Computer Science and Information Technologies
Vol. 7, No. 1, March 2026, pp. 20~29
ISSN: 2722-3221, DOI: 10.11591/csit.v7i1.p20-29

Development and performance evaluation of a CNN model for seagrass species classification in Bintan, Indonesia

Nurul Hayaty[1], Hollanda Arief Kusuma[2]
[1]Department of Informatics Engineering, Faculty of Engineering and Maritime Technology, University of Maritime Raja Ali Haji, Tanjungpinang, Indonesia
[2]Department of Electrical Engineering, Faculty of Engineering and Maritime Technology, University of Maritime Raja Ali Haji, Tanjungpinang, Indonesia

Article Info

Article history:

Received May 29, 2025
Revised Aug 12, 2025
Accepted Nov 28, 2025

Keywords:

Deep learning
Early stopping
Image classification
Marine biodiversity
Morphological similarity

ABSTRACT

This study presents the development and evaluation of a convolutional neural network (CNN) model for automated seagrass species classification in Bintan, Indonesia. The objective of this research is to examine how different train-validation data split ratios affect model accuracy and generalization performance. The CNN was trained under four configurations (60:40, 70:30, 80:20, and 90:10) to analyze the influence of training data volume on learning convergence and predictive capability. The results indicate that all configurations achieved high validation accuracy, with the best performance reaching 98.53% when using the 90:10 split. Evaluation on unseen data demonstrated that the 60:40 configuration provided the most consistent and reliable generalization. Performance variations were also affected by the morphological similarity between the classified species, which increases the challenge in correctly distinguishing certain classes. Overall, the findings confirm the effectiveness of CNN-based classification for supporting marine biodiversity monitoring and underline the importance of dataset composition in achieving optimal performance. Future improvements will focus on expanding data variability to enhance robustness in real-world scenarios.

Corresponding Author:

Nurul Hayaty
Department of Informatics Engineering, Faculty of Engineering and Maritime Technology
University of Maritime Raja Ali Haji
Tanjungpinang, Kepulauan Riau, Indonesia
Email: nurul.hayaty@umrah.ac.id

1. INTRODUCTION

Seagrass plays a crucial role in coastal ecosystems, providing habitat for marine biota, carbon sequestration, and coastal protection [1], [2]. In Bintan, seagrass ecosystems contribute significantly to fisheries and marine tourism. Seagrass beds in Bintan can increase fish resource availability by 9,049.3 kg per hectare annually, equivalent to a habitat value of Rp. 166,963,204.72. Seagrass also acts as a blue carbon sink, with the highest carbon storage found in Pengudang at 245.03 gC/m² or 348.26 MgC/ha.

The identification of seagrasses in coastal ecosystems face several challenges and opportunities for improvement. Manual identification of seagrass species poses challenges due to morphological similarity, variable imaging conditions, and the need for expert taxonomic knowledge. This research proposes an automated approach using convolutional neural networks (CNNs) to improve efficiency and accuracy in ecological surveys. Traditional manual methods, while commonly used, are time-consuming, costly, and

require a high level of specialized expertise. Environmental factors, such as variations in water clarity, light conditions, and seagrass morphology, further complicate the identification process [3].

Machine learning and image processing techniques are emerging as effective solutions to overcome these limitations. Ensemble-based machine learning methods, particularly rotation forests, have outperformed traditional maximum likelihood classifiers in mapping seagrass using Sentinel-2 imagery [4]. CNNs typically surpass the performance of traditional classification methods when applied to large-scale datasets, owing to their ability to automatically learn and extract hierarchical features directly from raw image data [5], [6]. Various approaches for seagrass detection and mapping have been identified, including still image, video data, acoustic image, and spectral image data-based techniques [6]–[8]. The transition from traditional manual approaches to digital imaging and machine learning techniques represents a transformative step forward in seagrass and marine vegetation monitoring efforts.

Recent studies have increasingly applied CNNs for seagrass classification and conservation. CNN-based models have been developed to detect and classify seagrass species from underwater imagery, achieving high levels of accuracy—often exceeding 90% [9], [10]. In addition to underwater applications, CNNs have also been utilized for analyzing high-resolution satellite imagery to map benthic habitats and monitor seagrass distribution in shallow marine environments [11]. For instance, Noman *et al*. [12] implemented a CNN-based approach that achieved an accuracy of 99.33% in seagrass classification. Ozaeta *et al*. [13] introduced a deep learning method using differentiable architecture search, reaching 93.72% accuracy in classifying five seagrass species from the Philippines. Meanwhile, Reus *et al.* [14] focused on seagrass segmentation, demonstrating the effectiveness of CNN features in estimating seagrass coverage, with an accuracy of 94.5%.

