Hough Line Transform

The way lines are detected is through the following steps:

1. Apply medianBlur filter (7, 7)
2. Apply GaussianBlur filter (7, 7)
3. Apply bilateralFilter (The previous steps are to get rid of the noise before doing edge detection)
4. Run Canny edge detector
5. Create the hough space of ρ and θ with increments of $\delta\rho$ and $\delta\theta$ speicfied from the arguements `num_rhos` and `num_thetas` .
6. Retreive (x, y) values of the edges in the image resulting from Canny.
7. Calculate ρ values by multiplying the matrices $(x, y) * (cos(\theta), sin(\theta))$
8. Build a histogram of the accumulator space and filter the lines based on the `threshold` specified.
9. Filter similar lines (Lines whose difference in ρ and θ is less than the `lines_similarity_threshold`), which I tuned by experimenting.

Rectangles Detection from Lines Resulting from Hough Transform

1. Find all pairs of parallel lines. A pair of lines $H_1(\rho_1, \theta_2)$ and $H_2(\rho_2, \theta_2)$ are parallel if $abs(\theta_1 - \theta_2) < \epsilon_\theta$
2. After finding all pairs of H_i and H_j statisfying the parallel condition, this generates what is called extended peaks $P(\alpha, \zeta)$ where: $\alpha = (\theta_i + theta_j)/2$ $\zeta = (\rho_i - rho_j)/2$
3. By looping over all P_is found, we need to find peaks that are perpedindecular to each other. P_k and P_l are perpedindecular iff:

$\Delta\alpha = ||\alpha_k - \alpha_l| - 90| < \epsilon_\alpha$ where ϵ_α is the angular threshold that determines whether two P_is are perspendicular or not.

In the code, $\epsilon_\theta = \epsilon_\alpha = 3 degrees$

1. The found 4 lines forming every rectangle are then passed to `get_lines_intersection` function which returns a point of intersection given two lines defined by 2 points each. The relation found here: https://en.wikipedia.org/wiki/Line%E2%80%93line_intersection#Given_two_points_on_each_line (https://en.wikipedia.org/wiki/Line%E2%80%93line_intersection#Given_two_points_on_each_line) is used to return the intersection point.
2. After getting the 4 bounding points for the rectangle, the function `minAreaReact` is used to return the correct format for a rectangle bounded by those points, which is then passed to `drawContours` to draw the rectangle on the image.
3. Only rectangles with `area < area_threshold` are considered valid, given that the `area_threshold` is tuned for the usecase of paper money in this assignment.

Circle Hough Transform

1. I build a 3D space of (a, b, θ), where a circle is defined as the following:

$$x = a + r * cos(\theta)$$

$$y = b + r * sin(\theta)$$

The flow is similar to the line hough transform except is scaled to searching a third dimension specified by the `r_range` for the radius.

Money Count

I count the money purely based on the area of the paper or the radius of the coin.

The coins I experimented with are 0.5 and 1 pound (Which are very close in radii, so the threshold isn't always accurate)

The papers I count are:

1. 1-pound paper
2. 20-pound (based on its height, which way more than the 1 pount paper)

The program is limited to those since this is what my test set is composed of, but it can definitely be scaled to all egyptian money after experimting with them and knowing their width and height

