach, the suggested system provides a smart system that allows users to differentiate valuable fish from situations where other fish species are present. The following are the article's contributions:

- To achieve the best results in the field of fish classification approach, it is crucial to integrate two technological concepts, IoT and machine learning.
- An intelligent method of classifying and dividing several fish species using machine learning and image recognition.
- A microprocessor, a weight detector, and an ultrasonic sensor are used to produce a smart fish-packing container.
- A clever method of real-time fish monitoring employing Android application, wireless connection for short-range, and Internet of Things technology for long-range.
- In machine learning, we have combined two neural network methods CNN and RNN, which gives more accurate and steady results.

Five sections make up this article. The overview of related works is provided in Section 2, the strategies for collecting and processing data and the comprehensive methodology are shown in Section 3, and the outcomes of our suggested system are shown in Section 4. Finally, section 5 serves as an illustration of this article's conclusion.

2. Comprehensive Literature Reviews

We have divided our literature review section into 3 sub-sections namely, deep learning based, computer vision and machine learning based, & CNN and IoT based.

2.1 Deep Learning based

Muslim and Zulkifli, developed a convolutional neural network topology for fish species categorization. CNN features are used to solve classification issues from labeled image data. With data augmentation, transfer learning was used to train CNN models to categorize fish photos. Employed dataset of 18,000 fish photos over 18 categories, 5,400 of which were used for validation (about 30%) and 12,600 for training (about 70%). AlexNet models exhibit an accuracy of 99.85% [24]. In a paper Elbatsh et al., exhibited recognizing fish species frequently found in the Sea of Mediterranean and, therefore, a cutting-edge smartphone application is suggested in the fish market. 2670 photograph constructed dataset, the data was separated into three portions, with training and validation dataset being picked from 80% of the dataset and testing dataset constructed from the remaining 20%. Used AlexNet for picture classification. RELU and SOFTMAX were selected for the primary network, and the model shows an accuracy of 80%. It might have included more datasets of other fish families, allowing the software to be utilized globally [25].

Smadi et al., presented an efficient fish classification approach based on convolutional neural networks in their paper study. The 2610 photos in the BD2019 fish dataset are divided into eight groups. 3 types of splitting (80%-20%, 75%-25%, and 70%-30%) were used. Stochastic gradient descent (SGD), adaptive momentum (Adam), Adaptive delta (AdaDelta), Adaptive max pooling (Adamax), and Root mean square propagation (Rmsprop) have been compared for CNN. Their improved CNN shows an accuracy of 98.46% [26]. Mostafa et al., developed an automated classification of Nile Tilapia fish using machine learning methods. A Dataset of 55 photos of non-Tilapia fish and 96 photographs of tilapia fish was used. 50% was used for training purposes, and the other was used for testing. Employed the support vector machines (SVMs) algorithm together with feature extraction methods based on the scale-invariant feature transform (SIFT) and sped-up robust features (SURF) algorithms. The greatest result was attained by an SVM classifier using a linear function, which had an accuracy of 94.4% [34].

Cueto et al., constructed a convolutional neural network, several varieties of koi fish are classified. Fifteen varieties, training, and testing datasets contained 1200 and 300 pictures, respectively. Eighty training photos and 20 testing images are utilized. The average accuracy of 84% was calculated using CNN based on system testing. The prior study also employed the same dataset used in the study [27]. Islam et al., created a combination of machine learning techniques to classify fish based on content. The experiment dataset "QUT fish data" contains 256 fish photos of 21 classes. Local Binary Pattern (LBP), Speeded-Up Robust Feature (SURF), and Scale Invariant Feature Transform (SIFT) are used to extract features from fish images. Color Coherence Vector is used to extract general characteristics from a fish image. For fish species prediction, Decision Tree, k-Nearest Neighbor (k-NN), Support Vector Machines (SVM), Naive Bayes, and Artificial Neural Networks (ANN) were used. and got an accuracy is 98.46% [28].

Chen et al., produced deep learning techniques to segment and identify fish in the Taiwanese fish market automatically. There was a total of 41 fish species in this data collection. One thousand twenty-five photos made up the data set, 20 photos in the training data set, and five images in the test data set for each class. Constructed a system that incorporates ResNet-based classification and cutting-edge segmentation. The system had accuracy rates of 85% for Top-1 and 95% for Top-5 [29]. Jose et al., created a super-learning ensemble of CNN-grouped 2D-LBP models for categorizing tuna. There are 612 photos total in the dataset, 205 of Bigeye tuna, 200 of Skipjack tuna, and 207 of Yellowfin tuna. To improve the photos, the contrast stretching method is used. Each fish picture's three areas are separated and enhanced before trained CNN. The output from the last convolutional layer is delivered to a grouped 2D-local binary pattern descriptor once the models have been fine-tuned (G2DLBP). has a 93.91% accuracy on a separate test dataset [30].

Kurniawan et al., produced Rastrelliger brachysoma and Rastrelliger kanagurta species classification using CNN. Four hundred thirty-four photos were used in the dataset as a training group, with 217 images for each class. For each class, 21 photos were in the validation group and 19 in the test group. Employed the CNN approach, with CNN1, CNN2, CNN3, and CNN4 as the 2 to 5 convolutional layer designs, respectively. The CNN models' accuracy got 94.7% [31]. Montalbo and Hernandez, developed fish species classification using augmented data and DCNN. There were 530 total photos in the dataset. Seventy-five photos for testing and 455 photos for training. Re-training, fine-tuning, and optimizing processes are used to the DCNN's VGG16 to improve classification accuracy. The model is 99% accurate overall. The study may be expanded by creating online and mobile applications [35].

Jafari et al., induced accurate identification among Natural and Farmed common carp. Data mining methods were used in the research to look at the carp species categorization within Iranian and cultured common carp. Dataset of 76 pictures, including 14 images for the Anzali, 27 for the Gomishan, 19 for the Miankaleh, and 14 for the farmed carp. Used Naïve Bayes classifier and LDA method to classify. Using a linear discriminant method, we got an accuracy of 81.1% [32]. Kottursamy, exhibited a Multi-scale CNN pathway for proper detection of underwater fixed fish images. The Fish4knowlegdet dataset was used in comparing the performance of the proposed model. An algorithm has been presented to divide the datasets into 80% and 20% for training and testing purposes to get perfect accuracy. This 20% division is allotted for testing and validation for robust object detection. And top accuracy gain in MS-CNN is 96.99% [16].

Alsmadi et al., developed fish recognition based on the extraction of reliable information from size and shape measurements using a neural network. The ANN was chosen with reliability. Dataset contains three hundred fifty unique diverse fish photos from 20 different fish families. Two hundred fifty-seven training photos and 93 testing images were used to create two datasets from these images. An accuracy of 86 percent was attained using the ANN linked to the back-propagation method [36]. Ogunlana et al., revealed support vector machine-based classification of fish. In the paper, a support vector machine (SVM)-is constructed to improve fish species classification. Numerous methods have been successfully applied to address these issues, such as (KNN), K-mean Clustering, and neural networks. Using two subsets, 76 fish as a training set and second 74 fish as a testing set. The classification accuracy of 78.59 percent. The study can upgrade by focus on increasing the classification rate and classifying fish for larger datasets and species [37].

X. Xu et al., disclosed a transfer learning, and SE-ResNet152 networks-based system that can identify imbalanced fish species on a small scale. The generalized feature extractor SE-ResNet152 model is built and transferred to the goal dataset. The six fish species are categorized using three different body, head, and scale images. The three fish species were shown accuracy each 98.80%, 96.67%, and 91.25%. The study does not consider the impact of a complicated backdrop on fish species classification [17]. Saleh et al., exhibited a self-supervised fish segmentation in underwater videos using transformers. Their proposed model outperforms existing (CNN)-based and Transformer-based self-supervised methods. Study tests utilizing the DeepFish Seagrass and YouTube-VOS datasets, three publicly accessible datasets. The study could be extended by utilizing bigger dataset and bigger datasets of underwater fish to reach that potentate's segmentation model [18]. Badawi, developed fish classification using acceptable feature extraction. The study introduced a hybrid genetic algorithm (GA) that combines a back-propagation method (GSB classifier) with the simulated annealing (SA) algorithm. Total used 500 fish photos were, 350 photos used neural learning stage, and 150 images used for testing. With diverse species in each of the 24 fish families. The GSB classifier shown 87.7 percent accuracy compared to the back-propagation (BP) algorithms of 82.9 percent [19].

Dialogo, exhibited a system of Leyte Gulf using convolutional neural network in fish species detection application (FiSDA). The 6918 fish photos are utilized for 467 different species of fish. A total of 4,000 training epochs were used to train the model. The accuracy rates for training and testing were 96.49% and obtained 82.81% before 82.58% validation accuracy with a harm value of 1.868. by using more datasets might could improve the performance of the model [20]. Paraschiv et al., introduced an image and video classification of underwater fish using a very small CNN. They used 51,260 photos. They distribute these photos among the six species. Separated the image to training, validation, and testing into (30,646, 10,413, 10,201). An appropriate working degree of accuracy of 42% for six fish species is provided by 12,000 parameters. If images were extracted from videos and added to the data using a simple object-tracking technique, the accuracy could rise to about 49%. Utilizing different datasets might enhance the model's performance [21].

Jin et al., developed fish classification system using deep learning method with DNA barcode sequences. Presented the ESK model, a revolutionary deep learning technique that combines the Flexible Net-Stacked Autoencoder (EN-SAE) and Kernel Density Estimation (KDE). Used three great families, Sciaenidae, Mugilidae, and Barbinae, 21, 23, 103 species, 13,7, and 9 genera. The average F1-Score for accuracy, recall, and precision reached 98.96 %, 97.57%, and 97.43%. The research can be extend by examine a more effective technique to enhance fish classification based on DNA barcode sequences [22]. Veiga et al., exhibited an effective technique for fish detection via autonomous spatial pseudo-labeling. The autonomous fish species classification method is based on Transfer Learning, The OzFish dataset images were divided into training, validation, and test subsets, following a ratio of the distribution of 80%, 15%, and 5%, respectively for their study, for a total of 1753 images their developed method achieved average of 93.11% accuracy [23].

Table 1: Comparison among related works.

Reference	Dataset	Model	Contribution	Accuracy (%)	Limitations
Elbatsh et al. [25]	Total Image:2670 Species:128	CNN	During the system implementation, a huge data set were collected which helps other researchers and scientist	80%	Dataset combined with around twenty images per species.
Mostafa et al. [34]	Total Image:151 Species:2	SVM classifier	used support vector machine (SVMs) algorithm in conjunction with feature extraction techniques for the Nile Tilapia fish classification.	94.4%	Dataset has been made up with a few numbers of fish images and species.
Cueto et al. [27]	Total Image:1500 Species:15	CNN	To extract the necessary textures from the image, the researchers examined RGB, HSV, and grayscale color spaces	84%	Quality of the photos isn’t up to the mark.
Islam et al., [28]	Total Image:256 Species:21	ANN	Five classifiers ANN, KNN, SVM, Naive Bayes, and Decision Tree were employed to improve the accuracy.	98.46%	each species contains only twelve images.
Chen et al. [29]	Total Image:1205 Species:41	ResNet Architecture based.	Constructed a system that incorporates ResNet-based classification and cutting-edge segmentation.	95%	This data was generated with fewer image per species.

Jose et al. [30]	Total Image:612 Species:3	CNN	Authors separated three regions of each fish images and enhanced before used as input to previously trained convolutional neural networks (CNN)	93.91%	Only three species have been used to comprise the dataset.
Jafari et al. [32]	Total Image:76 Species:4	Linear Discriminant	The study examined Iranian and farmed common carp species categorization using data mining techniques.	81.1%	Only 4 species were used.
Ogunlana et al. [37]	Total Image:150 Species:2	SVM	Used several techniques, including KNN, K-mean Clustering, and neural networks.	78.59%	The dataset consists only 2 species.
Badawi. [19]	Total Image:500 Species:24	Genetic algorithm, GSB classifier.	The study introduced a hybrid genetic algorithm (GA) that combines a back-propagation method (GSB classifier) with the simulated annealing (SA) algorithm.	87.7%	The dataset was created with around twenty-one images per species.

2.2 Computer vision and Machine learning based

Rathi et al., built classification of underwater fish species applying Convolutional Neural Network and Deep Learning. The dataset has a total of 21 species and 27,142 pictures. Before the training phase, the applied image processing helps eliminate debris, non-fish bodies, and underwater impediments from the photos. The second stage applied CNN's Deep Learning method to classify the different types of fish. obtained a 96.29% accuracy. Image enhancement methods might improve the algorithm and compensate for lost image characteristics [38]. Sharmin et al., constructed detection of local fish using machine vision. 180 images of 6 species are included in the Dataset collection. A group of 14 features comprises four different categories of characteristics. Used histogram-based technique, characteristics are first extracted. PCA is used to reduce the number of features. The study uses three classifiers—SVM, k-NN, and Ensemble—. The maximum accuracy, 94.2%, is provided by SVM. Low-resolution and oblique photos are not detected by the system [39].

Kartika and Herumurti, exhibited classification of koi fish using the HSV color system. The dataset utilized 281 different kinds of koi. Nine categories are used to group the data. divided into a training set (66%) and a test set (34%). The item and backdrop were separated using the K-Means algorithm. Naive Bayes method compared with SupportVector Machine (SVM), which both use the K-Fold Cross Validation. accuracy had a 97% influence on tests. Research can develop by categorization techniques based on color, structure, and texture [40]. Arivazhagan et al., demonstrated fruit identification using Texture and Color characteristics. 2635 fruits of 15 various classes made up the database. half used to teach the system and half used to the testing set. Orange sugar content prediction with an Interrelation coefficient of 0.83 and classification of an apple grade after drying with an accuracy of 95% each. The minimum distance criterion is utilized in the categorization process. Shapes and sizes can be mixed with colors and textures to enhance the identification system [41].

Alsmadi et al., introduced a revolutionary fish classification system in their study that combined robust feature selection, picture segmentation, and geometric parameter procedures with an ANN and DT. Using 1500 fish photos to compress 100 individual fish photos, 100 were utilized for training the NN, and 75 were utilized for testing. Fish categorization accuracy achieved 97.4 percent. The study couldn't solve color image distortion, noise, segmentation mistakes, overlap, and object occlusion [42]. Zhang et al., introduced a Fruit classification that used feedforward neural networks with computer vision. The article based on FSCABC–FNN method. The data set contains 1653 images from 18 different categories. The research included additional features to increase classification accuracy [43].

