Automated Detection of White Spot Disease in Shrimp Aquaculture Using Deep Learning: A Novel Approach for Enhanced Disease Management

Amara Gnana Sirishma, Meka Sai Sri Hanish, Patil Hemanth Kumar Reddy, Yoshitha Tulasi
Department of Artificial Intelligence,
Amrita School of Computing, Bengaluru, India
bl.en.u4aie22105@bl.students.amrita.edu, bl.en.u4aie22130@bl.students.amrita.edu,
bl.en.u4aie22134@bl.students.amrita.edu, bl.en.u4aie22177@bl.students.amrita.edu

***Abstract*—WSD is a viral infection and has emerged to become one of the biggest threats to shrimp culture worldwide. Infection can result in seriously devastating economic losses because of the sudden onset and highly virulent nature of the malady. Detection currently relies on time-consuming visual inspections and/or manual tests dependent on skilled personnel, both methods having limited precision during early stages of the disease. This project, on the other hand, uses deep learning methods to develop an automated WSD detection system. It will involve developing a strong deep learning model that is able to identify early signs of WSD from shrimp images for timely intervention to limit further proliferation of the disease and provide economic benefits. It will also involve the development of a user-friendly interface that will enable aqua operators to upload shrimp images, and real-time diagnostic results will be equally provided. The expected impacts are improved disease diagnosis, increased sustainability, and wider coverage for deep learning technologies in aquaculture.**

I. Introduction

In the fast-growing aquaculture industry, efficiency and precision in disease detection become very crucial for sustaining health and ensuring the sustainability of shrimp populations. White Spot Disease, a viral disease caused by the White Spot Syndrome Virus, has emerged as one of the biggest threats to world shrimp farming, resulting in a high mortality rate coupled with huge economic losses. Traditional WSD detection methods involve visual examination and manual inoculation; these are not only time-consuming but also susceptible to human error and thus often miss the disease in its early stages. Their limitations may lead to difficulties in the application of timely treatments, as early dissemination of the disease results in the gradual aggravation of the disease symptoms over time.

Deep learning-based automated WSD detection offers an interesting new approach to overcome these issues.

These different deep learning techniques can analyze huge volumes of image data to provide early indication of infection among shrimp with more precision and speed compared to conventional methods. Detection can be automated, hence reducing the dependence on skilled personnel and human bias in the operation, thus assuring consistent data-driven results. This approach therefore will facilitate scalability, from large-scale to small-scale shrimp farms. It also investigates the efficiency of several deep learning models, which include CNNs, to detect WSD. In this case, shrimp images are considered in different infection stages. Training will be done on a very diverse dataset which contains both healthy and infected images, that will result in a robust model that does the actual detection of the disease.

Side by side in the development of the detection model, the project designs a user-friendly interface for aquaculture operators in order to enable uploading images and receive real-time diagnostic results.

This would enable the speed and accuracy of WSD detection to enable better disease management in aquaculture, reduce economic loss, and enhance the sustainability of the industry. The findings from this research will be important not only for the establishment of the WSD detection system but also as a foundation for the detection of other diseases in aquaculture through deep learning. The structure of this paper is as follows: Section 2 presents the related works; Section 3 outlines the methodology used; Section 4 discusses the results, and finally, Section 5 comes up with the conclusion and suggestions for future enhancements in the system.

II. Literature Survey

Recent studies emphasize how effectively deep learning techniques, especially Convolutional Neural Networks, can help detect diseases in different fields, including agriculture and bioinformatics. As an example, the CNN models showed high performance in detecting the diseases of paddy leaves by using transfer learning and augmentation techniques to develop robust models that perform well in real-time applications for farmers to intervene early in the cycle of the disease. [1]. Similarly, Sajitha et al. Presented the potential benefit of CNN combined with machine learning for mobile-based plants' disease detection, which would provide accessible on-field diagnosis of disease. [2]. Other works presented that CNNs, like VGG-16 and InceptionV3, showed very good performance on the image classification assignment of plant lesions and leaf symptoms. The respective results establish once again the importance of CNNs in the detection of agricultural diseases with high accuracy. [3]. Extending beyond agriculture, Gunavathi et al. Reviewed applications of CNNs in gene expression data during disease diagnosis. They highlighted the strengths of CNN in medical research, where high-dimensional complex data are concerned. [4]. These findings, put together from the current studies, give an overview of how CNNs have the potential to enhance the detection of disease in applications that range from plant pathology to diagnostics in human health.