Recent research has focused on enhancing the performance of CNNs in classification tasks through data optimization. The impact of different train-test split ratios on CNN accuracy for EEG emotion recognition was investigated, revealing that an 80:20 split produced optimal results [15]. Likewise, Lecun *et al.* [16] demonstrated that modifying dataset configurations—such as class balance and data proportion—substantially improved CNN performance in chest X-ray image classification. A comprehensive study by Abadi *et al.* [17] evaluated class imbalance effects in CNN models, concluding that oversampling strategies offered the most stable solution to imbalance issues without inducing overfitting. Furthermore, Prechelt [18] proposed a rationale-based CNN model for text classification, which integrated sentence-level justifications and achieved high accuracy across multiple benchmark datasets. Together, these findings underscore the pivotal role of both data structuring and architectural design in maximizing the effectiveness of CNNs for classification tasks.

These findings highlight the significant potential of deep learning techniques in enhancing the accuracy and automation of seagrass classification. As seagrass ecosystems face increasing threats from human activities and climate change, such advancements are critical for improving the efficiency and scalability of monitoring and conservation efforts. By utilizing a representative dataset of seagrass images, machine learning models can be optimized to automatically detect and classify seagrass species, thus improving the efficiency of the monitoring process.

This research focuses on the development of a machine learning model to classify three seagrass species found in the coastal region of Bintan. Seagrass images taken from the study locations will undergo a series of preprocessing steps and will be used to train the machine learning model. Through this approach, it is expected that the research will make a meaningful contribution in simplifying the identification of seagrass species accurately and efficiently. The objective of this study is to design and evaluate the performance of a machine learning model in classifying seagrass species based on the available image dataset. Despite growing research on seagrass classification, there remains a lack of localized, species-specific models trained on curated datasets from Indonesia. This study addresses that gap by constructing a robust CNN trained on hand-labeled Bintan seagrass imagery, with performance evaluation across varied data compositions and deployment on mobile-friendly platforms such as TensorFlow Lite (TFLite) for lightweight inference in field environments.

2. METHOD

2.1. Research framework

This study employs a supervised deep learning approach utilizing a CNN architecture to classify three species of seagrass found in the coastal waters of Bintan: *Halodule uninervis, Syringodium isoetifolium,* and *Thalassia hemprichii*. CNNs are particularly suitable for image-based classification tasks due to their ability to automatically extract spatial features from raw pixel data [16]. The research methodology consists of nine primary stages, as illustrated in Figure 1. Image data were collected from a curated repository and grouped into species-specific classes. Preprocessing included resizing, normalization, and augmentation to

standardize inputs and enhance dataset diversity, followed by label encoding to prepare data for model training.

The dataset was split using four train-test configurations (60:40, 70:30, 80:20, 90:10) with stratified sampling to maintain class balance. The CNN architecture comprised convolutional and pooling layers followed by fully connected layers with dropout regularization. The model was trained using the Adam optimizer and categorical cross-entropy loss, with early stopping and checkpointing to prevent overfitting and preserve the best-performing weights.

Model performance was evaluated using accuracy and confusion matrix analysis on independent test data. The optimal model was then converted into TensorFlow Lite format to support lightweight deployment. Finally, inference was performed on unseen images, and predictions were compared against ground-truth labels.

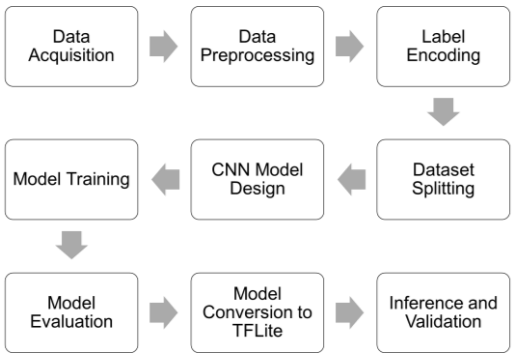

Figure 1. Research workflow for seagrass species classification using CNN

All experiments were executed on Google Colab using GPU acceleration. TensorFlow and Keras were applied for model development, while NumPy, Pandas, and PIL supported preprocessing and data manipulation. The complete workflow, implementation scripts, and model files are available in an open-access GitHub repository, ensuring reproducibility and scalability.

2.2. Data collection and dataset composition

The dataset used in this study consists of close-up photographs of individual seagrass leaves representing three tropical species: *H. uninervis, S. isoetifolium, and T. hemprichii.* All samples were collected from the coastal waters of Bintan Island, Indonesia. Unlike in-situ underwater imagery, each sample was manually extracted from its habitat, cleaned of debris, and photographed against a plain white background under controlled lighting conditions to minimize shadows and noise as shown in Figure 2. Each species was initially represented by 40 unique leaf images, resulting in a total of 120 original images across all classes. These images were stored in JPEG format and exhibited variations in orientation, size, shape, and texture—factors that contribute to species differentiation.

