Region growing is a pixel-based image segmentation technique that starts with seed pixels and grows a region by adding neighboring pixels that satisfy homogeneity criterion.

Homogeneity criterion: is a measure of similarity used to determine whether pixels or regions should be grouped together. It can be based on various properties such as intensity, color,

texture or other image features.

Region Growing:

import cv2

import numpy as np

from PIL import image

im = cv2.imread("/Object.jpg")

gray\_image = cv2.cvtcolor(im, cv2.color\_BGR2GRAY)

im = cv2.imread("/Object.jpg", cv2.color\_BGR2GRAY) -- Gray scale image

#Define or initalize region

def region\_growing(image, seed):

region = np.zeros\_like(gray\_image, dtype = np.uint8) -- Binary image to store the resulting region

visited = np.zeros\_like(gray\_image, dtype = np.uint8) -- Binary array to track visited pixels

queue = [seed]

-- Starting point for region growing (Queue: list to store pixels to be processed , initialized with seed point)

while queue:

current\_pixel = queue.pop(0)

x,y = current\_pixel

if visited[x,y] == 1:

continue

#Check homogeneity criterion (intensity difference)

if abs(int(image[x,y]) - int(image[seed])) < threshold:

region[x,y] = 255

visited[x,y] = 1

#Add neighbours coordinates to the queue

for i in range(-1,2):

for j in range(-1,2):

if 0 <= x + i < image.shape[0] and 0 <= y + j < image.shape[1]:

queue.append((x+i, y + j))

return region

seed\_point = (2,2)

threshold = 20

reult\_region = region\_growing(image, seed\_point)

cv2\_imshow(result\_region)

cv2.waitkey(0)

cv2.destroyAllWindows()

#Simple calculations on Pixels

import cv2

# Load the image

img = cv2.imread('/Object.jpg')

# Get the intensity of a pixel at (x, y)

x = 75 # Replace with the desired x-coordinate

y = 75 # Replace with the desired y-coordinate

intensity = img[y, x]

# Print the intensity value

print(f'The intensity of the pixel at ({x}, {y}) is {intensity}.')

x1 = 5

y1 = 5

intensity1 = img[y1, x1]

z = abs(int(image[x, y]) - int(image[x1,y1]))

z1 = int(image[x, y])

print(z1)

print(z)

Output: The intensity of the pixel at (75, 75) is [ 20 62 235].

109

146

Region splitting is a recursive image segmentation technique that divides an image into regions based on a homogeneity criterion.

The basic idea is to start with the entire image as one region and then recursively split into smaller regions untill stopping criteria are met.

Region merging is the reverse of region splitting, it starts with small regions and merges adjacent regions based on a homogeneity criterion until no further merging is possible.

Region splitting & Region merging

import cv2

import numpy as np

import matplotlib.pyplot as plt

# Region splitting

def region\_splitting(image, homogeneity\_threshold, merge\_threshold):

def homogeneity\_criterion(region):

if region is None:

return True # If empty, consider it homogeneous

elif region.size == 0:

return True

return np.std(region) < homogeneity\_threshold

def split\_region(region):

h, w = region.shape if len(region.shape) > 1 else (1, len(region))

return [

region[:h//2, :w//2],

region[:h//2, w//2:],

region[h//2:, :w//2],

region[h//2:, w//2:]

]

def process\_region(region):

if homogeneity\_criterion(region):

return [region]

else:

sub\_regions = split\_region(region)

result = []

for sub\_region in sub\_regions:

result.extend(process\_region(sub\_region))

return result

regions = process\_region(image)

# Display the result using matplotlib

for i, region in enumerate(regions):

if region is not None:

plt.imshow(region, cmap='gray')

plt.show()

regions = process\_region(image)

# Merge adjacent homogeneous regions

merged\_regions = []

for region in regions:

if len(merged\_regions) == 0:

merged\_regions.append(region)

else:

last\_region = merged\_regions[-1]

if homogeneity\_criterion(last\_region) and homogeneity\_criterion(region):

merged\_regions[-1] = np.vstack([last\_region, region])

else:

merged\_regions.append(region)

# Display the result using matplotlib

for i, region in enumerate(merged\_regions):

if region is not None:

plt.imshow(region, cmap='gray')

plt.show()

# Example usage:

image = cv2.imread('/Object.jpg', cv2.IMREAD\_GRAYSCALE)

region\_splitting(image, homogeneity\_threshold=20, merge\_threshold=50)