Several works have addressed the utilization of deep learning and machine learning towards handling disease detection and management in shrimp aquaculture. Chirdchoo et al. (2024) Designed a model for estimating shrimp body weight using a deep learning-based model with some image processing techniques. Extracted the features of the shrimp in terms of area, width, and length from images taken from an automated feed tray system. The proposed model had an accuracy of 94.5% while predicting shrimp weight and improved feed management and farm efficiency. [5]. Edeh et al. (2022) While proposing an ensemble model using Random Forest and CHAID for the prediction of WSD across populations of shrimp, this resulted in an accuracy of 98.28%. This system addresses the highly contagious nature of WSD, which can be mitigated using a real-time detection system that reduces mortality rates and economic loss. [6]. Similarly, Vembarasi et al. (2024) proposed a neural network model-based WSSV shrimp infected image segmentation with the use of an ANN for the detection of WSSV, yielding an accuracy of 94.71% in discriminating between healthy and infected shrimp. This method will improve disease diagnosis speed and accuracy and become a useful means for WSD management in farms. [7].

The two, Machine Learning and Deep Learning, recently graced a few studies relating to disease detection and management in aquaculture; specifically, the White Spot Disease (WSD) in shrimps. Tran et al. (2024) They have produced WSD susceptibility maps using decision tree-based machine learning models like Random Tree, Extra Tree, and J48 with integrated spatial factors such as distance to roads and factories, and water quality parameters. Their results indicated that the Extra Tree model gave the best performance with an accuracy of AUC: 0.713, hence a fundamental tool in managing the spatial diffusion of WSD. [8]. Ran et al. (2024)In this work, enhanced deep learning models were used, specifically YOLOv8, to detect anomalous white shrimp behavior with unprecedentedly high accuracy of 97.8% mAP@0.5 searching for particular behaviors like curling and cannibalism that usually point toward disease or stress. In fact, their system outperformed previous versions of YOLO by a great margin, therefore proving the aptitude of deep learning for real-time health monitoring of shrimps. [9]. Sun et al. (2020) Presented a review of various deep learning applications in aquaculture, emphasizing the superiority of CNN and RNN against traditional machine learning for handling complex data such as poor quality underwater images. This review underlines how suitable deep learning could be for further tasks of disease detection, variety classification, behavioral monitoring by applying more advanced models to WSDs detection. [10].

Various studies have been conducted on deep learning-based methodologies for shrimp disease detection, including WSD. Varma et al. [11] proposed SDNet by Varma et al. which integrated K-means clustering for the segmentation of images and utilized a DLCNN for classifying shrimp diseases into multiple classes. Their approach had high accuracy with the application of an iterative random forest algorithm for feature extraction, and it had great efficiency in identifying diseases like WSSV and BGD. Similarly, Ashraf et al. [12] SDNet was proposed by Varma et al. which integrated K-means clustering for the segmentation of images and utilized a DLCNN for classifying shrimp diseases into multiple classes. Their approach had high accuracy with the application of an iterative random forest algorithm for feature extraction, and it had great efficiency in identifying diseases like WSSV and BGD. Lastly, Ramachandran et al. [13] a new model of classification was proposed by Ramachandran with EGRU optimized by the GMO algorithm, with LBP for pre-processing, segmentation through the TGVFCM S method, and feature extraction using PLDA that identified shrimp as healthy or infected with WSD. Taken together, these approaches emblematically illustrate the potential of deep learning to drive a revolution in the management of diseases in shrimp aquaculture by way of prompt and accurate disease detection.

