Grenze International Journal of Engineering and Technology, January Issue

Sea Shell Classification

Aparna R[1], Pradeep L Ashtkar[2], Rajath R[3], Nishanth Kumar K S[4] and Sharath K M[5]
[1]Department of Information Science and Engineering, Siddaganga Institute of Technology, Tumakuru, India
Email: raparna@sit.ac.in
[2-5]Department of Information Science and Engineering, Siddaganga Institute of Technology, Tumakuru, India
Email: {pradeepla7976@gmail.com, rajathr2003@gmail.com, nishanthkumar188@gmail.com, shrathsharath52158@gmail.com}

***Abstract*—Seashell species classification plays a pivotal role in marine biology, ecology, and conservation efforts, facilitating the understanding of marine ecosystem diversity, monitoring changes, and forming conservation strategies. However, the complexity of seashell morphologies poses challenges, making manual identification laborious and hindering large-scale studies. Developing user-friendly, automated tools like image recognition algorithms could streamline classification processes, enhancing efficiency and enabling broader contributions to marine biodiversity understanding and conservation. This involves preprocessing a shell dataset, resizing images, assigning scientific labels, and extracting color, shape, and texture features. These features are crucial for classification, aiding in identifying various species within categories such as gastropods, bivalves, and cephalopods, ultimately contributing to the validation and improvement of classification algorithms for seashell species.**

***Keywords*—Sea shell, Convolutional Neural Network, k-nearest neighbour, Resnet.**

I. INTRODUCTION

Seashells are the protective outer layer formed by marine mollusks, composed mainly of calcium carbonate secreted by the mantle of the animal. Beyond mere protection, these shells are integral for mollusks in navigating their environment, offering structural support and assisting in buoyancy control. Their wide array of shapes, sizes, colors, and patterns mirrors the vast diversity of mollusk species found across oceans worldwide. For instance, the spiral conch shells and smooth, glossy cowrie shells exemplify this diversity. Scallop shells feature fan-like shapes, while nautilus shells are renowned for their intricate spirals and internal chambers. Abalone shells, with their iridescent shimmer, are highly prized in both art and jewelry. Seashells have captivated human imagination for centuries, serving as subjects for artistic expression, coveted adornments, and objects of scientific study. They serve as poignant reminders of the extraordinary beauty and biodiversity concealed beneath the ocean's surface.

Seashells, crafted from calcium carbonate secretions, serve as vital shields for marine life, providing protection against predators and environmental challenges while offering crucial structural support and aiding in buoyancy regulation. Their diverse characteristics, from spiral conchs to smooth cowries, mirror the rich diversity of mollusk species worldwide.

Scallop shells and nautilus shells showcase nature's creativity with their unique shapes and patterns. Abalone shells, prized for their iridescence, hold cultural and artistic significance.

Grenze ID: 01.GIJET.11.1.525

Throughout history, seashells have inspired human creativity and scientific inquiry, serving as reminders of the ocean's hidden beauty and biodiversity. They call us to explore and protect marine ecosystems for future generations.

The paper is organized as follows: Section II highlights the literature survey, In Section III system overview will be discussed, Section IV implementation details are emphasized and Section VI concludes the paper.

II. Related Works

Authors in paper [1] explore the comprehensive exploration of image classification methodologies, emphasizing the utilization of Convolutional Neural Networks (CNN) and Keras for automating the classification process. It underscores the significance of extracting discriminative features from different layers of the CNN to enhance classification accuracy. The work aims to implement and optimize the CNN model using Keras on the CIFAR10 dataset, utilizing stochastic gradient descent (SGD) for optimization. Platform independent techniques are employed, with Google Colab serving as the primary platform.

The paper [2] explores a comparative study of feature extraction methods in image classification, assessing their performance across different classifiers and classification scenarios. The research aims to identify the most effective technique for improving classification accuracy and providing implicit classification data.

