Capturing Image from Camera

Instead of using a saved image for Image Processing, OpenCV gives the provision to use an in-built webcam or any attached camera for the same. OpenCV provides certain functions through which camera can be initialized, each frame from the camera can be read and displayed, capture and save any frame, capture video from camera and save it.

Functions:

(a) cv2.VideoCapture(filename) or **cv2.VideoCapture(device) → VideoCapture object**

Use: To initialise or create an object for **VideoCapture** class in OpenCV that deals with camera related methods or functions.

Parameters:

- **filename:** name of the opened video file (e.g. video.avi) or image sequence
- **device:** id of the opened video capturing device (i.e. a camera index); if there is a single camera connected, just pass **0**

This function acts like a gateway to use other methods or functions of **VideoCapture** class, hence it should be called first while dealing with camera. The following functions of this class is then accessible via the object created by this constructor.

(b) cv2.VideoCapture.**read() → retval, image**

Use: To grab, decode and return the next video frame.

Parameters:

- **retval: True** if a frame is correctly grabbed by the camera and loaded into a variable as multi-dimensional NumPy array image; **False** if camera is not initialized properly, no frame is grabbed, camera has been disconnected, or there are no more frames in video file
- **image:** if camera is properly initialized and grabs frames, then this function returns the frame as multi-dimensional NumPy array image

This function is the most convenient method for reading video files or capturing data from decode and return the just grabbed frame.

(c) cv2.VideoCapture.**release()** → **None**

- **Use:** To close video file or feed from a capturing device and release it. This function also deallocates any associated memory.

(d) cv2.VideoCapture.**open(filename)** or cv2.VideoCapture.**open(device)** → **retval**

Use: To open a video file or a capturing device for video capturing.

Parameters:

- **filename:** name of the opened video file (e.g. video.avi) or image sequence
- **device:** id of the opened video capturing device (i.e. a camera index)

This function first calls cv2.VideoCapture.**release()** to close the already opened file or camera.

(e) cv2.VideoCapture.**isOpened()** → **retval**

Use: To check if video capturing is initialized.

Parameters:

- **retval:** **True** if the previous call to **VideoCapture** constructor or cv2.VideoCapture.**open()** succeeded, else returns **False**

Example Program 1: This program initializes the webcam of the system, reads the first frame from the camera into a variable and displays it in a window until any key is pressed. When a key is pressed, the camera is released, deallocates the associated memory and closes all windows.

```
# import OpenCV
import cv2

# initialize in-built web camera of the system,
# setting device id to 0
cap = cv2.VideoCapture(0)

# read the first frame from camera
ret, image = cap.read()

# display the captured frame in a window
cv2.imshow('captured frame', image)
```

```
# wait until any key is pressed
cv2.waitKey(0)

# close or release the camera, deallocate associated memory
cap.release()

# destroy or close all windows
cv2.destroyAllWindows()
```

Example Program 2: This program initializes the webcam of the system, reads the frames one after the another from the camera into a variable and displays it in a window in a while loop, hence a continuous video feed is displayed even if the camera is capturing single frame at a time. This is because the rate of displaying each captured frame in a window is much higher than our persistence of vision to differentiate between any frame. The while loop continues until **'ESC'** (Escape) key is pressed. When **ESC** key is pressed, the frame captured by the camera at that moment is saved as an image file in the current directory and the while loop exits. The camera is released, the associated memory is deallocated and all windows are closed.

```
# import OpenCV
import cv2

# initialize in-built web camera of the system,
# setting device id to 0
cap = cv2.VideoCapture(0)

# continuous loop to display every frame captured
# by the camera one after another
while(1):
    # read the frame from camera
    ret, image = cap.read()

    # display the captured frame in a window
    cv2.imshow('captured frame', image)

    # wait until 'ESC' key is pressed
    if cv2.waitKey(1) == 27:  # 27 - ASCII value for 'ESC' key
        # save the frame at this moment as a image file
        # in the current directory
        cv2.imwrite('captured frame.png', image)
        break  # end the while loop

# release the camera and destroy all windows
cap.release()
cv2.destroyAllWindows()
```