H. uninervis *S. isoetifolium* *T. hemprichii*

Figure 2. Example of each seagrass class data

2.3. Image augmentation and preprocessing

The limited raw dataset, consisting of only 40 images per species, was expanded using offline augmentation prior to model development. Each image underwent 16 transformation variations, resulting in 17 total versions per sample, including the original. This process increased the dataset to 680 images per class, yielding 2,040 images across the three species. Augmentation included geometric transformations

(flips, rotation, resize, zoom, shifts), photometric adjustments (brightness, contrast), and distortions (Gaussian noise and shear). Following augmentation, all images were imported into the Google Colab environment for preprocessing. Images were resized to 100×100 pixels, normalized to a pixel value range of [0, 1], and assigned numeric class labels for compatibility with the model's categorical loss function. The dataset was then split into training and validation subsets using stratified sampling while maintaining class balance across four train–test ratios: 60:40, 70:30, 80:20, and 90:10.

2.4. CNN model architecture

The CNN used in this study was developed to classify seagrass species based on single-leaf input images. The architecture was implemented using the Sequential API provided by TensorFlow's Keras library [17] and consists of convolutional layers for feature extraction followed by dense layers for final classification. The model accepts RGB images resized to 100×100×3. Feature learning is performed through three convolution–max pooling blocks with increasing filter depth (32, 64, and 128 filters) and rectified linear unit (ReLU) activations, each followed by a 2×2 max-pooling layer. The final convolution stage produces a 10×10×128 feature map, which is flattened before entering the classifier stage. The classifier consists of a Dense layer with 128 neurons, followed by a 0.4 dropout rate to mitigate overfitting. The output layer contains three neurons, representing the seagrass classes, with a softmax activation for probability-based prediction. In total, the model contains approximately 1.7 million trainable parameters.

2.5. Model training

The CNN model was trained using the TensorFlow-Keras framework in the Google Colab environment. Training was performed on the augmented dataset containing 680 images per class, resulting in a total of 2,040 images. Several train-validation split ratios were explored, namely 60:40, 70:30, 80:20, and 90:10, with class balance preserved through stratified sampling. The data was split into training and testing subsets under these four configurations, as shown in Table 1. Each composition maintained a total of 2,040 images but varied the allocation between training and testing sets.

The model was compiled using the Adam optimizer, selected for its adaptive learning rate and stable convergence behavior. The learning rate was set to 0.001, with standard hyperparameters: β_1 =0.9, β_2 =0.999, and ε =1e-7. The chosen loss function was sparse categorical crossentropy, suitable for multi-class classification with integer-encoded labels. Model performance during training was monitored using the accuracy metric.

Training was conducted in mini-batches of 32 samples, as defined by the batch_size parameter in the data generator. Although the maximum number of training epochs was set to 250, an early stopping strategy was implemented to dynamically terminate training. Two conditions were applied:

i) A custom callback that stopped training once validation accuracy reached 98%,
ii) A built-in EarlyStopping callback that monitored validation loss with a patience of 15 epochs, restoring the best weights upon termination.

This dual strategy allowed the model to avoid overfitting and reduced training time by halting the process when optimal performance was reached. A model checkpoint mechanism was also included to retain the weights corresponding to the highest validation accuracy observed during training.

Table 1. Distribution of training and testing images for different data split configurations

Data split	Training images	Testing images	Total
60:40	1,224	816	2,040
70:30	1,428	612	2,040
80:20	1,632	408	2,040
90:10	1,836	204	2,040

2.6. Model evaluation

Model evaluation was conducted using the validation datasets corresponding to each train-test split configuration (60:40, 70:30, 80:20, and 90:10). At the end of each training session, the model was evaluated using the model.evaluate() function, which returned the final validation loss and accuracy for the best-performing model checkpoint. The training history, which included loss and accuracy values for both training and validation sets across all epochs, was stored and visualized.

Line plots were generated, showing the progression of loss and accuracy throughout the training process. These visualizations offered insights into the model's convergence behavior, stability, and any indications of overfitting or underfitting. A bar chart was also created to summarize the final accuracy and loss values across all split configurations. This comparative visualization clarified how different training data

proportions affected the model's ability to generalize. All plots were saved and labeled according to their respective data split configurations, ensuring reproducibility and transparency in post-training analysis.

2.7. Model saving and inference

Upon completion of the training process for each data split configuration (60:40, 70:30, 80:20, and 90:10), the best-performing model from each configuration was saved in. tflite format and stored in a dedicated Google Drive directory. This ensured consistency during subsequent evaluation and enabled potential deployment without requiring retraining. Inference was performed using 11 previously unseen test images, which were not included in either the training or validation sets. These images were obtained from a publicly available GitHub repository and underwent the same preprocessing steps as the training data. Each saved model was reloaded and applied to classify the test images, and the predicted labels were compared against the ground truth to determine correctness. To facilitate performance evaluation, visual results were generated by displaying each test image alongside its prediction. Correct classifications were marked in green, whereas incorrect predictions were marked in red, and all outputs were archived for documentation purposes.

