Aquaponics Handbook: A Deep Convolutional Neural Network helper tool for Aquaponics

Introduction

Aquaponics is a typical bio-integrated method of production that links the cyclic nature of aquaculture with hydroponics, which shares a sustainable agricultural system (Panigrahi et al., 2016). It merges fish cultivation(aquaculture) and plant farming(hydroponics), without the presence of soil at the base of the setup. The mechanism works by combining these two agricultural sciences to sustain the recirculating system by natural biological cycles (nitrification) to supply nitrogen and reduce water inputs together with the non-renewable fertilizers (Tyson, n.d.). Several systems, techniques, design, scales and nomenclature within aquaponics have been developed throughout the years and it has already been commercialized and used in many scientific studies (Palm et al., 2018).

Producing plants hydroponically and farming fish using aqua­culture requires special methodologies to make sure that the respective systems are implemented and managed properly (Tyson, n.d.). The complexity of each system models is proportional to the parameters being considered and monitored across varying scales and purpose of implementation. Problems may occur when these parameters do not align with the acceptable thresholds that were initially set or external factors like the presence of disease-causing agents for the crops and unwanted chemicals in the waters are seen upon the actual implementation. In the real world, it is very difficult for fish farmers to recommend the perfect fish species for aquaculture in a specific aquaponics system (Islam et al., 2021). Land farmers may also be unfamiliar with the merger since growing crops through a land-based system is significantly different compared to an aquaponics environment. Whatever the case may be, investing into the production of an aquaponic system whether home-based or commercialized, is an expensive and risky venture if not equipped with the necessary guides and practices to not just make it work properly, but also sustainably maintaining its mechanisms.

Within the emergence of IoT based smart systems and the advancements in Artificial Intelligence, more innovative approaches would be developed to further advance the field of aquaponics and minimize the trade-offs enabling it to become a household or commercial venture. In this new era of science and technology, computer-aided diagnosis (CAD) systems are being used and developed for agricultural sciences along with the applications of Machine Learning and automation (Shrivastava et al., 2019). There are many applications of AI (Deep Learning and Machine Learning) in the aquaculture and hydroponics field and several journal articles have been published to provide empirical methods and sound practices that are based on gathered data and insights made through data mining (Dilmi & Ladjal, 2021; Islam et al., 2021; Kamilaris & Prenafeta-Boldú, 2018; Paymode & Malode, 2022; Shrivastava et al., 2019). Although both these fields share many common intersection with aquaponics, the implementation and scale are relatively different. In a general view, aquaponics encompasses both these fields and merges them to create a feedback and two player system that enables survival of various living agents within the model (Karimanzira et al., 2021).

Proper recommendations and guidelines for aquaponic farming are still lacking and requires more research investments focusing on possible production systems (type of fish × type of plant × densities × filtration system × hydroponic system × aquaculture system) and different varieties of crops (Tyson, n.d.). There are studies that integrate Internet of Things (IoT) devices within their aquaponic systems to gather data and make sense about the patterns that occur (Arvind et al., 2020). This intersection is commonly referred as “smart aquaponics” where data and information are core elements. However, measured data are oftentimes affected by complicated environmental factors that are usually nonlinear and various, which increases the complexity and difficulty of maintaining an accurate control system (Yang et al., 2021). The massive amount of data in smart aquaponics imposes a variety of challenges, such as multiple sources, data inconsistencies, data volatilities, and structural complexities. To find a way to convert this data into insights and imperatives, various sophisticated Machine Learning(ML) algorithms are being used. For image data, for example, Deep Learning can be utilized because it involves multilevel data representations, from low to high levels, in which high-level features are built on the low-level features and carry rich semantic information that can be used to recognize and detect targets or objects in the image (Yang et al., 2021). Deep Learning is commonly used together with other Machine learning models to makes sense of unstructured data like images or text. There are also other Machine Learning algorithms that can be used for structured or tabular datasets that can be utilized for regression or classification problems without the use of sophisticated Deep Learning techniques. By combining IoT, Data Mining, Artificial Intelligence and Data Science, it is possible to create a framework or standard that would aid beginning agriculturists or the common people to implement their own aquaponics system.

