



Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	SWTID1720333657
Project Title	Wce Curated Colon Disease Classification Using Deep
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	Data is comprised of colon disease images (Normal, Ulcerative Colitis, Polyps) from Kaggle.
Resizing	Train: 3200 belonging to 4 classes Test: 800 belonging to 4 classes
Normalization	1./255
Data Augmentation	Rotation range=40 width shift range=0.2 height shift range=0.2 shear range=0.2 zoom range=0.2 horizontal flip=True
Denoising	Applied denoising filters to reduce noise in the images

Edge Detection	-
Color Space Conversion	-
Image Cropping	Resize image to 224 x 224
Batch Normalization	-
Data Preprocessing Code Screenshots	
Loading Data	<pre> train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest') train_data = train_datagen.flow_from_directory('C:/Users/DNIN/Desktop/machine learning tutorial/train',target_size = (224,224),batch_size = 32,class_mode = 'categorical') Found 3200 images belonging to 4 classes. test_data = train_datagen.flow_from_directory('C:/Users/DNIN/Desktop/machine learning tutorial/test',target_size = (224,224),batch_size = 32,class_mode = 'categorical') Found 800 images belonging to 4 classes. </pre>
Resizing	 
Normalization	<pre> train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest') train_datagen </pre> <p><keras.src.legacy.preprocessing.image.ImageDataGenerator at 0x28252cc1520></p>
Data Augmentation	<pre> train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest') train_datagen </pre> <p><keras.src.legacy.preprocessing.image.ImageDataGenerator at 0x28252cc1520></p>

Denoising	<pre>def denoise_image(img): # Convert to grayscale (assuming grayscale is suitable for denoising) gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Apply non-local means denoising (replace with your preferred filter) denoised_img = cv2.fastNlMeansDenoising(gray, None, 10, 10, 7, 21) return denoised_img train_datagen.preprocessing_function = denoise_image train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest') # Add denoising to the data augmentation pipeline train_datagen.preprocessing_function = denoise_image train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest')</pre>
Edge Detection	-
Color Space Conversion	-
Image Cropping	<pre># Image Cropping (Can be implemented within flow_from_directory) train_data = train_datagen.flow_from_directory('C:/Users/MDIN/Desktop/machine learning tutorial/train', target_size=(224, 224), # Resize images to 224x224 batch_size=32, class_mode='categorical', subset='training' # Use 'validation' for validation data) Found 3200 Images belonging to 4 classes. test_data = train_datagen.flow_from_directory('C:/Users/MDIN/Desktop/machine learning tutorial/test', target_size=(224, 224), batch_size=32, class_mode='categorical') Found 800 Images belonging to 4 classes.</pre>
Batch Normalization	-