Paper [3] highlights the historical evolution and practical applications of Principal Components Analysis (PCA), emphasizing its delayed widespread adoption until the 1960s due to computational challenges. With the emergence of computers, the extensive utilization of PCA and other multivariate statistical methods became feasible. The authors address the need for precise numerical methods, particularly in calculating eigenvalues and eigenvectors, to ensure accuracy in technical applications. Furthermore, the increasing applications of PCA in Social Sciences led to enhanced interpret-ability of non-observable variables.

In paper [4], Convolutional Neural Networks (CNNs) are employed to classify handwritten digits from the MNIST dataset. By training and testing CNNs on these grayscale images, an impressive accuracy rate is achieved. However, the computational time required for processing these small grayscale images is notably high compared to standard JPEG images.

In paper [5], the abstract succinctly summarizes the transformative impact of deep learning in various domains, including speech recognition, visual object recognition, object detection, drug discovery, and genomics. Deep learning models, composed of multi preprocessing layers, learn representations of data with multiple levels of abstraction, significantly advancing state-of-the-art performance.

The work in paper [6] introduces a novel image classification approach, derived from the kNN strategy, tailored for images described by local features. Unlike traditional methods that compare entire images, the proposed approach assesses similarity between local features within the training set. This enables the utilization of global information present in the dataset and allows for more efficient and effective classification strategies, including the exploration of local feature cleaning techniques.

The paper [7] provides a comprehensive overview of a tutorial aimed at elucidating fundamental concepts of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks. Addressing a gap in existing literature, the tutorial offers a detailed derivation of the canonical RNN formulation from differential equations and introduces a precise statement for the RNN unrolling technique. It also discusses the challenges associated with training standard RNNs and proposes the transformation of RNNs into Vanilla LSTM networks to overcome these hurdles. Through logical arguments and detailed equations, the tutorial presents the LSTM system and its constituent entities, emphasizing ease of understanding.

The study in [8] highlights the accessibility of pre-trained ResNet-50 models on ImageNet and promotes transfer learning as a key strategy for efficiently addressing new problems. It underscores the effectiveness of retraining ResNet-50 over developing novel networks, emphasizing its versatility in diverse problem-solving scenarios.

III. System Overview

The system architecture is centered around the processing and analysis of seashell images to identify their species in real-time. It is depicted in Fig 1. The process initiates with the acquisition of an input image of the seashell, which subsequently undergoes data pre-processing to enhance its quality and prepare it for analysis. Following pre-processing, the image is input into a model that has been trained to recognize seashell species based on their color, texture, shape and other features.

The model scrutinizes these features to make real-time predictions concerning the species of the seashell. The outcomes of the analysis are then exhibited on a user-friendly interface.