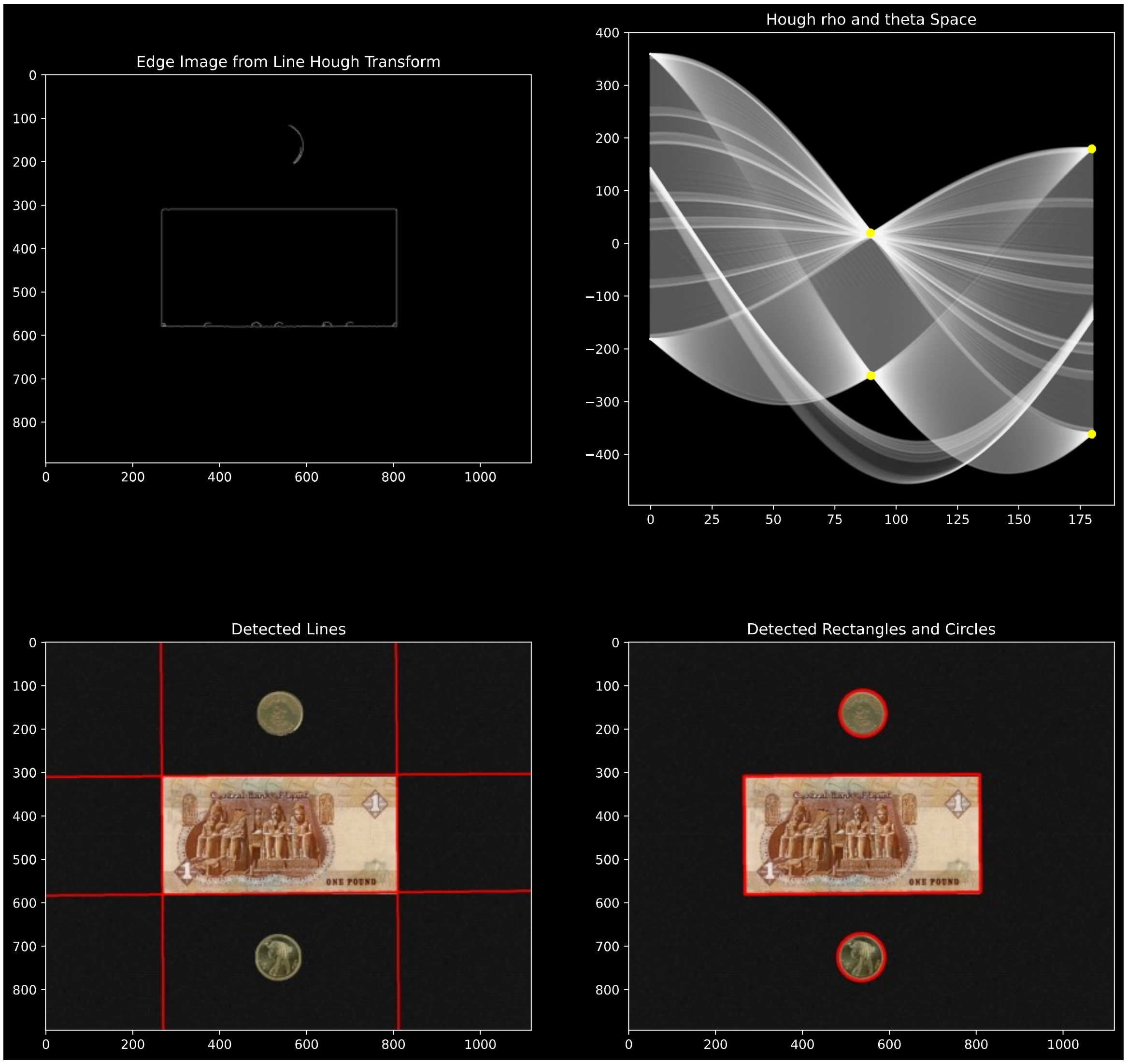
In [1]:

```
import cv2
from transform import hough_windowed_rectangle, hough_cricles
from utils import plot_hough_space, graph_output, graph_rectangles_and_
circles, countMoney
from skimage.io import imshow
import matplotlib.pyplot as plt
```

In [2]:

```
img = cv2.imread("cases/case1.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, rectangles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img,  r_range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circles)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, peaks)
```

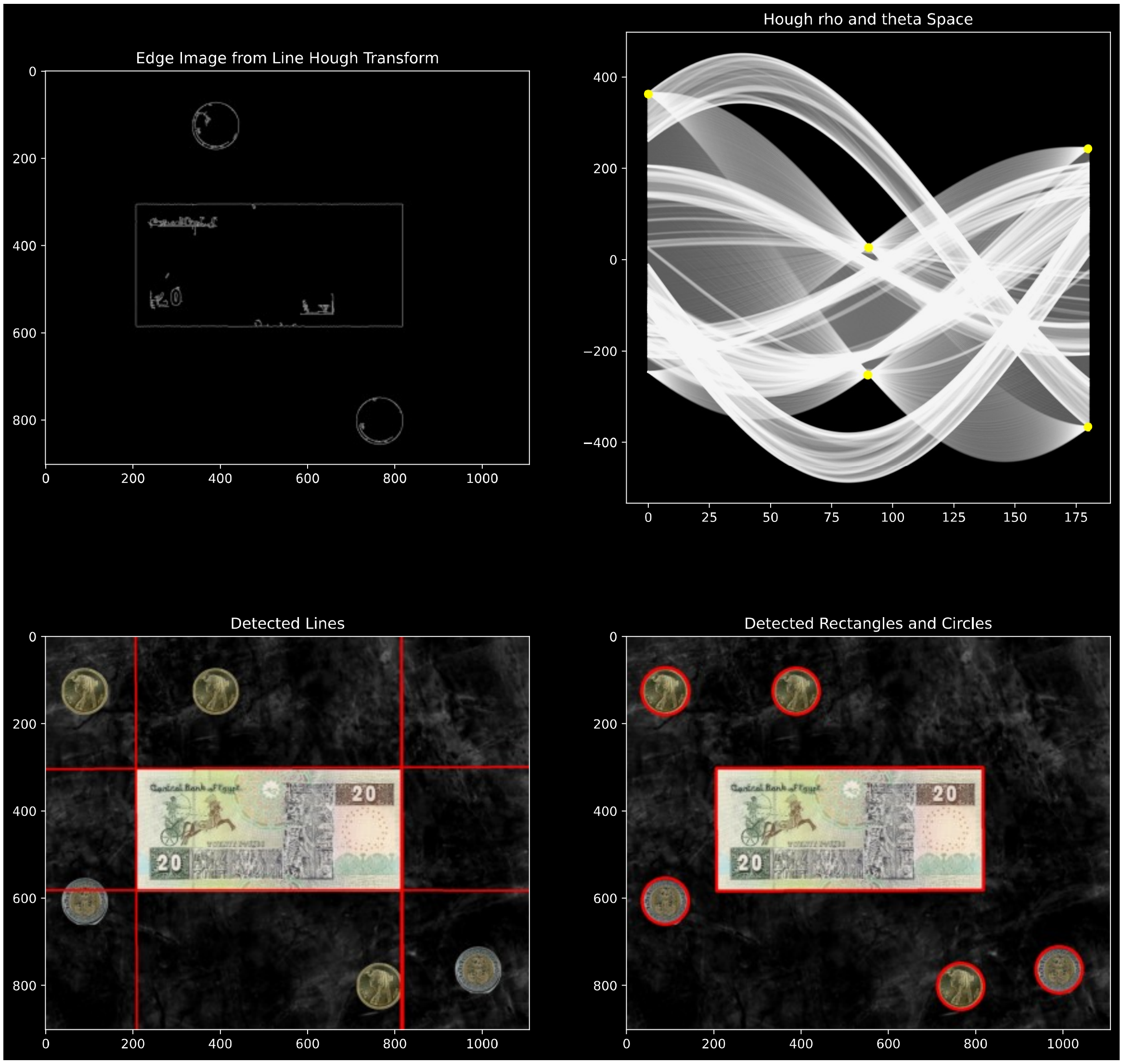
```
Detected Rectangle 1 width= 271 height= 541
Corner points=
 [[265 309]
 [806 305]
 [808 575]
 [267 580]]
Overall detected rectangles count =  1
Detected Circle 1 r= 53 cetner=( 538 , 165 )
Detected Circle 2 r= 54 cetner=( 534 , 724 )
Overall detected circles count =  2
Money Count in Pounds 2.0
```