Transforming data from previously acquired analytics or creating an image classification deep learning algorithm are just some of the possibilities that can be implemented through Artificial Intelligence that would help non-experts of the field have a data-driven decisions for implementing their own aquaponics system. As of the writing of this paper, there are only handful of AI-based papers that focuses solely on aquaponics and to the best of my knowledge, this is the first paper to provide a handbook for aquaponics covering several preliminaries using both Machine Learning and Deep Learning from the gathered datasets.

Methodology

The underlying step-by-step process followed when building the machine learning models are presented on Appendix 1. The processes within the workflows were thoroughly discussed and the models were specifically created for each respective feature present at the Aquaponics Handbook website.

The dataset

The varying datasets were all gathered from Kaggle.com. The largescale fish dataset was adapted from the fish segmentation and classification study by Ulucan, et al. The dataset is composed of raw fish images separated through nine different classes namely: Black Sea Sprat, Gilt Head Beam, Horse Mackerel, Red Mullet, Red Sea Bream, Sea Bass, Shrimp, Striped Red Mullet, and Trout. Images were collected via 2 different cameras, Kodak Easyshare Z650 and Samsung ST60. Therefore, the resolution of the images are 2832 x 2128, 1024 x 768, respectively (Ulucan et al., 2020). For each class, there are 1000 augmented images and their pair-wise augmented ground truths. However, this paper only considered the raw 1000 augmented images and completely disregarded the others. The sample images from randomly chosen classes are present in Figure 1.

A picture containing several

Description automatically generated

FIGURE I

The image dataset for plant disease classification was taken from a publicly known dataset called Plant Village from Pennsylvania State University that is available in many machine learning repositories. In this data-set, 38 different classes of plant leaf and background images are available. These classes include 13 different plant types some with many available images of different plant diseases. In this study, the proposed deep learning model utilized the augmented version of the dataset by Geetharamani and Arun Pandian in their paper “Identification of plant leaf diseases using a nine-layer deep convolutional neural network” (Geetharamani & Arun Pandian, 2019). Overall, the data-set contained 54, 305 images. The sample images from randomly chosen classes are presented in Figure 2.

A picture containing green, window

Description automatically generated

FIGURE II

Experimental Setup

The Image classification features of the aquaponic handbook website utilized the pre-trained Deep Convolutional Neural Network(CNN) model called MobileNetV2 as base model for transfer-learning (LeCun et al., n.d.; Sandler et al., 2019). This was implemented through the TensorFlow, Keras, and pandas libraries which use the Python programming language. The source code was written on Google Collaboratory which offers a free Graphics Processing Unit(GPU) that makes training and testing the models faster. The trained weights were then downloaded locally and integrated to a Streamlit backend which was used to create the website. The overall web application was then uploaded at GitHub and hosted at Streamlit sharing.

Image Classification

The schematic in Figure 3 depicts a potential view for the image classification and analysis of the available features in the aquaponics handbook website. Initially, plant leaf disease and market fish images are collected and classified into several categories. The image file paths were then arranged with its corresponding labels through a pandas Data frame to structure the data into two columns(file paths and labels) with the individual images as its rows. The data was then split into three categories namely: training, validation, and testing. Each category has a target size of 224 x 224, color mode is ‘rgb’, class mode is categorical, and the batch size is 32. The individual images were also preprocessed by scaling them to range from 0 to 1. Random flip, random rotation, random zoom, height adjustment, and width adjustment are some of the data augmentation techniques used. By using data augmentation methods, new sample photos are created from available photos to enhance and prepare the dataset (Paymode & Malode, 2022). In this case, only the images from the plant village dataset undergone data augmentation preprocessing. The photos are then used as input to the suggested approach for training the model in the following stage. After undergoing through transfer learning, the data from the base model is then connected to two Dense layers with 256 hidden fully connected(FC) layers and the activation function used was Rectified Linear Unit (ReLU). In the fish classification feature, only 128 fully connected layers are used within the two Dense layers. Both the FC layers are separated by a Dropout function with a value of 0.2 respectively (Srivastava et al., n.d.). The dropout function was only applied to the images in the plant village dataset since there is a tendency for the model to overfit in the training data. The newly trained architectural model will be used to anticipate previously unseen images. In this case, the model was evaluated on the test data from the previous split. Eventually the findings of the classifications are achieved.

