CS-518 Computer Vision

Assingment 1

Rhythm Jain
2019CSB1111

CollageCreate( )

This Function takes the address of the director which contains the images relative to the program directory.

First I Used Glob Module to input all the images in the specified directory.Then i randomly selected 6 images out of all the images and created another list called ImagesList.

Then I sorted the images according to Amount of color and edges present in an image which is based on the function var_color_edge( ).

Then I called the merge( ) function which takes the image list as input and the amount of borders of the images to be merged/blended and then returns the blended final image.

At last I save the final output image as Output_Img.jpg in the same directory.

var_color_edge( )

Takes input an image.

Since our image is a 3D matrix , I calculated the average intensity of each channel using matrix traversal and stored it in the variable named color .

Then to extract images, first I converted the image to grayscale image and then using canny edge detector I converted the image to an image with edges.Then in order to extract the percentage of edges in the image first i converted the image to binary image using threshold function and then I counted the number of pixels with intensity 0 and 1 since it is a binary matrix now. Here 0 Represents Edge (Black) and 1 Represents No Edge (white) .Then I took the percentage of black pixels that represents edges, that is pixels with value 0.

I stored the edge information in a variable named edge .

Then I returned the final variance using 50% color variance and 50% edge variance.

Return 0.5 * color + 0.5 * edge

merge( )

This function takes input as in image list and then gives us the final blended image as output.This functions called two helper functions :

1. merge_h( ):

   Input to this function is two images and a block size. Block size here represents the amount of borders (in pixels) of the images we need to blend.

   This function merges / blends two images horizontally .

   In order to blend images horizontally we need to blend the right border of the first image with the left border of the second image.So we extract the corresponding parts of the images using slicing.

   Then in order for the merge effect to look more authentic we have to blend more in the middle and less on the edges .For that I used a while loop and incremented value of ‘a’ as well as boundary each time till it ‘a’ reaches from 0 to 0.5 boundary reaches from block size to 0, and then I merged that part of the blended image.

   I Used the following algorithm in order to Blend:

   $$\text{Image1} * a + \text{Image1} * (1-a)$$

2. merge_v( ):

   Input to this function is two images and a block size. Block size here represents the amount of borders (in pixels) of the images we need to blend.

   The working of this function is exactly similar to the one defined above, the difference is that this function merges and blends the images vertically rather than horizontally.

Modules Used

1. skimage
2. matplotlib
3. numpy
4. glob
5. random