Experiment-4

Arithmetic And Logical Operation

Student Name : Pratibha **UID :** 18bcs6093

Branch : CSE-AI&ML **Section/Group :**1(Group-2)

Semester : 5 **Date of Performance :**

Subject Name: DIP **Subject Code :** CSF-336

1. **AIM/OVERVIEW of the Practical**

   In this experiment we have perform image arithmetic and logical operations using opencv. In this we have used the jupyter notebook software.

2. **Task to be Done**

   Various tasks which we have to performed in this experiment that are

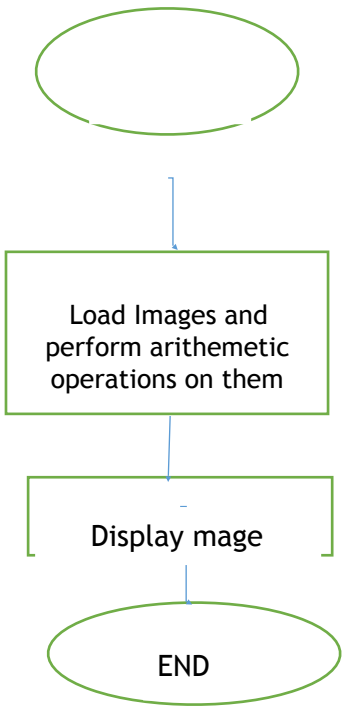
   - Read images
   - Perform arithmetic operation on images
   - Display image

3. **Required Libraries Or Softwares**

   Software – Jupyter Notebook

   Libraries – numpy, matplotlib, pil, scikit Image, opencv

4. **Algorithm/Flowchart**



Steps for Experiment/Practical

1. **# Image Addition**

```
import cv2
import numpy as np
imag1 = cv2.imread('./img1.png')
imag2 = cv2.imread('./img2.png')
add = imag1+imag2
cv2.imshow('addition window',add)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

2. **# Image Addition**

```
import cv2
import numpy as np
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
add = cv2.add(img1,img2)
cv2.imshow('addition window',add)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

3. **# Image Blending**

```
import cv2
import numpy as np
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
weighted = cv2.addWeighted(img1, 0.6, img2, 0.4, 0)
# Notation: addWeighted method, the parameters are the first image, the weight, the second image, that weight, and gamma, which is a measurement of light.
cv2.imshow('add weighted window',weighted)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

5. **# To show the mask inverse using threshold**

```
import cv2
import numpy as np
```

```
# Load two images

img1 = cv2.imread('./img1.png')

img2 = cv2.imread('./img2.png')

# I want to put logo on top-left corner, So I create a ROI

rows,cols,channels = img2.shape

roi = img1[0:rows, 0:cols]

# Now create a mask of logo and create its inverse mask, mask generly would be to grey scale

img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)

# add a threshold

# Threshold: it works is that, it will convert all pixels to either black or white, based on a threshold value.

ret, mask = cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)

cv2.imshow('mask window', mask)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

6. **# Bitwise Operation**

```
import cv2

import numpy as np

# Load two images

img1 = cv2.imread('./img1.png')

img2 = cv2.imread('./img2.png')

# I want to put logo on top-left corner, So I create a ROI

rows,cols,channels = img2.shape

roi = img1[0:rows, 0:cols ]

# Now create a mask of logo and create its inverse mask, mask generly would be to grey scale

img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)

# add a threshold

# Threshold: it works is that, it will convert all pixels to either black or white, based on a threshold value.

ret, mask = cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)

#This is a bitwise operation.black area of mask

mask_inv = cv2.bitwise_not(mask)

# bitwise_not : mask or not mask

# bitwise_and : when two are equal
```

```
# bitwise_or : When both values are true it runs, if only one value iss true it runs
# bitwise_xor : onle when one value is true
# we want to black out this area in the first image, and then take image 2 and replace it's contents in that empty spot.
# Now black-out the area of logo in ROI, bitwise is a lowlevel logical operation.
img1_bg = cv2.bitwise_and(roi,roi,mask = mask_inv)
# Take only region of logo from logo image.
img2_fg = cv2.bitwise_and(img2,img2,mask = mask)
dst = cv2.add(img1_bg,img2_fg)
img1[0:rows, 0:cols ] = dst
cv2.imshow('res window', img1)
cv2.imshow('mask_inv window', mask_inv)
cv2.imshow('img1_bg window', img1_bg)
cv2.imshow('img2_fg window', img2_fg)
cv2.imshow('dst window', dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

7. **# Performing bitwise operation without using mask inverse**