3. RESULTS AND DISCUSSION

3.1. Training and validation performance

The CNN model was evaluated under four train–validation split configurations: 60:40, 70:30, 80:20, and 90:10. The accuracy and loss curves provide insight into the model's learning stability and generalization behavior across these configurations. As shown in Figure 3, all accuracy curves increased steadily during the early epochs before reaching stabilization. The 90:10 split demonstrated the fastest convergence, achieving stable accuracy within four epochs, whereas the 60:40 split required 27 epochs and exhibited greater fluctuation, indicating slower learning stability.

Loss patterns further support these observations. The training and validation loss curves for the 90:10 and 80:20 configurations remained closely aligned as shown in Figure 3, suggesting effective learning with reduced overfitting. In contrast, the 60:40 and 70:30 splits displayed increasing divergence between training and validation loss as training progressed, indicating reduced generalization—likely caused by limited training sample diversity. Early stopping was employed to terminate training when no further improvement in validation metrics was observed as presented in Table 2. Although the training was initially set for 250 epochs, the mechanism halted training substantially earlier across all configurations—for example, at epoch 27 (60:40) and epoch 12 (90:10). Model weights were restored to the optimal checkpoint for each configuration, such as epoch 18 for the 80:20 split and epoch 12 for the 90:10 split, ensuring evaluation based on the best-performing state rather than the final training iteration. The rapid convergence seen in the 90:10 configuration can be attributed to the relatively larger training data proportion, which allowed the model to learn discriminative feature representations efficiently. Meanwhile, configurations with lower training data volumes required a longer optimization period and demonstrated greater sensitivity to overfitting.

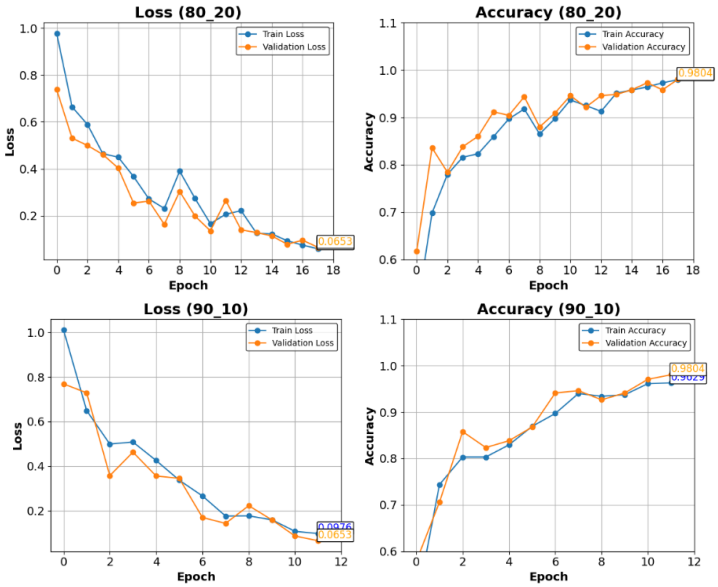

Figure 3. Accuracy plots for training and validation across the four data split configurations

Table 2. Early stopping efficiency

Data split	Epoch stopped	Best epoch restored
60:40	41	26
70:30	37	22
80:20	26	18
90:10	29	12

3.2. Model evaluation metrics

Model performance was quantitatively assessed on the validation datasets using the *model.evaluate()* function, which returned the final loss and accuracy values for each train-validation split configuration. Evaluation was performed on the checkpoint that yielded the highest validation performance, as determined by the early stopping mechanism discussed previously. As presented in Table 3, the 80:20 and 90:10 configurations achieved the best results, each with a validation accuracy of 98.04% and the lowest loss of 0.0653, indicating strong generalization and minimal overfitting. The 70:30 configuration followed closely with a validation accuracy of 95.59% and a loss of 0.1574, while the 60:40 configuration recorded the lowest performance, with a validation accuracy of 94.61%, and a loss of 0.1990.

In addition to accuracy and loss, other classification metrics such as precision, recall, and F1-score were also computed (weighted average across all classes) to provide a more comprehensive evaluation of model performance. Consistently, the 80:20 and 90:10 configurations achieved the highest precision (98.10% and 98.05%, respectively), recall (98.04% each), and F1-score (98.04% each), further confirming their superior performance. The 60:40 split yielded the lowest values across all metrics, with a precision of 94.62%, recall of 94.61%, and F1-score of 94.58%.

Table 3. Final validation accuracy and loss values for each train-validation split configuration

Data split	Accuracy	Loss	Precision (weighted avg)	Recall (weighted avg)	F1-score (weighted avg)
60:40	0.9461	0.1990	0.9462	0.9461	0.9458
70:30	0.9559	0.1547	0.9569	0.9559	0.9555
80:20	0.9804	0.0653	0.9810	0.9804	0.9804
90:10	0.9804	0.0653	0.9805	0.9804	0.9804

These results demonstrate a consistent trend: increasing the proportion of training data tends to enhance model performance. With more training examples, the CNN is able to learn more representative features and generalize better to unseen data. Although all configurations achieved over 94% accuracy, the gap of 3.43% between the 60:40 and the top-performing 80:20/90:10 configurations underscores the importance of training data volume in deep learning workflows. Notably, the 80:20 and 90:10 configurations not only delivered the best accuracy and lowest loss, but also achieved the most balanced precision-recall tradeoff. These findings emphasize that allocating a larger portion of data for training not only accelerates convergence but also improves model robustness and stability, reaffirming data composition as a critical factor in CNN performance.

