Digital Signal Processing Lab

Demo 22 - Exercise 1 (Video processing)

Saad Zubairi
shz2020

November 17, 2025

Solution

I started from the demo file `D5 - video operations/blur_video.py`. That script already opened the webcam, blurred each frame, handled the keyboard shortcuts, and saved a snapshot when `p` was pressed. To convert it into an edge detector I only had to swap the filtering step for a high-pass filter:

- Replaced the Gaussian blur with a 3×3 Laplacian-style kernel so high-frequency components (edges) are emphasized instead of suppressed.
- Applied `cv2.filter2D` with that kernel.

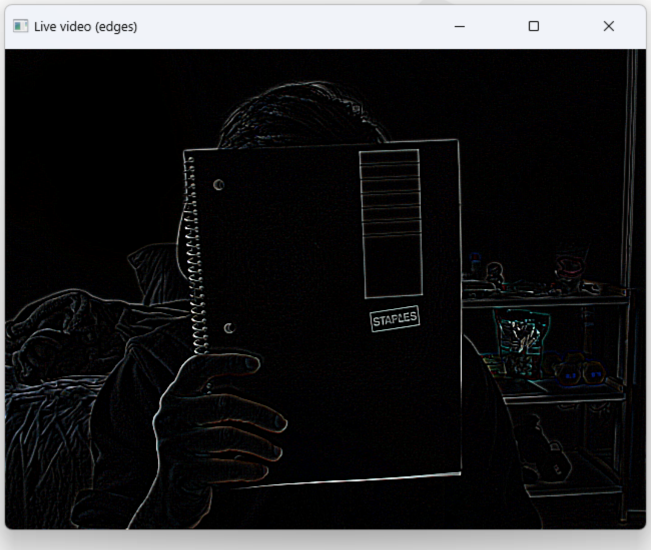
The rest of the loop (capture, display, save) stayed untouched.

```
kernel = np.array([[-1, -1, -1],
                   [-1,  8, -1],
                   [-1, -1, -1]])

[ok, frame] = cap.read()
frame = cv2.filter2D(frame, -1, kernel)
cv2.imshow('Live video (edges)', frame)
```

Snippet 1: Key change for edge detection

Screenshots



Addendum: Full implementation

```python
# blur_video.py
# Demonstrates 2D spatial filtering

import numpy as np
import cv2

cap = cv2.VideoCapture(0)

print("Switch to video window. Then press 'p' to save image, 'q' to quit")

# High-pass kernel highlights edges instead of smoothing
kernel = np.array([[-1, -1, -1],
                   [-1,  8, -1],
                   [-1, -1, -1]])

while True:

    [ok, frame] = cap.read()          # Read one frame

    # Use 2D filtering with the high-pass kernel to emphasize edges
    frame = cv2.filter2D(frame, -1, kernel)

    cv2.imshow('Live video (edges)', frame)

    key = cv2.waitKey(1)
    # key = key & 0xFF      # (May not be necessary)

    if key == ord('p'):
        cv2.imwrite('edges.jpg', frame)

    if key == ord('q'):
        break

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

Snippet 2: Full implementation