CS 136: Introduction to Computer Vision

Preprocessing of Coral Dataset

Sai Mounika Peteti, Sumukh Naveen Aradhya

INTRODUCTION

Preprocessing of images is an important step in computer vision workflows. This is useful because it ensures that raw data is optimized for further analysis. The main idea of our project is to preprocess the Coral image dataset in order to efficiently be able to detect and analyse various features present on the growing buds. This data is observed to have noise, irregular texture and also edge complexity.
The main focus of this project includes filtering of an image, detecting edges, and texture-based segmentation. To ensure modularity and ease of review, each preprocessing algorithm will be implemented as a separate module. All the resulting images and their corresponding code have been placed in labelled folders as per initial instructions.
The project is divided into two main parts:

1. Preprocessing
2. Creative Exploration

In the Preprocessing phase, we aim to perform image filtering, edge detection and image segmentation on grayscale coral images. In the creative exploration phase we aim to parameter tune various newer algorithms that haven't been discussed in class. Every implemented technique serves a specific purpose in the pre-processing of the coral dataset.
In the Creative Exploration phase, we aim to explore different methods such as skeletonization, template matching, k-means clustering, and histogram equalization.

OUR GOALS

This project will not only show the practical usages of computer vision techniques, but also systematic organization for the review. This work compares several algorithms, does their parameter tuning, and aims to provide insights about the efficacy of different preprocessing methods to prepare coral datasets for further processing.

METHODOLOGY

Part 1: Preprocessing with Standard Computer Vision Algorithms

During the preprocessing activity, we conducted various experiments using different computer vision algorithms that were taught in class to address key challenges like noise reduction, edge detection and texture-based segmentation. Each technique has been implemented as a standalone module to make sure it is clear and modular. Below are more details on the preprocessing methods implemented:

1.1. Image Filtering: Gaussian Filter

The main objective of this was to apply gaussian filters to smooth the coral images by reducing noise and unwanted high frequency details while making sure we preserved significant edges.
A 5x5 gaussian kernel has been generated with a standard deviation value of 1.0. The kernel values are computed using the Gaussian function
$G(x,y) = [1/(2\pi\sigma^2)]e\hat{}-[(x^2+y^2)/2\sigma^2]$.
The kernel is normalized to make sure the sum of all values is 1. Each pixel is convolved with the kernel, pixel values are weighted by kernel and summed to give a smoothed output.

1.2. Edge Detection

This is mainly done to extract boundaries and structural information from the given images. We've implemented two popular methods: Sobel and Canny.

1.2.1. Sobel Edge Detection

We compute intensity gradients in horizontal and vertical directions to identify edges.
Two 3x3 convolution kernels are used:

```
double sobelx[3][3]={{-1,0,1}, {-2,0,2}, {-1,0,1}}; //for horizontal detection
double sobely[3][3]={{-1,-2,-1}, {0,0,0}, {1,2,1}}; //for vertical detection
```

Gradient magnitude is calculated as: $|G| = \sqrt{(G_x^2 + G_y^2)}$ and is scaled to range [0, 255]. We also added new features to make sure the output edges are more prominent by first dilating the edges in a 3x3 neighborhood to thicken the edges and modified the edge intensity by multiplying it with a 1.5 factor and capping it at 255 to make sure the output edges are brighter.

1.2.2. Canny Edge Detection

Here we follow a multi-step process that includes, smoothing using Gauss filter, gradient calculation using Sobel operator, non-maximus suppression to thin edges by retaining local maxima along the gradient direction, hysteresis thresholding where we classify edges as strong, weak or suppressed, edge connectivity where weak connected to strong are preserved and others are suppressed. This produces precise connected edges. Similar to Sobel, we added new features to make sure the output edges are more prominent by first dilating the edges in a 3x3 neighborhood to thicken the edges and modified the edge intensity by multiplying it with a 1.5 factor and capping it at 255 to make sure the output edges are brighter.

1.2.3. Hough transform for circular object detection

A 3D hough space has been constructed having dimensions for center at (x,y). Edge pixels from input image vote in hough space for potential circle parameters. Peaks in hough space indicate presence of circles. Detected circles are then drawn to create the output image.

