Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING
Program: T. Y. B. Tech ECE Semester -6
Subject: Digital Image Processing DIP (PE-2) (ECE3013B)

Experiment No.1

Basic Operations on Images

Aim: To perform basic pixel-level operations on synthetic and natural digital images.

A. Arithmetic operations
B. Logical Operations
C. Resize
D. Rotation and Translation

Introduction:

A. Arithmetic operations

Arithmetic Operations like Addition, Subtraction, and Bitwise Operations(AND, OR, NOT, XOR) can be applied to the input images. These operations can be helpful in enhancing the properties of the input images. The Image arithmetics are important for analyzing the input image properties. The operated images can be further used as an enhanced input image, and many more operations can be applied for clarifying, thresholding, dilating etc of the image.

ADDITION OF IMAGE:

We can add two images by using function **cv2.add()**. This directly adds up image pixels in the two images.

These are the like image addition, subtraction, multiplication and division is possible. If there is two images I1 and I2 then addition of image can be given by:

$$I(x,y) = I1(x,y) + I2(x,y)$$

Where I(x,y) is resultant image due to addition of two images. x and y are coordinates of image. Image addition is pixel to pixel. Value of pixel should not cross maximum allow value that is 255 for 8 bit grey scale image. When it exceeds value 255, it should be clipped to 255.

To increase overall brightness of the image, we can add some conthe stant value depending on brightness required. In example program we will add value 50 to the image and compare brightness of original and modified image.

Syntax: **cv2.add(img1, img2)**

But adding the pixels is not an ideal situation. So, we use cv2.addweighted(). Remember, both images should be of equal size and depth.

SUBTRACTION OF IMAGE:

Just like addition, we can subtract the pixel values in two images and merge them with the help of cv2.subtract(). The images should be of equal size and depth.

Syntax: **cv2.subtract(src1, src2)**

To decrease brightness of image, we can subtract constant value. In example program, value 100 is subtracted for every pixel. Care should be taken that pixel value should not less than 0 (minus value). Subtract operation can be used to obtain complement (negative) image in which every pixel is subtracted from maximum value i.e. 255 for 8 bit greyscale image.

Multiplication operation can be used to mask the image for obtaining region of interest. Black and white mask (binary image) containing1s and 0s is multiplied with image to get the resultant image. Multiplication operation can also be used to increase brightness of the image **.Division operation** results into floating point numbers. So data type required is floating point numbers. It can be converted into integer data type while storing the image. Division can be used to decrease the brightness of the image. It can also use to detect change in image.

B. Logical Operations:

Bitwise logical operations can be performed between pixels of one or more than one image.

AND/NAND Logical operations can be used for following applications:

- Compute the intersection of the images
- Design of filter masks
- Slicing of grayscale images

OR/NOR logical operations can be used for the following applications:

- Merging of two images

XOR/XNOR operations can be used for the following applications:

- To detect a change in the gray level in the image
- Check the similarity of the two images
- NOT operation is used for:
- To obtain a negative image
- Making some features clear

Prerequisite: Arithmetic Operations on Images | Set-1
Bitwise operations are used in image manipulation and used for extracting essential parts in the image. In this article, Bitwise operations used are :

1. **AND**
2. **OR**
3. **XOR**

4. **NOT**

Also, Bitwise operations helps in image masking. Image creation can be enabled with the help of these operations. These operations can be helpful in enhancing the properties of the input images.
NOTE: The Bitwise operations should be applied on input images of same dimensions

Image resizing

Image resizing refers to the scaling of images. Scaling comes in handy in many image processing as well as machine learning applications. It helps in reducing the number of pixels from an image and that has several advantages e.g. It can reduce the time of training of a neural network as the more the number of pixels in an image more is the number of input nodes that in turn increases the complexity of the model.
It also helps in zooming in on images. Many times we need to resize the image i.e. either shrink it or scale it up to meet the size requirements. OpenCV provides us several interpolation methods for resizing an image.

Choice of Interpolation Method for Resizing:

- cv2.INTER_AREA: This is used when we need to shrink an image.
- cv2.INTER_CUBIC: This is slow but more efficient.
- cv2.INTER_LINEAR: This is primarily used when zooming is required. This is the default interpolation technique in OpenCV.

Syntax: cv2.resize(source, dsize, dest, fx, fy, interpolation)

Scaling an Image:- Scaling operation increases/reduces size of an image.

Rotating an image:- Images can be rotated to any degree clockwise or otherwise. We just need to define the rotation matrix listing the rotation point, degree of rotation and scaling factor.

Translating an Image:- Translating an image means shifting it within a given frame of reference.

Note
Write the Result and Conclusion in the last cell of the python code using markdown.

Questions
1. What do you mean by gray level
2. Write the expression to find the number of bits to store a digital image
3. Name types of resolutions w. r. to a digital image.
4. Specify the elements of the DIP system
5. Write any four applications of DIP