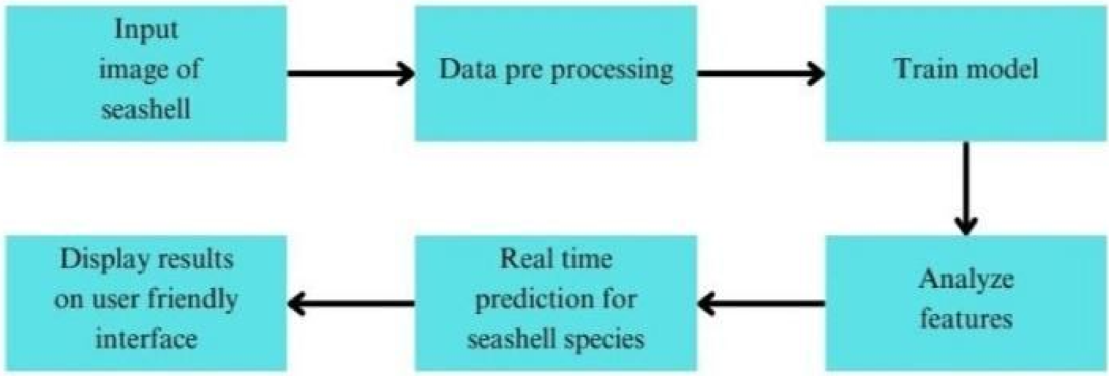

Fig. 1. The flow diagram of proposed system

Input Image of seashell: The seashell image can be captured either using a camera or selected from existing files. For mobile devices, users will be prompted to choose between using the camera or selecting from files. However, for desktop versions, image selection is limited to existing files, as many desktop cameras may have lower resolutions. Data pre-processing: The shell dataset comprises 8 to 25 species. Each sample consists of distinct views captured in JPG format. Every shell image is tagged with its scientific name and assigned a unique number. Subsequently, these images are resized to 300*400 pixels for further feature extraction and processing.
Train model: An already trained model is deployed on the server, providing seamless and instant predictions for seashell species. This model has been trained to recognize seashell species based on their visual features, enabling efficient and accurate analysis of input images. By leveraging this pre-trained model, the system can swiftly process incoming seashell images, making real-time predictions without the need for extensive computational resources on the user's end. Analyze features: In the feature analysis stage, various visual features are extracted from the seashell images using the existing model. These features include color, texture, and shape characteristics, which are crucial for identifying and classifying seashell species. The model analyzes these features to determine patterns and similarities that are indicative of specific species. By extracting and analyzing these features, the model can make accurate predictions about the species of the seashell.
Real time prediction for seashell species: Once the prediction is generated, the data is transformed and sent to the client browser. This real-time prediction capability allows users to receive immediate feedback on the species of the seashell, enhancing the overall user experience and usability of the system.
Display results on user-friendly interface: The predicted seashell species names are showcased on the client's browser, ensuring a seamless user experience. The interface is designed to be intuitive and easy to navigate, providing users with clear and concise information about the identified species. This user-friendly interface enhances the accessibility of the system, allowing users to quickly and easily access the results of the sea shell prediction.

IV. Implementation

The implementation of seashell species identification system covers data pre-processing steps to enhance image quality, followed by a description of the Convolutional Neural Network (CNN) architecture used for classification. The training process and model optimization methods are discussed, along with the implementation of the Flask framework for the web application. Various Python modules and libraries are highlighted for their roles in enhancing system functionality and performance. This section aims to provide a comprehensive overview of technical aspects, offering insights into the development and operation of our seashell species classification system.
Data Pre-processing: Initially, The system began by gathering and restructuring the shell images. The shell dataset comprises 8 species. Each sample consists of distinct views captured in JPG format. Every shell image is tagged with its scientific name and assigned a unique number. Subsequently, these images are resized to 300*400 pixels for further feature extraction and processing.
Feature extraction: The visual features like colour, shape and texture are extracted from the sample as depicted in Figure 2 and are described in further section.
Colour feature extraction: Color is a vital aspect of seashell identification and classification. To extract color features from seashell images, the RGB image is separated into its Red, Green, and Blue channels. This

division allows the user to examine the distribution of colors in each channel independently. Following this, a flood-fill algorithm is applied to generate a mask that isolates the seashell from its background, effectively filtering out extraneous color information.

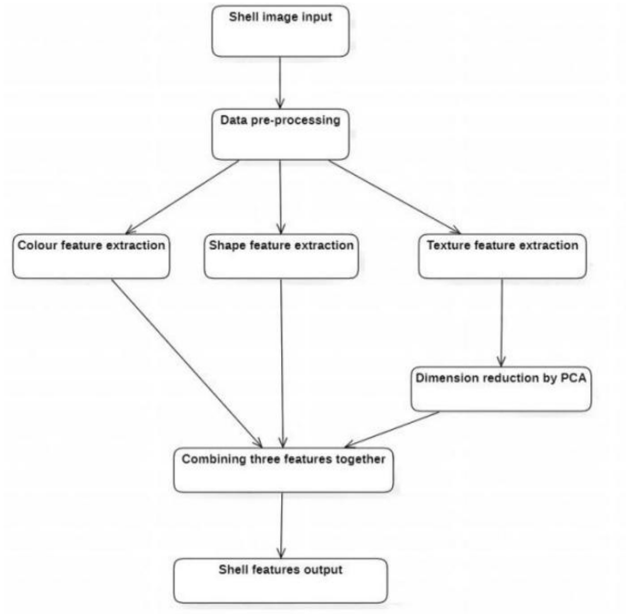

Fig. 2. Feature extraction flow diagram

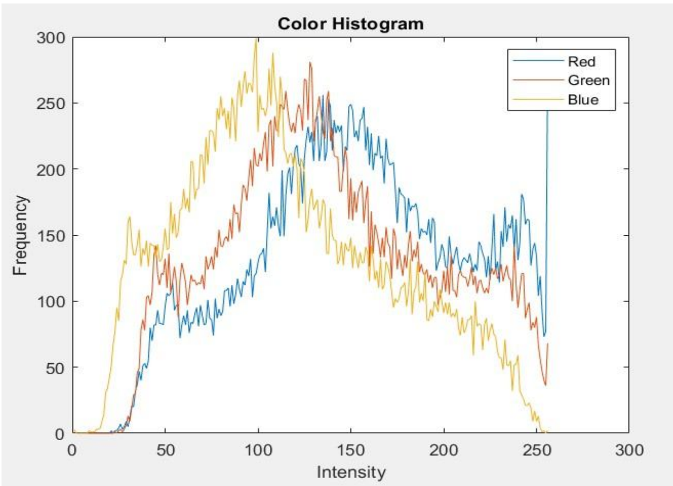

Fig. 3. Histogram

Histograms are essential in our seashell species identification system, providing insights into color distribution within images. Analyzing histograms of Red, Green, and Blue channels helps extract dominant colors, crucial for species differentiation. They aid in noise reduction and background subtraction, ensuring only relevant color data is considered. Histograms are instrumental in understanding seashell color characteristics, enhancing the accuracy and efficiency of the identification system. With the seashell isolated, histograms are computed for each color channel, representing the frequency of different color intensities in the image. These histograms provide valuable insights into the predominant colors present in the seashell. To enhance the accuracy of the analysis, the background pixels are subtracted from the histograms, ensuring that only the seashell's colors are considered.

Finally, the histogram values from the three color channels are combined into a single feature vector, encapsulating the seashell's color characteristics. This feature vector can then be utilized for diverse applications, such as species classification or comparison with other seashells. By extracting color features from seashell images, a deeper understanding of their unique color patterns was possible, facilitating their identification and furthering the study of these fascinating marine specimens. The histogram for species is as in the Figure 3.

Shape feature extraction: In order extract shape features from images of seashells, the color has to be read. Then a binary mask using a flood-fill algorithm is created, isolating the seashell from its background. After obtaining the binary image, the script labels the connected components (blobs) in the image and identifies the blob corresponding to the seashell. The centroid of this blob is calculated and used as a reference point for further analysis. Next, Identify the boundary of the seashell using the bwboundaries. It calculates the angle and distance of each boundary point relative to the centroid of the seashell. The angles and distances are plotted for visualization, providing insights into the shape characteristics of the seashell. Additionally, processes the boundary points to ensure they are in a monotonically increasing order of angles, which is essential for accurate feature extraction. In the final step, calculate the distances at every 5-degree interval around the centroid of the seashell by which we get contour distance. These distances serve as shape features, capturing the shell's outline in a concise and informative manner. It is ensured that the resulting feature vector contains 72 elements, padding with zeros if necessary. These shape features can be used for further analysis and classification of seashell species, contributing to a comprehensive understanding of their morphology and characteristics.

Texture Feature Extraction: Extraction of texture features from images of seashells, a crucial step in the analysis and classification of seashell species. It begins by reading in an image of a sea shell and converting it to facilitate further processing. A binary mask is then created to isolate the seashell from the background, and an active contour algorithm is applied to refine the segmentation, ensuring accurate boundary detection.

