```
# Healthcare Imaging Analysis – Brain Anomaly & Hand Fracture
Detection
import cv2
import numpy as np

# -------------------- Brain Anomaly Detection
--------------------

print("🔍 Starting Brain X-ray Anomaly Detection...")

brain_img = cv2.imread('brain.jpeg', cv2.IMREAD_GRAYSCALE)
if brain_img is None:
    print("❌ Brain image not found!")
else:
    blurred_brain = cv2.GaussianBlur(brain_img, (5, 5), 0)
    _, thresh = cv2.threshold(blurred_brain, 200, 255,
cv2.THRESH_BINARY)

    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    brain_output = cv2.cvtColor(brain_img, cv2.COLOR_GRAY2BGR)

    anomaly_found = False
    for cnt in contours:
        area = cv2.contourArea(cnt)
        if area > 200:
            anomaly_found = True
            x, y, w, h = cv2.boundingRect(cnt)
            cv2.rectangle(brain_output, (x, y), (x + w, y + h),
(0, 0, 255), 2)
            cv2.putText(brain_output, "Possible Anomaly", (x, y
- 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0,
255), 1)

    if anomaly_found:
        print("✅ Possible anomalies detected in brain X-ray.")
    else:
        print("✅ No significant anomalies detected in brain X-
ray.")

    cv2.imshow("Brain Anomaly Detection", brain_output)

# -------------------- Hand Fracture Detection
--------------------
```

```
print("\n🔍 Starting Hand X-ray Fracture Detection...")

hand_img = cv2.imread('handfracture.jpeg',
cv2.IMREAD_GRAYSCALE)
if hand_img is None:
    print("❌ Hand X-ray image not found!")
else:
    hand_img = cv2.resize(hand_img, (500, 500))
    blurred_hand = cv2.GaussianBlur(hand_img, (5, 5), 0)
    edges = cv2.Canny(blurred_hand, 50, 150)

    kernel = np.ones((5, 5), np.uint8)
    dilated = cv2.dilate(edges, kernel, iterations=1)

    contours, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    hand_output = cv2.cvtColor(hand_img, cv2.COLOR_GRAY2BGR)

    fracture_found = False
    for cnt in contours:
        area = cv2.contourArea(cnt)
        length = cv2.arcLength(cnt, True)
        if area < 1000 and length > 200:
            fracture_found = True
            x, y, w, h = cv2.boundingRect(cnt)
            cv2.rectangle(hand_output, (x, y), (x + w, y + h),
(0, 0, 255), 2)
            cv2.putText(hand_output, "Possible Fracture", (x, y
- 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0,
255), 1)

    if fracture_found:
        print("✅ Possible fractures detected in hand X-ray.")
    else:
        print("✅ No significant fractures detected in hand X-
ray.")

    cv2.imshow("Original Hand X-ray", hand_img)
    cv2.imshow("Edge Map", edges)
    cv2.imshow("Hand Fracture Detection", hand_output)

# -------------------- Display All --------------------
```

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```