Table 2: Comparison among best related works.

Reference	Dataset	Model	Contribution	Accuracy (%)	Limitations
Sharmin et al. [39]	Total Image:180 Species:6	SVM, KNN, and Ensemble.	A machine-vision based local freshwater fish recognition system was presented.	94.2%	The system isn't able to detect low-resolution and oblique photos.

Arivazhagan et al. [41]	Total Image:2635 Species:15	Feature extraction	The developed method can process, analyse, and recognize fruits based on color and texture.	[illegible]	The number of images in the dataset is slightly low.

2.3 CNN and IoT based

Allken et al., exhibited using a convolutional neural network trained on synthetic data to identify different types of fish using the Deep Vision trawl camera system. The Dataset collection contains 1,000 photos of each species, divided the dataset into a training, validation, and test set. The Keras library was combined with the TensorFlow deep learning framework in the study. A modern convolutional neural network model called InceptionV3 was implemented using Keras. The network was evaluated using a test data set of 2400 genuine photos, 800 of which represented each species. obtained a classification accuracy of 94% [44]. Chen and Chang, introduced storage of Koi fish using mobile cloud computing. Authors combined portable cloud computing and big data analytics methods, unsupervised texture pattern classification in fish tank videos, image retrieval based on SVM, and statistical methods for managing warehouses, in place of intrusive RFID-based systems. A tiny unit comprising a single title node and data node can detect a fish characteristic among 500,000 koi fishes in 7 seconds [45].

Rahman et al., introduced a deep learning-based automated waste management solution with IoT. In the dataset, 2527 photos made up the collection, including 501 glasses, 594 paper, 482 plastic, 403 cardboard, 137 rubbish, and 410 metal. 80% training set, 10% validation set, 10% test sets, 50% training set, 25% validation set, and 25% test sets were utilized to divide the data. The model suggests a clever technique to divide trash into digestible and undigested categories using a CNN. The hidden layers typically included convolutional, pooling, Rectified Linear Units (ReLU), and fully linked layers. The plan also presents the architectural layout of a smart garbage can that uses a microprocessor and several sensors. The suggested approach uses Bluetooth and IoT connection to monitor data. The suggested architecture, constructed on the CNN architecture, has an accuracy of classification of 95.3125% [46].

Table 3: Comparison among best existing work to fish management.

Reference	Dataset	Model	Contribution	Accuracy (%)	Limitations
Allken et al. [44]	Total Image:2400 Species:3	CNN	Keras library combined with the TensorFlow deep learning framework.	94%	Number of species that used in the study is comparatively low.

3. Methodology

We have divided our methodology section into 3 sub-sections namely, dataset and data collection, working principles (IoT), & deep learning.

3.1 Dataset and Data collection

There are no such datasets available on the internet including the specie used to create our dataset.

3.1.1 Dataset

Recognizing distinct fish species is one of the numerous uses for classifying fish. The research on various fish species benefits from the precise classification of fish [47]. In addition, the arrangement of fish is helpful for understanding fish behaviour and interspecies cooperation in a typical natural setting [48]. The categorization of fish species from photographs is a multiclass analysis showed and a fascinating study area in computer vision and machine learning [49]. Fish species classification may be automatically monitored thanks to automated fish categorization, which is crucial for fisheries research and to build automated packaging system [50]. To build an automated classification system the first needed thing is a dataset that is filled with plenty number of images of various species. In our work, we first created a dataset with unique species.

We have constructed a dataset of 8 different species, i.e., **Koi** *(Anabas testudineus)*, **Tengra** *(Batasio tengana)*, **Taki** *(Channa punctata)*, **Shing** *(Heteropneustes fossilis)*, **Chingri** *(Marcrobrachium malcoimsonii)*, **Baim** *(Mastacembelus armatus)*, **Pabda** *(Ompok bimaculatus)*, and **Punti** *(Puntius sophore)*. Our dataset contains 800 images, some are popular for farming in the local area, and some are found in the rivers of Bangladesh and other countries of the world. Each class contains 100 images. The dataset is multiclass and includes four main characteristics of fish species (head, Body, Scale, and Fin).

Using the BD Fish dataset, morphological characteristics of various fish species were manually determined:

Table 4: Short description of the eight species.

Species	Head	Body	Scales	Fin

Koi (*Anabas testudineus*)	long, broad head, Mouth: lower jaw is a tiny bit longer Teeth: present on jaw	A slim physique and a sleek conformation, laterally compressed, dark to pale greenish color.	Scales: from 9 to 11 below the lateral line, First lateral line: 15 to 16 up to the, Second lateral line: 10 to 12 up to the corresponding.	Long dorsal, Caudal fin: dark grey. Anal and pectoral fins: pale yellow. Pelvic fin pale: orange color Pectorals and caudal fin: rounded.
Tengra (*Batasio tengana*)	Head: angled and mouth low, a little longer in the upper jaw than the lower jaw, there are four sets of barbels.	Size: 90mm or 3.5, elongated and compressed, A short first dorsal spine. Caudal: forked	Well-defined and beginning well behind the rayed dorsal is the adipose fin.	Little adipose-fin: (12·6–12·8 v. 14·5–33·3%LS) Longer adipose-fin: (14.5–17.5 v. 12.6–12.8% LS) Slender caudal: (5·7–6·2 v. 6·7–11·8%LS)
Taki (*Channa punctata*)	Head: 3.3-3.9 Eyes: Small and situated at the before of the head. Eye diameter: 6.2-8.5	Body: elongated and tubular, No barbels, Maximum length: 30cm	Scale: Above the pelvis, the pectoral has a big, rounded caudal. 75% of the range of the pectoral fin is in the pelvis.	Dorsal: 30-32, Pectoral: 15-17 Pelvic: 2 .5. Anal:19-21
Shing (*Heteropneustes fossilis*)	Head: total length in 6.8-7.2 cm. The Osseous plate is present at the peak and sides of the depressed skull.	Maximum lengths: 30 cm Body: lengthens and contracts. Barbels: four pairs. Two additional respiratory organs (air sacs) Caudal: Rounded.	No scales on the body of this fish.	Pectoral fin rays: 6-7, Pelvic fin rays: 6, Anal fin rays: 64-69, Caudal fin rays: 16-18
Chingri (*Marcrobrachium malcoimsonii*)	Eyes large and sessile. 2 teeth and a biramous uropod on the rostrum. Middorsal blue, red with a red eye stalk base.	Carapace Length: 11–90 mm Body weight: 0.2–54.4 g. Transparent body, large posterior margin of the triangular telson	12.79 mm size Equal-sized teeth on the dorsal side of the rostrum, 10–11 teeth are holes on the dorsal side of the rostrum; the first teeth have 4-5 spines, the second teeth have three spines, then each tooth has two spines.	Two-toothed rostrum has 3–4 spines. Except for the fifth body, pleopod buds are ciliated. The fourth and third pleopods have red chromatophores.
Baim (*Mastacembelus armatus*)	Head is scaled, Nostril located near the front edge of the eye, great cleft, and thick lips.	Max length: 90.0 cm, published weight: 500.00 g. The body is relatively narrow, lengthened, and hardly compressed. There has a large dorsal, and that merges with the caudal fin.	All body is soft and lubricated and wrapped up in the whole body.	Dorsal spines: 33 – 40 Dorsal rays: 67-82 Anal rays: 67-83 and Vertebrae: 87 – 98. 1-3 darker spots on the limp brown body.
Pabda (*Ompok bimaculatus*)	The head is humble, and the snout is rounded. Eyes are little and covered by skin. The upper jaw is shorter than the lower jaw.	Max length: 45.0 cm; it has two twins of barbels. Maxillary barbs extend backward to the base of the anal fin. The body is covered in purple and yellowish markings.	This fish is totally scaleless.	Deeply forked and with a long top lobe, the caudal fin. Caudal stripped with the black marks on the near line behind the operculum. 57 or 58 branching rays around the anal fin.
Punti (*Puntius sophore*)	The upper jaw is a bit longer, and the mouth is tiny and terminal.	Maximum size 25 cm. Barbel is absent, and no rostral barbel. When alive, the body is silvery with a blackish spot	The lateral line has 25–26 scales, and there are moderate-sized scales between the lateral line and the dorsal fin.	Dorsal soft rays: 11 Anal fins including eight rays, Anal fin including five rays, generally with 22–28

		behind the upper scale and one on the caudal peduncle. Body thick and compressed laterally		pore scales on the flank. Eight branches, four unbranched rays, and one smooth and powerful simple ray.

Table 5: Some images used in the dataset.

Some sample fish images from BD Fish Dataset:			
 *Anabas testudineus*	 *Batasio tengana*	 *Channa punctata*	 *Heteropneustes fossilis*
 *Marcrobrachium malcoimsonii*	 *Mastacembelus armatus*	 *Ompok bimaculatus*	 *Puntius sophore*

Value of the Data

i. This dataset contains the only accurate species of fish images based on eight different classes, including **Taki** (*Channa punctata*), **Shing** (*Heteropneustes fossilis*), **Chingri** (*Marcrobrachium malcoimsonii*), **Baim** (*Mastacembelus armatus*), **Pabda** (*Ompok bimaculatus*), and **Punti** (*Puntius sophore*).
ii. Provides data to evaluate the efficacy of an algorithm for several fish species' characteristics, including body size and shape, fin shape and color, eye, medial line, face size and shape, and scale shape and size [51].
iii. This image set may be used to separate fish species to study the freshwater ecosystem and fish activity in fishery resources.
iv. An intricate multiclass picture dataset was created so that the researchers could test several derived features and track how well they performed [51].
v. Data was collected in a regulated environment, some with a constant white background.

Camera Specification Details

i. Some of the image capturing was done using an Asus Zenfone Max Pro M2 smartphone, a 12MP smartphone camera with a total resolution of 4000 x 3000 (Megapixels), and a sensor size of Sony IMX486 with an f/1.8 aperture. The scene setting on phones' cameras was chosen, with the snow scenario subsection selected since it was the option that worked best in the case's odd lighting conditions.
ii. Redmi Note 9: A quad rear camera array arrangement with a 48-megapixel camera with a resolution of 4290 x 3070 pixels and a sensor size with an aggregated f/1.8 aperture was used to take the remaining images. And some were taken using a 2 Megapixel Macro-Shooter with a 987 x 693 resolution. The scene was the setting for the phone's camera, and slow motion was chosen as the sub-category since it proved to be the most effective setting for odd lighting and steady shots

3.1.2 Data Collection

The images of **Pabda** (*Ompok bimaculatus*) and **Tengra** (*Batasio tengana*) were captured from different areas at the **Jamuna River** of **Sirajganj** district. The images of **Punti** (*Puntius sophore*), **Taki** (*Channa punctata*), and **Koi** (*Anabas testudineus*) were taken from the local fish market located near **Enayetpur,** Sirajganj. The remaining species **Shing** (*Heteropneustes fossilis*), **Chingri** (*Marcrobrachium malcoimsonii*), and **Baim** (*Mastacembelus armatus*), were captured from the local fish market of **Kashinathpur Bazar,** Pabna. The image of the BD fish dataset is of different resolutions, usually captured with an image dimension of **4000*2900 pixels** with **72 dpi** resolutions. Some images were captured with a **macro shooter** where the measurement of images is **987*693 pixels** with **72 dpi** resolutions.
We have faced some challenges in the time of our data collection. In the fish market situated near **Enayetpur**, we have yet to find all of our desired fish species. During that time, we searched the species of the fisherman of **Jamuna River**. But we only found some of the species. At that time, we decided to search for the species in the **Sirajganj** fish market. But we have yet to all the species we require. So, we move to the **Kashinathpur** fish market situated in **Pabna** district and find the rest of the species we need there. After collecting all the species of fish. We have faced some **challenges** in **image capturing**. The lighting condition wasn't good enough. At that time, we used two-cell phones flashlight to increase the brightness level so that the image could capture sharply adequately.

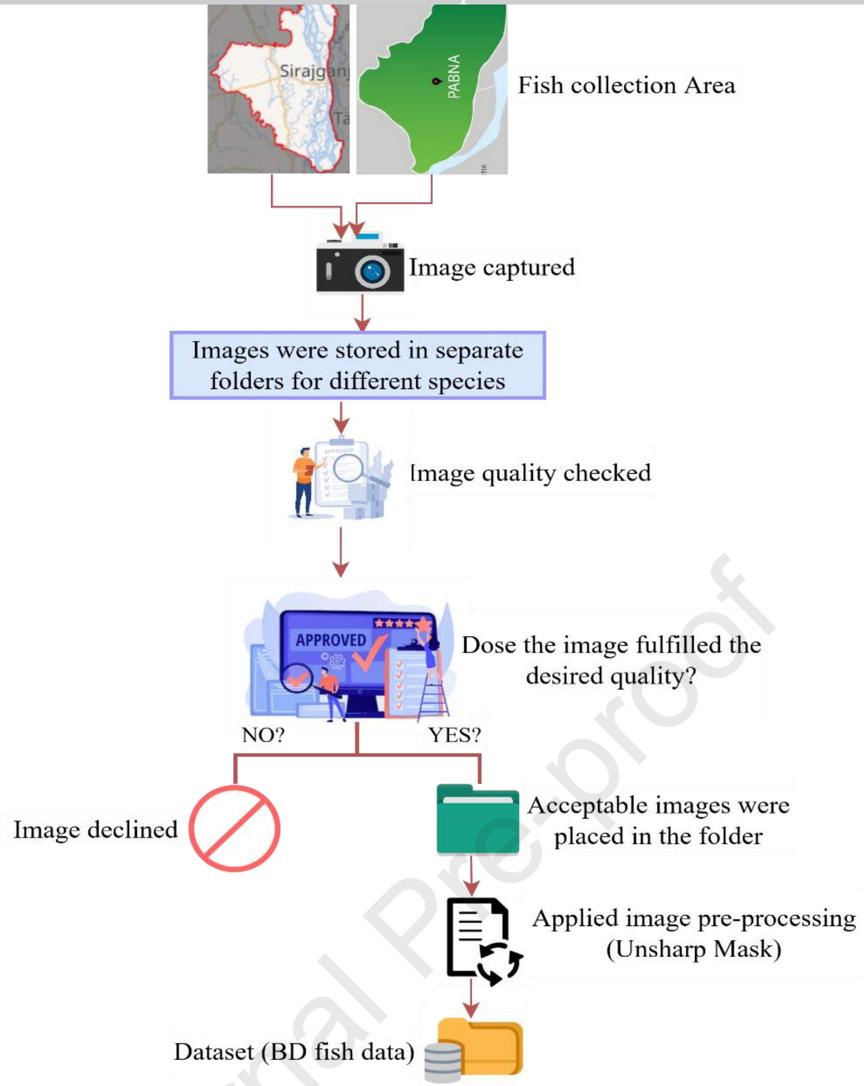


Figure 1: A block diagram of image collection to dataset creation.

