# Video processing and colour conversion

## 1.Read and Write Videos:

- Read Local Videos: You can use OpenCV’s `cv2.VideoCapture()` to read videos from files, and `cv2.VideoWriter()` to write videos back to disk.

- Example:

```
import cv2
cap = cv2.VideoCapture('video_file.mp4')
while(cap.isOpened()):
   ret, frame = cap.read()
   if ret:
      cv2.imshow('Frame', frame)
      if cv2.waitKey(1) & 0xFF == ord('q'):
         break
cap.release()
cv2.destroyAllWindows()
```

- Write to Video: To save a video, can define the video codec and resolution.

```
   out = cv2.VideoWriter('output_video.mp4', cv2.VideoWriter_fourcc(*'XVID'), 20.0, (640, 480))
   out.write(frame)
```

## 2. Read from DroidCam:

- To read video from DroidCam (which turns your phone into a webcam), can access it using the webcam index provided by DroidCam or by setting up a network stream.

**cap = cv2.VideoCapture(0) ....as a default webcam**

-Example:

```
import cv2
# Open the video capture from the first webcam (which should be DroidCam if set as default)
cap = cv2.VideoCapture(0)

if not cap.isOpened():
   print("Error: Could not open video stream")
   exit()

while True:
   ret, frame = cap.read()
   if not ret:
      print("Error: Failed to grab frame")
      break

   # Display the frame from DroidCam
```

```
    cv2.imshow('DroidCam Feed', frame)

    # Break the loop when the user presses the 'q' key
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the video capture object and close the display window
cap.release()
cv2.destroyAllWindows()
```

## 3. Read from Live Streaming (RTSP):

- You can use RTSP streaming to read live video feeds from cameras or streaming servers.
- Example of connecting to an RTSP stream:

```
    cap = cv2.VideoCapture('rtsp://username:password@IP_address:port/stream')
    while cap.isOpened():
        ret, frame = cap.read()
        if ret:
            cv2.imshow('Live Stream', frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    cap.release()
    cv2.destroyAllWindows()
```

-Stream URL and credentials must be correct.

## 4. Understand Colour Conversion:

- OpenCV supports many color conversions using “cv2.cvtColor()”. For example, converting from BGR to RGB or grayscale, we use the flag “cv.COLOR_BGR2GRAY”. Similarly for BGR to HSV, we use the flag “cv.COLOR_BGR2HSV”.

- Example of converting BGR to grayscale:

```
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

## 5. Difference Between Image Manipulation and Pixel Manipulation:

- Image Manipulation involves higher-level operations like resizing, rotating, cropping, and applying filters (blurring, sharpening, etc.).
- Pixel Manipulation involves direct access to pixel values, which is useful for fine-grained control like thresholding, color changes, or custom image filters.

- Image Manipulation example (Resizing):

```
    resized_image = cv2.resize(image, (width, height))
```

- Pixel Manipulation example (Changing pixel value):

```
    image[100, 100] = [255, 0, 0]  # Change the pixel at (100, 100) to red
```