In [3]:

```
img = cv2.imread("cases/case2.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, rectangles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img, region=20, threshold=15, r_range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circles)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, peaks)
```

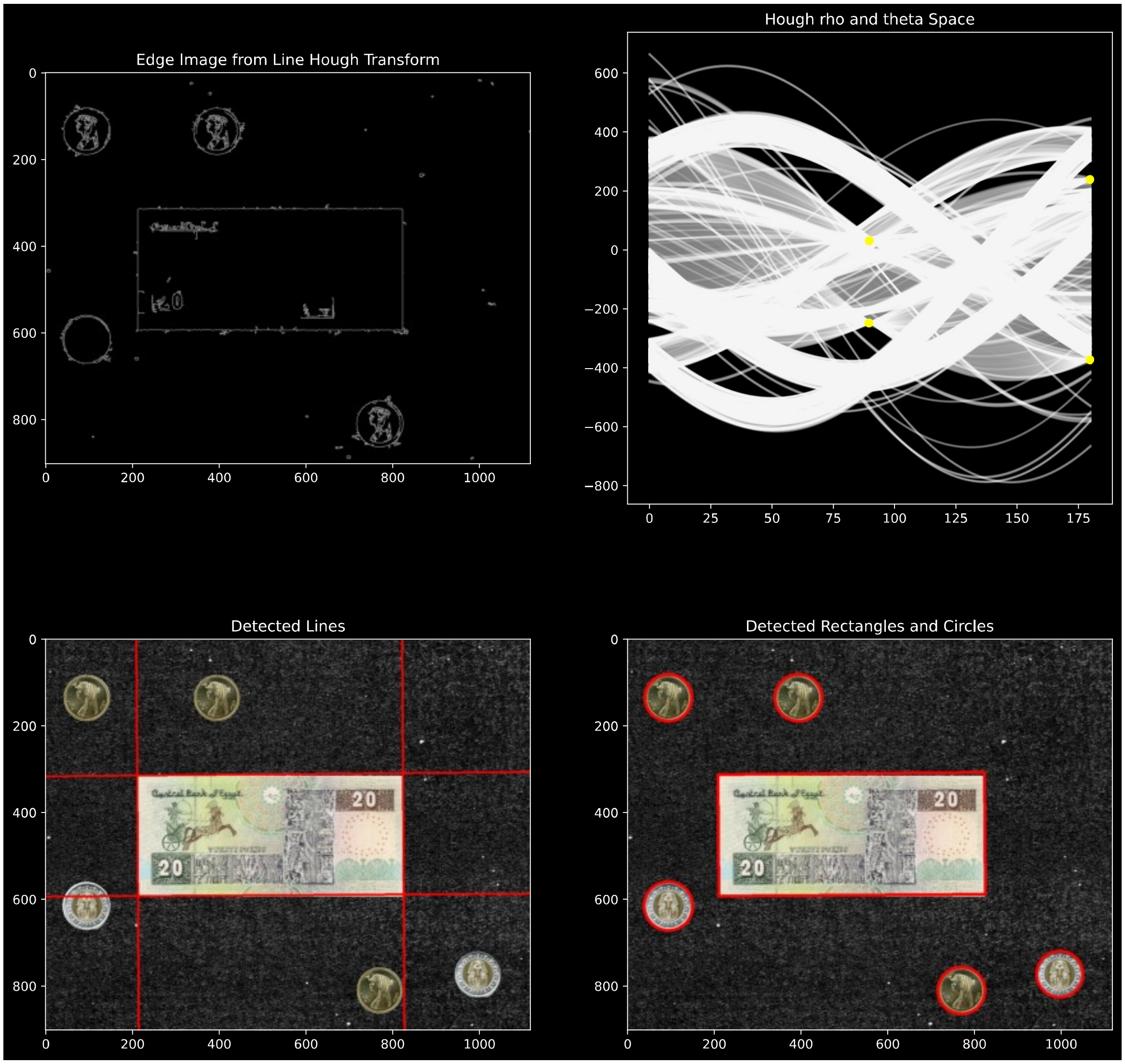
```
Detected Rectangle 1 width= 281 height= 609
Corner points=
 [[205 302]
 [815 300]
 [816 581]
 [206 583]]
Overall detected rectangles count =  1
Detected Circle 1 r= 53 cetner=( 388 , 126 )
Detected Circle 2 r= 53 cetner=( 89 , 606 )
Detected Circle 3 r= 53 cetner=( 989 , 763 )
Detected Circle 4 r= 53 cetner=( 764 , 801 )
Detected Circle 5 r= 54 cetner=( 89 , 126 )
Overall detected circles count =  5
Money Count in Pounds 22.5
```



