****

**EXPERIMENT 1**

|  |  |
| --- | --- |
| **Name: Shivam Bhosle** | **Roll No: 2** |
| **Subject: Computer Vision (CV)** | **Class/Batch: B1** |
| **Date of Performance: 16/08/24** | **Date of Submission:** |

|  |
| --- |
| **AIM** |

**To Perform Image Processing using Image Arithmetic Operations.**

|  |
| --- |
| **Theory** |

The important requirement in image arithmetic is that all (input and output) the images are of the same size MxM.

Arithmetic operations are done pixelwise. Let p = A(x,y) and q = B(x,y) be the pixel values to be operated on and r =I(x,y) be the result of the operation.

**Addition :-** I(x,y) = A(x,y) + B(x,y) → r = p + q

**Subtraction :-** I(x,y) = A(x,y) - B(x,y) → r = p - q

**Difference :-** I(x,y) = |A(x,y) - B(x,y)| → r = |p - q|

**Multiplication :-** I(x,y) = A(x,y) X B(x,y) → r = p x q

**Division :-**  I(x,y) = A(x,y) / B(x,y) → r = p / q

**Implementation issues :-** Digital images are stored as b - bit images. Hence, the range of values a pixel can take is restricted to the range [ 0, 1,.. (2b -1)]. With b= 8 this range is [0,1,..255]. The closed interval poses a problem when performing arithmetic operations in practice, as the results are not guaranteed to be within this interval. For an 8-bit image the intervals for the output pixel for each operation are:

Addition: r ∈ [0, (2x255=510)]

Subtraction: r ∈ [-255, 255]

Difference: r ∈ [0, 255]

Multiplication: r ∈ [0, (2552 = 65025)]

Division: r ∈ [0,∞]

Since we need r to be in [0,255], we will have an underflow or overflow. A final processing step is generally required to handle this problem.

There are two options:

Clipping- Map all negative pixel values ( r < 0) to zero and all values above 255 to 255.

Auto scaling - This operation remaps the range of r to fit to be in [0, 255] as follows. Let ra be the autoscaled value.

ra = 255 x (r - rmin)/(rmax-rmin)

Where, rmax and rmin are the maximum and minimum values of an arithmetic operation.



1. To learn to use arithmetic operations to combine images.
2. To study the effect of these operations on the dynamic range of the output image.
3. To study methods to enforce closure - force the output image to also be an 8 bit image.

**Procedure**

The experiment is design to understand and learn the image arithmetic concepts. This experiment consists five parts:

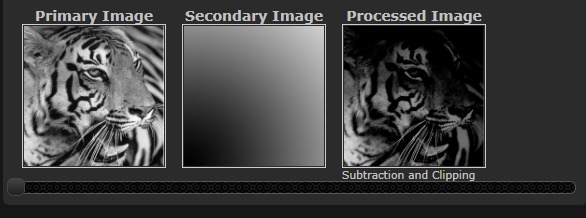
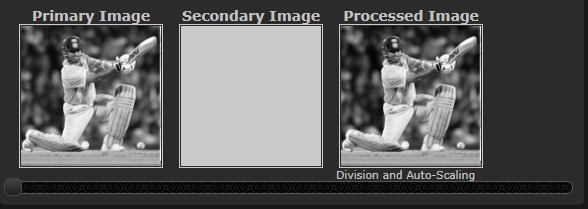
1. Image addition
2. Image Subtraction
3. Image Difference
4. Image Multiplication
5. Image Division

**Steps to run the experiments**

1. Select image from the mosaic using 'select imag' option
   1. Select region of the image to load it in the input image panel
   2. Select secondary image by choosing either of the 'dull' , 'bright' or 'gradient' images
2. Select one option from 'addition', 'subtraction', 'difference', 'multiplication' and 'division'
3. Select the one option from 'clipping' and 'auto-scaling'
4. Select run option to perform the operations
   1. Output result will be displayed in the output panel
   2. Along with intermediate results

**Output**

1. **Image Addition :-** 

1. **Image Subtraction :- **
2. **Image Difference :- **
3. **Image Multiplication :- **
4. **Image Division :- **

|  |
| --- |
| **Conclusion** |

From This Experiment we have learned about utilizing virtual labs for Image Processing and its operations namely Addition, Subtraction, Multiplication and Division.

|  |
| --- |
| **Assessment** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Timely Submission**  **(7)** | **Presentation**  **(06)** | **Understanding**  **(12)** | **Total**  **(25)** | **Sign** |
|  |  |  |  |  |