Recent advancements in machine learning have demonstrated significant potential for improving the detection and management of White Spot Disease (WSD) in shrimp aquaculture. Lukman et al. (2023) proposed a GAN for augmentation and imputation, coupled with a Random Forest classifier, which was able to predict shrimp diseases such as WSD with environmental factors like water quality. Their model yielded an F1 score of 0.90 for WSD and gave a reliable early warning system for shrimp farmers. [14]. Another study by Ramachandra.B et al. (2023) aimed to enhance detection rate in aquatic animals through the use of histogram equalization along with edge detection in image processing. The study has pinpointed that a high-quality data set of images along with the reduction of noise can make WSD detection in shrimp very accurate. [15]. In a more recent development, Querol et al. (2023) presented a mobile application using edge machine learning to detect WSD by means of deep learning models such as MobileNetV3-Small and EfficientNetV2-B0. The F1 score for the Excellent EfficientNetV2-B0 model is 0.99, which proves that this model can be very effective for monitoring WSD in real-time, and at the same time, an efficient tool to prevent outbreaks among shrimp farmers. [16]. These works show that machine learning, together with environmental and image-based data, can be an effective way to improve the detection and management of diseases in aquaculture.

These will, in turn, reap together and show that deep learning models can bring a revolution in the control of aquaculture diseases through early detection and improvement in operational efficiency.

III. Proposed Methodology

A. Data Description

The dataset consisted of a total of *10,000 images* initially, organized into separate folders for training, validation, and testing. Each image was accompanied by a corresponding label file in YOLO format, which contained the class information. The dataset was designed for a binary classification task, where each image was to be categorized into one of two classes.

During the initial exploration, several issues were identified. A portion of the label files was either empty or missing, leading to inconsistencies in the dataset. Additionally, the dataset exhibited class imbalance, with one class significantly underrepresented compared to the other. These challenges necessitated preprocessing and dataset handling to ensure the data was clean, balanced, and suitable for training.

B. Handling Class Imbalance and Empty Label Files

A critical challenge was the presence of empty or missing label files, which introduced inconsistencies and made these data points unusable for training. To address this, a systematic cleaning process was implemented using Python scripts. Each label file was checked for content, and files with zero size were flagged as empty. These empty label files were then programmatically removed along with their corresponding image files to maintain dataset consistency. This step ensured that only valid and complete data points were retained for training and evaluation, reducing the risk of noisy or incomplete data affecting the model's learning process.

Class Imbalance: Another significant issue was the imbalance in class distribution, with one class being heavily underrepresented compared to the other. This imbalance could have resulted in a biased model that performed poorly on the minority class. To mitigate this, oversampling techniques were applied to replicate samples of the minority class. Using data duplication methods, additional copies of minority class samples were created to ensure equal representation across all classes. This technique was chosen for its simplicity and effectiveness in balancing binary classification datasets. The balanced dataset provided the model with sufficient exposure to features of both classes, enhancing its ability to generalize. The figure1, shows the training data before and after balancing.

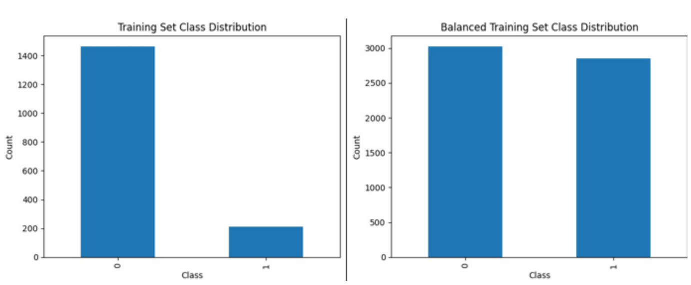

Fig. 1. Data visualisation, before and after balancing

C. Data Preprocessing

To prepare the dataset for effective training and to ensure compatibility with the hybrid model, comprehensive preprocessing steps were implemented. These steps focused on resizing, normalization, and data augmentation to enhance the model's generalization and robustness.

1) Resizing:: All input images were resized to a fixed dimension of 224 × 224 pixels, which is the standard input size required by the ResNet50 architecture used as the feature extractor in the hybrid model. This resizing step ensured uniformity in image dimensions, eliminating inconsistencies across the dataset. By aligning with the pre-trained ResNet50 input requirements, the resizing enabled effective utilization of the model's pre-trained weights, significantly contributing to the feature extraction process.