In [4]:

```
img = cv2.imread("cases/case2_medium.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, recta
ngles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img, region=20, threshold=15, r_
range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circ
les)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, pe
aks)
```

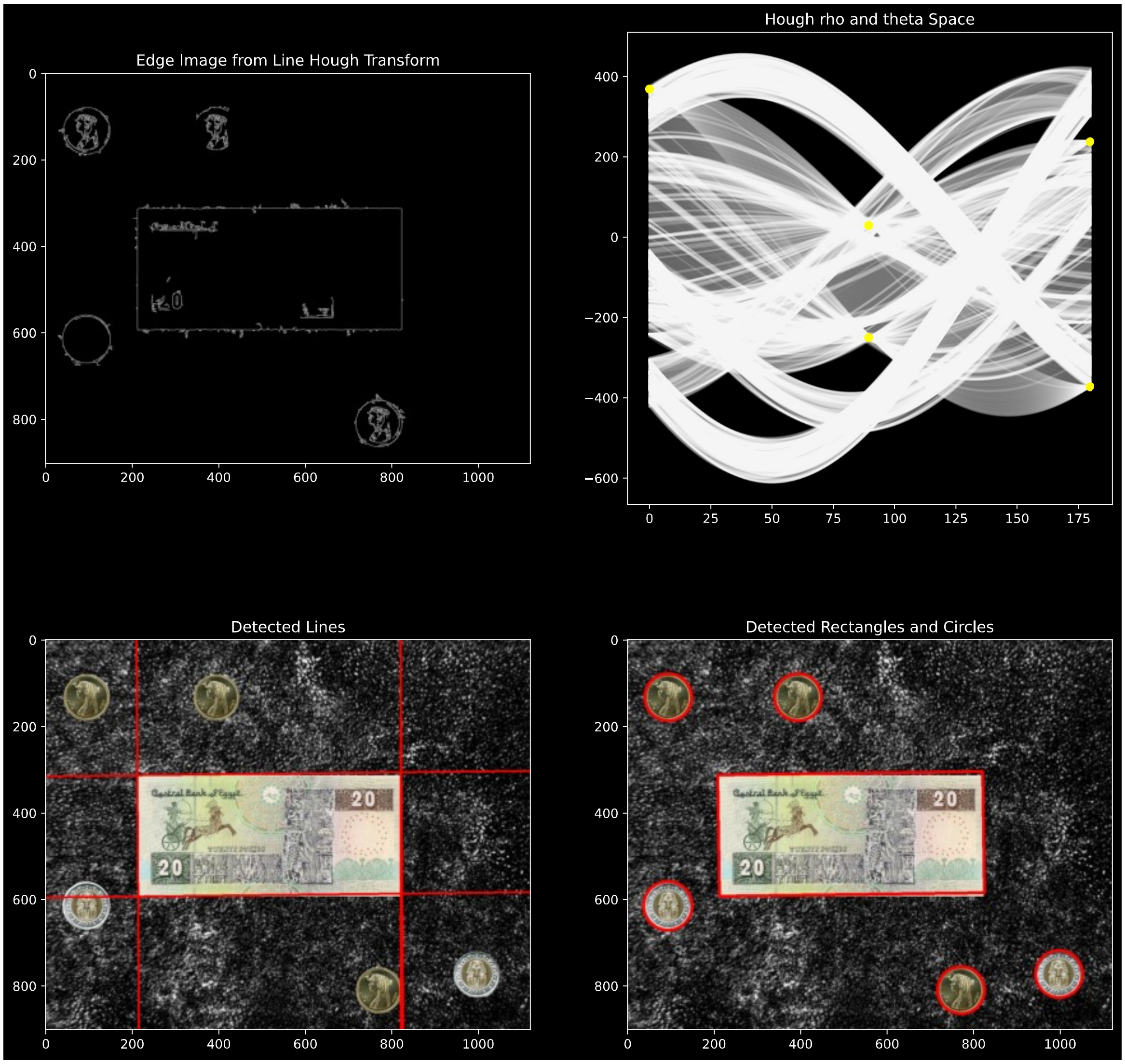
```
Detected Rectangle 1 width= 280 height= 612
Corner points=
 [[208 311]
 [821 306]
 [823 587]
 [210 591]]
Overall detected rectangles count =  1
Detected Circle 1 r= 53 cetner=( 994 , 771 )
Detected Circle 2 r= 54 cetner=( 94 , 134 )
Detected Circle 3 r= 54 cetner=( 393 , 134 )
Detected Circle 4 r= 54 cetner=( 768 , 809 )
Detected Circle 5 r= 56 cetner=( 94 , 614 )
Overall detected circles count =  5
Money Count in Pounds 23.0
```