3.1.3 Pre-processing of data

Unsharp masking of data

Unsharp mask filter is a versatile sharpening technique that enhances small details by removing low-frequency spatial data from a digital image, resulting in improved overall sharpness [52]. The Unsharp mask filter algorithm involves subtracting a blurred image (Unsharp mask) from the original picture. The Unsharp mask is created by filtering the image with a Gaussian low-pass filter using a kernel mask that corresponds to a two-dimensional Gaussian function *g (x, y)* (Eq. 1).

$$g(x,y)=\frac{1}{\sigma\sqrt{2\pi}}\,e^{-(x^2+y^2)/2\sigma^2} \tag{1}$$

The Gaussian filter's frequency range depends on the kernel mask size, determined by a parameter factor and Standard Deviation in pixels. Increasing the kernel size removes more frequencies from the Unsharp mask picture (Eq. 2).

$$F(x,y)=\frac{c}{2c-1}I(x,y)-\frac{(1-c)}{2c-1}U(x,y) \tag{2}$$

Equation 2 represents the intensity value of a pixel in the filtered image *F(x, y)* with the brightness values of the original and Unsharp mask images represented by *I(x,y)* and *U(x,y)*, respectively. The weights of the original and blurred images in the difference equation are determined by the parameter c. Unsharp masking produces good results in machine learning application

3.2 Working principles (IoT)

The methodology consists of two parts: Convolutional Neural Networks classify fish, and IoT-enabled smart fish containers monitor real-time data. Automated fish processing businesses are enabled by accurately classifying fish using deep learning algorithms for image categorization. Fine-tuned models are employed for classification of eight different species, reducing processing time. Fish types are identified from photos, and multiple sensors in fish containers gather measurements and transmit data for monitoring. Figure 2 displays a block diagram of the system.

This system uses a camera module to scan fish, pre-processes the images, and analyzes them using Raspberry Pi microprocessor. After classification, a servo motor moves the fish into the correct container. Real-time data monitoring is done through an Android app, and a roller transports the fish to the robotic servo arm. The camera module's flow chart is shown in Figure 3. The microcontroller uses a CNN architecture to classify the images, and the servo motor is instructed to place the fish in the right container based on the likelihood of it belonging to a particular species [46].

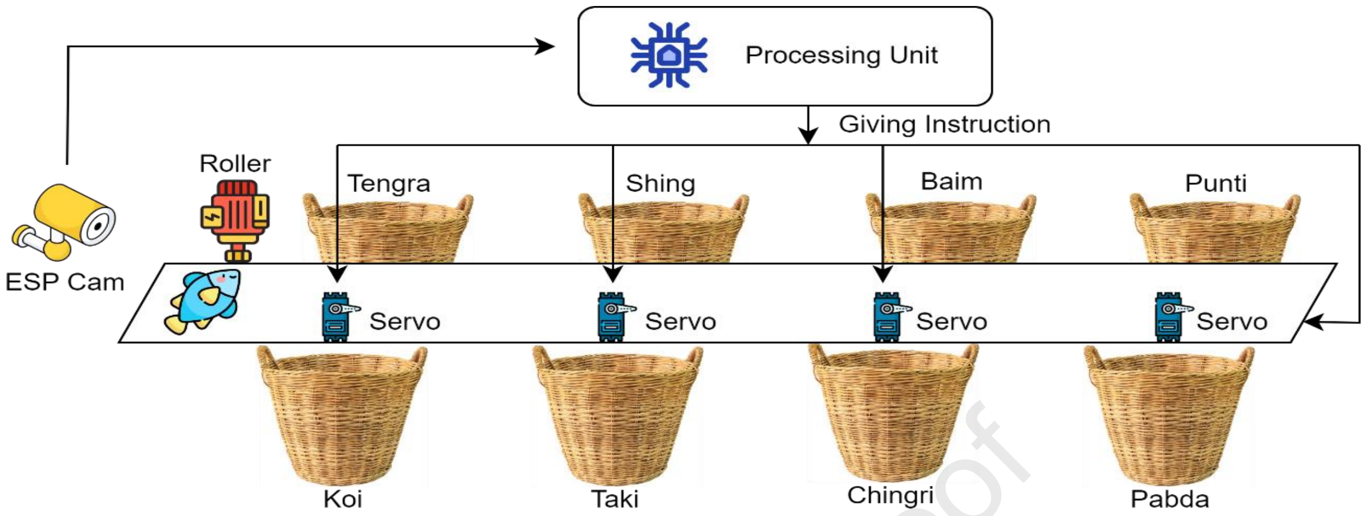


Figure 2: An IoT structure of the proposed smart fish processing system

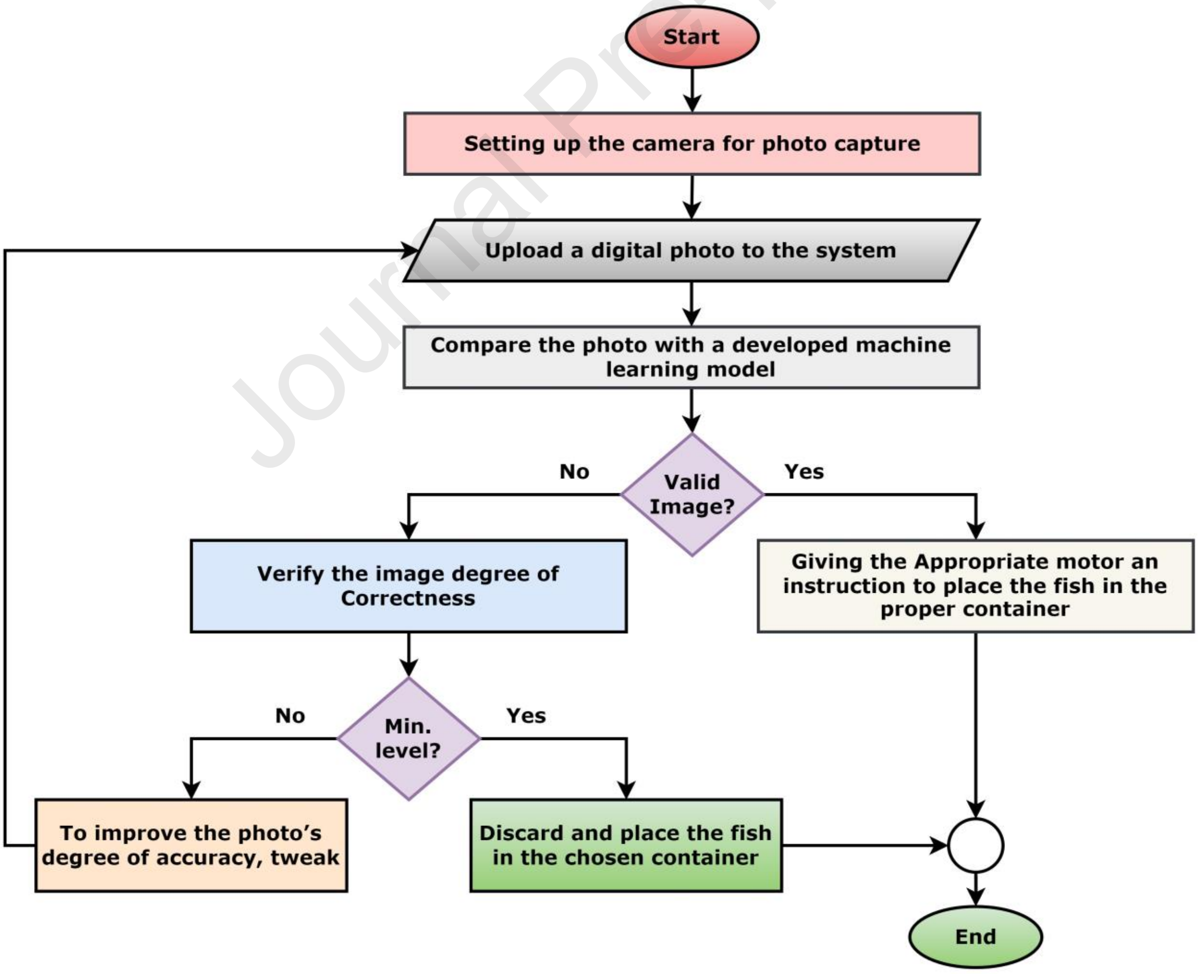


Figure 3: The working principles of Camera Module

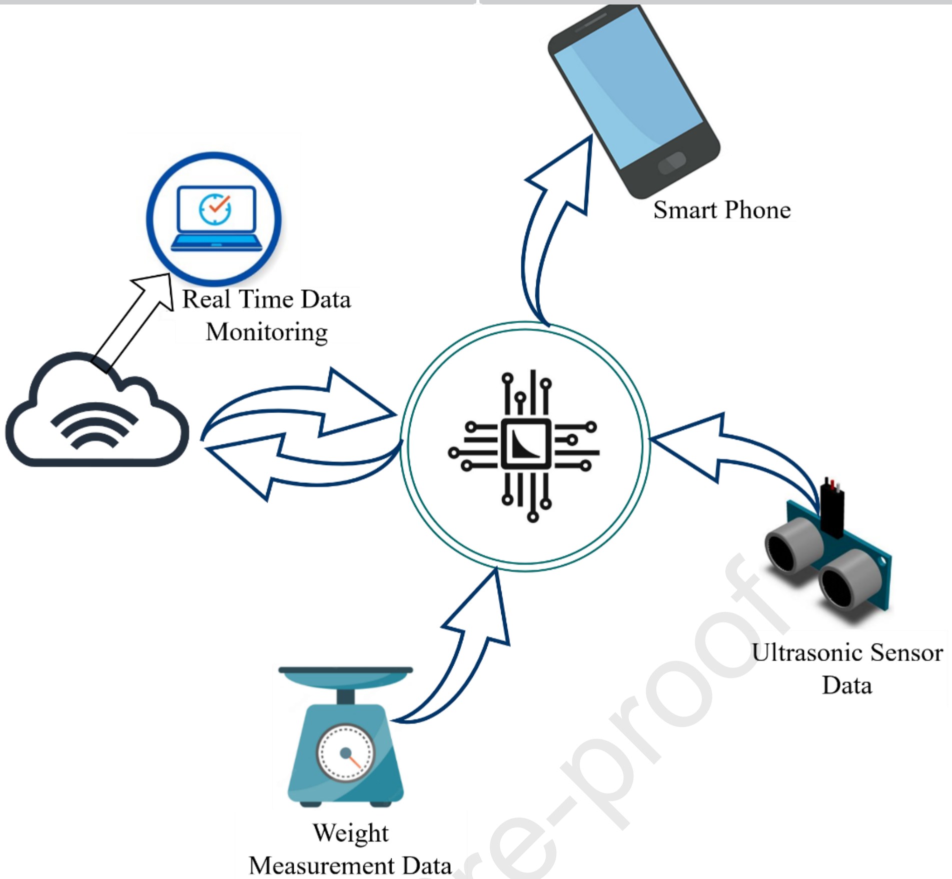


Figure 4: Block diagram and working process of smart fish packing container.

3.2.1 Application and operation of the fish packing container

The design philosophy and creation process of the suggested architecture for the smart fish packing container is described in this part.

The schematic diagram of the fish packing container we designed is shown in Figure 4. In this diagram, the general technique is programmed on a microcontroller called the "ESP8266" node MCU. To evaluate the empty level of the fish packing container, an ultrasonic sensor is connected to the microcontroller. In our created model, the ultrasonic sensor is located at the top. The empty level of the fish packaging container is determined by the transmission and reception of ultrasound. The microcontroller receives the computed value. At the base of the surface is also a load-measuring sensor. This sensor is in charge of figuring out the fish's weight in kilograms. The load-measuring sensor is aware of the timing of the approaching fish load. The value of the load increases as the number of fish in the fish packaging container increases over time. The microcontroller is then given the modified value. The Bluetooth-based Android application receives the results from the load measurement sensor and ultrasonic sensor. Using an android application, users can quickly see the fish's weight and the container's current empty layer. The corresponding values will be sent to the cloud server if an internet connection is available, and users will be able to view the data in real-time via an android application.

A sequence of the ultrasonic sensor's operation is shown in Figure 6. The ultrasonic sensor was first initialized by the system in this picture. The sensor transmits and receives ultrasound to determine the fish packing container's empty level. Additionally, the microcontroller gets the matching measured data from the ultrasonic sensor. The measured data is sent to the Android application when the Bluetooth connection status is positive. Checking the internet connection happens as a negative reaction to the Bluetooth connection that is currently active. The system will communicate the data via a cloud server to an Android mobile application if Wi-Fi access is available. A successful effort to obtain information from the cloud connection and Bluetooth results in the ability to view the information on a Mobile application.

We have established a threshold of the measure in order to verify the authenticity of an unused level. The solution will halt the relevant procedure once the fish reaches the threshold value and notify the user to pick up the package using application software. No information will be sent if the internet connection is down. To determine the weight of the fish container, a load-measuring sensor has been installed. A flowchart of the weight measurement sensor's operation is shown in Figure 7.

In the illustration, the system initially set the sensor up to read data. The weight measurement device computes the weight in kilograms and transmits the information to the microcontroller. The weight is checked by the microcontroller to see whether it exceeds a certain threshold. Bluetooth connectivity is discovered when the adverse reaction is greater than the maximum level. The system will send the information to the Mobile application if a Bluetooth connection exists. The system will proceed and verify internet connections if a Bluetooth connection is not feasible.

The system will stop the measurements, and no results will be delivered if there is no internet access at the time. The technology enables the transmission of a message to the Mobile application, instructing the user to swap out the pack in the fish packing container.

The system will transfer the weight that was calculated to the server if the system software has an internet connection. A successful attempt to obtain information from a remote server and a Bluetooth connection allowed for real-time monitoring of the data using an android application [46].

3.2.2 Functioning of Ultrasonic Sensor

The ultrasonic sensor emits 40,000 Hz ultrasound, which bounces back to the sensor if there is an obstruction [46]. The Trig needs to be set to a peak value for 10 microseconds to produce the ultrasound, which is picked up by the echo pin and output in microseconds. A schematic diagram of the ultrasonic device's operation is shown in Figure 5. [53].

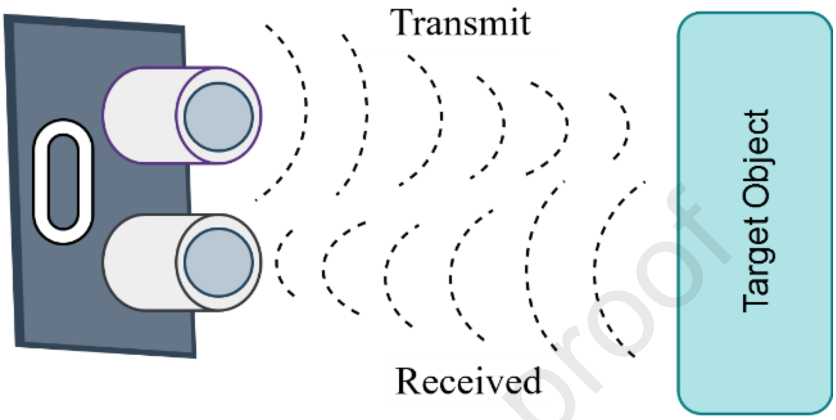


Figure 5: A block diagram of Ultrasonic sensor working principles.

The methodology used by the ultrasonic sensor, which is represented in Eq. (3), is based on an equation of sound reflection.

$$s = \frac{v * t}{2} \tag{3}$$

Where,

v = the velocity of sound is 340 meters per second.

t = the amount of time needed to send and receive.

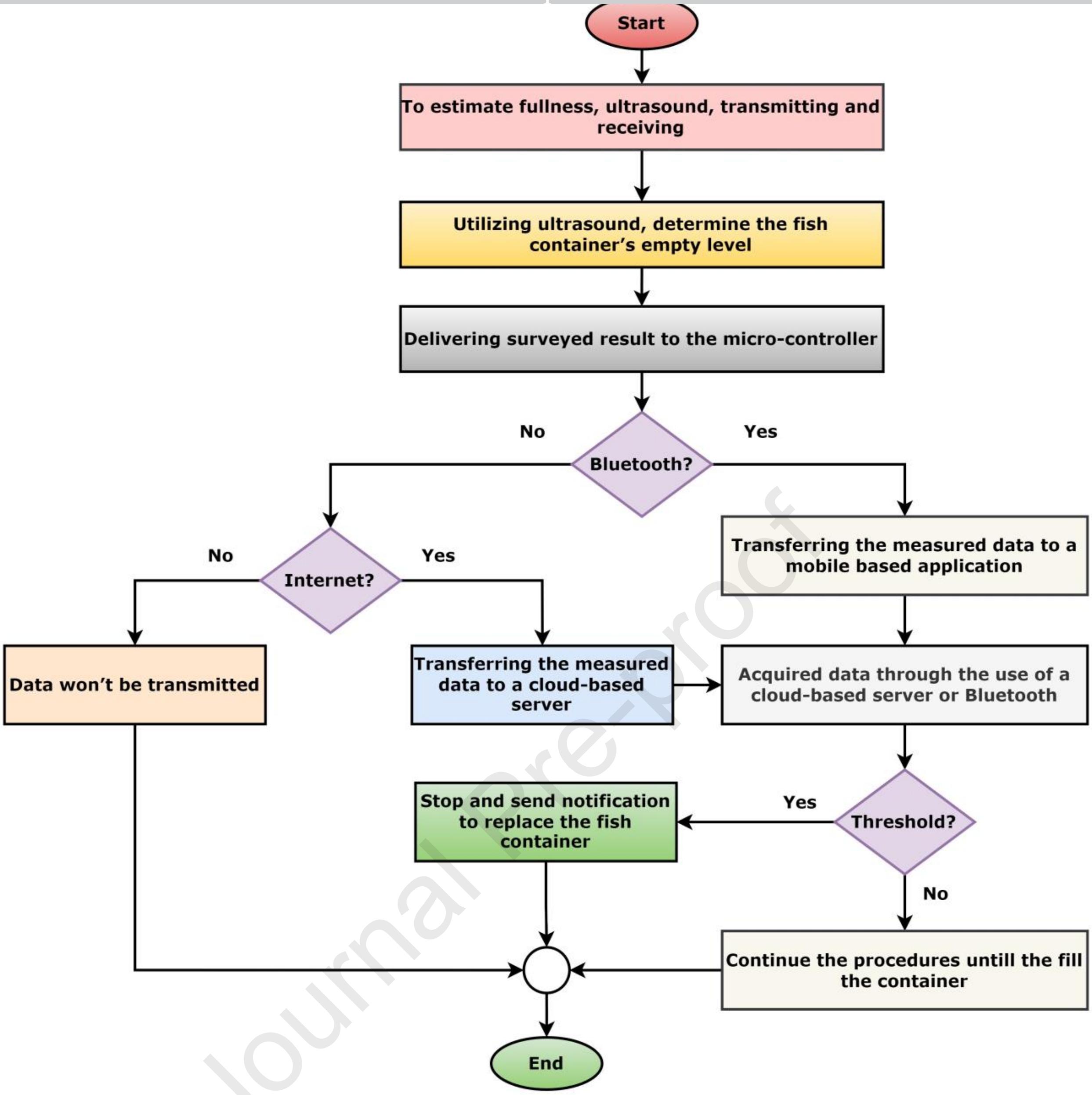


Figure 6: Diagram of the working principle of the Ultrasonic Sensor

3.2.3 Functioning of a weight measurement sensor

A weight measurement sensor is used to calculate the weight of the object using a formula that transforms the output voltage measured in mV/V. The user selects the preferred unit, such as kilograms or pounds. The load cell has a standard voltage level of 1.01.5mV/V, which provides a 5 kg capacity. The equation to translate the output voltage into weight is shown by Eq. (4), [54].

$$Measured\ force = A * measured\ voltage + B \tag{4}$$

Where,

$Measured\ voltage = 1.0 \pm 1.5mV/V$

So, the load cell's constant maximum value is 5

B=offset

Because the precise amount changes from the sensor to the sensor, it is crucial to compute the offset of every sensor. The offset value typically ranges from:

$$B = 0 - 5 * measured\ voltage \tag{5}$$

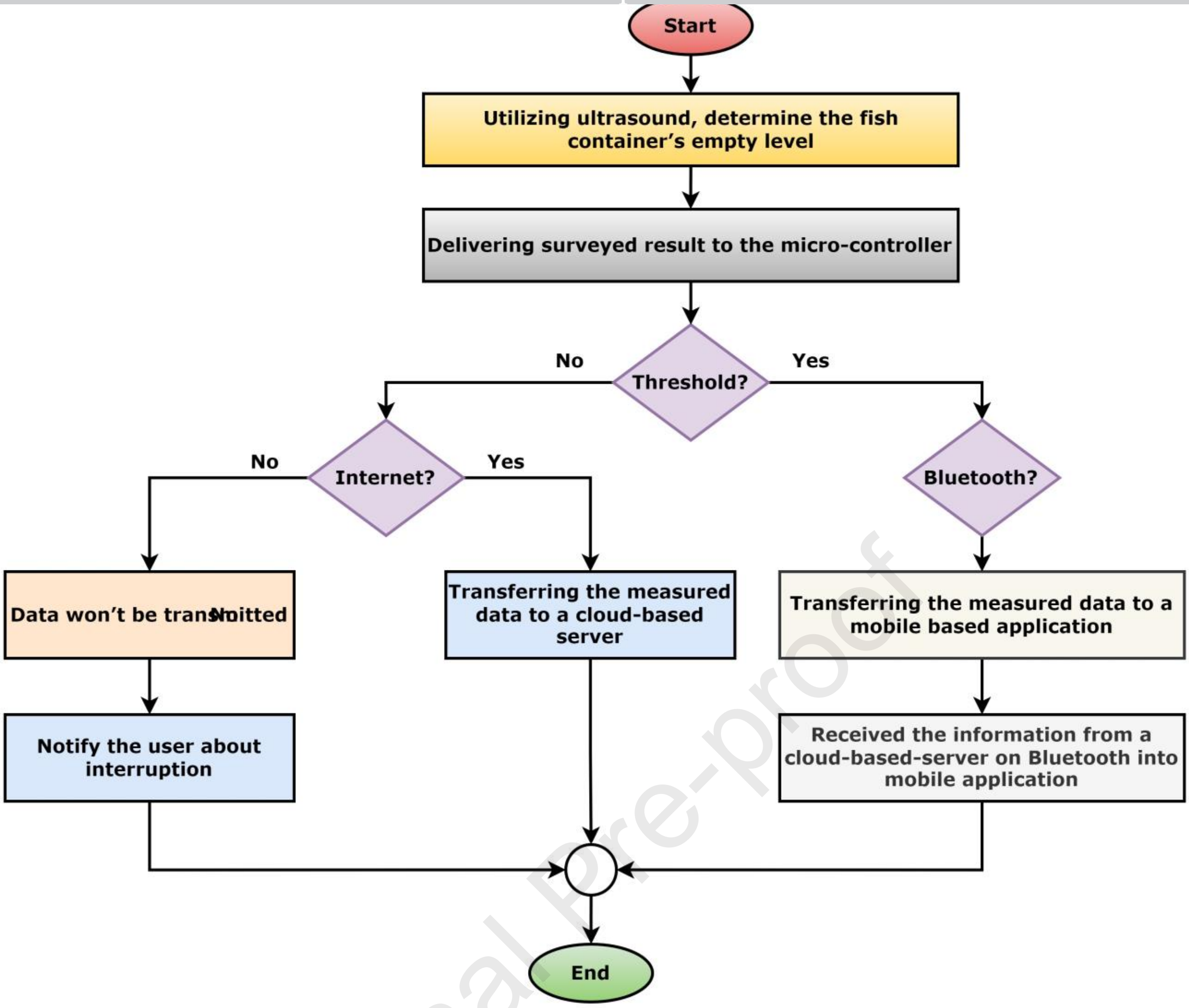


Figure 7: Diagram of the working principle of the weight measurement sensor

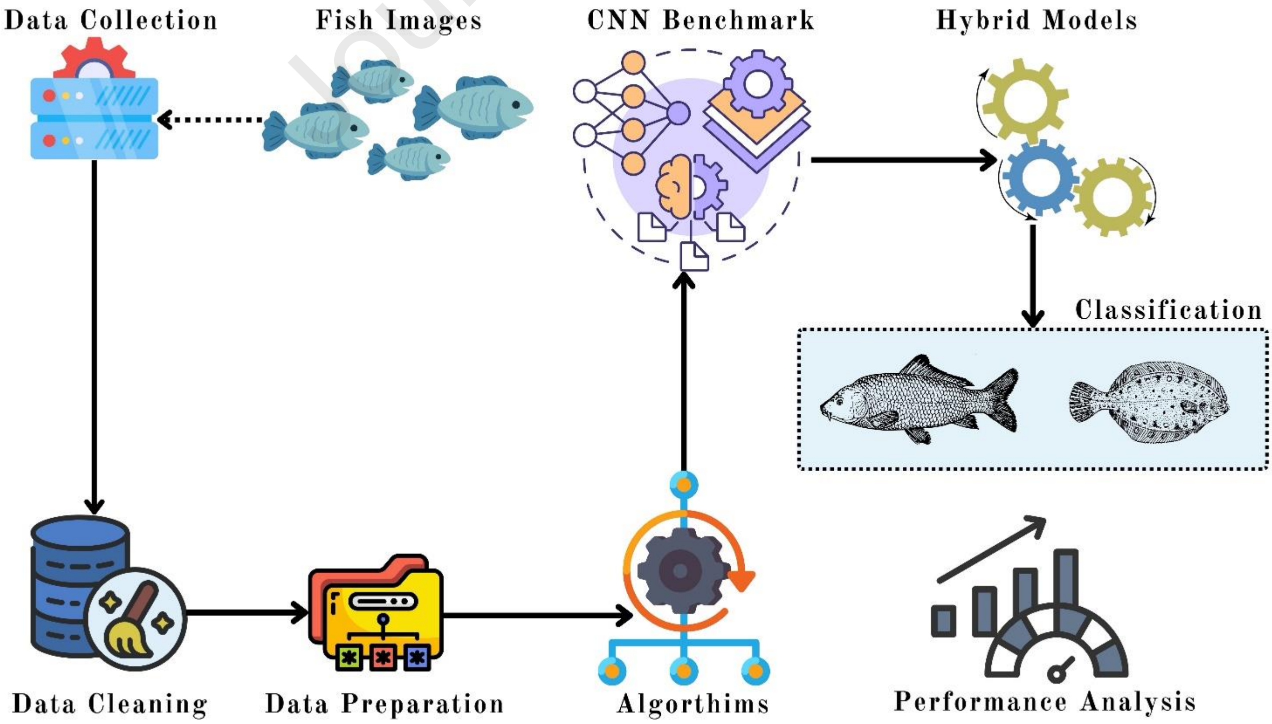


Figure 8: Block Diagram of Fish Classification process

3.3 Deep Learning

The data calculation process is divided into three sections: classification of fish using Convolutional Neural Network (CNN), measurement of the fish packing container's empty level using an ultrasonic sensor, and determination of the weight of the fish package using a weight measurement device.

3.3.1 Fish Classification process (CNN)

Xception: A depth-first convolutional neural network structure called Xception uses Depth wise Separable Convolutions. Francois Chollet, an employee of Google, Inc., introduced this network. The "extreme" variant of an Inception module is known as Xception [55].