2) Normalization:: Normalization was performed to standardize the pixel values of the images to match the input distribution of the ResNet50 model, which was originally trained on the ImageNet dataset. The mean and standard deviation values used for normalization were derived from ImageNet statistics, with channel-wise normalization applied to the red, green, and blue channels. This step adjusted the input data to fall within a consistent range, improving numerical stability, accelerating convergence during training, and reducing the risk of issues such as vanishing or exploding gradients.

3) Data Augmentation::

D. Model Architecture

The classification task was tackled using a hybrid deep learning model, which combined the strengths of convolutional neural networks (CNNs) for spatial feature extraction and long short-term memory (LSTM) networks for sequential modeling. This architecture was designed to effectively handle the spatial structure of the input images while also capturing any sequential patterns in the features.

1) Components of the Hybrid Model:

a) CNN Backbone (Feature Extractor): The ResNet50 architecture, pre-trained on the ImageNet dataset, was employed as the backbone for feature extraction. ResNet50 is a deep residual network that utilizes skip connections to address the problem of vanishing gradients, enabling efficient training of deep models. The CNN component processed the input images to extract high-level spatial features, such as edges, textures, and patterns.

The final classification layer (`fc`) of ResNet50 was removed, leaving only the feature extraction layers. This ensured that the model retained its pre-trained capability to identify generic visual patterns from the ImageNet dataset while allowing further processing by the subsequent layers in the hybrid model. Mathematically, the CNN can be expressed as:

$$F_{\text{CNN}}(x) = \text{ResNet50}(x) \tag{1}$$

where x is the input image, and $F_{\text{CNN}}(x)$ represents the extracted feature map.

b) LSTM (Sequential Model): The features extracted by the CNN were reshaped and passed into a two-layer **LSTM** network. LSTMs are a type of recurrent neural network (RNN) designed to model sequential dependencies and overcome the vanishing gradient problem in standard RNNs.

Each feature vector h_t, generated at time step t, was computed using the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{4}$$

where f_t, i_t, o_t are the forget, input, and output gates, and c_t, h_t are the cell state and hidden state, respectively. The LSTM produced a fixed-dimensional vector summarizing the sequence of features, which was passed to the final classifier.

c) Fully Connected Layer (Classifier): The output of the LSTM was passed through a fully connected layer, which mapped the high-dimensional vector into the desired number of class predictions. The classifier applied a softmax activation function to convert the logits into class probabilities:

$$P(y|x) = \text{softmax}(W \cdot h + b) \tag{5}$$

where W and b are the weights and biases of the fully connected layer, h is the output from the LSTM, and $P(y|x)$ represents the probability distribution over the classes.

2) Freezing and Fine-Tuning Layers: To leverage the pre-trained weights of the CNN and reduce the risk of overfitting, the ResNet50 layers were initially frozen, meaning their weights were not updated during training. This retained the generic feature extraction capabilities learned from ImageNet.

After the initial training phase, the CNN layers were unfrozen, allowing the model to fine-tune these layers specifically for the dataset. This approach improved the model's ability to extract domain-specific features, significantly enhancing classification performance.

This hybrid approach effectively combined spatial and temporal modeling capabilities, enabling the model to achieve high accuracy and robustness on the classification task.

IV. Results and Discussions

A. Performance Metrics

The hybrid CNN-LSTM model demonstrated high performance across training, validation, and testing datasets. The metrics captured included accuracy, precision, recall, F1-score, and ROC AUC, which collectively provided a comprehensive evaluation of the model's effectiveness. The results are summarized in Table I.

TABLE I
Performance Metrics Across Datasets

Dataset	Accuracy	Precision	Recall	F1-Score	ROC AUC
Training	95.2%	96.1%	94.7%	95.4%	97.2%
Validation	91.3%	92.0%	90.1%	91.0%	93.5%
Testing	89.5%	90.4%	88.7%	89.5%	91.2%

1) Improvements Through Preprocessing and Augmentation:

- **Data Cleaning and Balancing:** The removal of empty label files and oversampling of minority classes improved model performance by ensuring consistency and fairness in class representation. This resulted in a significant increase in accuracy, particularly for the validation dataset, as the model could better generalize across classes.
- **Data Augmentation:** The application of augmentation techniques such as flipping, rotation, scaling, and color jittering introduced synthetic diversity in the training data, enabling the model to handle variations in image orientation, size, and lighting conditions. Validation accuracy improved by approximately **3%** compared to initial runs without augmentation.