1.2.4. Edge detector performance evaluation

A reference edge map has been created for each input image using a separate ground_truth generator. Evaluation metrics considered are
Precision: TP/TP+FP (proportion of correctly detected edges among all detected edges)
Recall: TP/TP+FN (Proportion of correctly detected edges among all true edges)
F-measure: 2. (P.R)/(P+R) (Harmonic mean of precision and recall)
Quantitative results are derived to highlight the strengths and weaknesses of Sobel and Canny detectors.

1.3. Texture-Based Image Segmentation

Main goal of this was to segment the image into distinct regions based on texture characteristics. The input image is divided into fixed-size blocks and features like mean intensity, standard deviation and spatial coordinates are calculated for each block. K-means clustering groups blocks into texture regions. The segmented regions are visualized using distinct colors. The outputs have been saved for further analysis.

Part 2: Creative exploration with Computer Vision Algorithms

2.1 Skeletonization

The objective of skeletonization was to reduce the coral structures in the binary image to their skeletal form for complexity and branching analysis. This is achieved by iteratively removing boundary pixels while preserving connectivity, resulting in a thin, one-pixel-wide representation of the coral structure.
Use Case: Helps in analyzing coral branching complexity and structure for understanding growth patterns or health.
Implementation: The image was binarized, and iterative morphological thinning was applied until no further changes were observed. The resulting skeleton was saved for analysis.

```
// Structuring element for morphological operations
cv::Mat element = cv::getStructuringElement(cv::MORPH_CROSS, cv::Size(6, 6)); // Larger size
```

2.2 Template Matching

Template matching aimed to locate specific coral patterns in the larger image by sliding a template image over the input image and computing similarity scores. Edge detection was used to make the matching robust to texture and intensity variations.
Use Case: Detects recurring coral formations or regions of interest for targeted study in underwater imagery.
Implementation: Both images were converted to edge maps, and cv::matchTemplate was applied. Regions with high confidence were highlighted and saved.

2.3 K-means Clustering for coral regions

K-means clustering was employed to segment coral and non-coral regions in underwater images by grouping pixels based on their intensity values. This technique adapted to uneven lighting conditions to provide a more accurate segmentation.
Use Case: Identifies coral coverage in the field of view for monitoring coral reef density and health over time.
Implementation: Pixel intensities were grouped into two clusters, with the brighter cluster representing coral regions. The coral coverage percentage was calculated and the output visualized.

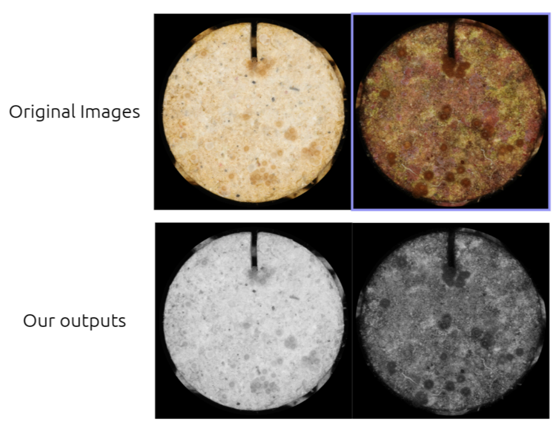
2.4 Histogram Equalization

Histogram equalization was used to enhance contrast in coral images with uneven lighting. Both global histogram equalization and CLAHE (Contrast Limited Adaptive Histogram Equalization) were applied to improve visibility of fine coral structures.
Use Case: Enhances visibility of coral features in low-contrast underwater images, aiding in structural and texture-based studies.
Implementation: The grayscale histogram was equalized globally using cv::equalizeHist, while localized enhancement was achieved with CLAHE. Enhanced images were saved for further processing.

