**Project 6 Coin Detection – Part 3 complete on all 3 images**

Name: Vivian Feng Period: 5 Date: 3/1/2023

Did you name your file l063.cpp (Lower case L, then 063)? Yes

Did you create the edge matrix? Yes

Did you create the gradient direction matrix? Yes

Did you create the imagev.ppm (visual of votes)? Yes

Did you use Bresenham's line algorithm for voting? Yes

Did you use the intersection with the border of the image or a similar approach when using Bresenham to vote? (fill in your answer)

I found the maximum radius of a coin. When voting with the Bresenham algorithm, I calculate the unit vector in direction of the gradient. Next, I vote along a segment with the length twice maximum radius centered at the edge pixel.

Does your application create imageCircles.ppm file? Yes

Does your application create imageCoins.ppm file? Yes

Does your code processes all command line arguments specified in the course materials? Yes

Does your code display on the screen and in results.txt a summary of your results? Yes

Did you test your code on terminal/gnu linux in c++11? Yes

Run your code (the same code you submit) on all 3 images I provided -then paste here the following:

* The initial image ( the one I provided), the imagev.ppm, imageCircles.ppm and imageCoins.ppm you obtained running your code, copy paste here the content of your results.txt file your code created

For the easy image:

Command line including all parameter values used for this image (including the ones you created with explanation of what is their meaning): (not a printscreen!)

./a.out -f image.ppm -lt 50 -ht 75 -TCircle 30 -q1t 30 -q2t 25 -q3t 25 -q4t 25 -minR 80 -maxR 120 -mincentdist 80

I divided the image into four quadrants, and each quadrant has its own threshold for vote counts. The sections of the image are divided as such: Diagram, engineering drawing

Description automatically generated

-q1t denotes the threshold for quadrant one. It is a percentage of the maximum intensity of the votes image. -q2t, -q3t, -q4t denote the same things respectively for each quadrant

-minR is the minimum Radius considered for the hough circle transform

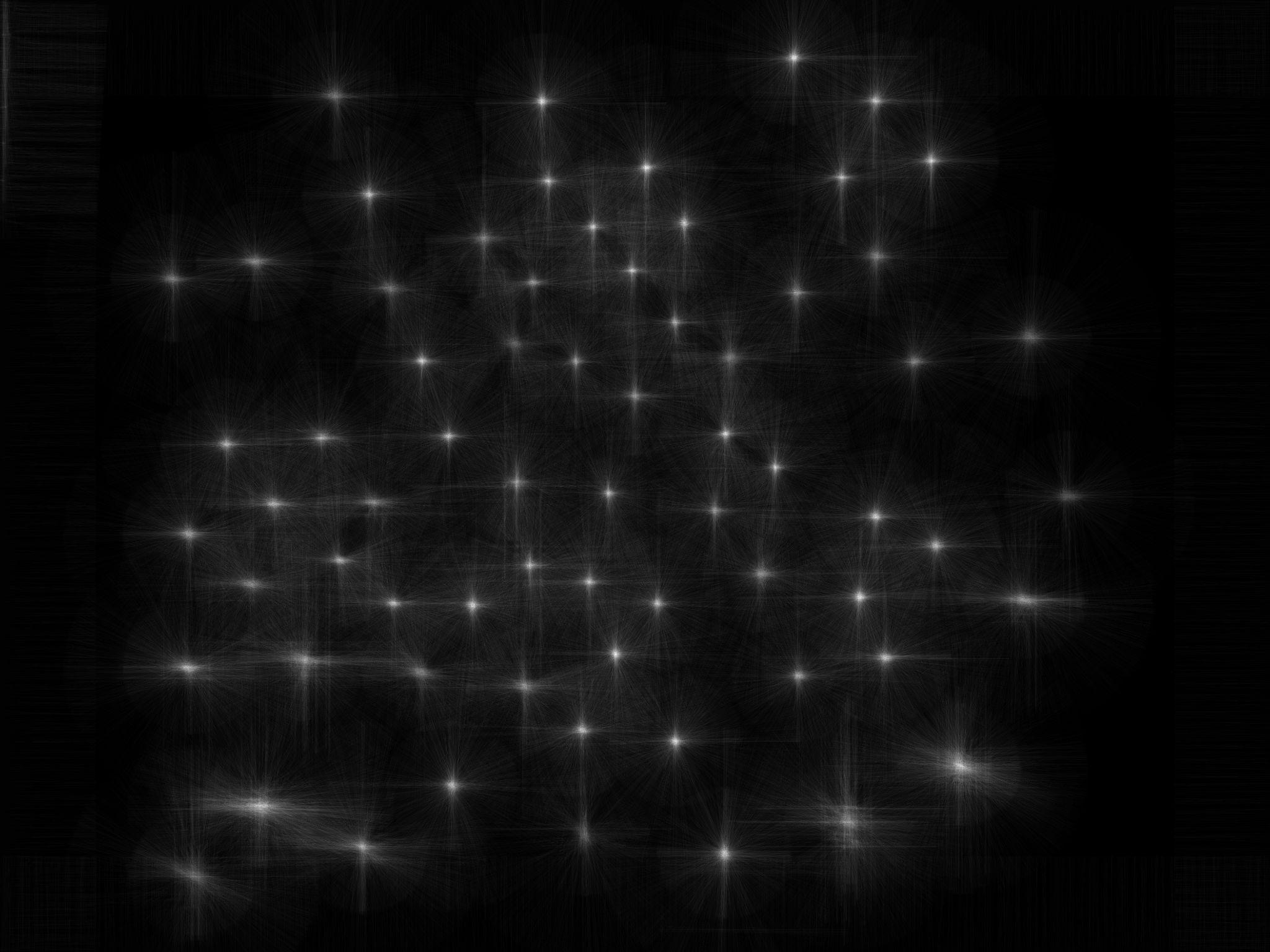
-maxR is the maximum Radius considered for the Hough Circle Transform