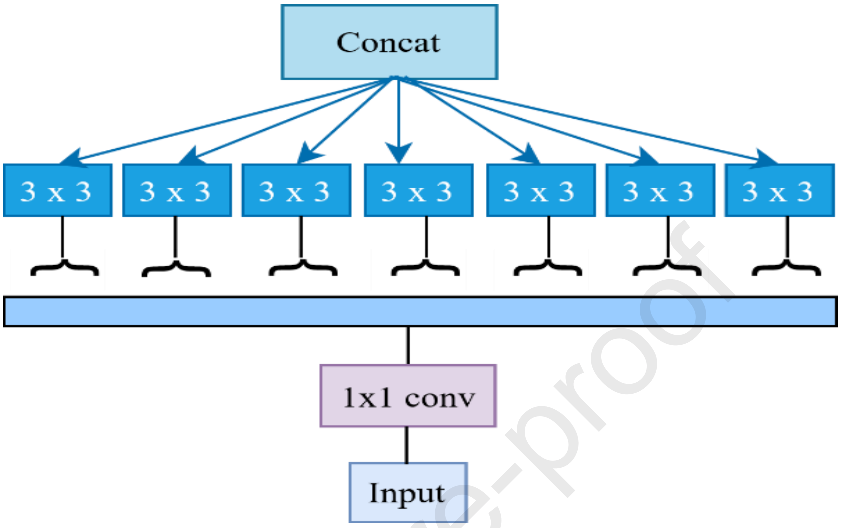


Figure 9: Xception architecture.

InceptionV3: The InceptionV3 convolutional neural network design uses Factorized 7 x 7 convolutions, Label Smoothing, and the inclusion of an auxiliary classifier to transport label information lower down the network, among other advances (Additionally, layers in the side head are batch-normalized) [56].

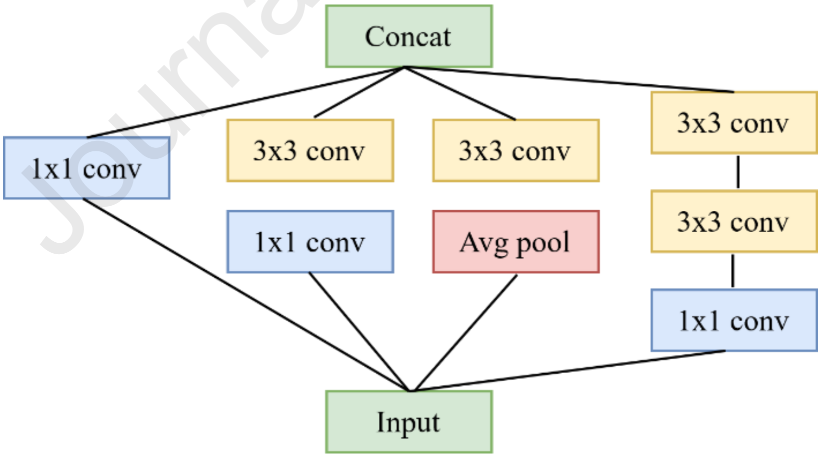


Figure 10: InseptionV3 architecture.

DenseNet121: 120 Convolutions with 4 AvgPool make up DenseNet121. Deeper layers can use features that were retrieved earlier because all levels, even those in the similar dense block, including transition layers, distribute their weights across numerous inputs [57].

$$X_l = H_l([X_0, X_1, \ldots, X_l\text{-}1]); \qquad (6)$$

Where the network consists of L layers, H_l (.) a non-linear metamorphosis X_l as the output of l^{th} layer and [X_0, X_1, …, X_l-1] represents the round-up of the feature-maps created in layer 0,…,$l-1$.

DenseNet169: Despite having 169 levels of depth, it has few parameters when compared to other architectures and structures effectively solve the disappearing convolution layers [58].

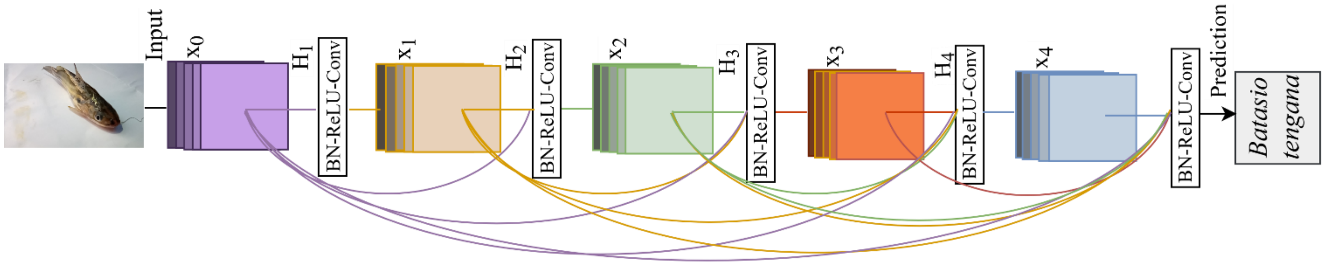


Figure 11: Dense block architecture.

DenseNet201: It has 201 layers and is called DenseNet201, a convolutional neural network. From the ImageNet collection, one may load a network that has been pre-trained with more than a million photos. The pre-trained network is able to categorize images into 1000 different item categories, including mice, pencils, and several animals [57].

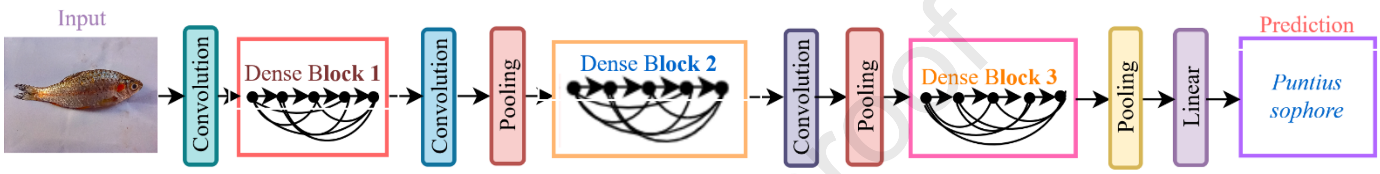


Figure 12: Three dense blocks in a deep DenseNet.

Inception-ResNet-v2: A convolutional neural structure called Inception-ResNet-v2 expands on the Inception family of architectures while incorporating residual connections (the Inception architecture's filter concatenation stage is replaced). A pricier hybrid Inception variant with much better recognition performance is Inception-ResNet-v2 [59].

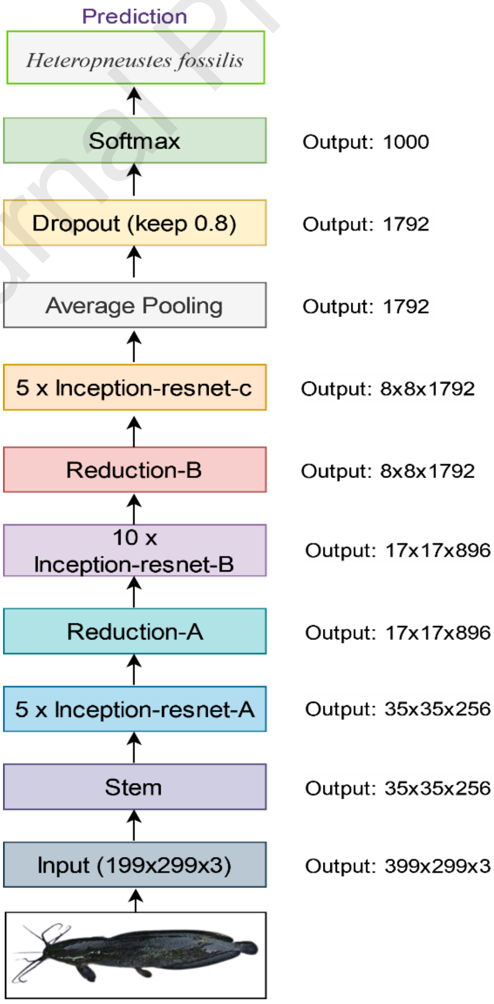


Figure 13: Inception-ResNet-v2

ResNet152V2: In figure 14, The information's quickest paths to spread are indicated by the grey arrows, correlating the additive element ‘’x_1″ from Eq. (7) (forward propagation) as well as the "1" additive element in Eq. (8) (backward propagation) [60].

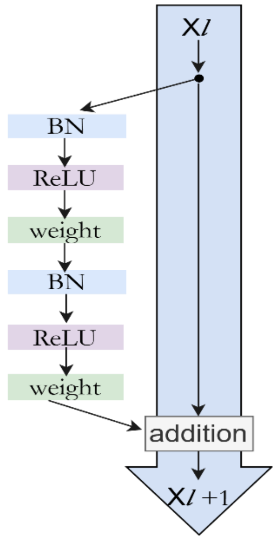


Figure 14: ResNet152V2

To create an identity mapping $f(y_l) = y_l$ where the weighted layers activation functions (ReLU and BN) are "pre-activated" as opposed to the "post-activated" activation that is typically the case [60].

$$x_L - xl + \sum_{i=l}^{L-1} F(x_i,\ W_i), \tag{7}$$

The property x_L, like any more profound element L, can be expressed as the feature x_l of every shallower component l with a residual function in the format of $\sum_{i-l}^{L-1} F$, showing that the model lies in between units L and l in a residual fashion. Any deep unit L has a feature called $x_L = x_0 + \sum_{i-0}^{L-1} F(x_1,\ W_i)$, which is the output of all previous residual functions added together [60].

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}(1 + \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_i, W_i) \tag{8}$$

The gradient $\frac{\partial \varepsilon}{\partial x_l}$ can also be divided into two additive parts, according to Eq. (8): an $\frac{\partial \varepsilon}{\partial x_L}$ expression that transmits information directly except taking into consideration any weight layers. Additionally, another term of $\frac{\partial \varepsilon}{\partial x_L}(\frac{\partial}{\partial x_l}\sum_{i-l}^{L-1} F)$ spreads throughout the weight layers. Information is directly transported back toward any shallower unit l according to the additive component of $\frac{\partial \varepsilon}{\partial x_L}$ [60].

3.3.2 The ConvLSTM Model

In several earlier research, the LSTM as a unique RNN structure has been shown to be reliable and effective for simulating long-range relationships [61] [62] [63] [64]. If the input gate is engaged, each time a new input is received, its information will be added to the cell. Furthermore, if the forget gate f_t is enabled, the previous cell state c_{t-1} can be "lost" throughout this procedure. The output gate also regulates whether the most recent cell output c_t will be transmitted to the ultimate state ht. It is possible to think of FC-LSTM as a multivariate LSTM in which the input, cell output, and states are all 1D vectors [65]. Below are the important equations where "o" stands for the Hadamard product:

$$\boldsymbol{i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i)} \tag{9}$$

$$\boldsymbol{f_t = \sigma\big(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f\big)} \tag{10}$$

$$\boldsymbol{c_t = f_t \circ\ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)} \tag{11}$$

$$\boldsymbol{O_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o)} \tag{12}$$

$$\boldsymbol{h_t = O_t \circ tanh(c_t)} \tag{13}$$

The FC-LSTM layer has limitations in handling spatial information and has redundancy. X Shi et al. proposed an extension of FC-LSTM called Convolutional LSTM (ConvLSTM) that uses convolutional structures in both input-to-state and state-to-state transitions to address this issue[65].

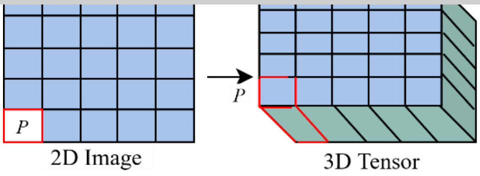


Figure 15: Transforming 2D image into 3D tensor

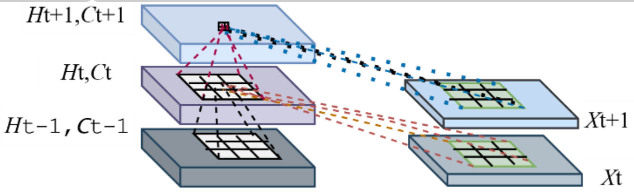


Figure 16: Inner structure of ConvLSTM

The weakness of FC-main LSTM is the use of complete connections in input-to-state and state-to-state transitions when no spatial information is recorded. The inputs X_1, ..., X_t, cell outputs C_1, ..., C_t, hidden states H_1, ..., H_t, and gates it, f_t, ... are 3D tensors who's last two dimensions are spatial dimensions in ConvLSTM, which helps solve this problem. Using convolution operator in state-to-state and input-to-state phases makes it easy to predict the future state of each cell in the grid by utilizing the inputs and previous states of its immediate neighbors (Figure 16) [58]. The equations for ConvLSTM are shown below, where '*' stands for the convolution function and '', as previously, for the Hadamard product [65].

3.3.3 Description of Proposed Model

Pre-trained DenseNet201: The model used a DenseNet201 architecture that was pre-trained on the ImageNet dataset for automatic feature extraction. This allowed the model to leverage the feature extraction capabilities of the DenseNet201 architecture, which has shown to perform well on image classification tasks. Three 1-D ConvLSTM Layers: The proposed ConvLSTM model has a total of three ConvLSTM layers, all of which are 1-D. The first three layers are ConvLSTM layers, which combine the properties of Convolutional Neural Networks (CNNs) and LSTMs. These layers can capture both spatial and temporal dependencies in the input data.

Neurons: In the first ConvLSTM layer, there are 128 neurons, in the next layer there are 256 neurons, and in the last layer, there are 128 neurons. This is a typical pattern for ConvLSTM models, where the number of neurons is gradually increased and then decreased.

Kernel size: In all three ConvLSTM layers, the kernel size is 2. This specifies the size of the sliding window that is used for convolution operations.

Flattened layer: After the three ConvLSTM layers, a flattened layer was applied, which flattens any input matrix into a one-dimensional vector. This prepares the data for the fully connected layers that follow.

Three Dense Layers: After the flattened layer, three dense layers were added with 1024, 512, and 512 neurons, respectively. These layers perform feature extraction on the output of the ConvLSTM layers.

ELU Activation Function: The ELU (Exponential Linear Unit) activation function was used in all three dense layers. This activation function has been shown to improve training performance and can prevent the "vanishing gradient" problem that can occur with other activation functions.

Output Layer: The output layer is a Dense layer with eight neurons, one for each class in the dataset. The SoftMax function was used for classification purposes, which converts the output of the model into a probability distribution over the eight classes.

