In [ ]:

```python
import cv2
import numpy as np

def is_fire_color(hsv_pixel):
    """
    Determines if a given HSV pixel is likely to be a fire pixel based on its color.
    Adjust the thresholds as necessary.
    """
    h, s, v = hsv_pixel
    if (h >= 0 and h <= 50) and (s >= 50 and s <= 255) and (v >= 200 and v <= 255):
        return True
    return False

def detect_fire(frame):
    """
    Detect fire in a video frame.
    """
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    fire_mask = np.zeros((frame.shape[0], frame.shape[1]), dtype=np.uint8)
    for y in range(hsv_frame.shape[0]):
        for x in range(hsv_frame.shape[1]):
            if is_fire_color(hsv_frame[y, x]):
                fire_mask[y, x] = 255

    return fire_mask

def main():
    video_path = "video_0.gif"  # Replace with your video file path or camera index
    cap = cv2.VideoCapture(video_path)

    if not cap.isOpened():
        print(f"Error: Could not open video file {video_path}")
        return

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            print("Error: Could not read frame. Exiting...")
            break

        fire_mask = detect_fire(frame)
        total_pixels = frame.shape[0] * frame.shape[1]
        fire_pixels = np.sum(fire_mask) // 255  # Number of fire pixels
        fire_percentage = (fire_pixels / total_pixels) * 100

        # Find contours in the fire mask
        contours, _ = cv2.findContours(fire_mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        for contour in contours:
            x, y, w, h = cv2.boundingRect(contour)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

        # Display the percentage of fire in the frame
        cv2.putText(frame, f"Fire Percentage: {fire_percentage:.2f}%", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

        cv2.imshow('Fire Detection', frame)

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

    cap.release()
    cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```