# singdetector

April 4, 2025

[15]:
```
import os
import time
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from skimage import io

from classifier import load_exemplars, roi_to_vector, classify_roi
from roifinder import roi_detector, visualize_refined_mask_with_rois

#  Jupyter
%matplotlib inline

def sign_detector(image_path, exemplar_path):
    image = io.imread(image_path)

    rois, binary_mask, opened_mask, refined_mask, hue_mask, sat_mask, val_mask,␣
 ↪rejection_info = roi_detector(image, debug=True)

    for info in rejection_info:
        print(info)

    fig, axes = plt.subplots(2, 3, figsize=(18, 12))
    ax = axes.ravel()

    ax[0].imshow(image)
    ax[0].set_title('Original Image')
    ax[0].axis('off')

    ax[1].imshow(hue_mask, cmap='gray')
    ax[1].set_title('Hue Mask')
    ax[1].axis('off')

    ax[2].imshow(sat_mask, cmap='gray')
    ax[2].set_title('Saturation Mask')
    ax[2].axis('off')

    ax[3].imshow(val_mask, cmap='gray')
```

```
    ax[3].set_title('Value Mask')
    ax[3].axis('off')

    ax[4].imshow(binary_mask, cmap='gray')
    ax[4].set_title('Binary Mask')
    ax[4].axis('off')

    ax[5].imshow(refined_mask, cmap='gray')
    ax[5].set_title('Refined Mask')
    ax[5].axis('off')

    plt.tight_layout()
    plt.show()

    categories, descriptors = load_exemplars(exemplar_path)
    detected_speeds = []

    for roi in rois:
        roi_img = image[roi[0]:roi[2], roi[1]:roi[3]]
        vector = roi_to_vector(roi_img)
        label, debug_msg = classify_roi(vector, categories, descriptors,␣
↪distance_threshold=1.2, debug=True)
        print(f"ROI at {roi} classified as {label}. {debug_msg}")

    fig, ax = plt.subplots(figsize=(10, 8))
    ax.imshow(image)
    for (minr, minc, maxr, maxc) in rois:
        roi_img = image[minr:maxr, minc:maxc]
        roi_vec = roi_to_vector(roi_img)
        label = classify_roi(roi_vec, categories, descriptors)

        if label == -1:
            continue

        detected_speeds.append(label)

        rect = patches.Rectangle((minc, minr), maxc - minc, maxr - minr,
                                 fill=False, edgecolor='red', linewidth=2)
        ax.add_patch(rect)
        ax.text(minc, minr, f"Speed: {label}", color='white',
                fontsize=12, verticalalignment='top')

    ax.set_title('Detected Regions with 1-NN Classification')
    ax.axis('off')
    plt.show()

    return detected_speeds
```

```
def group_test_images(folder_path, exemplar_path, group_number):
    """
       group_number    input()
    """
    groups = {
        1: ["40-0001x1.png", "40-0002x1.png"],
        2: ["50-0001x1.png", "50-0002x1.png", "50-0003x1.png", "50-0004x2.png",␣
↪"50-0005x1.png"],
        3: ["60-0001x1.png", "60-0002x1.png", "60-0003x1.png", "60-0004x2.png",␣
↪"60-0005x1.png"],
        4: ["80-0001x1.png", "80-0002x2.png", "80-0003x2.png", "80-0004x2.png",␣
↪"80-0005x1.png"],
        5: ["100-0001x1.png", "100-0002x1.png", "100-0003x1.png", "100-0004x2.
↪png", "100-0005x1.png"],
        6: ["120-0001x1.png", "120-0002x2.png", "120-0003x1.png", "120-0004x1.
↪png", "120-0005x1.png"],
        7: ["50-0002x1.png", "80-0003x2.png"]
    }

    if group_number not in groups:
        print(f"Group {group_number} not defined. Available groups are:␣
↪{list(groups.keys())}")
        return

    images_to_test = groups[group_number]
    print(f"\nNow testing group {group_number} with images: {images_to_test}")

    for img in images_to_test:
        full_path = os.path.join(folder_path, img)
        if not os.path.exists(full_path):
            print(f"File not found: {full_path}")
            continue

        print(f"\nProcessing {img} ...")
        detected_speeds = sign_detector(full_path, exemplar_path)
        count = len(detected_speeds)
        if count == 0:
            print("No speed signs detected.")
        elif count == 1:
            print(f"Detected 1 speed sign: {detected_speeds[0]} km/h")
        else:
            speeds_str = ", ".join([f"{s} km/h" for s in detected_speeds])
            print(f"Detected {count} speed signs: {speeds_str}")
```

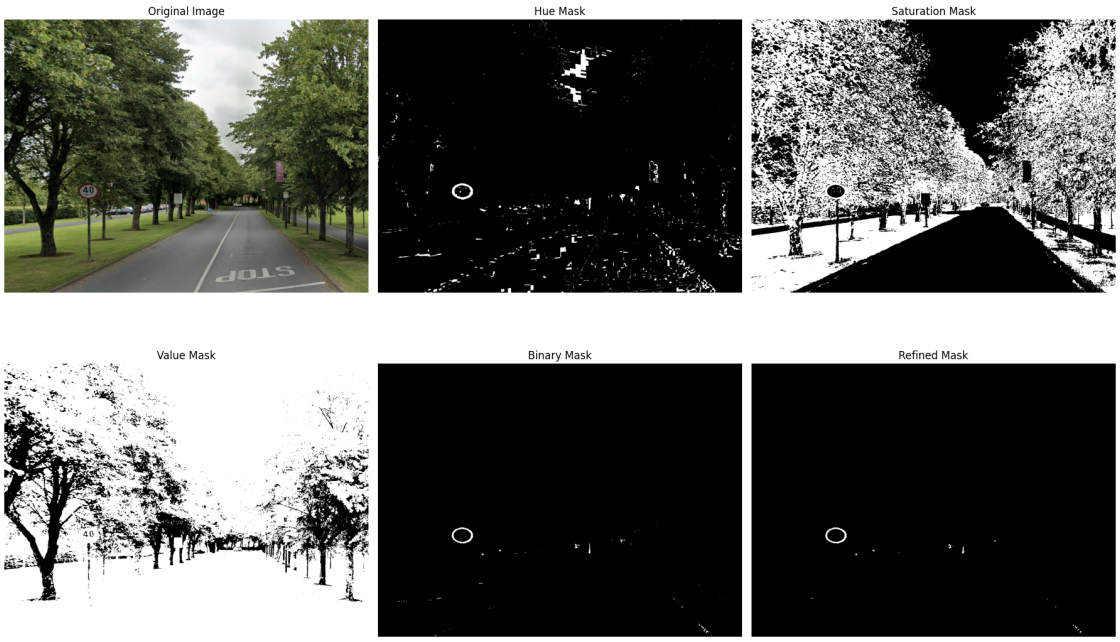
```
        plt.close('all')
        time.sleep(1)
```

[17]:
```
#    group
def group_test_images(folder_path, exemplar_path, group_number):
    groups = {
        1: ["40-0001x1.png", "40-0002x1.png"],
        2: ["50-0001x1.png", "50-0002x1.png", "50-0003x1.png", "50-0004x2.png",␣
 ↪"50-0005x1.png"],
        3: ["60-0001x1.png", "60-0002x1.png", "60-0003x1.png", "60-0004x2.png",␣
 ↪"60-0005x1.png"],
        4: ["80-0001x1.png", "80-0002x2.png", "80-0003x2.png", "80-0004x2.png",␣
 ↪"80-0005x1.png"],
        5: ["100-0001x1.png", "100-0002x1.png", "100-0003x1.png", "100-0004x2.
 ↪png", "100-0005x1.png"],
        6: ["120-0001x1.png", "120-0002x2.png", "120-0003x1.png", "120-0004x1.
 ↪png", "120-0005x1.png"],
        7: ["50-0002x1.png", "80-0003x2.png"]
    }

    if group_number not in groups:
        print(f"Group {group_number} not defined.")
        return

    images_to_test = groups[group_number]
    print(f"\n[Group {group_number}] Testing images: {images_to_test}")

    for img in images_to_test:
        full_path = os.path.join(folder_path, img)
        print(f"→ Now processing: {full_path}")

        if not os.path.exists(full_path):
            print(f"File not found: {full_path}")
            continue

        try:
            detected_speeds = sign_detector(full_path, exemplar_path)
            count = len(detected_speeds)
            if count == 0:
                print("No speed signs detected.")
            elif count == 1:
                print(f"Detected 1 speed sign: {detected_speeds[0]} km/h")
            else:
                speeds_str = ", ".join([f"{s} km/h" for s in detected_speeds])
                print(f"Detected {count} speed signs: {speeds_str}")

        except Exception as e:
```

```
            print(f"Error processing {img}: {e}")

        plt.close('all')
        time.sleep(1)

        #    group
def test_all_groups(folder_path, exemplar_path):
    for group_number in range(1, 8):  # group 1~7
        print(f"\n===== Testing Group {group_number} =====")
        group_test_images(folder_path, exemplar_path, group_number)
```

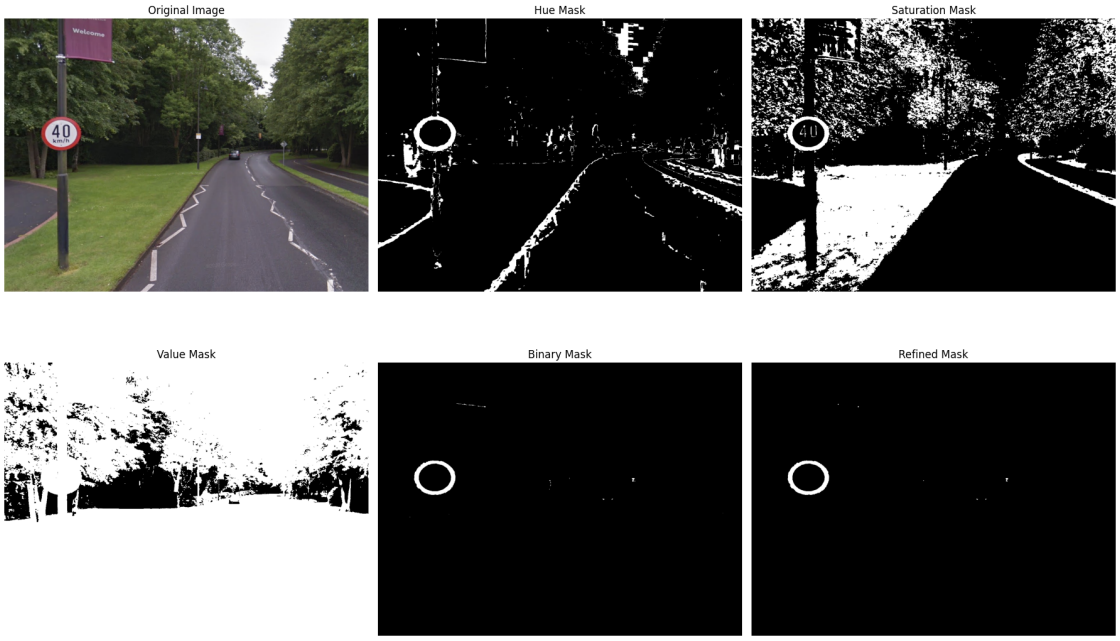
[ ]:
```
test_all_groups("image", "1-NN-descriptor-vects.npy")
```

```
===== Testing Group 1 =====

[Group 1] Testing images: ['40-0001x1.png', '40-0002x1.png']
→ Now processing: image\40-0001x1.png
Region 0: area = 1095. Width/Height = 1.32. Accepted.
Region 1: area = 18. Rejected: area too small (<=280).
Region 2: area = 25. Rejected: area too small (<=280).
Region 3: area = 54. Rejected: area too small (<=280).
Region 4: area = 5. Rejected: area too small (<=280).
Region 5: area = 105. Rejected: area too small (<=280).
Region 6: area = 31. Rejected: area too small (<=280).
Region 7: area = 5. Rejected: area too small (<=280).
Region 8: area = 29. Rejected: area too small (<=280).
Region 9: area = 5. Rejected: area too small (<=280).
Region 10: area = 8. Rejected: area too small (<=280).
Region 11: area = 5. Rejected: area too small (<=280).
Region 12: area = 8. Rejected: area too small (<=280).
Region 13: area = 11. Rejected: area too small (<=280).
Region 14: area = 33. Rejected: area too small (<=280).
Region 15: area = 10. Rejected: area too small (<=280).
Region 16: area = 8. Rejected: area too small (<=280).
Region 17: area = 26. Rejected: area too small (<=280).
```




```
ROI at (576, 259, 633, 334) classified as 40.0. Min distance: 0.473. Accepted.
```

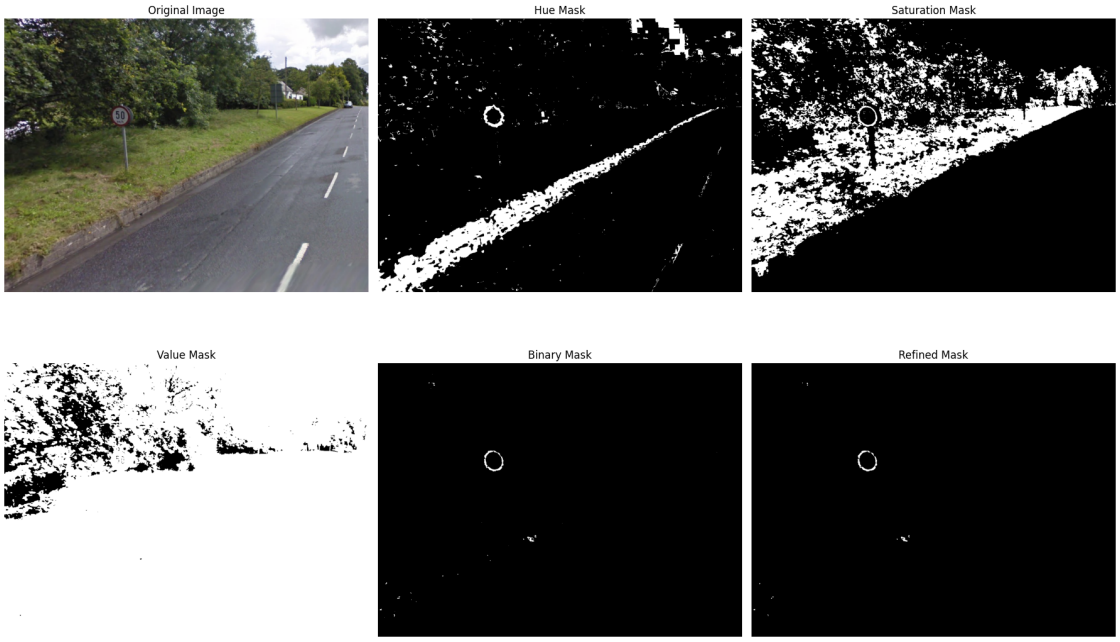
```
Detected 1 speed sign: 40.0 km/h
→ Now processing: image\40-0002x1.png
Region 0: area = 8. Rejected: area too small (<=280).
Region 1: area = 12. Rejected: area too small (<=280).
Region 2: area = 5242. Width/Height = 1.21. Accepted.
Region 3: area = 63. Rejected: area too small (<=280).
Region 4: area = 5. Rejected: area too small (<=280).
Region 5: area = 14. Rejected: area too small (<=280).
Region 6: area = 10. Rejected: area too small (<=280).
```




```
ROI at (344, 127, 465, 273) classified as 40.0. Min distance: 0.380. Accepted.
```

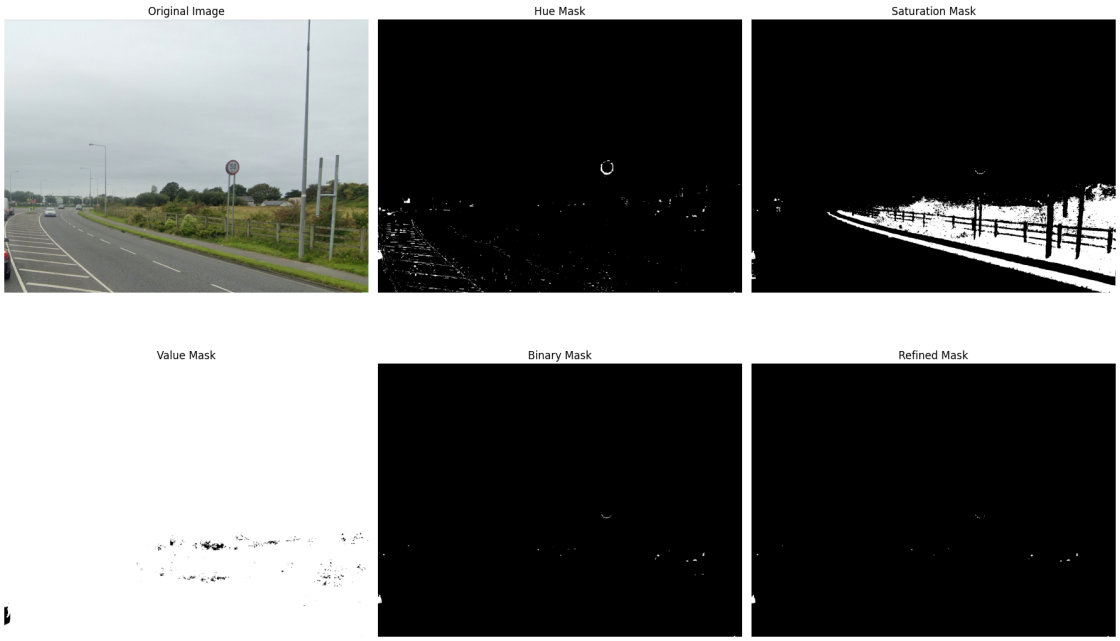

```
Detected 1 speed sign: 40.0 km/h

===== Testing Group 2 =====

[Group 2] Testing images: ['50-0001x1.png', '50-0002x1.png', '50-0003x1.png',
'50-0004x2.png', '50-0005x1.png']
→ Now processing: image\50-0001x1.png
Region 0: area = 5. Rejected: area too small (<=280).
Region 1: area = 16. Rejected: area too small (<=280).
Region 2: area = 8. Rejected: area too small (<=280).
Region 3: area = 295. Width/Height = 1.57. Accepted.
Region 4: area = 48. Rejected: area too small (<=280).
Region 5: area = 717. Width/Height = 1.28. Accepted.
Region 6: area = 26. Rejected: area too small (<=280).
Region 7: area = 173. Rejected: area too small (<=280).
Region 8: area = 10. Rejected: area too small (<=280).
Region 9: area = 10. Rejected: area too small (<=280).
Region 10: area = 11. Rejected: area too small (<=280).
Region 11: area = 10. Rejected: area too small (<=280).
Region 12: area = 21. Rejected: area too small (<=280).
```

Region 13: area = 5. Rejected: area too small (<=280).
Region 14: area = 5. Rejected: area too small (<=280).



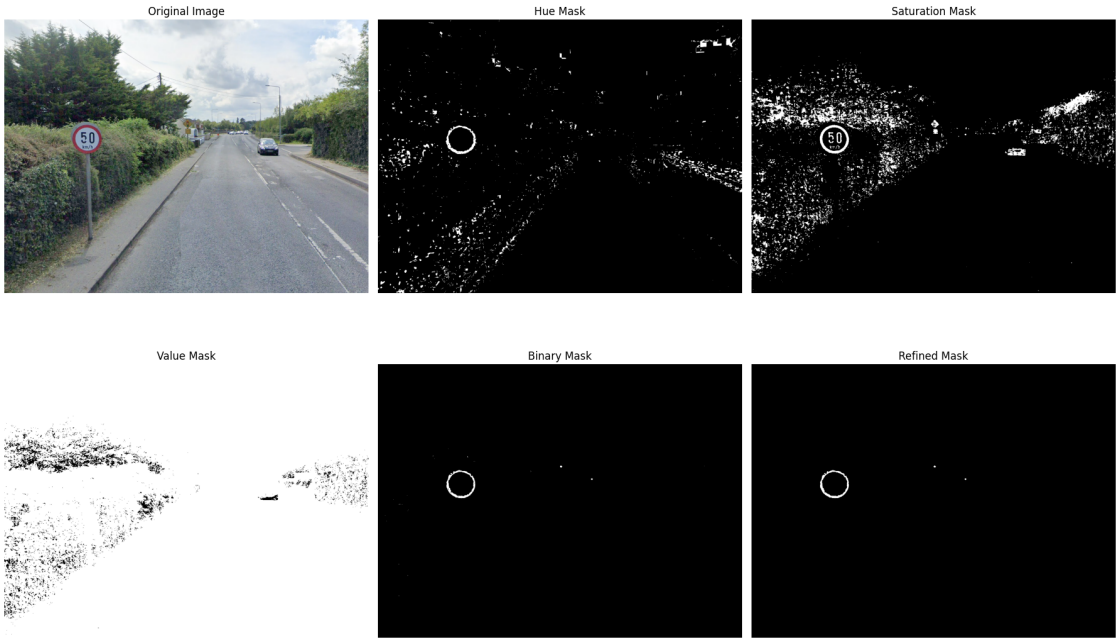

ROI at (307, 374, 335, 418) classified as -1. Min distance: 1.210. Distance 1.210 exceeds threshold 1.2.
ROI at (326, 373, 380, 442) classified as 50.0. Min distance: 1.081. Accepted.


```
Detected 1 speed sign: 50.0 km/h
→ Now processing: image\50-0002x1.png
Region 0: area = 8. Rejected: area too small (<=280).
Region 1: area = 5. Rejected: area too small (<=280).
Region 2: area = 10. Rejected: area too small (<=280).
Region 3: area = 5. Rejected: area too small (<=280).
Region 4: area = 5. Rejected: area too small (<=280).
Region 5: area = 5. Rejected: area too small (<=280).
Region 6: area = 12. Rejected: area too small (<=280).
Region 7: area = 41. Rejected: area too small (<=280).
Region 8: area = 20. Rejected: area too small (<=280).
Region 9: area = 34. Rejected: area too small (<=280).
Region 10: area = 5. Rejected: area too small (<=280).
Region 11: area = 5. Rejected: area too small (<=280).
Region 12: area = 44. Rejected: area too small (<=280).
Region 13: area = 20. Rejected: area too small (<=280).
Region 14: area = 304. Width/Height = 0.47. Accepted.
Region 15: area = 12. Rejected: area too small (<=280).
```




```
ROI at (813, 0, 843, 14) classified as 100.0. Min distance: 1.165. Accepted.
```

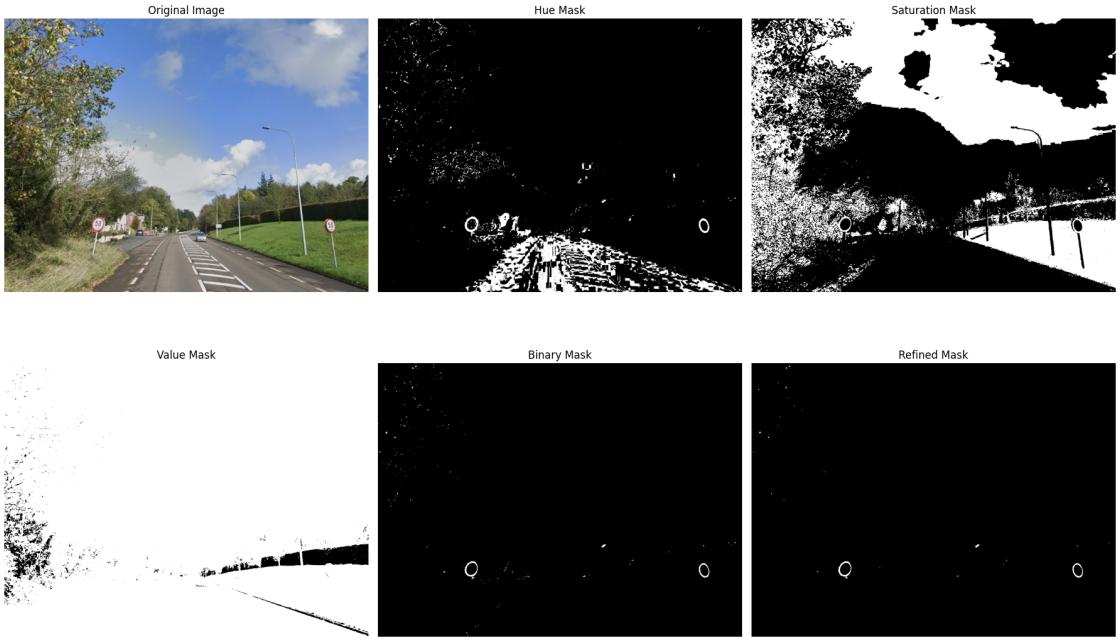
```
Detected 1 speed sign: 100.0 km/h
→ Now processing: image\50-0003x1.png
Region 0: area = 39. Rejected: area too small (<=280).
Region 1: area = 1896. Width/Height = 1.02. Accepted.
Region 2: area = 27. Rejected: area too small (<=280).
```




```
ROI at (372, 241, 471, 342) classified as 50.0. Min distance: 0.718. Accepted.
```


```
Detected 1 speed sign: 50.0 km/h
→ Now processing: image\50-0004x2.png
Region 0: area = 15. Rejected: area too small (<=280).
Region 1: area = 15. Rejected: area too small (<=280).
Region 2: area = 5. Rejected: area too small (<=280).
Region 3: area = 18. Rejected: area too small (<=280).
Region 4: area = 8. Rejected: area too small (<=280).
Region 5: area = 5. Rejected: area too small (<=280).
Region 6: area = 5. Rejected: area too small (<=280).
Region 7: area = 5. Rejected: area too small (<=280).
Region 8: area = 5. Rejected: area too small (<=280).
Region 9: area = 5. Rejected: area too small (<=280).
Region 10: area = 24. Rejected: area too small (<=280).
Region 11: area = 15. Rejected: area too small (<=280).
Region 12: area = 5. Rejected: area too small (<=280).
Region 13: area = 5. Rejected: area too small (<=280).
Region 14: area = 8. Rejected: area too small (<=280).
Region 15: area = 5. Rejected: area too small (<=280).
Region 16: area = 5. Rejected: area too small (<=280).
Region 17: area = 91. Rejected: area too small (<=280).
```

```
Region 18: area = 8. Rejected: area too small (<=280).
Region 19: area = 8. Rejected: area too small (<=280).
Region 20: area = 722. Width/Height = 0.75. Accepted.
Region 21: area = 524. Width/Height = 0.73. Accepted.
Region 22: area = 5. Rejected: area too small (<=280).
Region 23: area = 5. Rejected: area too small (<=280).
Region 24: area = 11. Rejected: area too small (<=280).
Region 25: area = 5. Rejected: area too small (<=280).
Region 26: area = 8. Rejected: area too small (<=280).
```




```
ROI at (695, 305, 756, 351) classified as 50.0. Min distance: 0.970. Accepted.
ROI at (702, 1128, 753, 1165) classified as 50.0. Min distance: 0.643. Accepted.
```


```
Detected 2 speed signs: 50.0 km/h, 50.0 km/h
→ Now processing: image\50-0005x1.png
Region 0: area = 5. Rejected: area too small (<=280).
Region 1: area = 21. Rejected: area too small (<=280).
Region 2: area = 5. Rejected: area too small (<=280).
Region 3: area = 5. Rejected: area too small (<=280).
Region 4: area = 5. Rejected: area too small (<=280).
Region 5: area = 37. Rejected: area too small (<=280).
Region 6: area = 15. Rejected: area too small (<=280).
Region 7: area = 56. Rejected: area too small (<=280).
Region 8: area = 8. Rejected: area too small (<=280).
Region 9: area = 8. Rejected: area too small (<=280).
Region 10: area = 10. Rejected: area too small (<=280).
Region 11: area = 625. Width/Height = 0.79. Accepted.
Region 12: area = 83. Rejected: area too small (<=280).
Region 13: area = 16. Rejected: area too small (<=280).
Region 14: area = 8. Rejected: area too small (<=280).
Region 15: area = 14. Rejected: area too small (<=280).
Region 16: area = 105. Rejected: area too small (<=280).
Region 17: area = 5. Rejected: area too small (<=280).
```

```
Region 18: area = 5. Rejected: area too small (<=280).
Region 19: area = 5. Rejected: area too small (<=280).
Region 20: area = 59. Rejected: area too small (<=280).
Region 21: area = 5. Rejected: area too small (<=280).
```