In [5]:

```
img = cv2.imread("cases/case2_hard.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, rectangles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img, region=20, threshold=15, r_range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circles)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, peaks)
```

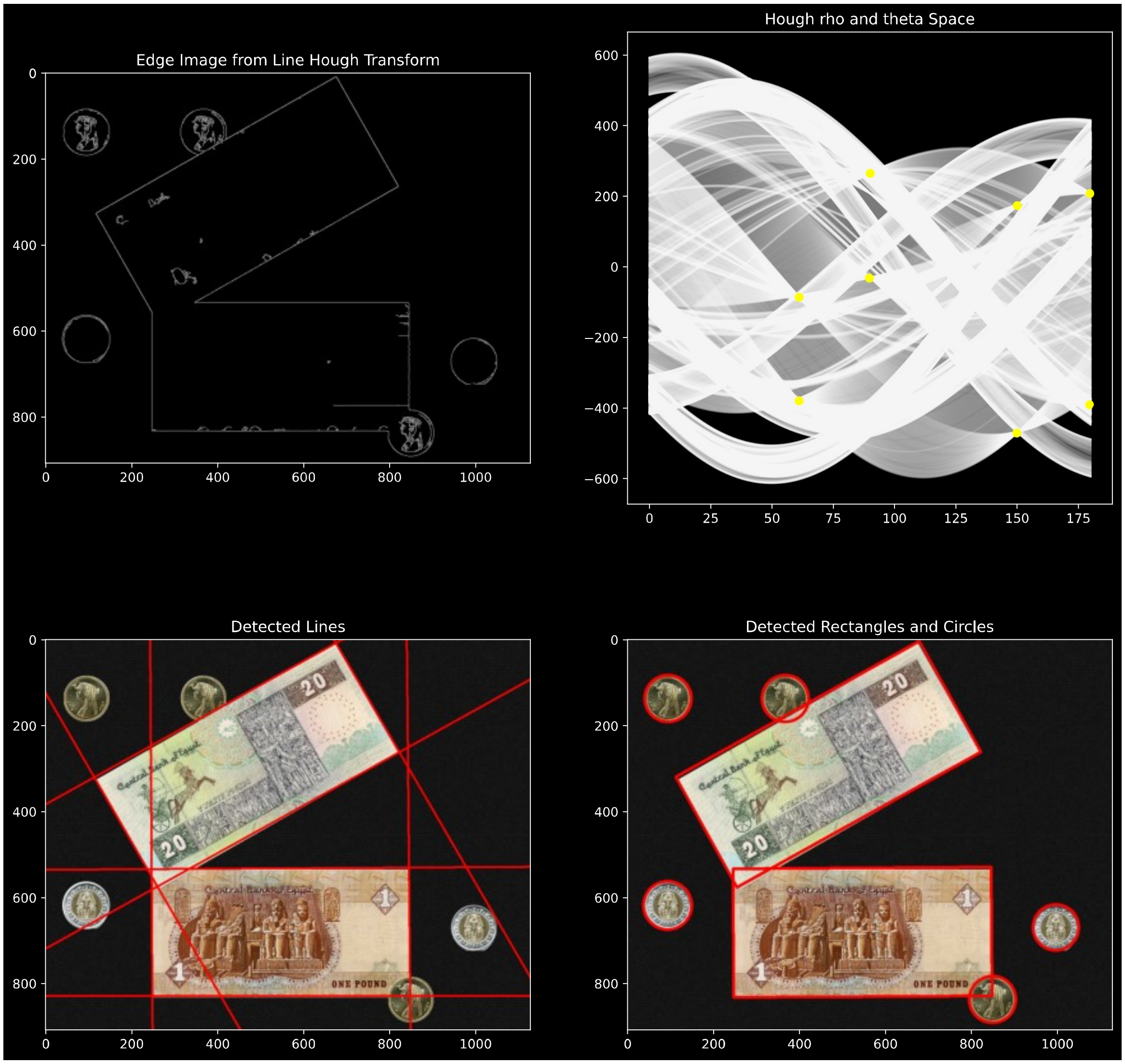
```
Detected Rectangle 1 width= 279 height= 611
Corner points=
 [[208 313]
 [819 305]
 [823 584]
 [212 592]]
Overall detected rectangles count =  1
Detected Circle 1 r= 54 cetner=( 94 , 134 )
Detected Circle 2 r= 54 cetner=( 393 , 134 )
Detected Circle 3 r= 54 cetner=( 995 , 771 )
Detected Circle 4 r= 54 cetner=( 769 , 808 )
Detected Circle 5 r= 56 cetner=( 93 , 614 )
Overall detected circles count =  5
Money Count in Pounds 23.0
```



