opencv ch 1-16

February 4, 2022

1 Ch.1 reading an image and displaying it

[ ]:
```
# import library
import cv2 as cv
```

[ ]:
```
img = cv.imread("resourses/image.jpg")
```

[ ]:
```
cv.imshow("Pehli_Image", img)
cv.waitKey(0)
```

[ ]:
```
-1
```

2 Ch.2 resizing image

[ ]:
```
img2 = cv.imread("resourses/image.jpg")
img2 = cv.resize(img2, (800,600))
```

[ ]:
```
cv.imshow("Pehli_Image", img2)
cv.waitKey(0)
cv.destroyAllWindows()
```

3 Ch.3 grayscale conversion

[ ]:
```
img3 = cv.imread("resourses/image.jpg")
img3 = cv.resize(img3, (800,600))
```

[ ]:
```
# grayscale conversion
gray_img3 = cv.cvtColor(img3, cv.COLOR_BGR2GRAY)
```

[ ]:
```
# display code
cv.imshow("orignal_Image", img3)
cv.imshow("gray_Image", gray_img3)

# delay code
cv.waitKey(0)
```

```
cv.destroyAllWindows()
```

4 Ch 4 Image into Black and white image

[ ]:
```
img4 = cv.imread("resourses/image.jpg")
img4 = cv.resize(img4, (800,600))
gray = cv.cvtColor(img4, cv.COLOR_BGR2GRAY)
(thresh, binary)= cv.threshold(gray, 127, 255, cv. THRESH_BINARY)
cv.imshow('original', img4 )
cv.imshow('gray', gray)
cv.imshow('Black and White', binary)
cv. waitKey(0)
cv.destroyAllWindows()
```

5 Ch 5 Save image

[ ]:
```
img5 = cv.imread("resourses/image.jpg")
img5 = cv.resize(img5, (800,600))
gray = cv.cvtColor(img5, cv.COLOR_BGR2GRAY)
(thresh, binary)= cv.threshold(gray, 127, 255, cv. THRESH_BINARY)
cv.imwrite('resourses/Image_gray.png', gray)
cv.imwrite('resourses/Image_bw.png', binary)
```

[ ]:
```
True
```

6 Ch 6 reading video

[ ]:
```
cap=cv.VideoCapture('resourses/video.mp4')
#indicator
if (cap.isOpened () == False):
    print("error in reading video")

    #reading and playing
while(cap.isOpened ()):
    ret, frame = cap.read()
    if ret == True:
        cv.imshow("Video", frame)
        if cv.waitKey(25) & 0XFF == ord('q'):
            break
    else:
        break
cap.release()
cv.destroyAllWindows ()
```

7 Ch 7 converting video to gray or Black and white

[ ]:
```
#converting video to gray or Black and white
import cv2 as cv
cap2 = cv.VideoCapture("dolphin.mp4")
while (True):
    ret, frame = cap2.read()
    grayframe = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    (thresh, binary) = cv.threshold(grayframe, 127, 255, cv.THRESH_BINARY)
    if ret == True:
       cv.imshow("Video", grayframe)
       if cv.waitKey(1) & 0XFF == ord('q'):
            break
    else:
        break
cap.release()
cv.destroyAllWindows()
```

8 Ch 8 video writing

[ ]:
```
cap=cv.VideoCapture('resourses/video.mp4')
# writing format, codec, video writer object and file output
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
out = cv.VideoWriter("resourses/Out_video.avi", cv.VideoWriter_fourcc('M', "j",␣
 ↪'P', 'G'), 10, (frame_width, frame_height))

#indicator
if (cap.isOpened () == False):
    print("error in reading video")

    #reading and playing
while(cap.isOpened ()):
    ret, frame = cap.read()
    if ret == True:
        out.write(frame)
        cv.imshow("Video", frame)
        if cv.waitKey(25) & 0XFF == ord('q'):
            break
    else:
        break
cap.release()
cv.destroyAllWindows ()
```