Next, the function identifies the seashell blob with in the image and calculates its centroid, providing a reference point for subsequent analysis. It then crops the seashell image into smaller 20x20-pixel samples, ensuring each sample contains a significant portion of the seashell for meaningful texture analysis. The ses amples are used to extract texture features using a Gabor filter, which analyzes the texture patterns at multiple orientations and scales.

The Gabor filter which is a filter widely used for analysis of texture and recognition. Here a diversion of this filter is used which is specified in Equations 1 and 2. This filter is applied with varying wavelengths to capture texture features at different scales. For each wavelength, the function extracts texture features such as average value, energy, and entropy, providing a comprehensive representation of the texture characteristics of the seashell. These features are computed for each sample and concatenated to form a feature vector, which can be used for further analysis and classification of sea shell species.

In summary, the texture extract function plays a crucial role in the analysis of seashell images by extracting meaningful texture features. By combining segmentation, cropping, and Gabor filtering techniques, the function enables the extraction of texture features at various scales, providing valuable insights into the texture characteristics of seashells for species classification and identification.

$$f(x,y,\omega,\sigma_x\sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} exp[(\frac{-1}{2}(\frac{x}{\sigma_x})^2(\frac{y}{\sigma_y})^2) + i\omega(x\cos\theta + y\sin\theta)] \longrightarrow \text{Eq. 1}$$

$$\gamma(x,y) = I(x,y) \times f(x,y,\omega,\sigma_x,\sigma_y) \longrightarrow \text{Eq. 2}$$

In Eq 1. σ represents spacial width, ω represents frequency, and θ is orientation and I(x, y) in Eq. 2 represents binary image of a seashell texture and γ(x, y) represents Gabor filter results after filtering

Combining three features together: After extracting color, shape, and texture features from seashell images, the next step is to assemble these features for further analysis. In this process, the extracted color features are loaded, followed by the shape and texture features. The texture features are then subjected to Principal Component Analysis (PCA) to reduce their dimensionality.

PCA identifies the most influential components in the texture features and projects them into a lower-dimensional space, capturing the most important variations in the data. In this case, the top 10 influential components are selected to represent the texture features.

The assembled data consists of the color, shape, and reduced-dimensional texture features, all combined into a single matrix. Before finalizing, standardization is applied to the features to ensure that each feature contributes equally to the analysis, regardless of its original scale. Finally, the complete dataset is saved for further use in modeling and analysis. This process of assembling and preparing the feature dataset is crucial for the success of any subsequent analysis or modeling task. By combining the color, shape, and texture features, the dataset provides a comprehensive representation of the seashell images, capturing both their visual appearance and structural characteristics. The application of PCA to the texture features further enhances the dataset by reducing its dimensionality, making it more manageable and computationally efficient for subsequent analysis. Overall,

