```python
# Import Libraries
import cv2
import sys
import numpy as np

PREVIEW = 0  # Preview Mode
BLUR = 1  # Blurring Filter
FEATURES = 2  # Corner Feature Detector
CANNY = 3  # Canny Edge Detector

# maxCorners --> Maximum Number of Corners Algorithm Will Return
# qualityLevel --> Minimum Threshold for Filtering Corner Features
# minDistance --> Minimum Distance Between Pixel Space. How Close Two Corner Features Can be in the List.
# blockSize --> Size of the Pixel Neighbourhood

feature_params = dict(maxCorners=500, qualityLevel=0.2, minDistance=15, blockSize=9)

s = 0
if len(sys.argv) > 1:
    s = sys.argv[1]

image_filter = PREVIEW
alive = True

windows_name = "Camera Filters"
cv2.namedWindow(windows_name, cv2.WINDOW_NORMAL)
result = None

source = cv2.VideoCapture(s)

while alive:
    has_frame, frame = source.read()
    if not has_frame:
        break

    # Use flipCode==0 (vertically),  flipCode>0 (Horizontaly), flipCode<0 (vertical & Horizontal)
    frame = cv2.flip(frame, 1)

    if image_filter == PREVIEW:
        result = frame
    elif image_filter == CANNY:
        # cv2.Canny(img, Threshold_lower, Threshold_upper) --> to detect the edges in an image
        result = cv2.Canny(frame, 80, 150)
    elif image_filter == BLUR:
        # cv2.blur(src, ksize, dst, anchor, borderType) --> A tuple representing the blurring kernel size
        result = cv2.blur(frame, (13, 13))
    elif image_filter == FEATURES:
        result = frame
        frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        corners = cv2.goodFeaturesToTrack(frame_gray, **feature_params)
        if corners is not None:
            for X, Y in np.float32(corners).reshape(-1, 2):
                cv2.circle(result, (X, Y), 10, (0, 255, 0), 1)

    cv2.imshow(windows_name, result)

    key = cv2.waitKey(0)

    if key == ord("Q") or key == ord("q") or key == 27:
        alive = False
    elif key == ord("C") or key == ord("c"):
        image_filter = CANNY
    elif key == ord("B") or key == ord("b"):
        image_filter = BLUR
    elif key == ord("F") or key == ord("f"):
        image_filter = FEATURES
    elif key == ord("P") or key == ord("p"):
        image_filter = PREVIEW

source.release()
cv2.destroyWindow(windows_name)
```