9 Ch 9 capture a webcam

[ ]:
```
# how to capture a webcam inside python
# Step-1 Import libraries

import cv2 as cv
import numpy as np

# Step-2 Read the frames from Camera

cap = cv.VideoCapture(0) #webcam no.1

# read until the end
# Step-3 Display frame by frame
while(cap.isOpened()):
     #capture frame by frame
     ret, frame = cap.read ()
     if ret == True:
           # to display frame
           cv. imshow("Frame", frame)
           # to quit with q key
           if cv.waitKey(1) & 0XFF == ord('q'):
                break
     else:
           break

# Step-4 release or close windows easily
cap.release()
cv.destroyAllWindows ()
```

10 Ch 10 change color of webcam

[ ]:
```
import cv2 as cv
import numpy as np
cap = cv.VideoCapture(0)
while(True):
    (ret, frame) = cap.read()
    gray_frame = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    (thresh, binary)= cv.threshold(gray_frame, 127, 255, cv. THRESH_BINARY)

    cv.imshow("OriginalCam", frame)
    cv.imshow("GrayCam", gray_frame)
    cv.imshow("BinaryCam", binary)
    if cv.waitKey(1) & 0XFF == ord('q'):
        break
cap.release()
```

```
cv.destroyAllWindows ()
```

11 Ch 11 Writing videos from cam

[ ]:
```
# Writing videos from cam
import cv2 as cv
import numpy as np
cap=cv.VideoCapture(0)

# writing format, codec, video writer object and file output
# Obtain frame size information using get() method
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
frame_size = (frame_width,frame_height)
fps = 15

out = cv.VideoWriter("resourses/cam_video.avi", cv.VideoWriter_fourcc('M', "j",␣
 ↪'P', 'G'), 15, frame_size)

# fourcc: 4-character code of codec, used to compress the frames (fourcc)
# fps: Frame rate of the created video stream


if (cap.isOpened () == False):
    print("error in reading video")

    #reading and playing
while(cap.isOpened ()):
    ret, frame = cap.read()
    if ret == True:
        out.write(frame)
        cv.imshow("Video", frame)
        if cv.waitKey(25) & 0XFF == ord('q'):
            break
    else:
        break
cap.release()
out.release()
cv.destroyAllWindows ()
```

12 Ch 12 Setting of camera or video

[ ]:
```
# Setting of camera or video
import cv2 as cv
import numpy as np
cap = cv.VideoCapture(0)
cap.set(10, 100) # 10 is the key to set brightness
cap.set(3, 640) # width key 3
cap.set(4, 480) #height key 4
while (True):
     ret, frame = cap.read()
     if ret == True:
          cv.imshow("frame", frame)
          if cv.waitKey(1) & 0XFF == ord('q'):
               break
     else:
          break
cap.release()
cv.destroyAllWindows ()
```

13 Ch 13 basic functions

[ ]:
```
# basic functions or manipulation in oen cv
import cv2 as cv
img = cv.imread("resourses/image.jpg")
img = cv.resize(img, (800,600))
#resize
resized_img = cv.resize(img, (350, 250))
# gray
gray_img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
# binary
(thresh, binary)= cv.threshold(gray_img, 127, 255, cv.THRESH_BINARY)
# blurred image
blurr_img = cv.GaussianBlur(img, (7,7), 0)
# edge detection
edge_img = cv.Canny(img, 53,53)
# thickness of lines
mat_kernel = np.ones((3,3), np.uint8)
dilated_img= cv.dilate(edge_img, (mat_kernel), iterations=1)
# Make thinner outline
ero_img = cv.erode(dilated_img, mat_kernel, iterations=1)
#cropping we will use numpy library not open cv
print("The size of our image is: ", img.shape)
cropped_img = img[0:500, 150:400]
```

```
#cv.imshow("Original", img)
#cv.imshow("Resized", resized_img)
#cv.imshow("Gray", gray_img)
#cv.imshow('Black and White', binary)
#cv.imshow("Blurr", blurr_img)
#cv.imshow("edge", edge_img)
#cv.imshow("Dilated", dilated_img)
#cv.imshow("Erosion", ero_img)
cv.imshow("Cropped", cropped_img)


cv.waitKey(0)
cv.destroyAllWindows ()
```

```
The size of our image is:  (600, 800, 3)
```

