

PROGRAM 7

Program Title

Capture Live Image, Mark Region of Interest (ROI), and Display using OpenCV

Aim

To capture a live image using a camera, mark a Region of Interest (ROI), and display the image using OpenCV.

Components / Requirements

- Jetson Nano
 - USB Camera
 - OpenCV library
-

Pre-Code Setup (Libraries & Prerequisites)

OpenCV Installation

```
sudo apt update  
sudo apt install python3-opencv -y
```

Camera Availability Check

```
ls /dev/video*
```

Expected device:

```
/dev/video0
```

Algorithm / Logic

1. Initialize the USB camera using OpenCV.

2. Continuously capture live frames from the camera.
3. Define static ROI coordinates.
4. Draw a rectangular ROI on each frame.
5. Display the image with ROI overlay.
6. Save the image with ROI overlay.
7. Terminate the program when the user presses the 'q' key.
8. Release camera resources and close all OpenCV windows.

Libraries Used

- cv2
- time

Program Code

```
import cv2
import time

# ===== CONFIG =====
SAVE_PATH = "/home/aiml/Pictures/live_annotated.jpg"
# =====

def LiveROI():
    print("Opening camera...")
    cam = cv2.VideoCapture(0)

    if not cam.isOpened():
        print("Camera not accessible")
        return

    time.sleep(2) # Camera warm-up

    print("Live feed started (press 'q' to quit)")

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

        if not ret:
            print("Frame not received")
            break

        # Get frame dimensions
        h, w, _ = frame.shape
```

```

# ----- ROI (Region of Interest) -----
start_point = (0, int(h * 0.7))      # Bottom-left
end_point = (int(w * 0.3), h)        # Bottom-right

# Draw ROI rectangle (overlay only)
cv2.rectangle(
    frame,
    start_point,
    end_point,
    (0, 0, 255), # Red color (BGR)
    thickness=3
)

# Add annotation label
cv2.putText(
    frame,
    "pen_stand",
    (start_point[0] + 5, start_point[1] - 10),
    cv2.FONT_HERSHEY_SIMPLEX,
    0.6,
    (255, 0, 0), # Blue color (BGR)
    2
)

# Display live annotated feed
cv2.imshow("Live ROI Annotation", frame)

# Exit and save frame on 'q' key press
if cv2.waitKey(1) & 0xFF == ord('q'):
    cv2.imwrite(SAVE_PATH, frame)
    print("Final annotated frame saved")
    break

cam.release()
cv2.destroyAllWindows()
print("Camera released, program ended")

# ===== MAIN =====
if __name__ == "__main__":
    LiveROI()

```

Output / Result

- Live camera feed is displayed with a rectangular ROI overlay.

- The image with ROI overlay is saved as `roi_output.jpg`.
 - The program terminates when the user presses the 'q' key.
-

Possible Viva / Exam Questions

1. What is a Region of Interest (ROI)?
 2. How is ROI marked in OpenCV?
 3. Why is `cv2.waitKey()` used?
 4. How does the program terminate?
-

Conclusion

The program successfully demonstrates capturing live images, marking a Region of Interest using OpenCV, and displaying and saving the overlaid image.