DIP Assignment 1

Dr. Akhtar Jamil

Name:	**Fahad Ramzan**
Section:	B
Roll No:	20i-0443
Date of Sub:	2023.10.01

1 Question 1

Introduction

First of all, I mounted drive in my Colab environment. I have uploaded the data.rar to my drive and then extracted it into a folder named Dataset. Later, I used that path to access every image and use it in my question to perform image processing tasks.

Proposed Solution Question 1

This code performs several key tasks using the OpenCV library. Firstly, it imports OpenCV. Next, it defines a function called classify_shape that evaluates whether a given contour represents a "Square" or a "Rectangle" based on the aspect ratio of its enclosing rectangle. The code then loads an image from a file and converts it into a binary format using a threshold. After identifying contours in the binary image, it iterates through each contour, approximating them to polygons. The classify_shape function is used to classify each shape as a square or rectangle based on their aspect ratios. Further calculations are done to determine the perimeter, area, and centroid of each contour, with these parameters and the shape classification being displayed. Finally, the code visually annotates the image by drawing the contours, centroids, and shape classifications before displaying the modified image.

Relevance to Lectures

During our recent lectures, we delved into important concepts such as thresholding to create binary representations of images and the extraction of contours to identify edges. Building on this foundation, we have applied the knowledge acquired from these lectures to our current task. Additionally, we have drawn upon the concepts of image moments explored in lecture 5-2 to calculate both the areas and centroids of these identified contours. This process enables us to analyze and characterize shapes within the image more effectively.

2 Question 2

Introduction

First of all, I mounted drive In my Colab environment. I have uploaded the data.rar to my drive and then extracted it into a folder named Dataset. Later, I used that path to access every image and use it in my question to perform image processing tasks.

Proposed Solution Question 2

The code first defines a function to count black pixels in an image and another function to compare the counts between the two images. It then reads both images and converts them to black and white using a specific threshold value. This conversion helps distinguish the dark and light parts of the images.

After this conversion, it counts how many black pixels are in each image and compares the counts. If Image A has more black pixels, it will tell you that Image A has more dark areas, and it might mention something about the content of Image A. If Image B has more black pixels, it will say the same for Image B. If both images have the same number of black pixels, it simply states that they are equal in terms of dark areas.

In summary, this code serves as a tool for comparing the darkness or blackness of two images and provides insights into which one contains more dark regions, helping in image analysis and processing tasks.

Relevance to Lectures

The concept of thresholding are used to convert image into a binary images with black and white pixels only. Moreover than image moments are used to count the number of black pixels the more the black pixels the more chance that it's a girl.

3 Question 3

Introduction

First of all, I mounted drive In my Colab environment. I have uploaded the data.rar to my drive and then extracted it into a folder named Dataset. Later, I used that path to access every image and use it in my question to perform image processing tasks.

Proposed Solution Question 3

It employs a method known as Sobel edge detection, which helps identify edges or boundaries within the images—these edges often indicate transitions between light and dark areas. The code calculates the average strength of these edges in both images, providing a measure of how clear or blurry each image appears.

After applying Sobel edge detection and computing the average edge strengths, the code reports the results. It tells you the edge strength for both images and identifies which image exhibits stronger edges. If Image 1 has more pronounced edges, it declares Image 1 as clearer. Conversely, if Image 2 has stronger edges, it designates Image 2 as the clearer one. In the event that both images possess similar edge strengths, the code indicates that they are equally clear (or blurry).

In essence, this code evaluates image clarity by assessing the presence and strength of edges in the images, a common technique in image processing to determine image sharpness and quality.

Relevance to Lectures

In the lectures as we studied about the Sobel operator or kernel or a first order derivative filter is known for detection of edges. Once edge is detected the edge strength is calculated using the formula. The greater the edge strength the clearer the image is.

s.

4 Question 4

Introduction

First of all, I mounted drive In my Colab environment. I have uploaded the data.rar to my drive and then extracted it into a folder named Dataset. Later, I used that path to access every image and use it in my question to perform image processing tasks.

Proposed Solution Question 4

This code is an image processing tool that focuses on quantifying and visualizing the areas of color bars within an input image. It follows a step-by-step process to achieve this. First, it loads the image containing the color bars. Next, it defines specific regions of interest (ROIs) for each color bar, essentially creating rectangular areas around them. These ROIs are essential for isolating and processing each color bar separately. The code then proceeds to analyze each ROI individually. For each ROI, it converts the selected region to grayscale and applies a thresholding technique to create a binary mask that highlights the color bar's shape. The contours of the color bar within the binary mask are detected and used to calculate the area of the color bar. To ensure accuracy, the code filters out small noise regions. Additionally, it annotates the original image by drawing bounding rectangles around the color bars and placing text annotations displaying the calculated areas at the centroids of the contours. Finally, the modified image with annotations is displayed, providing a clear visual representation of the color bar areas. In essence, this code facilitates image-based area measurements, which can be valuable in various applications such as quality control, image analysis, and data extraction.

Relevance to Lectures

In the last lecture as we studied about the image segmentation and bounding boxes using that concept. Contours are calculated and then each bars area is calculated and then using the concept of image moments in lecture 5-2. We calculate the area and centroid.

5 Question 5

Introduction

First of all, I mounted drive In my Colab environment. I have uploaded the data.rar to my drive and then extracted it into a folder named Dataset. Later, I used that path to access every image and use it in my question to perform image processing tasks.

Proposed Solution Question 5

This image processing solution is designed to perform two key tasks on an input image containing color bars and a red arrow symbol. Firstly, it calculates the areas of individual color bars within the image. To achieve this, it employs techniques like grayscale conversion, thresholding, and contour detection to isolate and quantify the area of each color bar. Secondly, it determines the percentage of each color bar covered by the red arrow symbol. This is achieved by creating a specific color range for the red arrow, identifying it within the color bars, and calculating the proportion of each color bar's area occupied by the arrow. The code segments and displays the red arrow within each color bar for debugging purposes. Overall, this solution combines various image processing techniques to provide quantitative measurements of color bar areas and the extent of coverage by a symbol, which can be invaluable for tasks such as quality assessment or data extraction from images.

Relevance to Lectures

In the last lecture as we studied about the image segmentation and bounding boxes using that concept. Contours are calculated and then each bars area is calculated and then using the concept of image moments in lecture 5-2. We calculate the area and centroid and the area of interest.

6 Question 6

Introduction

First of all, I mounted drive In my Colab environment. I have uploaded the data.rar to my drive and then extracted it into a folder named Dataset. Later, I used that path to access every image and use it in my question to perform image processing tasks.

Proposed Solution Question 6

It employs the HSV (Hue, Saturation, Value) color space to enhance color-based segmentation. The code defines specific HSV ranges for three different colors: "Blue," "Peach," and "Green." Each color range is used to create masks that isolate regions of the image matching the specified color characteristics. The code then identifies contours within these color-matched regions and filters out small contours based on a minimum area threshold.

The identified segments are saved as separate images, each containing objects of a specific color. Green bounding boxes are drawn around these segments for visualization, and the maximum width and height of each segmented object are determined and displayed. This process facilitates the extraction and analysis of objects with different colors from the original image.

Additionally, segmented images are saved in a directory named "segmented_images," allowing further inspection and analysis of individual color regions. This code demonstrates the practical application of color-based segmentation in DIP, which is valuable for various tasks such as object recognition, medical image analysis, and quality control.

Relevance to Lectures

In the last lecture as we studied about the segmentation and using the concept of image moments we calculate the maximum width and height of the segmented images.