14 Ch 14 How to draw lines, and shapes in python

[ ]:
```
# How to draw lines, and shapes in python
import cv2 as cv
import numpy as np
# Draw a canvas e is for Black
img = np.zeros((600,600)) # black
img1 = np.ones((600,600))
# print size
print("The size of our canvas is: ", img.shape)
# print(img)

#adding colors to the canvas
colored_img = np.zeros( (600,600, 3), np.uint8)   #color channel addition

colored_img[:] = 255,0, 255 #color complete image

colored_img[150:230, 180:500] = 255,168, 10 # part of image to be colored

#adding line
cv.line(colored_img, (0,0), (colored_img.shape[0], colored_img.shape[1]),␣
 ↪(255,0,0) ,3) #croosed line
cv.line(colored_img, (100,100), (300, 300), (255,255,50) ,3) # part line

#adding rectangles
cv.rectangle(colored_img, (50, 100), (300, 400), (255,255,255), 3) # draw
cv.rectangle(colored_img, (50, 100), (300, 400), (255,255,255), cv.FILLED) #␣
 ↪draw

# adding circle
```

```
cv.circle(colored_img, (400,300), 50, (255,100,0), 5) # draw
cv.circle(colored_img, (400,300), 50, (255,100,0), cv.FILLED) # draw

#adding text
y0, dy, text = 500,50, "Python ka Chilla \non Codanics Youtube Channel"
for i, line in enumerate(text.split('\n')):
    y = y0 + i*dy
    cv.putText(colored_img, line, (50, y ), cv.FONT_HERSHEY_SIMPLEX, 1,␣
 ↪(255,255, 0), 2, cv.LINE_AA, False)


# kv.imshow("Black", img)
# cv.imshow("White", img1)
cv.imshow("Colored", colored_img)

cv.waitKey(0)
cv.destroyAllWindows ()
```

```
The size of our canvas is:  (600, 600)
```

15 Ch 15 Resolution of cam

[ ]:
```
# Reolution of cam
import cv2 as cv
import numpy as np
cap = cv.VideoCapture(0)

cap = cv.VideoCapture(0)

#resolution HD (1280x720)
def hd_resolution():
    cap.set (3, 1280)
    cap.set(4, 720)
def sd_resolution():
    cap.set(3, 640)
    cap.set(4, 480)
def fhd_resolution():
    cap.set(3, 1920)
    cap.set(4, 1080)

cap.set(cv.CAP_PROP_FPS, 30)

#sd_resolution()
hd_resolution()
#fhd_resolution()

while(True):
```

```
    ret, frame = cap.read()
    if ret ==True:
        cv. imshow("Camera", frame)
        if cv.waitKey(1) & 0XFF == ord('q'):
            break

    else:
       break
cap.release()
cv.destroyAllWindows ()
```

16 Ch 16 Saving HD recording of Cam steaming

[ ]:
```
# Saving HD recording of Cam steaming

import cv2 as cv
import numpy as np
cap = cv.VideoCapture(0)

cap = cv.VideoCapture(0)

#resolution HD (1280x720)
def hd_resolution():
    cap.set (3, 1280)
    cap.set(4, 720)
def sd_resolution():
    cap.set(3, 640)
    cap.set(4, 480)
def fhd_resolution():
    cap.set(3, 1920)
    cap.set(4, 1080)


#sd_resolution()
hd_resolution()
#fhd_resolution()

# Obtain frame size information using get() method
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
frame_size = (frame_width,frame_height)
fps = 30

out = cv.VideoWriter("resourses/cam_video.avi", cv.VideoWriter_fourcc('M', "j",␣
 ↪'P', 'G'), 30, frame_size)
```

```
# fourcc: 4-character code of codec, used to compress the frames (fourcc)
# fps: Frame rate of the created video stream


if (cap.isOpened () == False):
    print("error in reading video")

    #reading and playing
while(cap.isOpened ()):
    ret, frame = cap.read()
    if ret == True:
        out.write(frame)
        cv.imshow("Video", frame)
        if cv.waitKey(25) & 0XFF == ord('q'):
            break
    else:
        break
cap.release()
out.release()
cv.destroyAllWindows ()
```