Coral Health Detection Model Using SVM

Introduction:

This project aims to classify coral images as either "healthy" or "bleached" using machine learning techniques. The model uses a Support Vector Machine (SVM) classifier with feature extraction from a pre-trained VGG16 network to distinguish between the two categories. The goal is to improve the accuracy of predictions on coral health using image-based features.

Technologies and Libraries Used:

- **Python**: The core programming language used for developing this model.
- **OpenCV**: Used for loading, processing, and resizing images.
- **scikit-learn**: Utilized for the machine learning tasks, such as SVM classification, data splitting, and evaluation.
- **TensorFlow/Keras**: Used for the pre-trained VGG16 model, which extracts meaningful features from the images.
- **Matplotlib**: For visualizing the results, including accuracy and classification reports.
- **Seaborn**: For enhanced data visualization, particularly in generating pie charts.

Model Workflow

1. Data Collection:

- The dataset consists of images of healthy and bleached corals stored in separate folders.
- Each image is labeled either "healthy" (0) or "bleached" (1).

2. Data Preprocessing:

- The images are resized to **224x224 pixels** to match the input size expected by the VGG16 model.
- The color format is converted from **BGR (used by OpenCV) to RGB (used by Keras)**.
- Preprocessing is applied using preprocess_input from Keras to prepare images for the VGG16 model.

3. Feature Extraction:

- A pre-trained **VGG16 model** is used to act as a feature extractor.
- The model is used to extract features from the images, and these features are then flattened to **one-dimensional** vectors.

4. Data Normalization:

- The extracted features are normalized using **StandardScaler** to improve the performance of the SVM classifier.

5. Model Training:

- The dataset is split into training (80%) and testing (20%) sets using **train_test_split** from **scikit-learn**.
- An **SVM** classifier is trained using the **linear kernel (C=0.1)**, and the model is fitted on the training features.

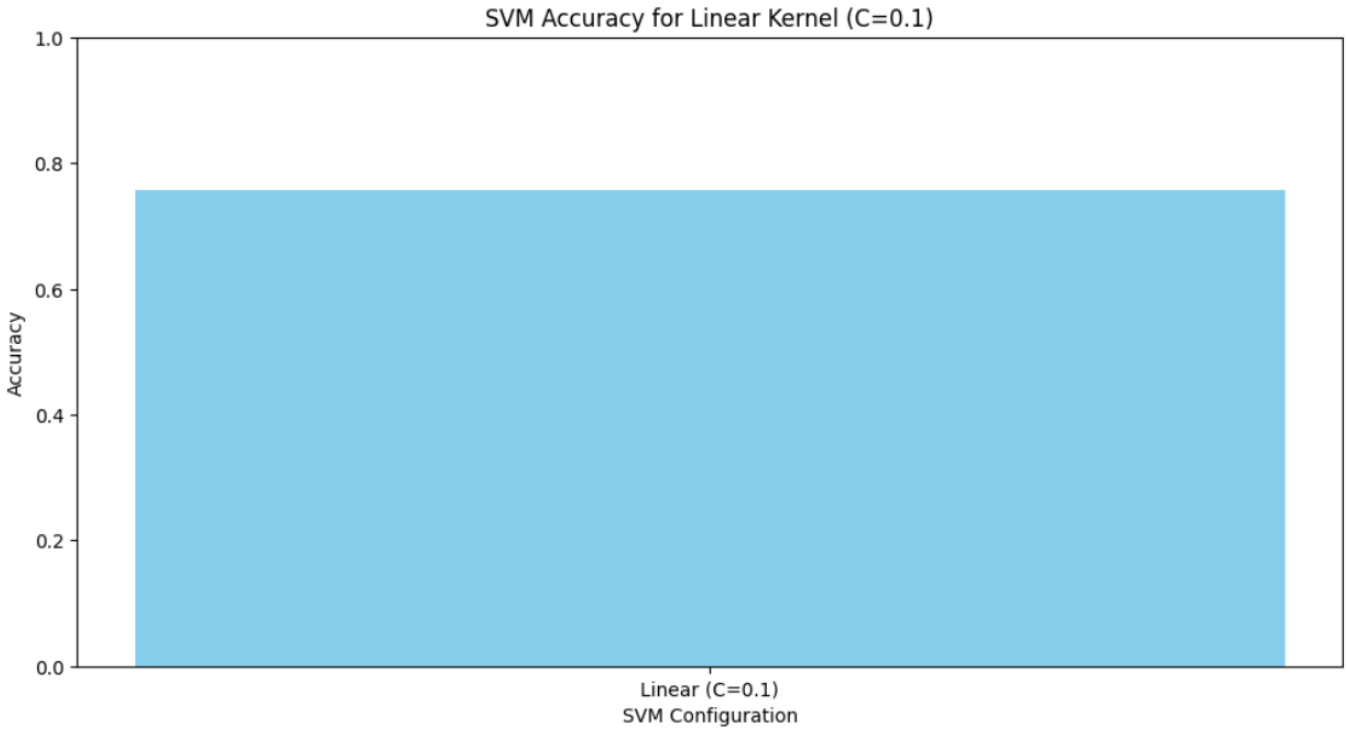
6. Model Evaluation:

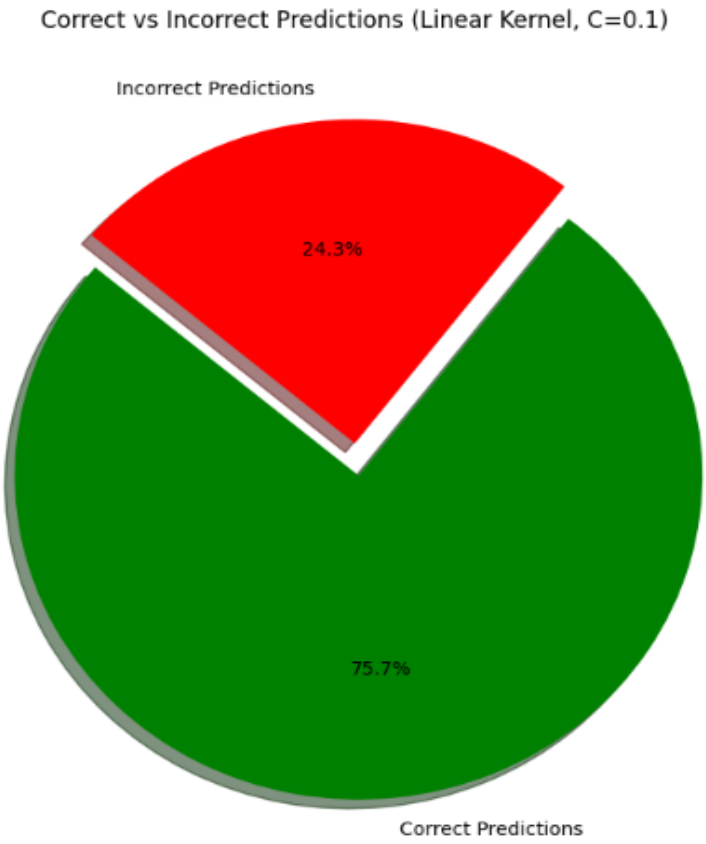
- The trained SVM model is evaluated on the test set, and the accuracy is calculated.
- A **classification report** is generated using classification_report from scikit-learn to show precision, recall, and F1-score for both classes.

```
'degree': 3, 'gamma': 'scale', 'kernel': 'linear'
Config: {'kernel': 'linear', 'C': 0.1}, Accuracy: 0.76, Time: 34.55 seconds
              precision    recall  f1-score   support

           0       0.71      0.78      0.74        83
           1       0.81      0.74      0.77       102

    accuracy                           0.76       185
   macro avg       0.76      0.76      0.76       185
weighted avg       0.76      0.76      0.76       185
```

7. Visualization:

- The accuracy of the model for the linear kernel is visualized using a **bar chart**.
- A **pie chart** is generated to show the proportion of correct and incorrect predictions.
- The **execution time** for the linear kernel model is plotted.





8. Results:

- **Accuracy**: The model achieved an accuracy of **76%** (75%-79%) with the linear kernel (C=0.1). The model completed the task in approximately **34.55 seconds** (28sec - 40sec).
- **Support Vectors**: The model used a total of **185 support vectors**, with the number of support vectors for each class (healthy and bleached).

Classification Report

Bleached (Class 1):	**Healthy (Class 0)**:
Precision: 0.81	Precision: 0.71
Recall: 0.74	Recall: 0.78
F1-score: 0.77	F1-score: 0.74
Support: 102	Support: 83

NOTE: we have tried to config which parameter will give the best output, the parameters are Best Kernel and Parameters:
{'C': 10, 'degree': 2, 'gamma': 'scale', 'kernel': 'rbf'}

```
# GridSearchCV
param_grid = {
    'kernel': ['linear', 'poly', 'rbf'],
    'C': [0.1, 1, 10],  # Regularization parameter
    'gamma': ['scale', 'auto'],  # Kernel coefficient
    'degree': [2, 3, 4],
}
```

Single Image Prediction

A dedicated process was implemented to evaluate the model on individual coral images. The process involves two key steps:

- **Image Preprocessing**: The input image is resized to 224x224 pixels, converted from BGR to RGB format, and preprocessed to match the input requirements of the VGG16 model.

- **Prediction**: Once the image is preprocessed, features are extracted using the VGG16 model, and these features are normalized using the pre-fitted scaler. The **trained SVM classifier** then predicts whether the coral is "Healthy Coral" or "Bleached Coral".

The model correctly predicts **8 out of the 10 images**, demonstrating its reliability for single-image predictions.

Challenges we are facing

In cases where coral forms in clusters, the model encounters difficulties due to the high level of noise in the images, which makes it challenging to accurately classify the coral as healthy or bleached.