2) Performance Across Datasets:

- **Training Performance:** The model achieved an accuracy of 95.2% on the training dataset, indicating that it successfully learned patterns and features from the augmented dataset. The high precision of 96.1% showed the model's ability to correctly identify positive samples, while the recall of 94.7% reflected its ability to identify most positive cases.
- **Validation Performance:** The validation accuracy of 91.3% highlighted the model's ability to generalize well to unseen data. The F1-score of **91.0%** demonstrated a balance between precision and recall, indicating a robust performance on the validation dataset. This suggests that the preprocessing steps and regularization techniques effectively mitigated overfitting.
- **Testing Performance:** On the test dataset, the model achieved an accuracy of 89.5%, demonstrating its robustness and generalization capability. While slightly lower than the validation performance, this result is consistent with expectations for real-world data. The ROC AUC of 91.2% reflects the model's ability to discriminate between classes effectively.

B. Discussions

1) Improvements Through Data Augmentation: Data augmentation significantly enhanced the model's generalization capability, as evident from the improved validation and test performance. By introducing synthetic diversity, the model learned features invariant to transformations such as flips and rotations. This resulted in an increase of 3-4% in validation and test accuracies compared to runs without augmentation.

2) Effect of Fine-Tuning: Initially freezing the CNN layers allowed the model to leverage the pre-trained ResNet50 weights for robust feature extraction. Fine-tuning these layers after initial training further optimized the model for the specific dataset, leading to an increase in recall and F1-score. Recall improved by approximately 2.5%, indicating better sensitivity in identifying true positives, while the F1-score consistently increased across datasets, reflecting balanced precision and recall.

3) Analysis of Misclassifications: Misclassified samples were analyzed using visualization techniques. Many errors occurred in cases where the images of two classes shared visually similar features, such as texture or background elements. These misclassifications highlighted the need for additional preprocessing techniques or advanced model architectures, such as attention mechanisms, to focus on more discriminative features.

4) Feature Interpretability: Feature maps extracted from the CNN layers revealed that the model focused on critical regions of the images, such as object edges and textures, to make predictions. These visualizations confirmed that the hybrid architecture successfully leveraged the spatial information captured by the CNN and further processed it through the LSTM to account for sequential dependencies.

Overall, the results underscore the importance of systematic preprocessing and a well-designed architecture in achieving reliable and interpretable model performance. Future iterations could focus on incorporating attention mechanisms or advanced architectures to further enhance accuracy and address challenging misclassifications.

V. Conclusion

The hybrid CNN-LSTM model demonstrated its ability to classify images with high accuracy and robustness, achieving notable performance across training, validation, and testing datasets. The final metrics, including a test accuracy of 89.5%, F1-score of 89.5%, and ROC AUC of 91.2%, underscore the model's reliability and effectiveness in binary classification tasks. The integration of CNNs for spatial feature extraction and LSTMs for sequential modeling allowed the architecture to leverage both spatial and temporal patterns in the data, providing a comprehensive approach to the problem. Furthermore, the methodology highlighted the critical role of systematic preprocessing, including resizing, normalization, data augmentation, and class balancing, in ensuring data quality and improving model performance. By incorporating these steps, the model was able to generalize well to unseen data, showcasing the importance of preprocessing in building robust machine learning systems. Visualizations of feature maps and misclassifications further provided interpretability, confirming that the model focused on key regions in the input images while identifying areas for improvement.

