Data Collection and Preprocessing Phase

Date	19 July 2024
Team ID	SWTID1720110092
Project Title	Beneath the Waves: Unraveling Coral Mysteries Through Deep Learning
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	The data consists of images of coral reefs collected from various sources, labeled with categories related to reef health and environmental conditions.
Resizing	Resize images to a specified target size (224x224 pixels) using TensorFlow's ImageDataGenerator.
Normalization	Normalize pixel values to a specific range (0 to 1) using rescale=1./255 in ImageDataGenerator.
Data Augmentation	Apply augmentation techniques such as rotation, shifting, zooming, and flipping to increase dataset diversity and model robustness..
Denoising	Not applicable for image data in this context.
Edge Detection	Not applicable for image data in this context.

Color Space Conversion	Not applicable for image data in this context.
Image Cropping	Not applicable as the images are already resized to a fixed size.
Batch Normalization	Apply batch normalization to the input of each layer in the neural network to improve training efficiency and convergence.
Data Preprocessing Code Screenshots	

Loading Data	`import pandas as pd`
	`train_df = pd.read_csv('D:/train_updated.csv').sample(200)`
	`test_df = pd.read_csv('D:/test_updated.csv').sample(50)`

Resizing	`train_datagen = ImageDataGenerator(`
	`rescale=1./255,`
	`rotation_range=20,`
	`width_shift_range=0.1,`
	`height_shift_range=0.1,`
	`shear_range=0.1,`
	`zoom_range=0.1,`
	`horizontal_flip=True,`
	`fill_mode='nearest'`
	`)`

Normalization	`test_datagen = ImageDataGenerator(rescale=1./255)`

Data Augmentation	`train_generator = train_datagen.flow_from_dataframe(`
	`dataframe=train_df,`
	`x_col='local_filename',`
	`y_col='label',`
	`target_size=IMG_SIZE,`
	`batch_size=BATCH_SIZE,`
	`class_mode='categorical'`
	`)`

Batch Normalization	`from tensorflow.keras.layers import BatchNormalization` `x = BatchNormalization()(x)`