Transfer Learning

The Deep Learning model’s optimization and training is a computationally intensive and time-consuming operation. As mentioned earlier, a powerful graphics processing unit(GPU) is required for training the model, as well as large amounts of data. However, transfer learning, which is deployed in deep learning, solves these problems (Paymode & Malode, 2022). The pre-trained Convolutional Neural Network (CNN) used in transfer learning is optimized for one task and transfers knowledge to different modes (Nevavuori et al., 2019; Paymode & Malode, 2022). The images from the gathered datasets were compromised of different file sizes. Due to this, it was resized to a size of 224 x 224 with three channels to cater its ‘rgb’ type. The pre-trained CNN used is MobileNetV2 was used to find patterns within the input images and it’s corresponding final layers have to be connected to a dense layer. The final layers before the softmax is a 11 x 11 Dense layer for the Fish classification, and a 38 x 38 Dense for the plant disease classification. The basic picture preparation is necessary for the transfer learning considerations with the data augmented images. Figure 4 shows the whole Convolutional Neural Network architecture used to train the images. A screenshot of a computer

Description automatically generated with medium confidence

FIGURE III

Histogram

Description automatically generated with medium confidence

FIGURE IV

Model Evaluation

The test dataset gathered from the splitting of the dataset during the data preparation stage was used to evaluate the trained model. Classification metrics from scikit-learn library were also used. This includes the classification reports and confusion matrix. Throughout the training of the model, the training accuracy, training loss, validation accuracy, and validation loss were also tracked to determine if the model is overfitting or underfitting.

Aquaponics Handbook Website

The saved weights from the trained models were downloaded locally. The image classification features were made through the Streamlit library that use the Python programming language. The finished website was then uploaded to GitHub pages together with the images, weights, and helper python files.

Results and Discussions

Metrics

The metrics used to evaluate how well the model performs are precision, recall, F1 score and the confusion matrix. Higher precision leads to less false positives, higher recall leads to less false negatives, f1-score is the combination of precision and recall, usually a good overall metric for classification model. When comparing predictions to truth labels, a confusion matrix can be used to see where the model gets confused.

* Precision(P)

The fraction of true positives (TP, correct predictions) from the total amount of relevant results, i.e., the sum of TP and false positives (FP). For multi-class classification problems, P is averaged among the classes (Kamilaris & Prenafeta-Boldú, 2018).

*P=TP/(TP+FP)* (1)

* Recall(R)

The fraction of TP from the total amount of TP and false negatives (FN). For multi-class classification problems, R gets averaged among all the classes (Kamilaris & Prenafeta-Boldú, 2018).

R=TP/(TP+FN)

* F1 score(F1)

The harmonic mean of precision and recall. For multi-class classification problems, F1 gets averaged among all the classes (Kamilaris & Prenafeta-Boldú, 2018). It is mentioned as F-measure in (Minh et al., 2017).

F1=2 \* (TP \* FP)/(TP+FP)

Fish Classification Evaluation

The images from the fish classification was trained using the proposed Deep Convolutional Neural Network. The model scored a test accuracy of 99.78% and a test loss of 0.00722. The timeline for the accuracy and loss of the training and validation datasets are presented on Figure 5 and Figure 6 respectively. The corresponding classification reports are presented on Table 1 and the confusion matrix on Figure 7. Sample predictions on the test data are presented on Figure 8.

Line chart

Description automatically generated with medium confidence

FIGURE V

Chart, line chart

Description automatically generated

FIGURE VI

Table 1

Classification reports of the fish dataset after evaluating the model with the test data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class Name | Precision | Recall | F1-score | Support |
| Black Sea Sprat | 1.00 | 1.00 | 1.00 | 206 |
| Gilt-Head Bream | 0.99 | 0.99 | 0.99 | 188 |
| Hourse Mackerel | 1.00 | 1.00 | 1.00 | 202 |
| Red Mullet | 1.00 | 0.99 | 1.00 | 187 |
| Red Sea Bream | 0.99 | 1.00 | 1.00 | 196 |
| Sea Bass | 1.00 | 1.00 | 1.00 | 225 |
| Shrimp | 1.00 | 1.00 | 1.00 | 208 |
| Striped Red Mullet | 0.99 | 0.99 | 0.99 | 183 |
| Trout | 1.00 | 1.00 | 1.00 | 205 |

