Lab 2

Working with Videos in OpenCV

Learning Objectives:

- *Read and write videos using OpenCV (cv2.VideoCapture, cv2.VideoWriter).*
- *Access and process webcam video frame by frame.*
- *Draw shapes and overlay text on video frames.*
- *Convert video frames to grayscale.*
- *Save processed videos with different codecs.*
- *Prompt Engineering (Bonus Task).*

Introduction:

This lab focuses on handling videos in OpenCV, including reading and writing videos, accessing the webcam, processing frames, drawing shapes, and saving video output using different codecs. By the end of this lab, you will also implement edge detection on video frames as a bonus task. These techniques are widely used in real-world applications such as surveillance, motion detection, and video analysis.

Exercise 1: Reading and Writing Videos

Task A) **Read and Display a Video File**
Real-World Application: Used in video analysis software to process and analyze recorded footage, such as security surveillance or traffic monitoring

```
import cv2

# Open video file
cap = cv2.VideoCapture('Video.mp4')

while True:
    ret, frame = cap.read()

    if not ret:
        break

    cv2.imshow('Video Playback', frame)

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

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

B) Write Video File
Real-World Application: Used in media production and streaming services to process and export video files in various formats.

```
import cv2

# Open video file
cap = cv2.VideoCapture('input_video.mp4')
# Define codec and create VideoWriter object for MP4
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter('output_video.mp4', fourcc, 20.0, (640, 480))

while True:
    ret, frame = cap.read()

    out.write(frame)

    """There Is A missing code here which display the webcam stream frame,
write it"""

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

Exercise 2: Accessing Webcam and Processing Frames

Real-World Application: Commonly used in any real time AI-based demo you would like to present

```
import cv2

# Open video file
cap = cv2.VideoCapture("""Find what shall we write here to open the webcam""")

while True:
    ret, frame = cap.read()

    cv2.imshow("Webcam", frame)

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

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

Exercise 3: Drawing Shapes and Adding Text

Real-World Application: Commonly used in any real time AI-based demo you would like to present and visualization to it

```
import cv2
# Open webcam
cap = cv2.VideoCapture(0)
while cap.isOpened():
    """There Is A missing code here which read the frame, write it"""

    # Draw a rectangle
    cv2.rectangle(frame, (50, 50), (300, 300), (0, 255, 0), 3)
    # Add text
    cv2.putText(frame, 'Webcam Feed', (50, 40), cv2.FONT_HERSHEY_SIMPLEX, 1,
(255, 0, 0), 2)

    cv2.imshow('Shapes and Text', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Exercise 4: Converting Video Frames to Grayscale

Real-World Application: Real-World Application: Used in medical imaging and object detection systems where color is unnecessary or distracting.

```
import cv2

cap = cv2.VideoCapture('input_video.mp4')
#Another type of video extension XVID
fourcc = cv2.VideoWriter_fourcc(*'XVID')

out = cv2.VideoWriter('gray_video.avi', fourcc, 20.0, (640, 480),
isColor=False)
while cap.isOpened():
    ret, frame = cap.read()


    gray_frame = """There Is A missing code here which convert the frame to
grayscale, write it"""
    out.write(gray_frame)
    cv2.imshow('Grayscale Video', gray_frame)

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

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

Bonus Task: Advanced Video Edge Detection and Recording

This task will challenge your ability to transform ideas into functional code, simulate real-world scenarios, and refine your troubleshooting skills. Imagine you need to implement a video processing system and want AI assistance to write the code. Your goal is not only to generate the code but also to test, debug, and optimize it for accuracy and performance.

Your Objective

Write a Python program that performs the following real-time video processing tasks:

✅ Open the webcam and capture the live video stream.
✅ Record the processed video while applying real-time transformations.
✅ Flip the video horizontally to create a mirrored effect.
✅ Apply edge detection to highlight contours dynamically.
✅ Overlay red text with your name at the center of the frame.
✅ Stop the recording when the 'q' key is pressed.
✅ Save the recorded video in a specified directory for later use.

Real-World Application

This exercise mimics real-world applications such as motion tracking, security surveillance, and live video effects used in media and augmented reality. It also simulates the software development workflow, requiring you to generate, test, and troubleshoot your code like a real developer.

💡 **Pro Tip: If the code doesn't work as expected, analyze the errors, modify the approach, and refine it step by step. Debugging is part of the learning process!** 🚀

References

- **OpenCV: Reading and Writing Videos**
- **OpenCV: Drawing Functions**
- **OpenCV: Edge Detection**
- **Prompt Engineering: Video**