In [6]:

```
img = cv2.imread("cases/case3.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, recta
ngles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img, region=20, threshold=15, r_
range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circ
les)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, pe
aks)
```

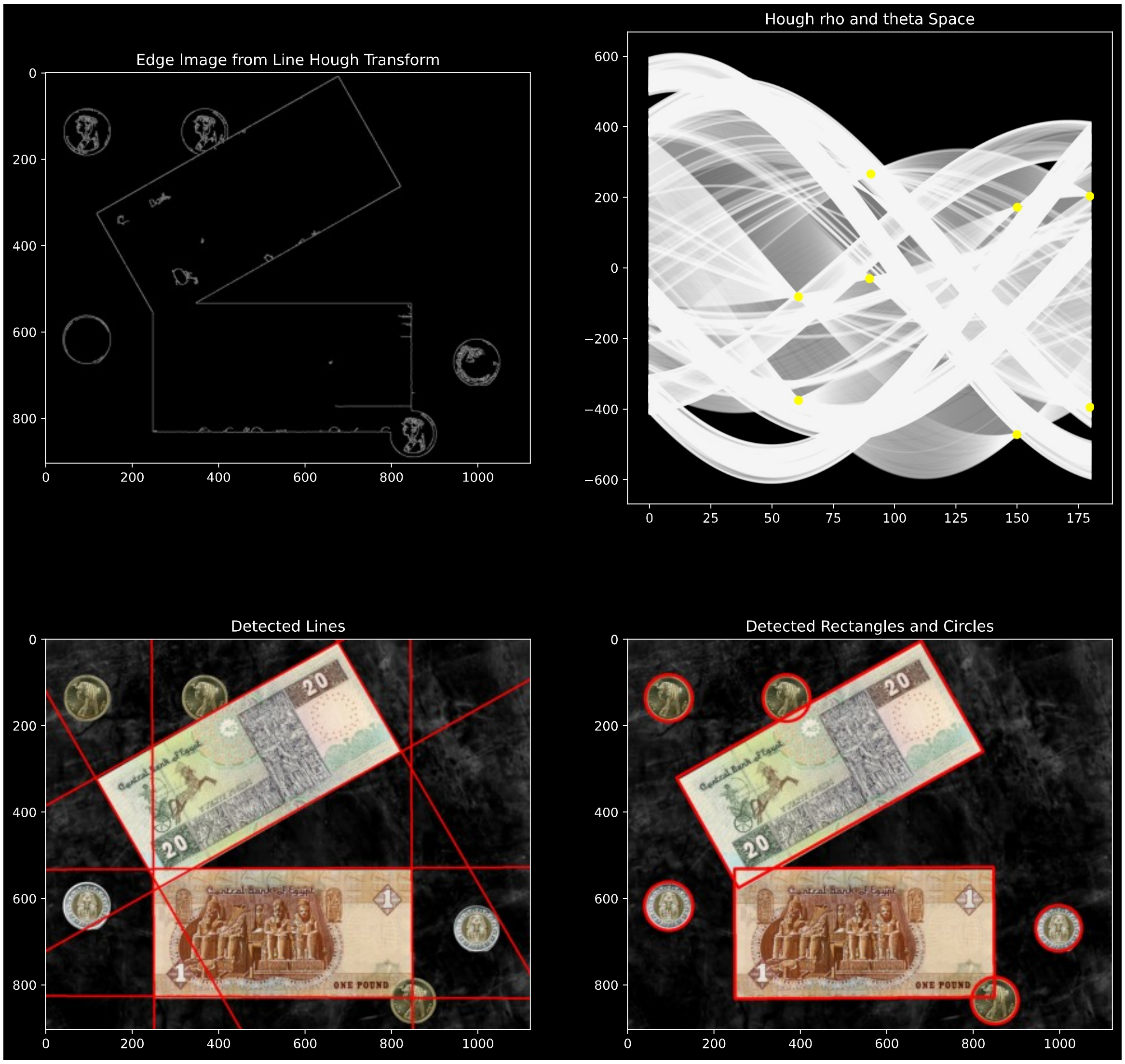
```
Detected Rectangle 1 width= 292 height= 647
Corner points=
 [[112 320]
 [677   5]
 [820 262]
 [254 576]]
Detected Rectangle 2 width= 298 height= 599
Corner points=
 [[245 533]
 [844 529]
 [846 828]
 [246 831]]
Overall detected rectangles count =  2
Detected Circle 1 r= 53 cetner=( 994 , 669 )
Detected Circle 2 r= 54 cetner=( 94 , 138 )
Detected Circle 3 r= 54 cetner=( 366 , 138 )
Detected Circle 4 r= 54 cetner=( 847 , 836 )
Detected Circle 5 r= 56 cetner=( 93 , 618 )
Overall detected circles count =  5
Money Count in Pounds 24.0
```