```
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
# I want to put logo on top-left corner, So I create a ROI
rows,cols,channels = img2.shape
roi = img1[0:rows, 0:cols ]
# Now create a mask of logo and create its inverse mask also
img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(img2gray, 10, 255, cv2.THRESH_BINARY)
mask_inv = cv2.bitwise_not(mask)
# Now black-out the area of logo in ROI
img1_bg = cv2.bitwise_and(roi,roi,mask = mask_inv)
# Take only region of logo from logo image.
img2_fg = cv2.bitwise_and(img2,img2,mask = mask)
# Put logo in ROI and modify the main image
dst = cv2.add(img1_bg,img2_fg)
img1[0:rows, 0:cols ] = dst
```

cv2.imshow('res',img1)

cv2.waitKey(0)

cv2.destroyAllWindows()

4. The command that we have learned today in the program :

In this program we have learnt various command performing the arithmetic or logical operations on image.

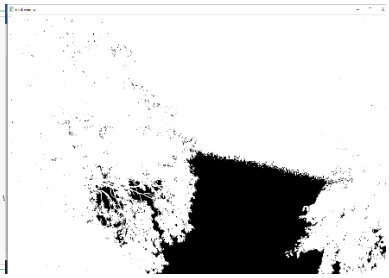
5. Output

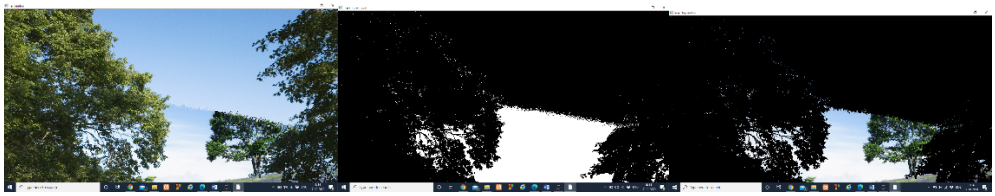
```
In [*]: # Importing required modules
import cv2
import numpy as np
imag1 = cv2.imread('./img1.png')
imag2 = cv2.imread('./img2.png')
add = imag1+imag2
cv2.imshow('addition window',add)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

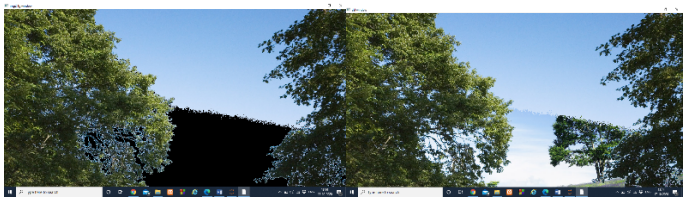
```
In [*]: # Importing required modules
import cv2
import numpy as np
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
add = cv2.add(img1,img2)
cv2.imshow('addition window',add)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
# Importing required modules
import cv2
import numpy as np
# Considering both the imgaes of indentical size 500 x 250
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
weighted = cv2.addWeighted(img1, 0.6, img2, 0.4, 0)
# Notation: addWeighted method, the parameters are the first
cv2.imshow('add weighted window',weighted)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

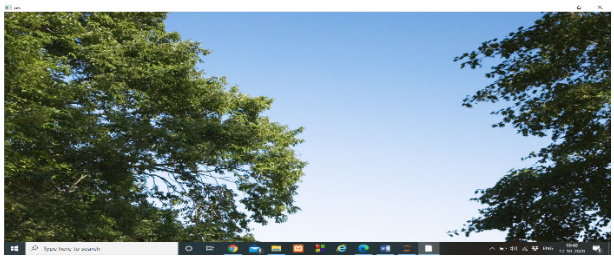
In [*]:
```
# Importing required modules
import cv2
import numpy as np
# Load two images
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
# I want to put logo on top-left corner, So I create a ROI
rows,cols,channels = img2.shape
roi = img1[0:rows, 0:cols]
# Now create a mask of logo and create its inverse mask, mask generly would be to grey scale
img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
# add a threshold
# Threshold: it works is that, it will convert all pixels to either black or white, based on a threshold v
ret, mask = cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)
cv2.imshow('mask window', mask)
cv2.waitKey(0)
cv2.destroyAllWindows()
```







In [4]:
```
# Importing required modules
import cv2
import numpy as np
# Load two images
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
# I want to put logo on top-left corner, So I create a ROI
rows,cols,channels = img2.shape
roi = img1[0:rows, 0:cols ]
# Now create a mask of logo and create its inverse mask, mask generly would be to grey scale
img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
# add a threshold
# Threshold: it works is that, it will convert all pixels to either black or white, based on a threshold value.
ret, mask = cv2.threshold(img2gray, 220, 255, cv2.THRESH_BINARY_INV)
#This is a bitwise operation.black area of mask
mask_inv = cv2.bitwise_not(mask)
# bitwise_not : mask or not mask
# bitwise_and : when two are equal
# bitwise_or : When both values are true it runs, if only one value iss true it runs
# bitwise_xor : onle when one value is true
# we want to black out this area in the first image, and then take image 2 and replace it's contents in that empty spot.
# Now black-out the area of logo in ROI, bitwise is a lowlevel logical operation.
img1_bg = cv2.bitwise_and(roi,roi,mask = mask_inv)
# Take only region of logo from logo image.
img2_fg = cv2.bitwise_and(img2,img2,mask = mask)
dst = cv2.add(img1_bg,img2_fg)
img1[0:rows, 0:cols ] = dst
cv2.imshow('res window', img1)
cv2.imshow('mask_inv window', mask_inv)
cv2.imshow('img1_bg window', img1_bg)
cv2.imshow('img2_fg window', img2_fg)
cv2.imshow('dst window', dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



In [*]:

```
# Load two images
img1 = cv2.imread('./img1.png')
img2 = cv2.imread('./img2.png')
# I want to put logo on top-left corner, So I create a ROI
rows,cols,channels = img2.shape
roi = img1[0:rows, 0:cols ]
# Now create a mask of logo and create its inverse mask also
img2gray = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(img2gray, 10, 255, cv2.THRESH_BINARY)
mask_inv = cv2.bitwise_not(mask)
# Now black-out the area of logo in ROI
img1_bg = cv2.bitwise_and(roi,roi,mask = mask_inv)
# Take only region of logo from logo image.
img2_fg = cv2.bitwise_and(img2,img2,mask = mask)
# Put logo in ROI and modify the main image
dst = cv2.add(img1_bg,img2_fg)
img1[0:rows, 0:cols ] = dst
cv2.imshow('res',img1)
cv2.waitKey(0)
cv2.destroyAllWindows()
```