3.3. Inference and error analysis

Inference was performed using 33 previously unseen images of seagrass leaves, comprising 11 samples for each species. Each image was preprocessed following the same procedure as during training, including resizing to 100×100 pixels and pixel normalization. The best-performing models from each data split configuration (60:40, 70:30, 80:20, and 90:10) were reloaded to generate predictions. Class labels were derived using the *argmax* function on the model outputs and compared to the ground truth. A visual summary of these predictions is shown in Figure 4, Green text indicates correct predictions, while red text marks misclassifications. Each group represents predictions from one training configuration.

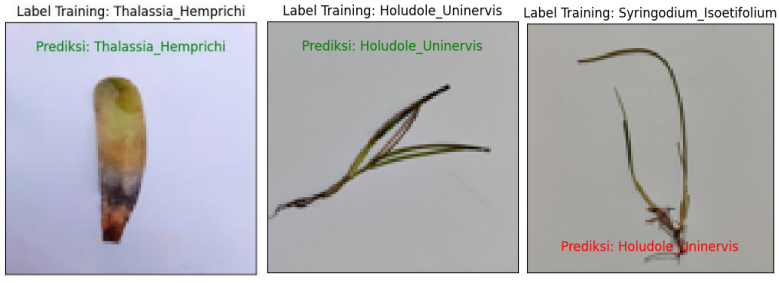

Figure 4. Inference results on unseen seagrass leaf images

As presented in Table 4, the model trained using the 60:40 configuration demonstrated the highest inference accuracy, correctly classifying 31 out of 33 test samples, resulting in performance of 94%. Models trained with the 80:20 and 90:10 splits followed closely, each achieving an accuracy of 85%, whereas the 70:30 configuration yielded the lowest performance at 82%. These outcomes diverge from the validation phase results, where the 80:20 and 90:10 configurations exhibited superior accuracy. This discepancy indicates that inference performance may be solely dependent on the size of the training set but also on the extent to which the training data distribution captures the characteristics of the unseen test instances.

Table 4. Inference accuracy based on the number of correct predictions for each data split configuration

	Precision				Recall				F1-score			
	60:40	70:30	80:20	90:10	60:40	70:30	80:20	90:10	60:40	70:30	80:20	90:10
H. uninervis	0.91	0.69	0.71	0.71	0.91	0.82	0.91	0.91	0.91	0.75	0.8	0.8
S. isoetifolium	0.92	0.82	0.9	0.9	0.92	0.75	0.75	0.75	0.92	0.78	0.82	0.82
T. hemprichii	1	1	1	1	1	0.91	0.91	0.91	1	0.95	0.95	0.95
accuracy									0.94	0.82	0.85	0.85
macro avg	0.94	0.84	0.87	0.87	0.94	0.83	0.86	0.86	0.94	0.83	0.86	0.86
weighted avg	0.94	0.84	0.87	0.87	0.94	0.82	0.85	0.85	0.94	0.83	0.86	0.86

A more detailed per-class analysis reveals distinct classification behavior. *T. hemprichi* consistently achieved perfect precision (1.00) across all configurations and maintained a high recall score (0.91), culminating in an F1-score of 0.95. The species' broad and distinguishable leaf morphology likely facilitated its consistent classification. Conversely, *S. isoetifolium* and *H. uninervis* – which share similar elongated leaf structures and coloration – were more susceptible to misclassification. Nevertheless, improvements were observed under specific configurations. In particular, the 60:40 split enabled *Syringodium* to achieve precision and recall values of 0.92, while *Halodule* obtained its highest F1-score of 0.91 under the same split.

Across all train–test configurations, the confusion matrices reveal consistent classification trends: *T. hemprichii* is reliably and almost perfectly identified in every split, indicating strong morphological distinctiveness and high model separability. In contrast, misclassifications occur primarily between *H. uninervis* and *S. isoetifolium*, whose visual similarities lead to reciprocal errors in all configurations. The 60:40 split provides the most balanced and stable performance, while the 70:30, 80:20, and 90:10 splits show slight increases in confusion between these two species, though overall accuracy remains high. Taken together, the results show that the model is robust across dataset proportions, with errors driven not by data volume but by the intrinsic similarity between specific species pairs. A visual confusion matrix on test data is shown in Figure 5.

