Function Documentation:`is_contour_inside`

1 Description

The `is_contour_inside` function determines whether all points of one contour (`contour1`) are inside another contour (`contour2`). It uses OpenCV's `cv2.pointPolygonTest` to perform this check.

2 Function Definition

```python
def is_contour_inside(contour1, contour2):
    # Check if all points of contour1 are inside contour2
    for i in range(len(contour1)):
        # Using cv2.pointPolygonTest to check each point of contour1 against contour2
        point = (int(contour1[i][0][0]), int(contour1[i][0][1]))
        if cv2.pointPolygonTest(contour2, point, False) < 0:
            return False
    return True
```

3 Function Explanation

3.1 Step-by-Step Breakdown

Function 1: Check Containment

Check if all points of `contour1` are inside `contour2`.

```python
for i in range(len(contour1)):
    # Using cv2.pointPolygonTest to check each point of contour1 against contour2
    point = (int(contour1[i][0][0]), int(contour1[i][0][1]))
    if cv2.pointPolygonTest(contour2, point, False) < 0:
        return False
```

Explanation: The function iterates through each point in `contour1` and uses `cv2.pointPolygonTest` to check if the point is inside `contour2`. If any point is found outside, the function returns `False`. If all points are inside, it returns `True`.

4 Conclusion

The `is_contour_inside` function provides a way to verify the spatial relationship between two contours by checking if all points of one contour are contained within another contour. This can be useful for various image processing and geometric applications.