Detect Differences between two Images- By

Niladri Mullick-17uec081

Approach and explanation-

Below is our two images as example.
We first read and display the images using opencv.
Then we need to change the shape of both the images to a constant value which must be same (in this case (300*300*3)).

Image 1

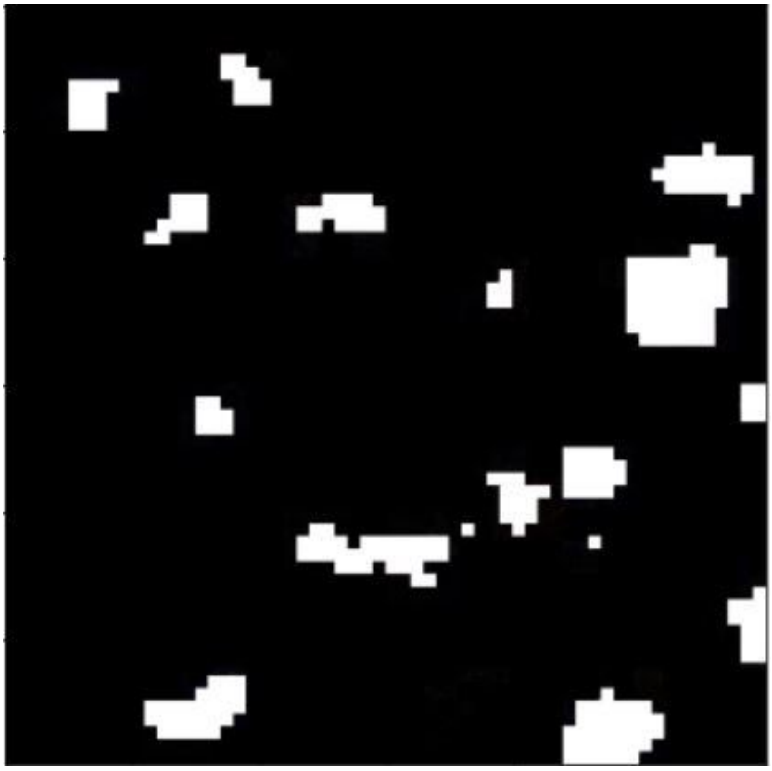
Image 2

The approach is to first take pixel by pixel difference between the two images using the difference function of pillow library..

Then we need to draw contours around the places where there are differences between two images. But the problem is that since the differences are in the form of blobs, hence there would be overlapped and non-uniform and so the bounding boxes would be overlapping.
To resolve this issue, we modified the difference image. The idea is to make the regions of the image where there are differences to be uniform and continuous.

For this we have defined a filter of size a*a (say 5pixel * 5pixel, this value is dynamic and can be changed) and parsed the filter through the difference image. Whenever the sum of all values within the filter is greater than a threshold value (say 500, it is also a hyperparameter and can be altered), then we have changed the color of entire filter to white, in effect this makes the

entire region where blobs existed to a continuos white-space thereby the contours can be uniform and non overlapping.



Areas where differences exists

Here the filter size (a*a) and the threshold_value are hyperparameters, we will use multiple combinations of these values to get the best set of bounding boxes as explained later

Next we draw contours around the white spaces using the opencv function- “cv2.findCountours” which returns a list of contour coordinates. We then use these contour coordinates to draw the bounding boxes by using the cv2.boundingrect which returns a list of tuples containing the coordinates of bounding boxes and rectangle function of open cv to finally draw the bounding boxes around the image.

In the final cell of the code, the user will be prompted for a number and he will have to enter the no. of differences expected from the image(say 15 for this image, generally this value should be provided in the question). (this will be used to auto-tune the values of hyperparameters.)
After getting the expected differences value , the code will run various combinations of hyperparameters(filter-size and threshold-value (explained above)) and try to match the no. of bounding boxes to the expected differences,something similar to (but not exactly) grid-search concept of machine learning. For example if the expected differences is 15 then the no. of bounding boxes should be as close to 15 as possible and so we will try different combinations of hyperparameters to obtain the no. of bounding boxes to as close to 15 as possible.

NOTE- When prompted enter the no. expected differences (15 for this one room_image)

1. For room image no of expected differences= 15
2. For leaf image no. of expected differences= 5
3. For fruits image, no.of expected differences= 5
4. For things image, no.of expected differences= 7

Then we will obtain our best 3 sets of images with bounding boxes outlining the differences,(this could be increased or decreased to more or less than 3 also),generally the first set would be our answer but we will manually compare the 3 sets (which now have the differences outlined) to see which one gives the best result.

1.

2.

3.

Finally the result is displayed in the final cell.

Final Answer