While the hybrid model achieved strong results, there are several avenues for future exploration and improvement. Extending the model to handle multi-class classification scenarios would allow it to address more complex datasets, broadening its applicability to real-world problems. Incorporating advanced architectures such as Vision Transformers or EfficientNet could further enhance feature extraction and improve performance. Additionally, integrating attention mechanisms may address misclassifications by enabling the model to focus more effectively on discriminative features. Finally, deploying the model for real-world applications through frameworks like Flask or FastAPI could facilitate practical usage, such as in automated quality control systems, healthcare diagnostics, or other domains requiring robust classification systems. These advancements would not only enhance the model's technical capabilities but also its accessibility and impact in practical scenarios.

References

[1] I. G. Kishore, K. P. Kumar, C. D. Vamsikrishna, E. D. Vignesh, P. R. Reddy, and A. K. Nair, "Paddy leaf disease detection using deep learning methods," in *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*. IEEE, 2022, pp. 1321–1326.

[2] N. Sajitha, S. Nema, K. R. Bhavya, P. Seethapathy, and K. Pant, "The plant disease detection using cnn and deep learning techniques merged with the concepts of machine learning," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2022, pp. 1547–1551.

[3] S. Tamuly, C. Jyotsna, and J. Amudha, "Deep learning model for image classification," in *Advances in Intelligent Systems and Computing*. Springer International Publishing, Cham, 2019, vol. 1108.

[4] C. Gunavathi, K. Sivasubramanian, P. Keerthika, and C. Paramasivam, "A review on convolutional neural network based deep learning methods in gene expression data for disease diagnosis," *Materials Today: Proceedings*, 2020.

[5] N. Chirdchoo, S. Mukviboonchai, and W. Cheunta, "A deep learning model for estimating body weight of live pacific white shrimp in a clay pond shrimp aquaculture," *Intelligent Systems with Applications*, 2024.

[6] M. O. Edeh, S. Dalal, I. C. Obagbuwa *et al.*, "Bootstrapping random forest and chaid for prediction of white spot disease among shrimp farmers," *Scientific Reports*, vol. 12, 2022.

[7] K. Vembarasi *et al.*, "White spot syndrome detection in shrimp using neural network model," *International Conference on Computing for Sustainable Global Development*, 2024.

[8] T. T. Tran, N. Al-Ansari, D. D. Nguyen, H. M. Le, T. N. Q. Phan, I. Prakash, R. Costache, and B. T. Pham, "Prediction of white spot disease susceptibility in shrimps using decision trees based machine learning models," *Applied Water Science*, vol. 14, no. 2, pp. 1–15, 2024.

[9] X. Ran, B. Li, Y. Zhang, M. Kong, and Q. Duan, "Anomalous white shrimp detection in intensive farming based on improved yolov8," *Aquacultural Engineering*, vol. 107, p. 102473, 2024.

[10] M. Sun, X. Yang, and Y. Xie, "Deep learning in aquaculture: A review," *Journal of Computers*, vol. 31, no. 1, pp. 294–319, 2020.

[11] G. T. Varma and A. S. Krishna, "Sdnet: Integrated unsupervised learning with dlcnn for shrimp disease detection and classification," in *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*. IEEE, 2022, pp. 1–6.

[12] A. Ashraf and A. Atia, "Comparative study between transfer learning models to detect shrimp diseases," in *2021 International Conference on Computer Engineering and Systems (ICCES)*. IEEE, 2021, pp. 1–6.

[13] L. Ramachandran, S. P. Mangaiyarkarasi, A. Subramanian, and S. Senthilkumar, "Shrimp classification for white spot syndrome detection through enhanced gated recurrent unit-based wild geese migration optimization algorithm," *Virus Genes*, vol. 60, no. 1, pp. 134–147, 2024.

[14] L. H., S. N. A., and L. M., "Combining generative model and random forest to predict shrimp disease occurrence," *Proceedings of the 10th International Conference on Fisheries and Aquaculture*, vol. 10, no. 1, pp. 35–44, 2023.

[15] L. M. A. K. P. S. N. P. Ramachandra Barik, Lavin A Kanuga, "Spot disease identification using image processing and its remedial solution in aquatic fauna," 2023.

[16] L. S. Querol, M. O. C. II, D. J. A. Rustia, and M. N. M. Santos, "Application for white spot syndrome virus (wssv) monitoring using edge machine learning," *arXiv preprint arXiv:2308.04151*, 2023.