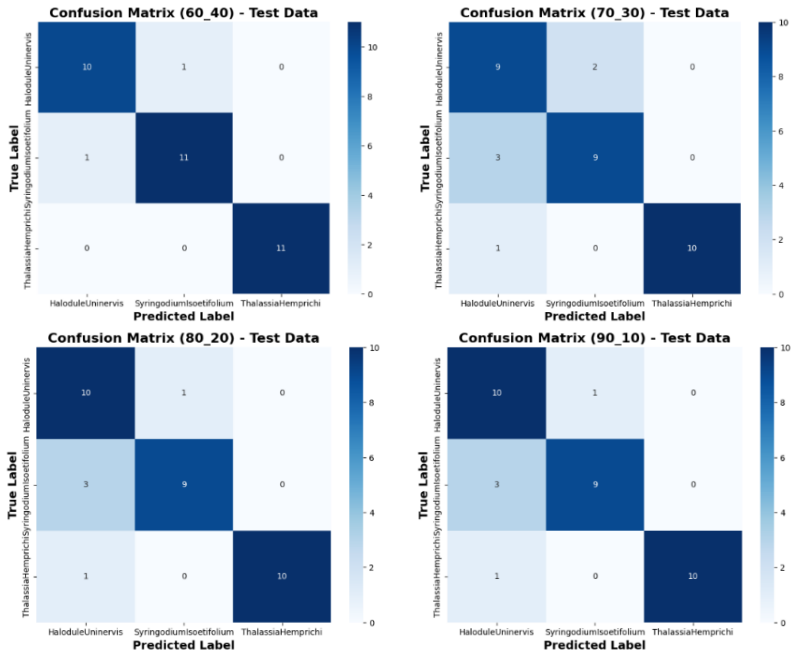

Figure 5. Confusion matrix

Across all three species, the 60:40 model consistently reported the highest metrics in terms of precision, recall, and F1-score, leading to the best macro and weighted averages at 94%. In contrast, the remaining configurations (70:30, 80:20, and 90:10) produced average macro and weighted scores around 87%. These findings suggest that while larger training datasets generally enhance validation accuracy, the representativeness of the training data plays a critical role in real-world inference performance.

3.4. Discussion and implications

The CNN model demonstrated fast convergence, stable training behavior, and high classification accuracy across all configurations. The integration of early stopping and dropout regularization played crucial roles in preventing overfitting, as evidenced by the close alignment of training and validation curves, especially in 80:20 and 90:10 configurations. These findings are consistent with best practices in CNN-based image classification [18]–[20]. The smooth progression of training loss and accuracy without oscillation or instability is consistent with well-regularized training behavior [21]. Despite limited data, the model effectively learned discriminative features, yielding reliable performance across configuration. This finding aligns with prior studies on CNN applications in marine and environmental domains [22]–[25], where high performance was achieved using moderate-size datasets. Inference results further confirm the model's robustness, with all configurations achieving accuracy ≥86.7%. These results are consistent with prior studies demonstrating high CNN performance in marine classification tasks, even with moderate dataset sizes [26]–[28].

Misclassifications, however, expose an important limitation in CNN performance for morphologically similar species. The consistent confusion between *H. uninervis* and *S. isoetifolium*—also observed in prior works [9] — highlights the need for either more diverse training samples or additional context (e.g., habitat, background, multi-angle views) to improve separability. On the other hand, the consistently correct classification of T. hemprichii reinforces the model's strength when applied to morphologically distinct classes [13], [27], [29], [30].

From an applied perspective, the proposed model demonstrates strong potential for supporting automated seagrass monitoring and biodiversity assessment in Bintan. Given its lightweight architecture, implementation on portable platform such as TensorFlow Lite is feasible and could support in-situ identification using mobile or embedded devices. Nonetheless, improvements are necessary for more challenging classification scenarios. Future directions may include integrating multi-angle or multi-scale imagery [26], adding contextual environmental features such as depth or location metadata [31], and leveraging ensemble or semi-supervised approaches to enhance generalization on real-world underwater data [9]. Taken together, the findings presented in this study demonstrate the model's practical viability and highlight the importance of data quality, species distinctiveness, and contextual augmentation as key considerations in advancing seagrass classification efforts using deep learning.

4. CONCLUSION

This study presented the development and performance evaluation of a CNN model for the classification of three seagrass species—*H. uninervis*, *S. isoetifolium*, and *T. hemprichii*—in Bintan, Indonesia. The model was trained using varying train-validation splits (60:40, 70:30, 80:20, and 90:10), demonstrating robust convergence and high classification accuracy across all configurations. The 90:10 configuration yielded the highest validation accuracy (98.53%) and lowest validation loss (0.08881), indicating strong model performance during training. Inference on unseen test images confirmed the model's generalization capability, achieving up to 86.7%. although the 80:20 and 90:10 splits excelled during the validation phase, the 60:40 split produced the most stable and reliable performance during inference, suggesting better generalization to real-world data. This highlights the importance of not only dataset size but also the representational quality and distribution of training samples. However, recurring misclassifications between *H. uninervis* and *S. isoetifolium* – species with sublte morphological differences – points to the limitations of single-view image input. In contrast, *T. hemprichii* was consistently classified with high precision, affirming the model's efficacy in identifying morphologically distinct species. Overall, the findings support the suitability of CNN-based approaches for automated seagrass classification in marine monitoring applications. Future improvements may focus on incorporating richer input modalities, increasing dataset diversity, and exploring model ensembles to further improve accuracy and reliability in complex underwater environments.