Input Shape: The input shape of the model was (256, 256, 3), which means that the model accepts 256x256 pixel images with three color channels (RGB).

Input Shape of the First ConvLSTM Layer: The input shape of the first ConvLSTM layer of the proposed model was (8, 7, 128). This means that the first layer processed inputs with a temporal sequence length of 8 and a spatial size of 7x128.

Activation Function Used in the ConvLSTM Layers: The activation function used in the ConvLSTM layers is ReLU (Rectified Linear Unit), which is a commonly used activation function in deep learning.

Optimizer: The optimizer used during training was RMSProp (Root Mean Square Propagation), which is an optimization algorithm that adapts the learning rate based on the gradient history.

Loss Function: The loss function used during training was categorical_crossentropy, which is commonly used for multi-class classification tasks.

Batch Size: The batch size used during training was 32, which means that the model was trained on batches of 32 images at a time.

Training: Training continued until 10 consecutive no improvements in validation loss performance. This is a common technique for preventing overfitting and ensuring that the model has converted to a stable solution.

Learning Rate: The learning rate used during training was 0.0001, which specifies how quickly the weights of the model are updated during training.

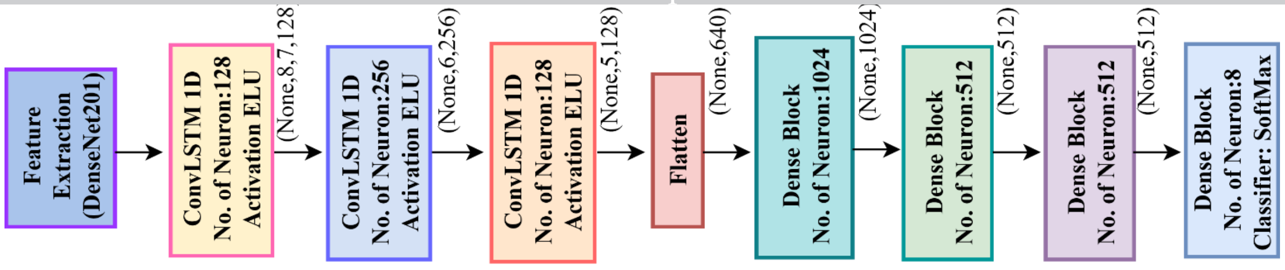


Figure 17: Our Proposed (DenseNet201 + ConvLSTM) Model.

Exponential Linear Units (ELUs): ELU is an activation function developed for neural networks that can reduce computing complexity while bringing mean unit activations closer to zero. It accelerates learning by reducing bias shift effect and guarantees a deactivation state that is resistant to noise. ELUs exhaust to a negative value with lower inputs, reducing forward propagated variance and information [66].

The exponential linear unit (ELU) with $0 < \alpha$ is

$$f(x) = \begin{cases} x & if\ x > 0 \\ \alpha\,(\exp(x) - 1) & if\ x \le 0 \end{cases}, \quad f'(x) = \begin{cases} 1 & if\ x > 0 \\ f(x) + \alpha & if\ x \le 0 \end{cases}. \tag{14}$$

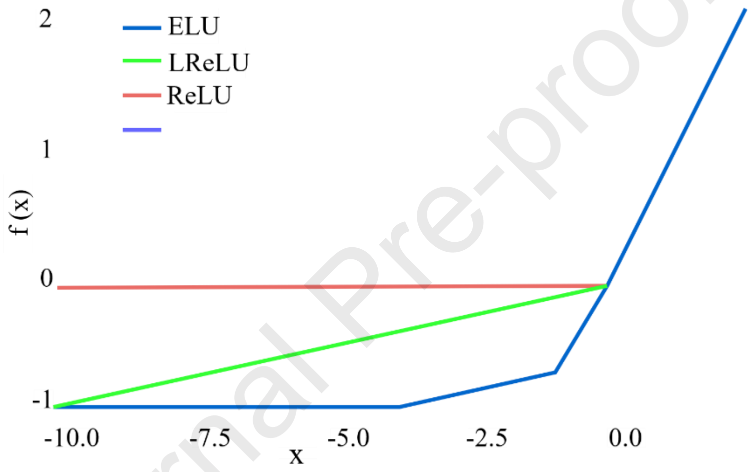


Figure 18: Exponential Linear Units (ELUs).

SoftMax: The output of a preceding layer is converted into a vector of probabilities using the SoftMax output function. The categorization of many classes frequently uses it [67].
If we have a weighted vector w and an input vector x, we have:

$$p(y = j \mid x) = \frac{e^{x^T w_j}}{\sum_{k=1}^{K} e^{x^T wk}} \tag{15}$$

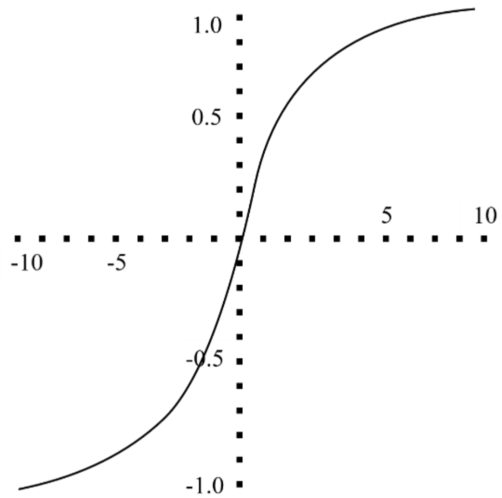


Figure 19: SoftMax function.

4. Result and Discussion

This subsection is classified into two interconnected parts. The first section describes the experimented data analysis with the deep learning models and performance analysis. The second section provides the data analysis with the IoT based solution.

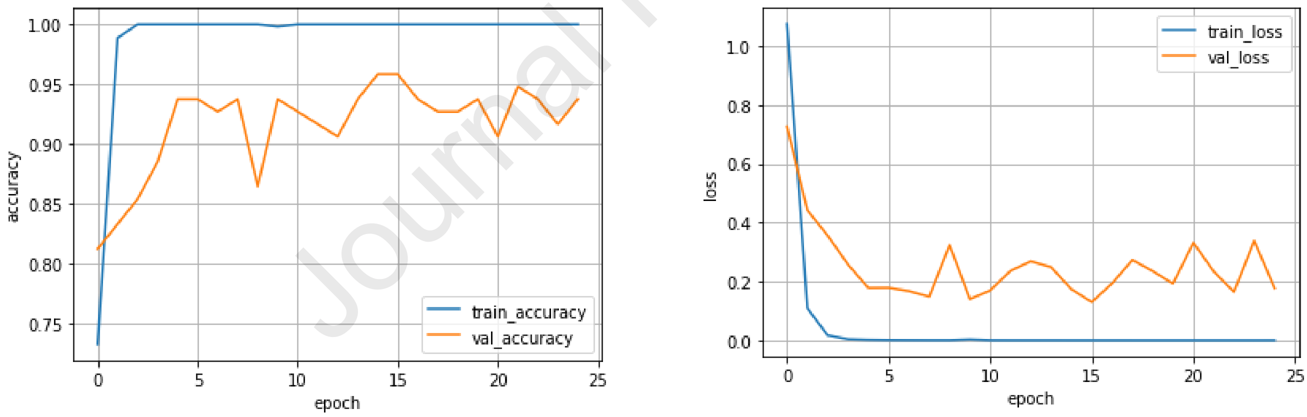
4.1 Data Analysis with Deep Learning

This portion discusses findings of our exhibited system, including impact of object detection, accuracy rate, training curve, comparison between CNN models with our proposed combined CNN and RNN model. It also presents experimental results from weight measurement and ultrasonic sensors, and images of the constructed model with a detailed analysis.

Table 6: Performance comparison in term of aaccuracy

Model Name	Before Unsharp Mask		After Unsharp Mask	
Architecture	*Validation (%)*	*Test (%)*	*Validation (%)*	*Test (%)*
InceptionV3	93.75	94.00	93.75	94.00
Xception	96.88	92.00	95.83	96.00
DenseNet121	96.88	96.00	96.88	93.00
DenseNet169	96.88	96.00	94.79	91.00
DenseNet201	94.79	96.00	98.96	93.00
InceptionResNetV2	96.88	95.00	92.71	91.00
ResNet152V2	91.67	89.00	92.71	91.00
Proposed	-	-	93.75	97.00

This experiment compared pre-trained CNN models and a proposed combined CNN and RNN model in predicting fish species using transfer learning. The models were trained with 70% data, validated with 15% data, and tested with 15% data. The models were evaluated with a confusion matrix, learning curves, and classification accuracy. Dense architecture and Xception had the best performance, with DenseNet201 chosen for the proposed model. The proposed model achieved an accuracy of 97%, outperforming all other models.



(a) (b)

Figure 20: **(a)** Training and Validation Accuracy for the proposed hybrid model. **(b)**Training and Validation Loss for the proposed hybrid model.

We have used a ten count-no-improvement process in our machine's training process; if the validation accuracy is not increased, then stop the training process.

Table 7: Output of Precision

Model Name		Class							
		Anabas testudineus	*Batasio tengana*	*Channa punctata*	*Heteropneustes fossilis*	*Marcrobrachium malcoimsonii*	*Mstacembelus armatus*	*Ompok bimaculatus*	*Puntius sophore*

Before Unsharp Mask	Xception	1.00	0.94	0.87	0.75	1.00	0.93	0.94	1.00
	InceptionV3	0.93	0.94	0.88	0.94	0.94	1.00	0.88	1.00
	DenseNet121	0.93	1.00	0.93	0.88	1.00	1.00	1.00	0.93
	DenseNet169	1.00	0.88	0.87	0.94	1.00	1.00	1.00	1.00
	DenseNet201	1.00	0.83	0.88	1.00	1.00	1.00	1.00	1.00
	Inception ResNetV2	0.88	0.88	0.93	1.00	1.00	1.00	0.94	1.00
	ResNet152V2	1.00	0.94	0.81	0.74	0.92	0.93	0.88	1.00
After Unsharp Mask	Xception	1.00	0.88	0.82	0.82	1.00	1.00	1.00	1.00
	InceptionV3	1.00	0.94	0.88	0.94	0.94	1.00	0.88	1.00
	DenseNet121	0.93	1.00	0.88	0.82	1.00	0.94	0.93	1.00
	DenseNet169	1.00	0.88	0.81	0.71	1.00	1.00	1.00	1.00
	DenseNet201	1.00	1.00	0.79	0.83	1.00	1.00	1.00	1.00
	Inception ResNetV2	0.88	0.83	0.87	1.00	1.00	1.00	0.79	1.00
	ResNet152V2	1.00	0.88	0.88	0.71	1.00	0.94	1.00	1.00
	Proposed	1.00	0.94	0.94	0.93	0.93	1.00	1.00	1.00

Table 8: Output of Recall

	Model Name	**Class**							
		Anabas testudineus	*Batasio tengana*	*Channa punctata*	*Heteropneustes fossilis*	*Marcrobrachium malcoimsonii*	*Mstacembelus armatus*	*Ompok bimaculatus*	*Puntius sophore*
Before Unsharp Mask	Xception	0.93	1.00	1.00	1.00	0.87	0.93	1.00	0.73
	InceptionV3	0.93	0.93	0.93	1.00	1.00	1.00	1.00	0.67
	DenseNet121	0.93	1.00	0.87	0.93	1.00	1.00	1.00	0.93
	DenseNet169	1.00	1.00	0.87	1.00	0.93	1.00	1.00	0.87
	DenseNet201	1.00	1.00	0.93	1.00	0.87	1.00	1.00	0.87
	Inception ResNetV2	0.93	1.00	0.87	1.00	1.00	1.00	1.00	0.80
	ResNet152V2	0.93	1.00	0.87	0.93	0.80	0.93	0.93	0.73
After Unsharp Mask	Xception	1.00	1.00	0.93	1.00	1.00	1.00	1.00	0.73
	InceptionV3	0.93	1.00	0.93	1.00	1.00	1.00	1.00	0.67
	DenseNet121	0.93	1.00	1.00	0.93	0.87	1.00	0.93	0.80
	DenseNet169	0.87	1.00	0.87	1.00	0.80	1.00	1.00	0.73
	DenseNet201	1.00	1.00	1.00	1.00	0.73	1.00	1.00	0.73
	Inception ResNetV2	1.00	1.00	0.87	1.00	1.00	1.00	1.00	0.40
	ResNet152V2	0.87	1.00	1.00	1.00	0.80	1.00	0.93	0.80
	Proposed	1.00	1.00	1.00	0.93	0.87	1.00	0.93	1.00

Table 9: Output of F1-Score

	Model Name	**Class**							
		Anabas testudineus	*Batasio tengana*	*Channa punctata*	*Heteropneustes fossilis*	*Marcrobrachium malcoimsonii*	*Mstacembelus armatus*	*Ompok bimaculatus*	*Puntius sophore*
Before Unsharp M	Xception	0.97	0.97	0.87	0.86	0.93	0.93	0.97	0.85
	InceptionV3	0.93	0.97	0.90	0.97	1.00	1.00	0.94	0.80
	DenseNet121	0.93	1.00	0.90	0.90	1.00	1.00	1.00	0.93
	DenseNet169	1.00	0.94	0.87	0.97	0.97	1.00	1.00	0.93
	DenseNet201	1.00	0.91	0.90	1.00	0.93	1.00	1.00	0.93

	Inception ResNetV2	0.90	0.94	0.90	1.00	1.00	1.00	0.97	0.89
	ResNet152V2	0.97	0.97	0.84	0.82	0.86	0.93	0.90	0.85
After Unsharp Mask	Xception	1.00	0.94	0.87	1.00	1.00	1.00	1.00	0.85
	InceptionV3	0.97	0.97	0.90	0.97	0.97	1.00	0.94	0.80
	DenseNet121	0.93	1.00	0.94	0.87	0.93	0.97	0.93	0.89
	DenseNet169	0.93	0.94	0.84	0.83	0.89	1.00	1.00	0.85
	DenseNet201	1.00	0.97	0.88	0.91	0.85	1.00	1.00	0.85
	Inception ResNetV2	0.94	0.91	0.87	1.00	1.00	1.00	0.88	0.57
	ResNet152V2	0.93	0.94	0.94	0.83	0.89	0.97	0.97	0.89
	Proposed	1.00	0.97	0.97	0.93	0.90	1.00	0.97	1.00

Comparison of results between CNN-based models and a proposed model with confusion matrix analysis and accuracy is presented using Table 6, Table 7, Table 8, and Table 9. The Xception architecture yielded 92% accuracy before the unsharp masking training process, with some misclassifications. After the unsharp masking process, accuracy improved to 96%. The proposed model, which combines DenseNet201 with ConvLSTM, achieved an accuracy of 97% and demonstrated superior performance in most classes.