this step plays a fundamental role in preparing the data for classification, clustering, or any other analysis aimed at understanding and categorizing seashell species.
Convolutional Neural Networks implementation: Convolutional Neural Networks (CNNs) are a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs are composed of multiple layers, each serving a specific purpose in feature extraction and classification. The typical architecture of a CNN includes convolutional layers, pooling layers, and fully connected layers.
In a CNN, the convolutional layers are responsible for detecting various features in the input image. Each convolutional layer consists of multiple filters that convolve across the input image, producing feature maps that highlight different aspects of the image, such as edges, textures, or patterns. These feature maps are then passed through activation functions, such as RELU, to introduce non- linearity.
Pooling layers are used to reduce the spatial dimensions of the feature maps while retaining important information. Common pooling operations include max pooling, which selects the maximum value from a set of values, and average pooling, which calculates the average value. Pooling helps to make the learned features more robust and invariant to small changes in the input.
Fully connected layers, also known as dense layers, are typically found towards the end of a CNN. These layers take the flattened output of the previous layers and perform classification based on the extracted features. Each neuron in a fully connected layer is connected to every neuron in the previous layer, allowing the network to learn complex relationships between features, Multiple layers will be as depicted in Figure 4.
To incorporate the extraction of color, shape, and texture features into a CNN, additional layers can be added in between the standard convolutional and pooling layers. For example, specific layers can be designed to extract color features using techniques like histogram analysis or color channel separation. Shape features can be extracted using contour detection algorithms or shape descriptor calculations. Texture features can be extracted using filters like Gabor filters or by analyzing the spatial distribution of pixel intensities. Integrating these feature extraction techniques into the CNN architecture enhances its ability to learn discriminative features for seashell species classification.
K-nearest neighbors implementation: The K-Nearest Neighbors (KNN) algorithm is a versatile tool in machine learning, ideal for classifying shell data. Operating on a non- parametric, instance-based approach, it assigns a class label to a data point by majority vote among its k-nearest neighbors in the feature space. KNN’s simplicity and effectiveness suit datasets with clear decision boundaries and few classes. The process involves partitioning the dataset into training and test sets, training the KNN model on extracted features, and iteratively adjusting the number of neighbors, k, to optimize accuracy. Evaluation metrics like accuracy and F-score gauge performance across different parameter settings and data partitions. The top-performing KNN model is saved for future use, ensuring reproducibility and facilitating further analysis on new shell data.
Residual Network implementation: ResNet-50, a groundbreaking convolutional neural network, drives our seashell species classification efforts. Developed by Microsoft, ResNet-50’s 50 layers and skip connections resolve gradient issues, enabling deep network training with superior performance. In our approach, ResNet-50 extracts high-level features from seashell images, including color, shape, and texture, merging them into a comprehensive vector.
These features feed into a classification layer to identify seashell species. Through rigorous training and fine-tuning on seashell datasets, ResNet-50 elevates our classification system to achieve unmatched accuracy and robustness.

V. RESULT ANALYSIS

Test train split: Here 80/20 test train split is used because when the testing set decreases, the model’s accuracy tends to increase, as it is better evaluated on previously unseen and more representative data of real-world scenarios. Using an 80%-20% split for training and testing strikes a balance by providing enough data for training effectively while offering a substantial amount for evaluating generalization capabilities. This split leads to improved model performance due to better generalization on new, unseen samples. Adjusting the training and testing set proportions played a crucial role in improving the model’s accuracy, with the 80%-20% split providing the most representative evaluation, resulting in the high accuracy, additionally, the 80:20 split helps in detecting potential overfitting issues.
Result analysis for CNN model: Figure 4 presents performance metrics for 8-species classification task, demonstrating an accuracy of 88%, precision of 88%, recall of 88%, and an F1 score of 87%. The presented confusion matrix in Figure 5 illustrates the performance metrics for 8-species classification task. Figure 6 presents performance metrics for 25-species classification task, demonstrating an accuracy of 67%, precision of

68%, recall of 67%, and an F1-score of 67%. The presented confusion matrix in Figure 7 illustrates the performance metrics for 25-species classification task.

Result analysis for KNN model: Figure 8 presents performance metrics for 8-species classification task, demonstrating an accuracy of 83%, precision of 84%, recall of 83%, and an F1-score of 83%. The presented confusion matrix in Figure 9 illustrates the performance metrics for 8-species classification task. Figure 10 presents performance metrics for 25-species classification task, demonstrating an accuracy of 55%, precision of 55%, recall of 55%, and an F1 score of 54%. The presented confusion matrix in Figure 11 illustrates the performance metrics for 25-species classification task.

Result analysis for ResNet model: In Resnet model, the performance metrics for 8-species classification task, demonstrates an accuracy of 82%, precision of 80%, recall of 81%, and an F1-score of 80%.

