OpenCV Tutorial
- OpenCV Tutorial
- OpenCV Installation
- Read & Save Images
- Basic Operation On images
- OpenCV Resize Image
- OpenCV Image Rotation
- OpenCV Drawing Functions
- OpenCV Blob Detection
- Canny Edge Detection
- OpenCV Gaussian Blur
- OpenCV Image Filters
- OpenCV Image Threshold
- OpenCV Contours
- OpenCV Mouse Event
- OpenCV Template Matching
- OpenCV Erosion & Dilation
- **OpenCV Video Capture**
- Face Recognition & Face Detection
- Limitations in Face Detection

← prev | next →

OpenCV VideoCapture

OpenCV provides the **VideoCature()** function which is used to work with the Camera. We can do the following task:

- Read video, display video, and save video.
- Capture from the camera and display it.

Capture Video from Camera

OpenCV allows a straightforward interface to capture live stream with the camera (webcam). It converts video into grayscale and display it.

We need to create a **VideoCapture** object to capture a video. It accepts either the device index or the name of a video file. A number which is specifying to the camera is called device index. We can select the camera by passing the O or 1 as an argument. After that we can capture the video frame-by-frame.

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

while(True):
    # Capture image frame-by-frame
    ret, frame = cap.read()

    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Display the resulting frame
    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

The **cap.read()** returns a boolean value(True/False).It will return True, if the frame is read correctly.

Playing Video from file

We can play the video from the file. It is similar to capturing from the camera by changing the camera index with the file name. The time must be appropriate for **cv2.waitKey()** function, if time is high, video will be slow. If time is too less, then the video will be very fast.

```
import numpy as np
import cv2

cap = cv2.VideoCapture('filename')

while(cap.isOpened()):
    ret, frame = cap.read()
## will open the camera in the grayscale mode
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

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

cap.release()
cv2.destroyAllWindows()
```

Saving a Video

The **cv2.imwrite()** function is used to save the video into the file. First, we need to create a VideoWriter object. Then we should specify the **FourCC** code and the number of frames per second (fps). The frame size should be passed within the function.

FourCC is a 4-byte code used to identify the video codec. The example is given below for saving the video.

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)

# Define the codec and create VideoWriter object
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))

while(cap.isOpened()):
    ret, frame = cap.read()
    if ret==True:
        frame = cv2.flip(frame,0)

        # write the flipped frame
        out.write(frame)

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

# Release everything if job is finished
cap.release()
out.release()
cv2.destroyAllWindows()
```

It will save the video at the desired location. Run the above code and see the output.

Next Topic Face Recognition & Face Detection

← prev | next →