- mincentdist – distance in pixels between centers for two circles to be considered duplicates

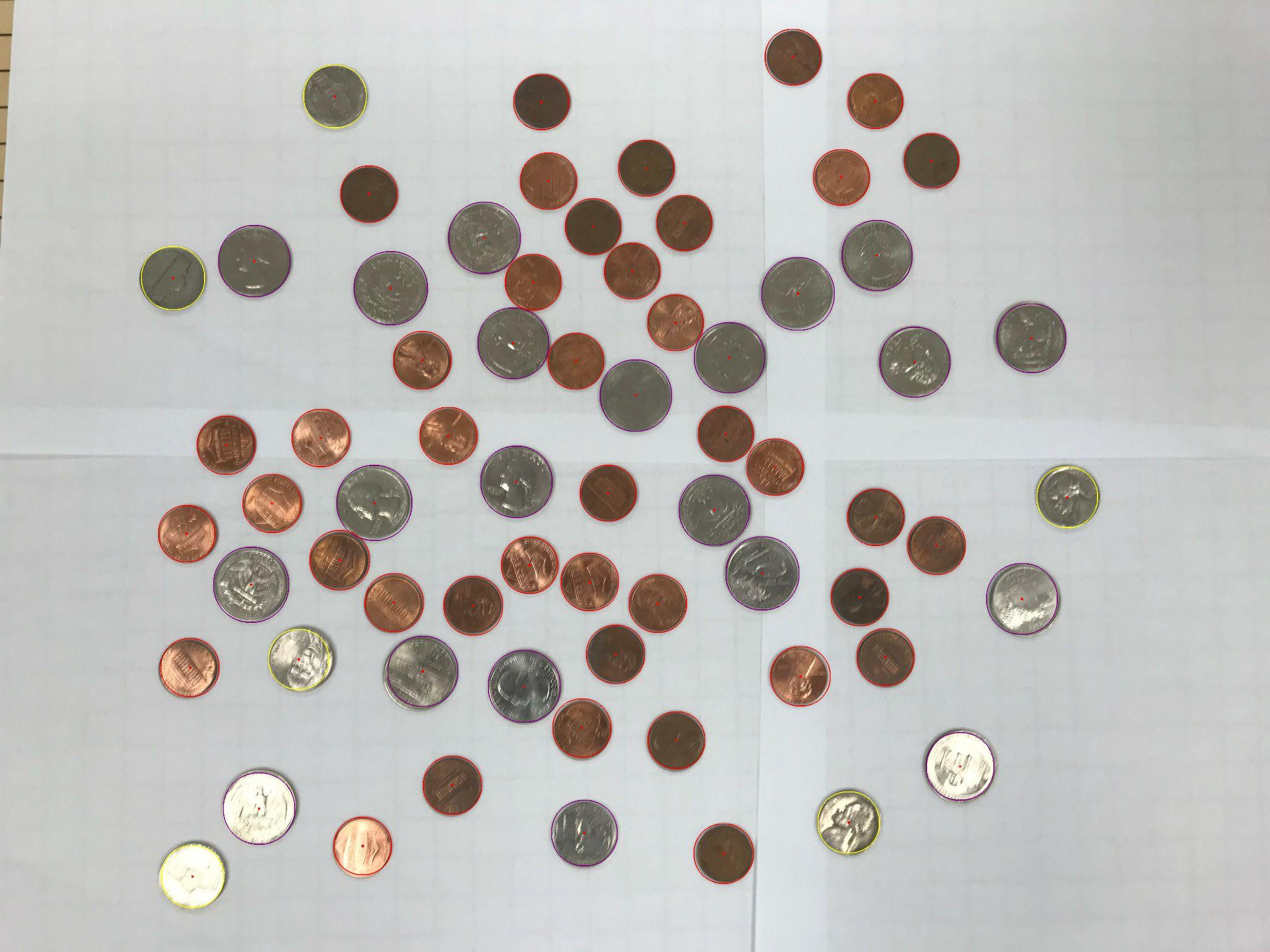
Initial image:



imagev.ppm:



imageCoins.ppm:



imageCircles.ppm:



Content of results.txt: (NOT a printscreen so I can copy/paste if I need)

penny - 41

nickel - 6

dime - 0

quarter - 21

half dollar - 0

dollar - 0

Total: $5.96

For the medium image:

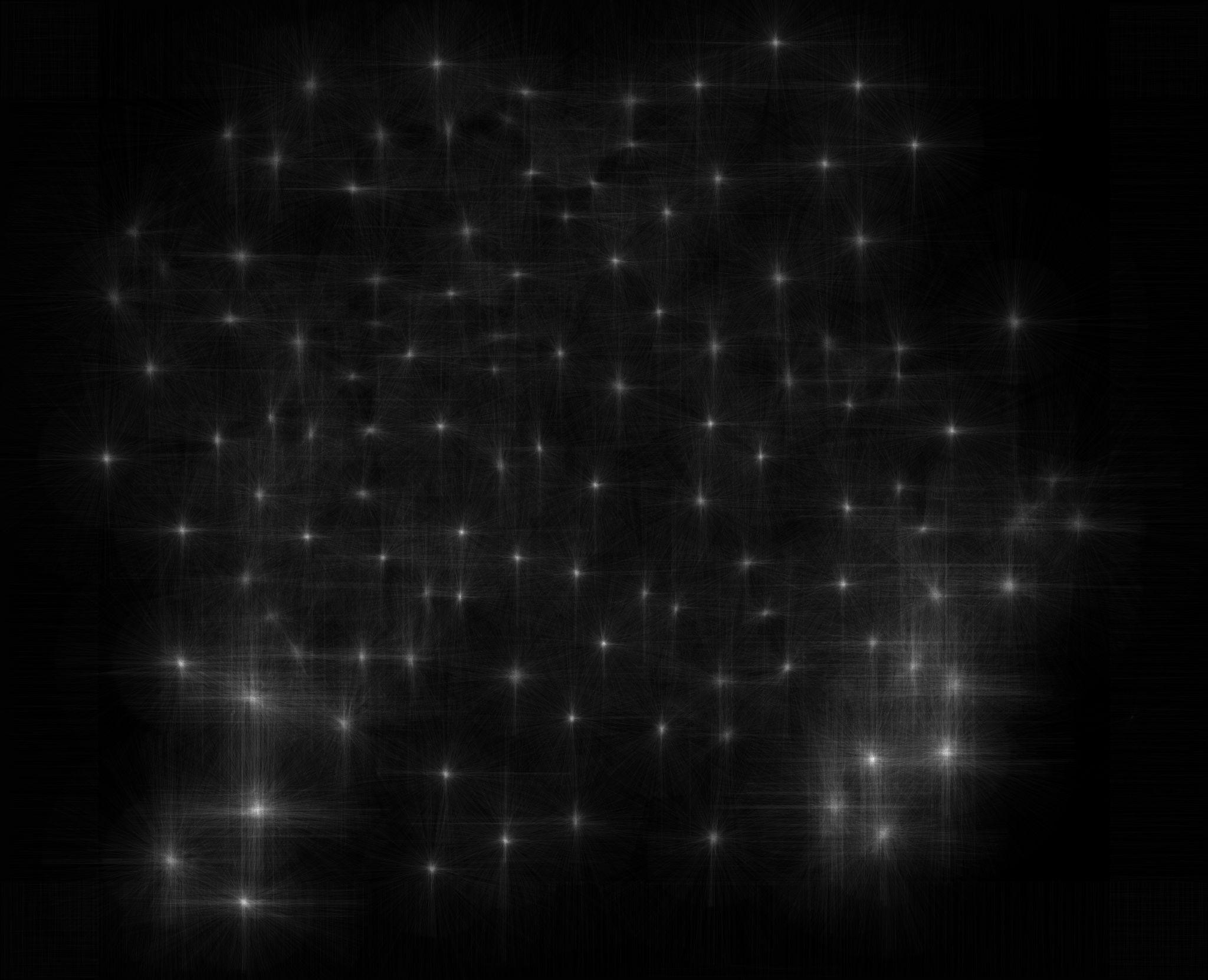
Command line including all parameter values used for this image (including the ones you created with explanation of what is their meaning): (not a printscreen!)

./a.out -f coinsHarder.ppm -lt 75 -ht 100 -q1t 20 -q2t 20 -q3t 30 -q4t 30 -minR 80 -maxR 120 -TCircle 22

Initial image:



imagev.ppm:



imageCoins.ppm:



imageCircles.ppm:



Content of results.txt: (NOT a printscreen so I can copy/paste if I need)\

penny - 64

nickel - 20

dime - 0

quarter - 31

half dollar - 0

dollar - 0

Total: $9.39

For the hard image:

Command line including all parameter values used for this image (including the ones you created with explanation of what is their meaning): (not a printscreen!)

./l063 -f coinsHardest.ppm -lt 40 -ht 60 -q1t 35 -q2t 30 -q3t 25 -q4t 25 -minR 70 -maxR 180 -hard -mincentdist 70 -TCircle 30

-hard indicates if the image is part of the hard coins image. The algorithm for identifying coins is slightly different for the hard image

I divided the image into four quadrants, and each quadrant has its own threshold for vote counts. The sections of the image are divided as such: Diagram, engineering drawing

Description automatically generated

-q1t denotes the threshold for quadrant one. It is a percentage of the maximum intensity of the votes image. -q2t, -q3t, -q4t denote the same things respectively for each quadrant