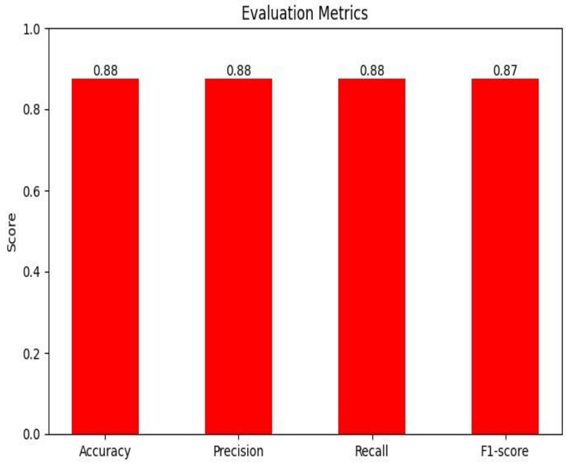

Fig. 4. CNN parameters performance For 8-species Classification

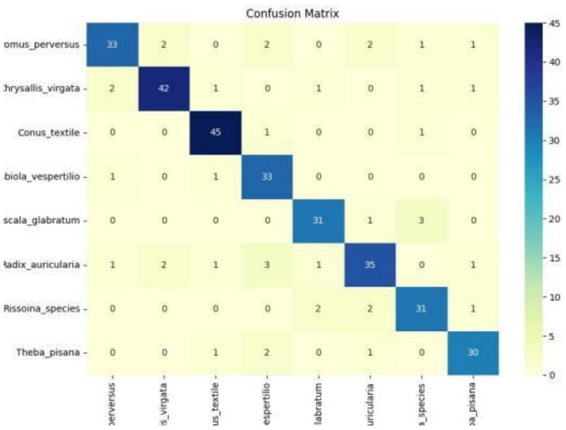

Fig. 5. Confusion Matrix for 8 species classification

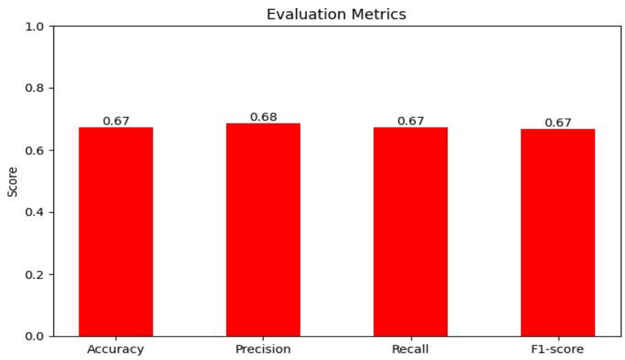

Fig. 6. CNN parameters performance for 25-species Classification

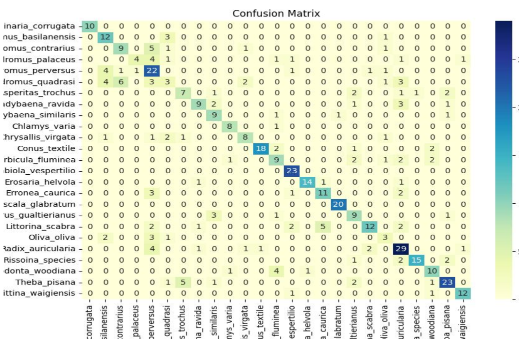

Fig. 7 Confusion Matrix for 25-species classification

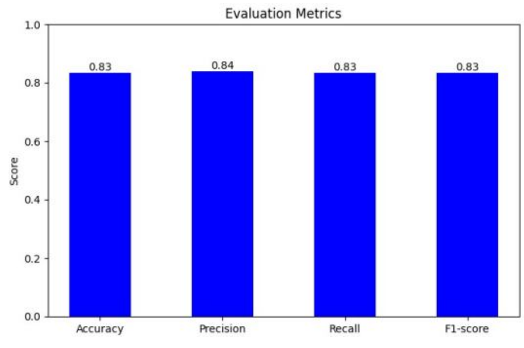

Fig. 8. KNN parameters performance For 8-species Classification

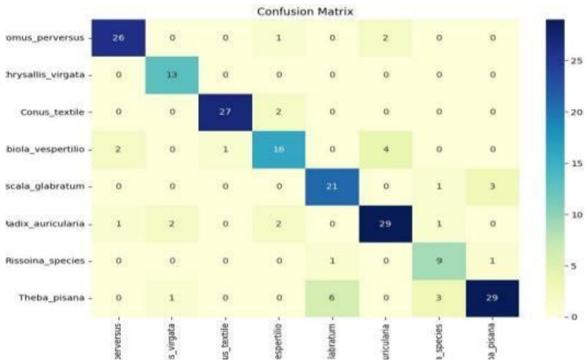

Fig. 9 Confusion Matrix for 8-species classification

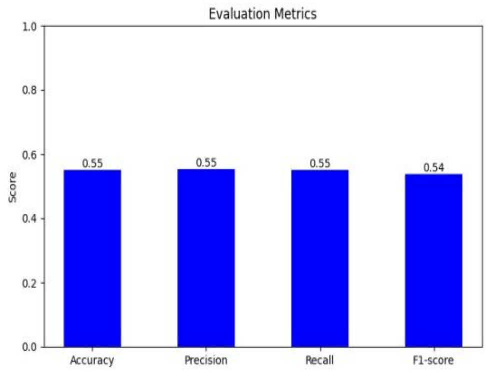

Fig. 10. KNN parameters performance for 25-species Classification

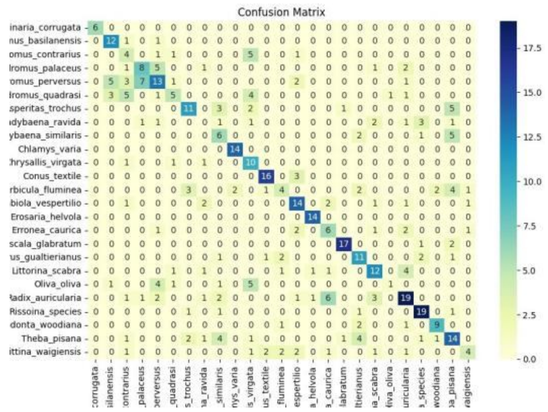

Fig. 11 Confusion Matrix for 25-species classification

V. CONCLUSION

In conclusion, the paper proposes a comprehensive system for sea shell classification utilizing image analysis techniques. By integrating color, shape, and texture features extracted from shell images, this system offers a robust framework for accurate classification of sea shell species. Leveraging methods such as Centroid Contour Distance for shape extraction and Gabor filter for texture analysis enhances the discriminative power of the classification model. Through meticulous preprocessing steps, including grayscale conversion and background masking, the effectiveness of feature extraction is ensured. The proposed system demonstrates promising