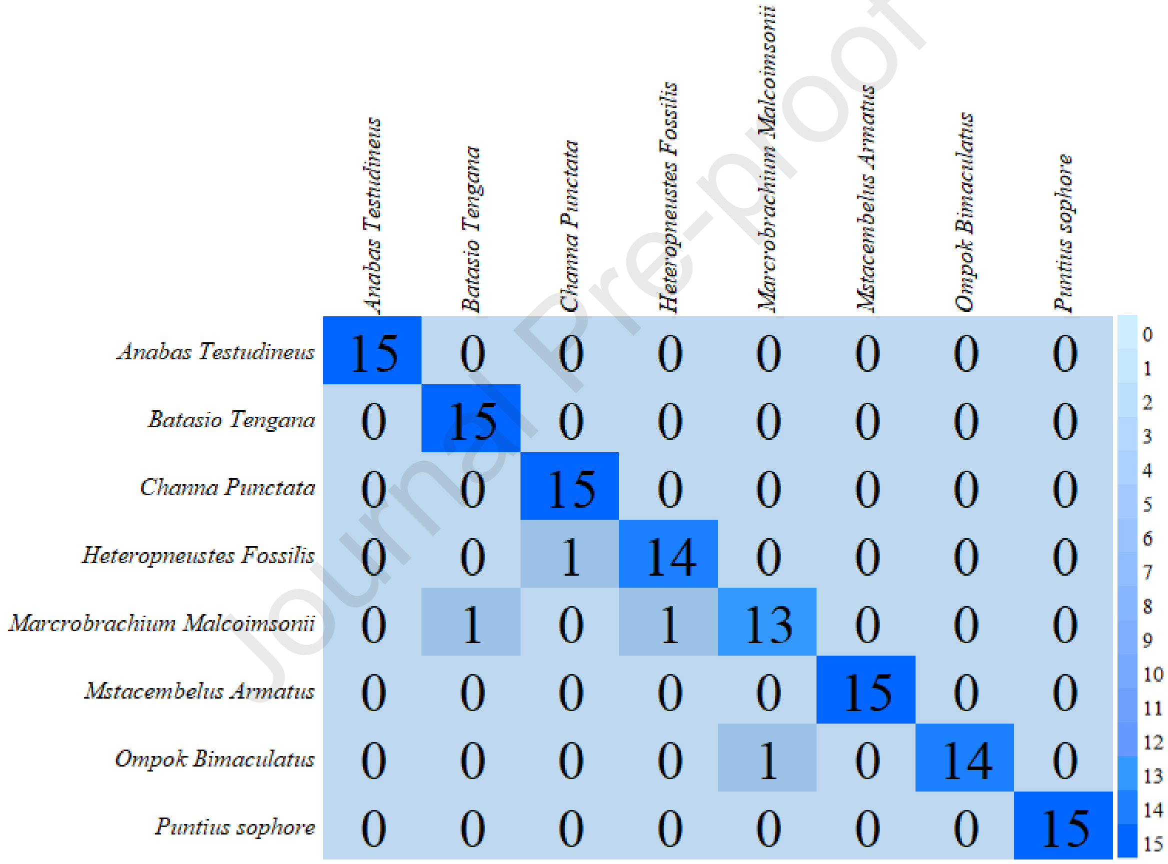


Figure 21: Confusion Matrix of our proposed Hybrid model.

4.1.1 Outcome of ResNet152V2, InceptionResNetV2 and InceptionV3 Architecture

In ResNet152V2 architecture, before the unsharp mask process, the accuracy was 89% with 49 epochs completed. From the confusion matrix, we found some misclassified images in different classes. After applying the Unsharp mask process, the accuracy improved, and all images in some classes were correctly classified. In InceptionV3 architecture, before the Unsharp mask process, the accuracy was 94% with 29 epochs completed.

For each of the five fish species, all 15 test images are correctly classified in the InceptionV3, InceptionResNetV2, and ResNet152V2 architectures. In the *Anabas testudineus* class, 14 images are correctly classified, and one is misclassified as *Heteropneustes fossilis*. In the *Channa punctata* class, 14 images are correctly classified, and one is misclassified as *Batasio tengana* in InceptionV3 and one each is misclassified as Anabas testudineus and *Batasio tengana* in InceptionResNetV2. In the *Puntius* sophore class, 10 images are correctly classified, and one is misclassified as Anabas testudineus and two each are misclassified as *Channa punctata* and *Ompok bimaculatus* in InceptionV3. In InceptionV3, the training period completed with 24 epochs and achieved an accuracy of 94%. In InceptionResNetV2, before applying the Unsharp mask to the images, 36 epochs were completed during the training period, and an accuracy of 95% was achieved.

After applying Unsharp mask to the images in the dataset and retraining the InceptionResNetV2 architecture for 20 epochs, an accuracy of 91% was achieved. The confusion matrix showed that for *Anabas testudineus*, *Batasio tengana*, *Heteropneustes fossilis*, *Marcrobrachium malcoimsonii*, *Mstacembelus armatus*, and *Ompok bimaculatus* classes, all 15 test images were correctly classified. In Channa punctata, 13 images were correctly classified, but one each was misclassified as *Anabas testudineus* and *Batasio tengana*, and in *Puntius sophore*, 6 images were correctly predicted, but one image was misclassified as *Anabas testudineus*, every two images were miss-matched with *Batasio tengana* and *Channa punctata*, and four images were miss-classified as *Ompok bimaculatus*.

4.1.2 Outcome of DenseNet121, DenseNet169 and DenseNet201 Architecture:

Before applying Unsharp mask, the DenseNet121 architecture was trained for 37 epochs and achieved an accuracy of 96%. The confusion matrix showed that for *Batasio tengana, Marcrobrachium malcoimsonii, Mstacembelus armatus, and Ompok bimaculatus* classes, all 15 test images were correctly classified. For *Anabas testudineus*, 14 images were correctly classified, but one was misclassified as *Heteropneustes fossilis*. In *Channa punctata*, 13 images were correctly classified, but one each was mis-matched with *Heteropneustes fossilis* and *Puntius sophore*. In *Heteropneustes fossilis*, all 14 images were correctly predicted, but one was misclassified as *Anabas testudineus*. In *Puntius sophore*, 14 test images were correctly classified, but one was mis-matched with *Channa punctata*.

After applying unsharp mask and retraining the DenseNet121 architecture for 37 epochs, the accuracy was 93%. The confusion matrix showed that for *Batasio tengana, Channa punctata*, and *Mstacembelus armatus* classes, all the test images were correctly predicted. For Anabas testudineus, 14 images were correctly classified, but one was mis-matched with *Heteropneustes fossilis*. In Heteropneustes fossilis, all 14 images were correctly predicted, but one was misclassified as *Anabas testudineus*. In *Marcrobrachium malcoimsonii* class, 13 images were correctly classified, but two images were wrongly predicted as *Heteropneustes fossilis*. In *Ompok bimaculatus* class, 14 images were correctly recognized, but one was misclassified as Mstacembelus armatus. In Puntius sophore class, 12 images were correctly recognized, but two images were misclassified as *Channa punctata* and one image was miss-matched as *Ompok bimaculatus*.

Before applying Unsharp mask, the DenseNet169 architecture was trained for 25 epochs and achieved an accuracy of 96%. The confusion matrix showed that for *Anabas testudineus, Batasio tengana, Heteropneustes fossilis, Mstacembelus armatus*, and *Ompok bimaculatus* classes, all 15 test images were correctly classified. In *Channa punctata* class, 13 test images were well recognized, but one each was misclassified as *Batasio tengana* and *Heteropneustes fossilis*.

After applying the Unsharp mask to the images and retraining the DenseNet201 architecture, a total of 37 epochs were executed and an accuracy of 93% was achieved. From the confusion matrix analysis, in the *Anabas testudineus* class, 14 images were correctly classified and one misclassified as *Heteropneustes fossilis*. In *Batasio tengana, Mstacembelus armatus*, and *Ompok bimaculatus* classes, all 15 test images were correctly categorized. In *Channa punctata* class, 13 images were correctly classified and two misclassified as *Heteropneustes fossilis*. In *Heteropneustes fossilis* class, all 14 images were correctly predicted, but one was misclassified as *Anabas testudineus*. In *Marcrobrachium malcoimsonii* class, 13 images were correctly classified, and two were wrongly predicted as *Channa punctata*. In Puntius sophore class, 12 images were correctly recognized, but one was misclassified as *Batasio tengana*, and two were misclassified as *Channa punctata*.

4.1.3 Outcome of Proposed (DenseNet201+ConvLSTM) Architecture

Proposed CNN's DenseNet201 model to combine with the RNN's ConvLSTM to predict a more accurate and steady result, In the training process 26 epochs completed and we have found an accuracy of 97%, and from confusion matrix analysis, each of *Anabas testudineus, Batasio tengana, Channa punctata, Mstacembelus armatus*, and *Puntius sophore* class all the 15 test images have perfectly recognized as their class, in *Heteropneustes fossilis* class 14 images correctly categorized but one misclassified as *Channa punctata*, now *Marcrobrachium malcoimsonii* 13 test images correctly predicted and one each image wrongly predicted as *Batasio tengana* and *Heteropneustes fossilis*, in *Ompok bimaculatus* class 14 images classified correctly but one miss categorized as *Marcrobrachium malcoimsonii.*

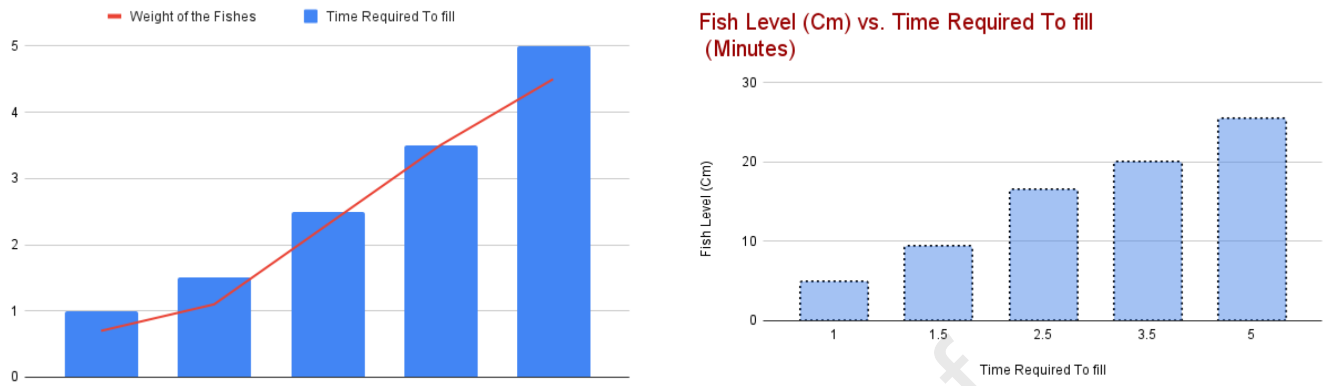
4.2 Data Analysis with IoT based Solution

This subsection presents the experimental data analysis. Table 10 shows the corresponding data analysis with the IoT based solutions. In this experiment we have taken a container to store the as per proposed architecture. The container successful store the container with the fishes indicating the empty level of the container along with the weight of the fishes. Figure 22 shows the experimental data with respect to time vs. fish level as well as time vs. weight of the fishes.

Table 10: Data Analysis with IoT based components

Number Trials	Time Required to fill (Minutes)	Fish Level (Cm)	Container Empty Level (%)	Weight of the Fishes (KG)
1	1.0	5.0	Above 85%	0.70
2	1.5	9.5	Above 75%	1.10

3	2.5	16.5	Above 55%	2.50
4	3.5	20.0	Below 35%	3.50
5	5.0	25.5	Below 15%	4.50



(a) (b)

Figure 22: (a) Time vs. Weight of the Fishes (b) Time vs. Fish Level

4.3 Comparison of studies

Table 11: Comparison among previous work and this work

Reference	Dataset	Model	Contribution	Accuracy (%)	Limitations
Elbatsh et al. [25]	Total Image:2670 Species:128	CNN	During the system implementation, a huge data set were collected which helps other researchers and scientist	80%	Dataset combined with around twenty images per species.
Jose et al. [30]	Total Image:612 Species:3	CNN	Authors separated three regions of each fish images and enhanced before used as input to previously trained convolutional neural networks (CNN)	93.91%	Only three species have been used to comprise the dataset.
G. G. Dialogo. [20]	Total Image:6918 Species:467	CNN	Training accuracy increased from previous studies using CNN model	82.81	Dataset built with on an average fourteen images per species.
Kurniawan et al. [31]	Total Image:434 Species:2	CNN	Employed the ENN approach, with CNN1, CNN2, CNN3, and CNN4 as the 2 to 5 convolutional layer designs, respectively	94.70	Dataset consisted of only two species.

Now in Table 11, we summarised some previous research that is mostly related to our work. Most of the datasets used in these studies utilized a few numbers of species or the number of images was relatively low. In our study, we built a dataset containing Bangladeshi fish species, and the datasets available on the internet don't contain the species we used in our study. And finally, we have developed a hybrid model combining CNN and RNN approaches and IoT-based advanced solutions.

5. Conclusion and Possible Future Contributions

Fish categorization leads to an automated machine-based fish separation system. Deep learning and the Internet of Things (IoT) each offer an efficient solution for classification and real-time data monitoring. This study focuses on the creation of an embedded system based on Deep Learning and IoT principles. The proposed technique is divided into interrelated sections. The first section describes the working principles of DL as they relate to dataset construction, model analysis, and overall system architecture. A new dataset of eight different Bangladeshi fish species. During the DL process, two sets of datasets were created, setup-1(S1) containing original images and setup-2(S2) containing Unsharp masked photos. Then, on both benchmarking setups, seven traditional ImageNet pertained state-of-the-art deep learning models: InceptionV3, Xception, DenseNet121, DenseNet169, DenseNet201, InceptionResNetV2, and ResNet152V2. This study found that the DL models and IoT-based components produced satisfactory results. The best benchmark accuracy for setup-1 was 96% for all of the DenseNet121, DenseNet169, and DenseNet201 architecture, and for setup-2, it was 96% for the Xception model. Finally, we built a hybrid (CNN+ Convolutional LSTM) model with a 97% accuracy that outperformed all of the previously mentioned state-of-the-art methods.