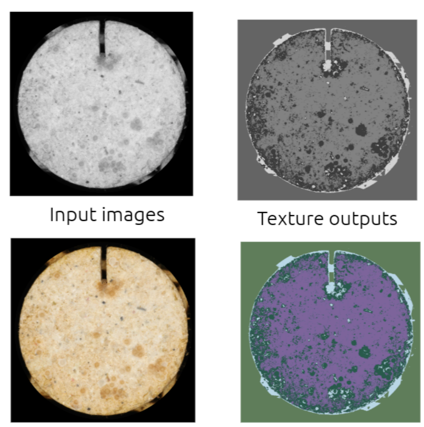
RESULTS

All results for the entire dataset have been uploaded in the zip folder attached with this report. For simplicity, we are adding only a few results in the report.

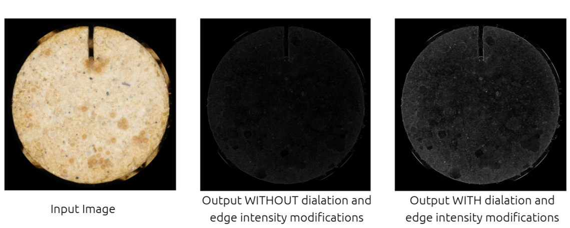
Gaussian Filter:



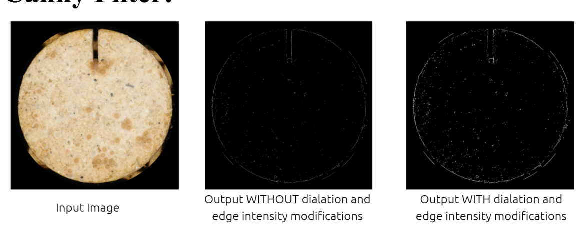
Texture Analysis:



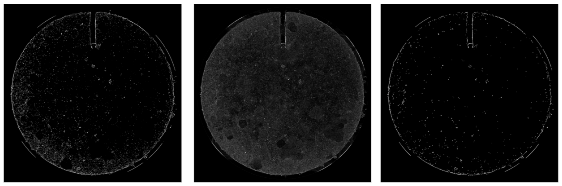
Sobel Filter:



Canny Filter:



Edge Detector Performance Evaluation:



Ground Truth   Sobel Output   Canny Output

Sobel Evaluation:

Precision: 0.714

Recall: 0.231

F-Measure: 0.349

Canny Evaluation:

Precision: 0.692

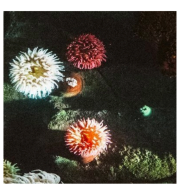
Recall: 0.414

F-Measure: 0.518

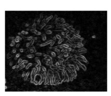
Skeletonization:

Original Image   Skeletonized Result

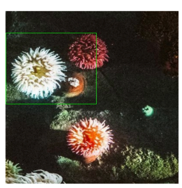
Template Matching



Original Image



Template

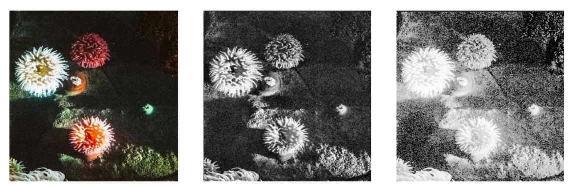


Result

K-means clustering

Original Image   K-means Clustered Binary Image

Histogram Equalization:



Original Image   Result of Global Histogram Equalization   CLAHE (Contrast Limited Adaptive Histogram Equalization)

Conclusion

This project demonstrates the critical role of image preprocessing in computer vision applications, particularly for analyzing complex datasets like coral images. By systematically implementing and evaluating standard techniques such as Gaussian filtering, Sobel and Canny edge detection, and texture-based segmentation, we addressed challenges related to noise, edge complexity, and irregular textures. The creative exploration phase showcased the potential of advanced methods like skeletonization, template matching, k-means clustering, and histogram equalization to enhance analysis and derive deeper insights into coral structures. The modular design and systematic organization of the project ensure ease of review and reusability. Overall, this work provides a comprehensive comparison of preprocessing methods, along with valuable insights into their efficacy, paving the way for more robust downstream tasks in coral image analysis and broader computer vision applications.