results in distinguishing between different shell species, offering potential applications in marine biodiversity research and conservation efforts. Further refinement and validation of the system holds promise for enhancing its performance and applicability in real-world.

REFERENCES

[1] Akhil Kumar, Kartik P Singh, Sawan Kumar, and L Vetrivendan. Image classification in python using keras. In Proceedings of Data Analytics and Management: ICDAM 2021, Volume 1, pages 541–556. Springer, 2022.

[2] Seyyid Ahmed Medjahed. A comparative study of feature extraction methods in images classification. International journal of image, graphics and signal processing, 7(3):16, 2015.

[3] AN Dar. Principal component analysis (pca)(using eigen decomposi- tion). Gsj, 9(7):240–252, 2021.

[4] Muthukrishnan Ramprasath, M Vijay Anand, and Shanmugasun- daram Hariharan. Image classification using convolutional neural networks. International Journal of Pure and Applied Mathematics, 119(17):1307–1319, 2018.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015.

[6] Giuseppe Amato and Fabrizio Falchi. knn based image classification relying on local feature similarity. In Proceedings of the Third In- ternational Conference on Similarity Search and Applications, pages 101–108, 2010.

[7] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenom- ena, 404:132306, 2020.

[8] Brett Koonce and B Efficientnet Koonce. Convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization. Springer, 2021.