While studying with the proposed models, this study includes some drawbacks. We utilize only eight species of fishes. More fish species can be added to extend this research further. The number of images per species needs to be better. Increasing the number of images per species could improve classification accuracy. Also, the processing techniques applied to the dataset aren't enough. We aim to engage several different image processing techniques. However, the proposed solution can be workable in real-time solution.

Reference

[1] J. W. R. Chong *et al.*, "Use of image processing and digital algorithm for microalgae identification," *Bioresource Technology,* p. 128418, 2022.

[2] D. Y. Y. Tang *et al.*, "Application of regression and artificial neural network analysis of Red-Green-Blue image components in prediction of chlorophyll content in microalgae," *Bioresource Technology,* vol. 370, p. 128503, 2023.

[3] J. W. R. Chong, K. S. Khoo, K. W. Chew, H.-Y. Ting, and P. L. Show, "Trends in digital image processing of isolated microalgae by incorporating classification algorithm," *Biotechnology Advances,* p. 108095, 2023.

[4] A. P. Peter *et al.*, "Artificial intelligence model for monitoring biomass growth in semi-batch Chlorella vulgaris cultivation," *Fuel,* vol. 333, p. 126438, 2023.

[5] N. K. Singh *et al.*, "Artificial intelligence and machine learning-based monitoring and design of biological wastewater treatment systems," *Bioresource technology,* p. 128486, 2022.

[6] "Fish." Banglapedia. https://en.banglapedia.org/index.php/Fish#:~:text=In%20Bangladesh%20there%20are%20401,etc)%20are%20now%20becoming%20rare. (accessed 01 November, 2022).

[7] T. Ambulkar, "AQUACULTURE IN BANGLADESH: USING SEAFOOD TO OVERCOME HUNGER," Blog, Report NOVEMBER 3, 2016 2016. Accessed: April 04, 2023. [Online]. Available: https://borgenproject.org/aquaculture-in-bangladesh/#:~:text=With%20its%20close%20proximity%20to,essential%20fatty%20acids%20and%20protein.

[8] I. Mahmud, "Bangladesh globally ranks third in fish farming," in *Prothom Alo English*, ed. Prothom Alo: Matiur Rahman, 2022, p. 01.

[9] M. Sowman, "Subsistence and small-scale fisheries in South Africa: A ten-year review," *Marine Policy,* vol. 30, no. 1, pp. 60-73, 2006.

[10] C. Béné and S. Heck, "Fish and food security in Africa," 2005.

[11] B. Akande and Y. Diei-Ouadi, *Post-harvest losses in small-scale fisheries*. Food and Agriculture Organization of the United Nations, 2010.

[12] R. P. Mramba and K. E. Mkude, "Determinants of fish catch and post-harvest fish spoilage in small-scale marine fisheries in the Bagamoyo district, Tanzania," *Heliyon,* vol. 8, no. 6, p. e09574, 2022.

[13] U. Hamidu, "Assessment of the marine artisanal fisheries in Tanzania mainland," 2014.

[14] S. Maulu, O. J. Hasimuna, C. Monde, and M. Mweemba, "An assessment of post-harvest fish losses and preservation practices in Siavonga district, Southern Zambia," *Fisheries and aquatic sciences,* vol. 23, pp. 1-9, 2020.

[15] J. Brownlee, *Machine learning mastery with Python: understand your data, create accurate models, and work projects end-to-end*. Machine Learning Mastery, 2016.

[16] K. Kottursamy, "Multi-scale CNN Approach for Accurate Detection of Underwater Static Fish Image," *Journal of Artificial Intelligence,* vol. 3, no. 03, pp. 230-242, 2021.

[17] X. Xu, W. Li, and Q. Duan, "Transfer learning and SE-ResNet152 networks-based for small-scale unbalanced fish species identification," *Computers and Electronics in Agriculture,* vol. 180, p. 105878, 2021.

[18] A. Saleh, M. Sheaves, D. Jerry, and M. R. Azghadi, "Transformer-based Self-Supervised Fish Segmentation in Underwater Videos," *arXiv preprint arXiv:2206.05390,* 2022.

[19] U. A. Badawi, "Fish classification using extraction of appropriate feature set," *International Journal of Electrical & Computer Engineering (2088-8708),* vol. 12, no. 3, 2022.

[20] G. G. Dialogo, "Fish Species Detection Application (FiSDA) in Leyte Gulf Using Convolutional Neural Network," *Proceedings of Engineering and Technology Innovation,* vol. 19, p. 16, 2021.

[21] M. Paraschiv *et al.*, "Classification of Underwater Fish Images and Videos via Very Small Convolutional Neural Networks," *Journal of Marine Science and Engineering,* vol. 10, no. 6, p. 736, 2022.

[22] L. Jin, J. Yu, X. Yuan, and X. Du, "Fish classification using DNA barcode sequences through deep learning method," *Symmetry,* vol. 13, no. 9, p. 1599, 2021.

[23] R. J. Veiga *et al.*, "Autonomous Temporal Pseudo-Labeling for Fish Detection," *Applied Sciences,* vol. 12, no. 12, p. 5910, 2022.

[24] M. MUSLIM and M. F. R. ZULKIFLI, "CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES PERFORMANCE EVALUATION FOR FISH SPECIES CLASSIFICATION," *Journal of Sustainability Science and Management,* vol. 16, no. 5, pp. 124-139, 2021.

[25] K. Elbatsh, I. Sokar, and S. Ragab, "WikiFish: Mobile App for Fish Species Recognition Using Deep Convolutional Neural Networks," in *2021 The 4th International Conference on Computational Intelligence and Intelligent Systems*, 2021, pp. 13-18.

[26] A. Al Smadi, A. Mehmood, A. Abugabah, E. Almekhlafi, and A. M. Al-Smadi, "Deep convolutional neural network-based system for fish classification," *International Journal of Electrical and Computer Engineering,* vol. 12, no. 2, p. 2026, 2022.

[27] M. S. Cueto, J. M. B. Diangkinay, K. W. B. Melencion, T. P. Senerado, H. L. P. Taytay, and E. R. E. Tolentino, "Classification of different types of koi fish using convolutional neural network," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021: IEEE, pp. 1135-1142.

[28] S. M. Islam, S. I. Bani, and R. Ghosh, "Content-based Fish Classification Using Combination of Machine Learning Methods," 2021.

[29] C.-H. Chen, L.-H. Chen, and C.-Y. Chen, "Automatic Fish Segmentation and Recognition in Taiwan Fish Market using Deep Learning Techniques," *Journal of Imaging Science and Technology,* vol. 65, no. 4, pp. 40403-1-40403-10, 2021.

[30] J. A. Jose, C. S. Kumar, and S. Sureshkumar, "Tuna classification using super learner ensemble of region-based CNN-grouped 2D-LBP models," *Information Processing in Agriculture,* vol. 9, no. 1, pp. 68-79, 2022.

[31] K. Kurniawan, B. Sedayu, A. Hakim, and I. Erawan, "Classification of Rastrelliger kanagurta and Rastrelliger brachysoma using Convulational Neural Network (CNN)," in *IOP Conference Series: Earth and Environmental Science*, 2022, vol. 969, no. 1: IOP Publishing, p. 012017.

[32] O. Jafari *et al.*, "Integration of morphometrics and machine learning enables accurate distinction between wild and farmed common carp," *Life,* vol. 12, no. 7, p. 957, 2022.

[33] I. M. Hassoon, "Fish Species Identification Techniques: A Review," *Al-Nahrain Journal of Science,* vol. 25, no. 2, pp. 39-44, 2022.

[34] M. M. M. Fouad, H. M. Zawbaa, N. El-Bendary, and A. E. Hassanien, "Automatic nile tilapia fish classification approach using machine learning techniques," in *13th international conference on hybrid intelligent systems (HIS 2013)*, 2013: IEEE, pp. 173-178.

[35] F. J. P. Montalbo and A. A. Hernandez, "Classification of fish species with augmented data using deep convolutional neural network," in *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, 2019: IEEE, pp. 396-401.

[36] M. K. Alsmadi, K. B. Omar, S. A. Noah, and I. Almarashdeh, "Fish recognition based on robust features extraction from size and shape measurements using neural network," *Journal of Computer Science,* vol. 6, no. 10, p. 1088, 2010.

[37] S. Ogunlana, O. Olabode, S. Oluwadare, and G. Iwasokun, "Fish classification using support vector machine," *African Journal of Computing & ICT,* vol. 8, no. 2, pp. 75-82, 2015.

[38] D. Rathi, S. Jain, and S. Indu, "Underwater fish species classification using convolutional neural network and deep learning," in *2017 Ninth international conference on advances in pattern recognition (ICAPR)*, 2017: Ieee, pp. 1-6.

[39] I. Sharmin, N. F. Islam, I. Jahan, T. Ahmed Joye, M. Rahman, and M. Habib, "Machine vision based local fish recognition," *SN Applied Sciences,* vol. 1, no. 12, pp. 1-12, 2019.

[40] D. S. Y. Kartika and D. Herumurti, "Koi fish classification based on HSV color space," in *2016 International Conference on Information & Communication Technology and Systems (ICTS)*, 2016: IEEE, pp. 96-100.

[41] S. Arivazhagan, R. N. Shebiah, S. S. Nidhyanandhan, and L. Ganesan, "Fruit recognition using color and texture features," *Journal of Emerging Trends in Computing and Information Sciences,* vol. 1, no. 2, pp. 90-94, 2010.

[42] M. K. S. Alsmadi, K. B. Omar, S. A. Noah, and I. Almarashdah, "Fish recognition based on the combination between robust feature selection, image segmentation and geometrical parameter techniques using Artificial Neural Network and Decision Tree," *arXiv preprint arXiv:0912.0986,* 2009.

[43] Y. Zhang, S. Wang, G. Ji, and P. Phillips, "Fruit classification using computer vision and feedforward neural network," *Journal of Food Engineering,* vol. 143, pp. 167-177, 2014.

[44] V. Allken, N. O. Handegard, S. Rosen, T. Schreyeck, T. Mahiout, and K. Malde, "Fish species identification using a convolutional neural network trained on synthetic data," *ICES Journal of Marine Science,* vol. 76, no. 1, pp. 342-349, 2019.

[45] C.-L. Chen and C.-C. Chang, "Koi Fish Warehousing Based on Mobile Cloud Computing," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2019: IEEE, pp. 173-178.

[46] M. W. Rahman, R. Islam, A. Hasan, N. I. Bithi, M. M. Hasan, and M. M. Rahman, "Intelligent waste management system using deep learning with IoT," *Journal of King Saud University-Computer and Information Sciences,* 2020.

[47] S. Hasija, M. J. Buragohain, and S. Indu, "Fish species classification using graph embedding discriminant analysis," in *2017 International Conference on Machine Vision and Information Technology (CMVIT)*, 2017: IEEE, pp. 81-86.

[48] J. Hu, D. Li, Q. Duan, Y. Han, G. Chen, and X. Si, "Fish species classification by color, texture and multi-class support vector machine using computer vision," *Computers and electronics in agriculture,* vol. 88, pp. 133-140, 2012.

[49] S. Marini, E. Fanelli, V. Sbragaglia, E. Azzurro, J. Del Rio Fernandez, and J. Aguzzi, "Tracking fish abundance by underwater image recognition," *Scientific reports,* vol. 8, no. 1, pp. 1-12, 2018.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[51] S. Z. H. Shah *et al.*, "Fish-Pak: Fish species dataset from Pakistan for visual features based classification," *Data in brief*, vol. 27, p. 104565, 2019.

[52] J. C. R. Kenneth R. Spring, Matthew Parry-Hill, Thomas J. Fellers, and Michael W. Davidson. "Unsharp Mask Filtering." EVIDENT. https://www.olympus-lifescience.com/en/microscope-resource/primer/java/digitalimaging/processing/unsharpmask/#:~:text=The%20unsharp%20mask%20filter%20algorithm%20involves%20subtraction%20of%20an%20unsharp,a%20Gaussian%20low%2Dpass%20filte (accessed 01 November, 2022).

[53] Dejan. "Ultrasonic Sensor HC-SR04 and Arduino – Complete Guide." HowToMechatronics. https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/ (accessed 01 November, 2022).

[54] "3133 - Micro Load Cell (0-5kg) - CZL635." Datasheet. https://www.robotshop.com/media/files/pdf/datasheet-3133.pdf (accessed 01 November, 2022).

[55] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.

[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.

[57] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[58] C. Zuluaga. "Transfer learning using Keras with DenseNet-169." Medium. https://medium.com/@carlosz22/transfer-learning-using-keras-with-densenet-169-91679300f94a (accessed 07 September, 2022).

[59] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, 2016: Springer, pp. 630-645.

[61] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[62] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[63] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013: PMLR, pp. 1310-1318.

[64] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems,* vol. 27, 2014.

[65] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems,* vol. 28, 2015.

[66] C. e. al. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)." Papers With Code. https://paperswithcode.com/method/elu#:~:text=The%20Exponential%20Linear%20Unit%20(ELU)%20is%20an%20activation%20function%20for,but%20with%20lower%20computational%20complexity. (accessed 10 October, 2022).

[67] "Softmax." Papers With Code. https://paperswithcode.com/method/softmax (accessed 13 October, 2022).

- To classify and divide several fish species using machine learning and image recognition.
- To develop a smart fish-packing container.
- To combine two neural network methods CNN and RNN, which gives more accurate and steady results.

AUTHOR DECLARATION TEMPLATE

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We further confirm that any aspect of the work covered in this manuscript that has involved either experimental animals or human patients has been conducted with the ethical approval of all relevant bodies and that such approvals are acknowledged within the manuscript.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from (wahidtuhin0@gmail.com)

Signed by all authors as follows:

Corresponding Author	Signature	Date
Md. Wahidur Rahman		19/02/2023