-minR is the minimum Radius considered for the hough circle transform

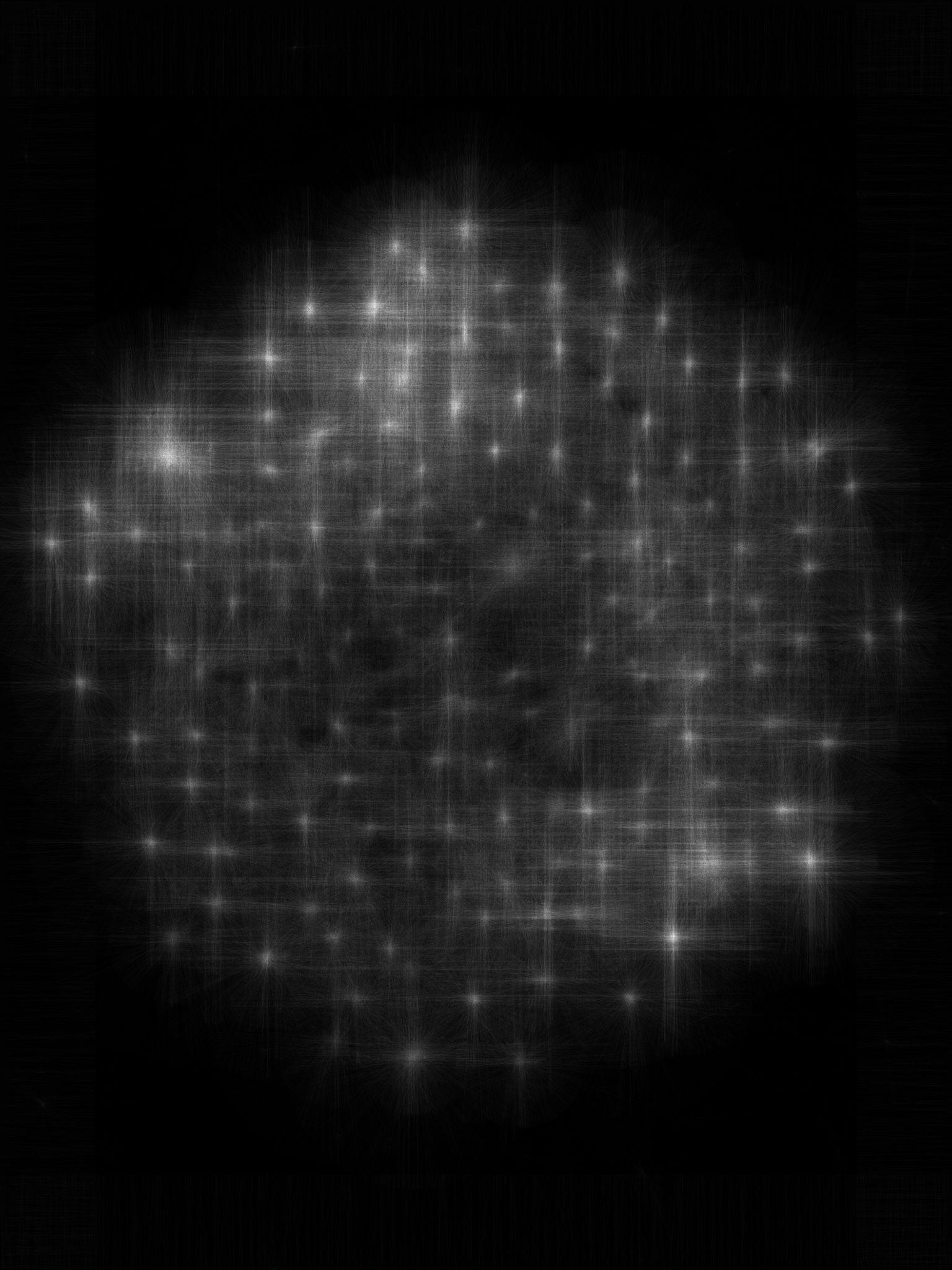
-maxR is the maximum Radius considered for the Hough Circle Transform

- mincentdist – distance in pixels between centers for two circles to be considered duplicates

Initial image:



imagev.ppm:



imageCoins.ppm:



imageCircles.ppm:



Content of results.txt: (NOT a printscreen so I can copy/paste if I need)

penny - 82

nickel - 22

dime - 53

quarter - 0

half dollar - 0

dollar - 7

Total: $14.22