ACKNOWLEDGMENTS

The author gratefully acknowledges Farida for assistance during fieldwork in Bintan, particularly in seagrass sample collection and organization. Appreciation is also extended to local authorities and community members for their cooperation and access to research sites.

FUNDING INFORMATION

The author declares that this research was conducted independently and received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Nurul Hayaty	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
Hollanda Arief Kusuma		✓	✓	✓	✓			✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

The code, training data, test data, training and validation sample generations, prediction results, and figures for loss and accuracy used in this study are publicly available in the GitHub repository: https://github.com/hollandakusuma/LamunMachineLearningCNN. Python scripts and Jupyter notebooks (.ipynb) are also provided in the repository to facilitate reproducibility and further analysis.

REFERENCES

[1] L. C. C.- Unsworth and R. Unsworth, "A call for seagrass protection," *Science*, vol. 361, no. 6401, pp. 446–448, 2018.
[2] M. Gross, "Save our seagrasses," *Current Biology*, vol. 30, no. 16, pp. R905–R907, 2020, doi: 10.1016/j.cub.2020.07.081.
[3] C. J. Miller, S. J. Campbell, and S. Scudds, "Spatial variation of Zostera tasmanica morphology and structure across an environmental gradient," *Marine Ecology Progress Series*, vol. 304, pp. 45–53, 2005, doi: 10.3354/meps304045.
[4] N. T. Ha, M. M.- Harris, T. D. Pham, and I. Hawes, "A comparative assessment of ensemble-based machine learning and maximum likelihood methods for mapping seagrass using sentinel-2 imagery in Tauranga Harbor, New Zealand," *Remote Sensing*, vol. 12, no. 3, 2020, doi: 10.3390/rs12030355.
[5] S. Dahiya, R. Tyagi, and N. Gaba, "Comparison of ML classifiers for image data," *EasyChair Preprint No 3815*, 2020, [Online]. Available: https://easychair.org/publications/preprint/KnC4.
[6] M. Moniruzzaman, S. M. S. Islam, M. Bennamoun, and P. Lavery, "Deep learning on underwater marine object detection: a survey," in *Advanced Concepts for Intelligent Vision Systems (ACIVS 2017)*, Cham, Switzerland: Springer, 2017, pp. 150–160, doi: 10.1007/978-3-319-70353-4_13.
[7] M. S. Hossain, J. S. Bujang, M. H. Zakaria, and M. Hashim, "The application of remote sensing to seagrass ecosystems: an overview and future research prospects," *International Journal of Remote Sensing*, vol. 36, no. 1, pp. 61–114, 2015, doi: 10.1080/01431161.2014.990649.
[8] M. U. Gumusay, T. Bakirman, I. T. Kizilkaya, and N. O. Aykut, "A review of seagrass detection, mapping and monitoring applications using acoustic systems," *European Journal of Remote Sensing*, vol. 52, no. 1, pp. 1–29, 2019, doi: 10.1080/22797254.2018.1544838.
[9] Y. S. Deshmukh, R. G. Tambe, R. D. Chintamani, S. S. Bhosale, and S. Muthuraj, "Seagrass detection and classification from underwater images using deep transfer learning," *Advances in Nonlinear Variational Inequalities*, vol. 28, no. 1S, pp. 129–136, 2025, doi: 10.52783/anvi.v28.2233.
[10] H. Mohamed, K. Nadaoka, and T. Nakamura, "Semiautomated mapping of benthic habitats and seagrass species using a convolutional neural network framework in shallow water environments," *Remote Sensing*, vol. 12, no. 23, pp. 1–18, 2020, doi: 10.3390/rs12234002.
[11] D. Perez *et al.*, "Quantifying seagrass distribution in coastal water with deep learning models," *Remote Sensing*, vol. 12, no. 10, 2020, doi: 10.3390/rs12101581.
[12] M. K. Noman, S. M. J. Jalali, S. M. S. Islam, and P. Lavery, "Ofda-CNN: a novel metaheuristic algorithm-based deep CNN for multi-species seagrass classification," *SSRN*, 2023, doi: 10.2139/ssrn.4348793.
[13] M. A. A. Ozaeta, A. C. Fajardo, F. P. Brazas, and J. A. M. Cantal, "Deep learning approach for seagrass species classification," *2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, pp. 250–254, 2024, doi: 10.1109/GECOST60902.2024.10474987.

