TOPIC : Artificial Intelligence with Python - Jahnavi N

In [ ]:
```
conda create -n tensorflow
pip install keras
```

In [ ]:
```
conda --version
```

In [ ]:
```
pip install opencv-python
```

In [1]:
```
import cv2
```

In [ ]:
```
print(cv2.__version__)
```

In [11]:
```
import cv2

#path = r'C:\Users\JAHNAVI\Downloads\aipic.jpg'

img = cv2.imread('e.jpg',cv2.IMREAD_GRAYSCALE)

cv2.imshow('river', img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [12]:
```
import cv2

img = cv2.imread('img1.jpg',33)
cv2.imshow('red car', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

cv2.IMREAD_COLOR: It specifies to load a color image. Any transparency of image will be neglected. It is the default flag. Alternatively, we can pass integer value 1 for this flag. cv2.IMREAD_GRAYSCALE: It specifies to load an image in grayscale mode. Alternatively, we can pass integer value 0 for this flag. cv2.IMREAD_UNCHANGED: It specifies to load an image as such including alpha channel. Alternatively, we can pass integer value -1 for this flag.

In [13]:
```
print('Image Dimensions :', img.shape)
```

```
Image Dimensions : (135, 240, 3)
```

cv2.cvtColor()

https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html

In [14]:
```
import cv2

path = r'C:\Users\JAHNAVI\Downloads\images.jpg'
```

```
img = cv2.imread(path,1)


cv2.imshow('rainbow', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [18]:
```
import cv2

path = r'C:\Users\JAHNAVI\Downloads\images.jpg'

img = cv2.imread(path,1)

img=cv2.cvtColor(img, cv2.COLOR_BGRA2RGBA)

cv2.imshow('rainbow', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [19]:
```
import cv2

path = r'C:\Users\JAHNAVI\Downloads\images.jpg'

img = cv2.imread(path,1)

img=cv2.cvtColor(img, cv2.COLOR_BGR2HSV )

cv2.imshow('rainbow', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [ ]:
```

```

VideoCapture

In [26]:
```
cap = cv2.VideoCapture("video.mp4")
while(cap.isOpened()):
    ret, frame = cap.read()
    if ret == True:
        cv2.imshow('frame', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break
cap.release()
cv2.destroyAllWindows()
```

ret is a boolean variable that returns true if the frame is available. frame is an image array vector captured based on the default frames per second defined explicitly or implicitly

In [27]:
```
import cv2
vid = cv2.VideoCapture(0)

while(True):
```

```
    ret, frame = vid.read()

    cv2.imshow('frame', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

vid.release()
cv2.destroyAllWindows()
```

Blurring

In [30]:

```
import cv2

img = cv2.imread('img1.jpg')

cv2.imshow('Original Image', img)

cv2.waitKey(0)
blurImg = cv2.blur(img,(10,10))
cv2.imshow('blurred image',blurImg)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Image write

In [32]:

```
import cv2

image = cv2.imread('e.jpg')
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2.imwrite('now.jpg', image_gray)
```

Out[32]: True

LINE ON IMAGES

In [37]:

```
import cv2

image = cv2.imread('images.jpg')
start_point = (0, 0)
end_point = (5000,5000)
color = (0, 255, 255)
thickness = 9
img = cv2.line(image, start_point, end_point, color, thickness)
cv2.imshow("image",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Circles on images

In [40]:

```
import cv2
```

```
image = cv2.imread('images.jpg')
center_coordinates = (120, 50)
radius = 100
color = (255, 255, 0)
thickness = 2
image = cv2.circle(image, center_coordinates, radius, color, thickness)
cv2.imshow('IMAGE',image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [ ]: