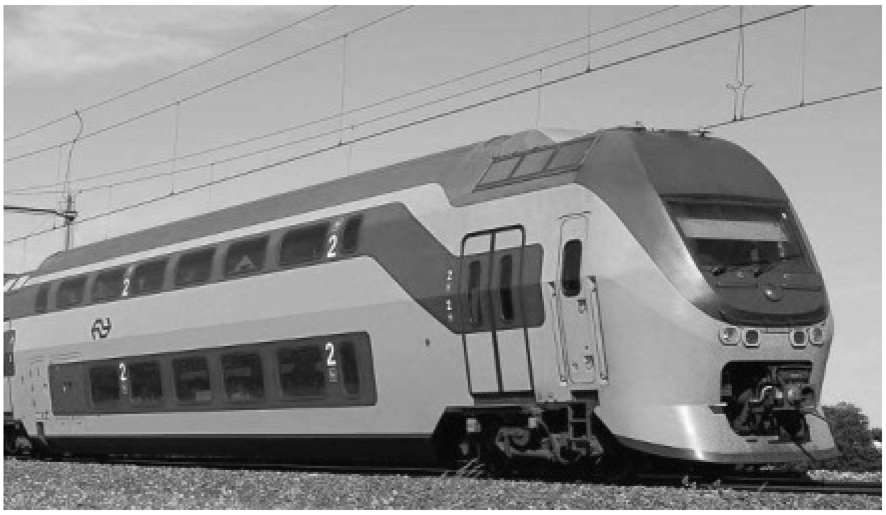
Assignment 2: Regions and Lines

Jon Reboiro and Ander Amigorena

14/10/2025

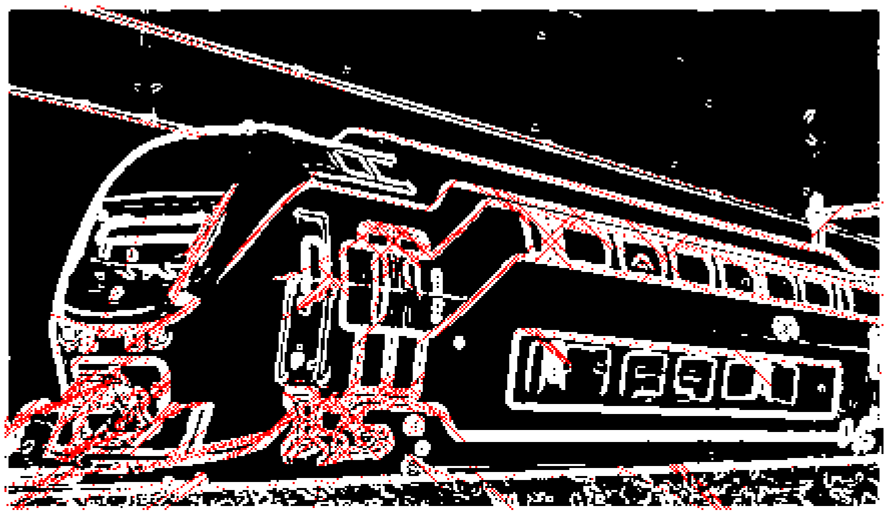
Image A from Assignment 1:



Task 1 – Binary line detection:

- 1. Parameter combination

  Edge magnitude threshold: 50
  Minimum intensity: 1
  Peak threshold: 0.3
  Minimum segment length: 15
  Maximum gap: 2

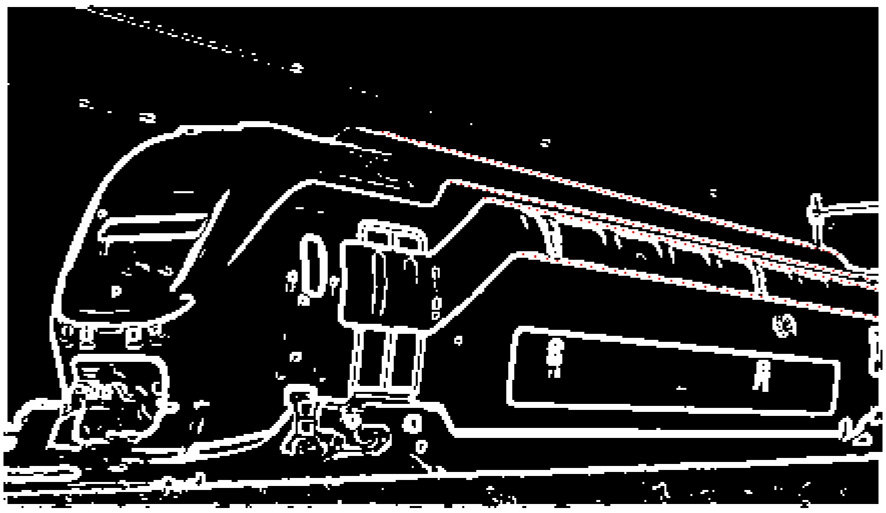


The first combination, with low thresholds and short segment constraints, was expected to produce a highly detailed and sensitive result, capturing even the weakest edges at the cost of introducing some noise.

So looking at the image, it is even denser than predicted, with an excessive presence of red lines covering several textured regions. The algorithm successfully captured fine structures but also misinterpreted noise and minor brightness variations as valid edges, which makes it a bit messy.

- 2. <u>Parameter combination</u>

  Edge magnitude threshold: 100
  Minimum intensity: 1
  Peak threshold: 0.5
  Minimum segment length: 30
  Maximum gap: 3

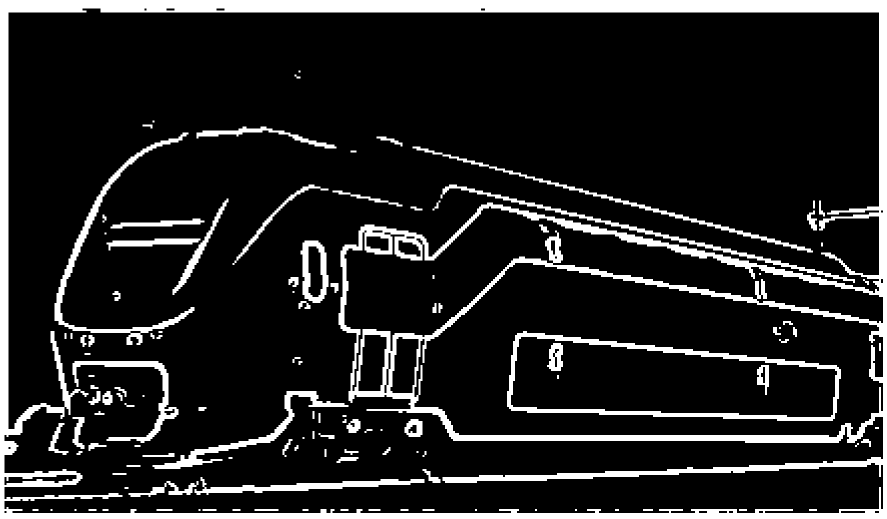


The second combination, a more balanced one, was anticipated to deliver a moderate level of sensitivity, enough to detect the most relevant structures while filtering out weaker edges.

The image fulfils expectations most closely: the few red lines are well-defined, continuous, and mainly aligned with meaningful geometric elements in the scene; pronounced lines in the middle and top of the train. The balance between sensitivity and selectivity produces a clean visualization, confirming the effectiveness of this parameter set for general-purpose use.

- 3. <u>Parameter combination</u>

Edge magnitude threshold: 150
Minimum intensity: 1
Peak threshold: 0.7
Minimum segment length: 50
Maximum gap: 5



The third combination, with the highest thresholds and longest minimum segment length, was expected to show a very clean and simplified result that only emphasized the most dominant and continuous lines in the image.

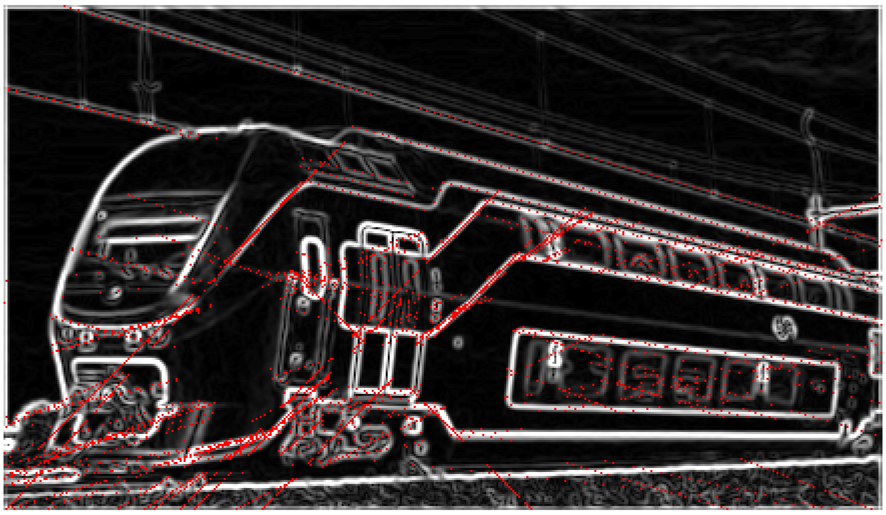
This third image, however, is sparser than expected, virtually all red overlays have disappeared, showing that the chosen thresholds were too restrictive for this specific image. While it theoretically meets the goal of emphasizing only strong lines, in practice it filtered out almost everything, producing a visually empty result.

In summary, the outcomes validate the general behavior anticipated from each configuration but also reveal practical differences in magnitude. The high-sensitivity setup is overly inclusive, the balanced one achieves the best compromise between completeness and clarity, and the low-sensitivity configuration demonstrates that excessive filtering can eliminate essential information entirely; so the second combination of parameters might be the best one in this case.

Task 2 – Grayscale line detection:

- 1. <u>Parameter combination</u>

  Minimum intensity: 20
  Peak threshold: 0.3
  Minimum segment length: 15
  Maximum gap: 2



This setup is very sensitive, and tries to find every possible line. Because of the low thresholds, even weak edges from shadows, reflections, and surface textures are treated as real lines. The result is that we get too many detections: some correct, but many false positives, we can see in the windows or at the front of the train, for example.

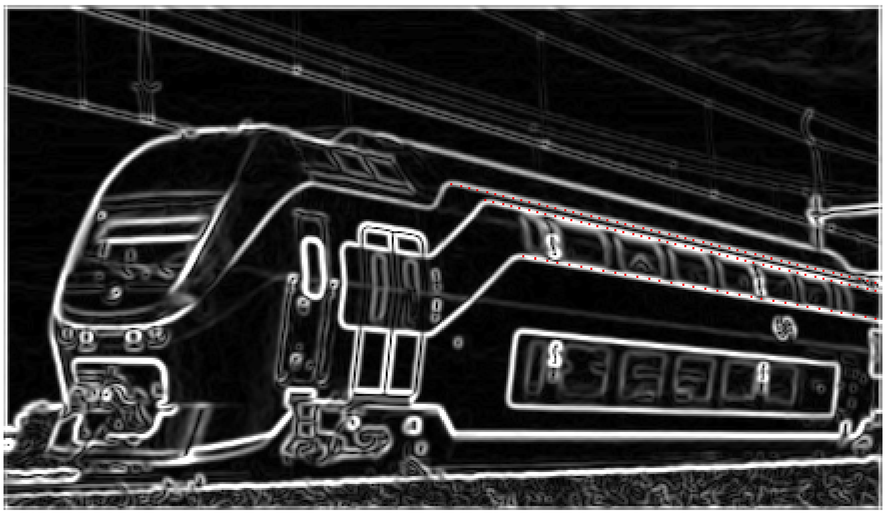
Short segment length (15 pixels) allows tiny noisy fragments to count as lines, and the very small gap value (2) means that real long edges are broken into multiple smaller ones.

In short, this configuration detects almost everything, but can't tell the difference between real edges and random noise.

- 2. <u>Parameter combination</u>

  Minimum intensity: 50

Peak threshold: 0.5
Minimum segment length: 25
Maximum gap: 3



We chose this combination of parameters in search for balance, but it leans towards being conservative. Only the strongest lines of the train are detected, so almost every detected line is correct, but many real edges are missed.

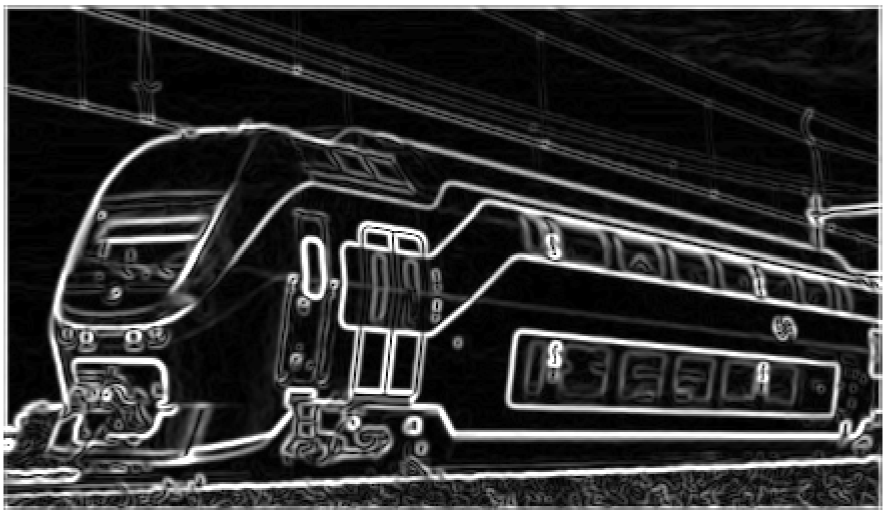
Raising the minimum intensity filters out weaker but still valid edges, and the moderate peak threshold (0.5) requires solid evidence in the accumulator.

This works great for simple scenes but struggles with real-world complexity, such as this train. In the image, most true lines are soft or partially hidden by lighting and perspective.

Overall, this setup gives clean but incomplete results, what we have obtained is accurate, but not enough at all.

- 3. <u>Parameter combination</u>

  Minimum intensity: 80
  Peak threshold: 0.7
  Minimum segment length: 40
  Maximum gap: 5



This configuration is far too strict. All real edges in the train image are filtered out before they can even be considered.

With such a high intensity threshold and long minimum segment length, the algorithm only reacts to extremely strong, continuous lines, which do not really exist in this image.

Even though the maximum gap (5) helps a bit, the other parameters eliminate too much. This gives us the result: no lines detected at all.

This is still useful, as it shows that when thresholds are too high, the algorithm stops being useful, it becomes blind to real but imperfect features.

We're in the same case as in the first task; Binary Task Detection. Comparing the three results, we'd say that the second combination is the most appropriate one, even though it's still too restrictive and lots of lines are not detected. The best combination might be somewhere between the first two combinations.

Choice tasks

1. Implement the flood fill algorithm. The output should be similar to the application of sequential region labeling. When Apply is clicked in the GUI for this item, the visualized output should be (1) an image with black for background and different gray values for each different foreground region and (2) the total number of foreground regions shown in a label or text box. (10 points)

2. Implement a function that given a binary edge map image and a list of (r, theta)-pairs as input, finds line crossings between all pairs and overlays them on the input edge map. When Apply is clicked in the GUI for this item, the visualized output should be an image in which these crossings are overlaid on the input edge map with red color. (10 points)