In [7]:

```
img = cv2.imread("cases/case4.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, rectangles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img, region=20, threshold=15, r_range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circles)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, peaks)
```

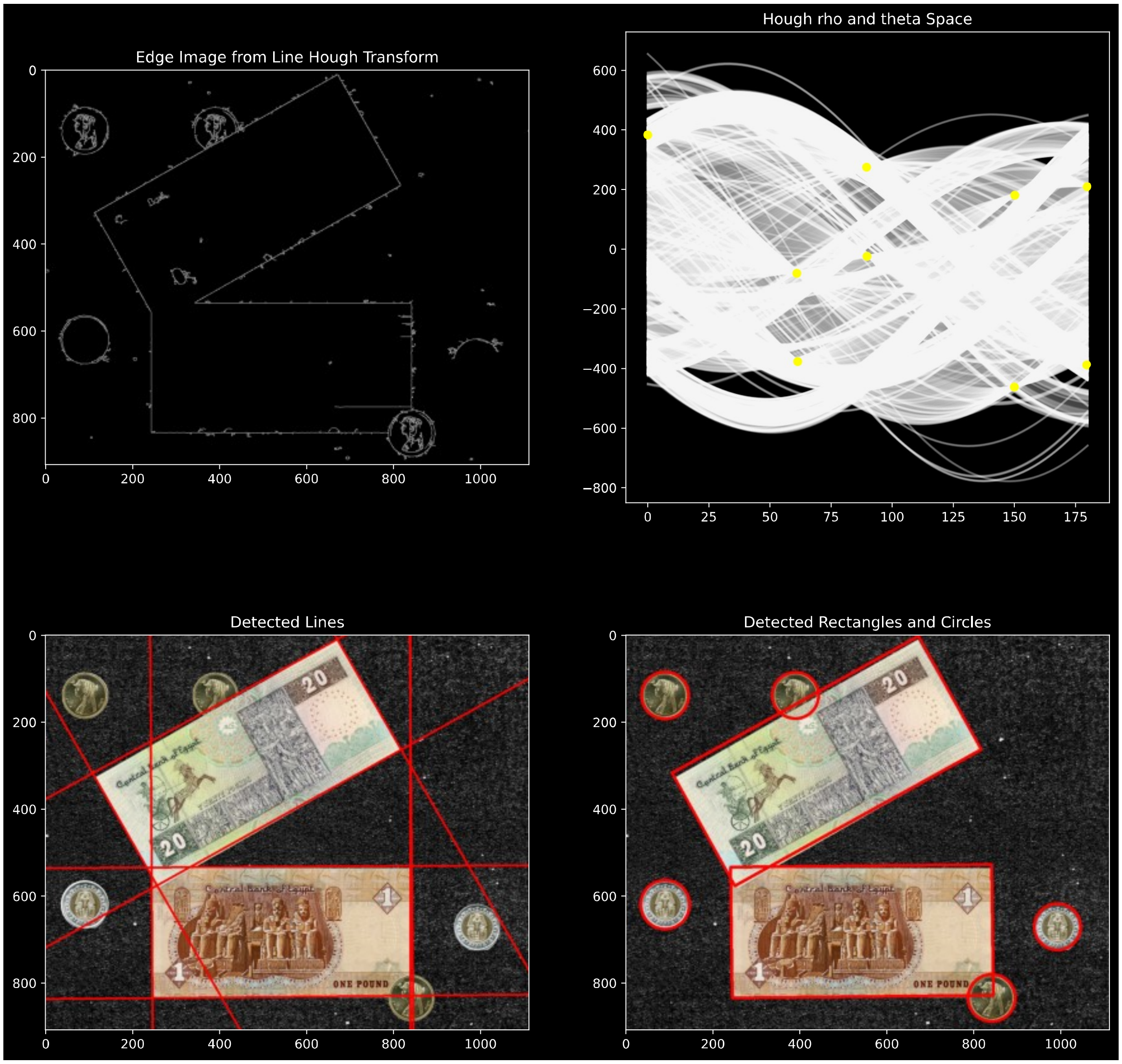
```
Detected Rectangle 1 width= 292 height= 646
Corner points=
 [[113 321]
 [678   5]
 [821 260]
 [256 575]]
Detected Rectangle 2 width= 300 height= 598
Corner points=
 [[247 531]
 [845 527]
 [847 827]
 [248 831]]
Overall detected rectangles count =  2
Detected Circle 1 r= 53 cetner=( 995 , 668 )
Detected Circle 2 r= 54 cetner=( 95 , 137 )
Detected Circle 3 r= 54 cetner=( 367 , 137 )
Detected Circle 4 r= 54 cetner=( 848 , 835 )
Detected Circle 5 r= 56 cetner=( 95 , 617 )
Overall detected circles count =  5
Money Count in Pounds 24.0
```