Chart

Description automatically generated with medium confidence

FIGURE VII

A picture containing text, several

Description automatically generated

FIGURE VIII

Plant Disease Classification Evaluation

The images from the plant disease classification was trained using the proposed Deep Convolutional Neural Network. The model scored a test accuracy of 96.35% and a test loss of 0.11303. The timeline for the accuracy and loss of the training and validation datasets are presented on Figure 9 and Figure 10 respectively. The corresponding classification reports are presented on Table 2 and the confusion matrix on Figure 11. Sample predictions on the test data are presented on Figure 12

A picture containing graphical user interface

Description automatically generated

FIGURE IX

A picture containing chart

Description automatically generated

FIGURE X

Table 2

Classification reports of the plant village dataset after evaluating the model with the test data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class Name | Precision | Recall | F1-score | Support |
| Apple scab | 0.97 | 0.81 | 0.89 | 124 |
| Apple Black rot | 0.99 | 0.98 | 0.99 | 128 |
| Apple Cedar - apple rust | 0.96 | 0.92 | 0.94 | 60 |
| Apple healthy | 0.97 | 1.00 | 0.98 | 336 |
| Blueberry healthy | 0.99 | 1.00 | 0.99 | 326 |
| Cherry(including sour) Powdery mildew | 1.00 | 0.99 | 0.99 | 227 |
| Cherry(including sour) healthy | 1.00 | 0.99 | 1.00 | 180 |
| Corn(maize) Cercospora leaf spot - Gray leaf spot | 0.83 | 0.81 | 0.82 | 110 |
| Corn(maize) Common rust | 0.98 | 1.00 | 0.99 | 255 |
| Corn(maize) Northern Leaf Blight | 0.90 | 0.90 | 0.90 | 201 |
| Corn(maize) healthy | 1.00 | 1.00 | 1.00 | 237 |
| Grape Black rot | 0.98 | 0.97 | 0.97 | 228 |
| Grape Esca (Black Measles) | 0.97 | 0.98 | 0.98 | 256 |
| Grape Leaf blight (Isariopsis Leaf Spot) | 1.00 | 0.97 | 0.98 | 232 |
| Grape healthy | 0.99 | 1.00 | 0.99 | 82 |
| Orange Haunglongbing (Citrus greening) | 1.00 | 1.00 | 1.00 | 1085 |
| Peach Bacterial\_spot | 0.99 | 0.99 | 0.99 | 469 |
| Peach healthy | 1.00 | 0.97 | 0.98 | 59 |
| Pepper, bell – Bacterial spot | 0.95 | 0.94 | 0.95 | 212 |
| Pepper, bell – healthy | 0.98 | 0.99 | 0.98 | 302 |
| Potato Early blight | 0.96 | 0.98 | 0.97 | 202 |
| Potato Late blight | 0.93 | 0.95 | 0.94 | 204 |
| Potato healthy | 1.00 | 0.85 | 0.92 | 39 |
| Raspberry healthy | 0.98 | 1.00 | 0.99 | 89 |
| Soybean healthy | 0.99 | 1.00 | 0.99 | 980 |
| Squash Powdery mildew | 1.00 | 0.99 | 1.00 | 366 |
| Strawberry Leaf scorch | 0.97 | 1.00 | 0.98 | 235 |
| Strawberry healthy | 1.00 | 0.99 | 0.99 | 86 |
| Tomato Bacterial spot | 0.90 | 0.96 | 0.93 | 417 |
| Tomato Early blight | 0.85 | 0.67 | 0.75 | 189 |
| Tomato Late blight | 0.93 | 0.95 | 0.94 | 380 |
| Tomato Leaf Mold | 0.91 | 0.91 | 0.91 | 201 |
| Tomato Septoria leaf spot | 0.87 | 0.89 | 0.88 | 360 |
| Tomato Spider mites Two-spotted spider mite | 0.90 | 0.93 | 0.91 | 316 |
| Tomato Target Spot | 0.86 | 0.88 | 0.87 | 259 |
| Tomato Yellow Leaf Curl Virus | 0.99 | 0.99 | 0.99 | 1028 |
| Tomato mosaic virus | 0.94 | 0.82 | 0.88 | 73 |

Chart

Description automatically generated

FIGURE XI

A picture containing text, several

Description automatically generated

FIGURE XII