[14] G. Reus *et al.*, "Looking for seagrass: deep learning for visual coverage estimation," *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO), Kobe, Japan*, pp. 1–6, 2018, doi: 10.1109/OCEANSKOBE.2018.8559302.
[15] D. P. Rini and W. K. Sari, "Optimizing hyperparameters of CNN and DNN for emotion classification based on EEG signals," *International Journal on Information and Communication Technology (IJoICT)*, vol. 10, no. 1, pp. 1–12, 2024, doi: 10.21108/ijoict.v10i1.857.
[16] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[17] M. Abadi *et al.*, "TensorFlow: a system for large-scale machine learning," *OSDI'16: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, pp. 265–283, 2016.
[18] L. Prechelt, "Early stopping — but when?," in *Neural Networks: Tricks of the Trade*, Berlin, Germany: Springer, pp. 53–67, 2012, doi: 10.1007/978-3-642-35289-8_5.
[19] R. Caruana, S. Lawrence, and L. Giles, "Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping," *Advances in Neural Information Processing Systems*, 2001.
[20] G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, and N. Srivastava, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts: The MIT Press, 2016.
[22] Z. Cao, J. C. Principe, B. Ouyang, F. Dalgleish, and A. Vuorenkoski, "Marine animal classification using combined CNN and hand-designed image features," *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–6, 2016, doi: 10.23919/oceans.2015.7404375.
[23] E. L. White, H. Klinck, J. M. Bull, P. R. White, and D. Risch, "One size fits all? adaptation of trained CNNs to new marine acoustic environments," *Ecological Informatics*, vol. 78, 2023, doi: 10.1016/j.ecoinf.2023.102363.
[24] J. Freeman, "Content search within large environmental datasets using a convolution neural network," *Computers and Geosciences*, vol. 139, 2020, doi: 10.1016/j.cageo.2020.104479.
[25] E. C. Orenstein and O. Beijbom, "Transfer learning and deep feature extraction for planktonic image data sets," *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, vol. 6, no. 4, pp. 1082–1088, 2017, doi: 10.1109/WACV.2017.125.
[26] J. Elsäßer, L. Weihl, V. Cheplygina, and L. T. Nielsen, "SeagrassFinder: deep learning for eelgrass detection and coverage estimation in the wild," *Ecological Informatics*, vol. 90, 2025, doi: 10.1016/j.ecoinf.2025.103200.
[27] S. Raine, R. Marchant, P. Moghadam, F. Maire, B. Kettle, and B. Kusy, "Multi-species seagrass detection and classification from underwater images," *2020 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2020, doi: 10.1109/DICTA51227.2020.9363371.
[28] I. Fawwaz, Y. Yennimar, N. P. Dharsinni, and B. A. Wijaya, "The optimization of CNN algorithm using transfer learning for marine fauna classification," *sinkron*, vol. 8, no. 4, pp. 2236–2245, Oct. 2023, doi: 10.33395/sinkron.v8i4.12893.
[29] T. Eerola *et al.*, "Survey of automatic plankton image recognition: challenges, existing solutions and future perspectives," *Artificial Intelligence Review*, vol. 57, no. 5, Apr. 2024, doi: 10.1007/s10462-024-10745-y.
[30] J. Yang, M. Cai, X. Yang, and Z. Zhou, "Underwater image classification algorithm based on convolutional neural network and optimized extreme learning machine," *Journal of Marine Science and Engineering*, vol. 10, no. 12, Dec. 2022, doi: 10.3390/jmse10121841.
[31] N. Abid *et al.*, "Seagrass classification using unsupervised curriculum learning (UCL)," *Ecological Informatics*, vol. 83, Nov. 2024, doi: 10.1016/j.ecoinf.2024.102804.

BIOGRAPHIES OF AUTHORS

Nurul Hayaty is a lecturer in the Department of Informatics Engineering at Universitas Maritim Raja Ali Haji (UMRAH), Tanjungpinang, Indonesia. She holds a Bachelor's degree in Computer Engineering and a Master's degree in Computer Science (M.Cs). Her research interests encompass artificial intelligence, image processing, machine learning, and data mining, with a particular focus on applications in environmental informatics and smart systems. She has contributed to various studies, including rainfall prediction using support vector machines and neural network models for mangrove species classification. In the current study, she led the conceptualization, data curation, model development, and manuscript drafting processes. Her work continues to bridge computational methods with practical applications in environmental monitoring and decision support systems. She can be contacted at email: nurul.hayaty@umrah.ac.id.

Hollanda Arief Kusuma is a lecturer at the Department of Electrical Engineering, Universitas Maritim Raja Ali Haji (UMRAH), Tanjungpinang, Indonesia. He holds a Bachelor's and Master's degree from IPB University (Institut Pertanian Bogor), specializing in Marine Science and Technology. His academic and research interests lie in marine technology, geospatial information systems, underwater robotics, and remote sensing applications for coastal environments. He is actively involved in the development of open-source oceanographic instruments and the integration of internet of things (IoT) technologies for real-time coastal monitoring. Over the past few years, he has led and participated in numerous research projects focusing on tidal pattern analysis, water quality assessment, and environmental data collection using embedded systems. In this study, he contributed to project supervision, methodology development, data validation, and manuscript review and editing. He can be contacted at email: hollandakusuma@umrah.ac.id.