In [8]:

```
img = cv2.imread("cases/case4_medium.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, rectangles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img, region=20, threshold=15, r_range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circles)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, peaks)
```

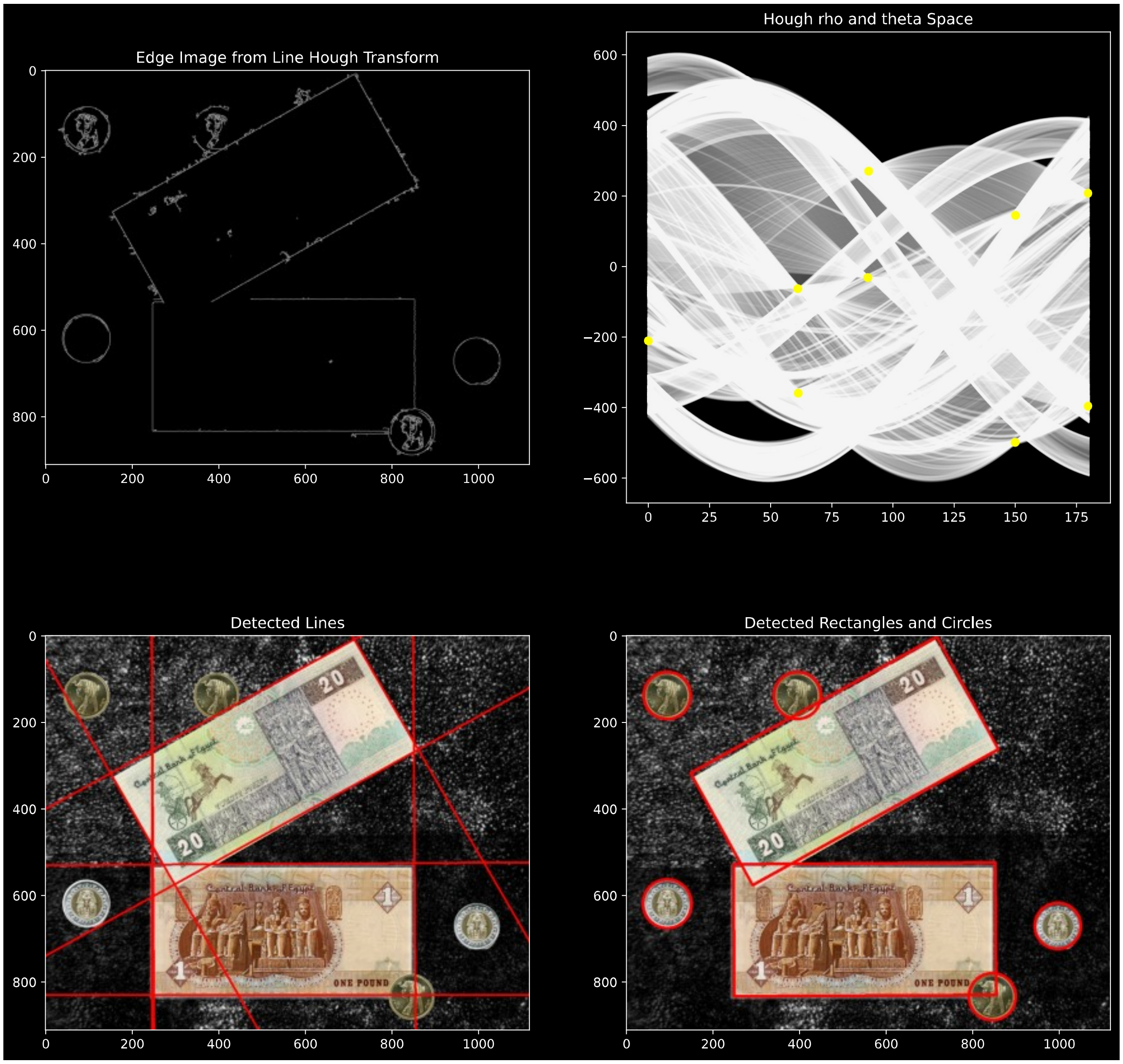
```
Detected Rectangle 1 width= 296 height= 645
Corner points=
 [[106 317]
 [671   3]
 [815 263]
 [250 576]]
Detected Rectangle 2 width= 300 height= 599
Corner points=
 [[240 533]
 [839 527]
 [843 827]
 [244 833]]
Overall detected rectangles count =  2
Detected Circle 1 r= 53 cetner=( 389 , 139 )
Detected Circle 2 r= 53 cetner=( 990 , 670 )
Detected Circle 3 r= 54 cetner=( 90 , 139 )
Detected Circle 4 r= 54 cetner=( 839 , 833 )
Detected Circle 5 r= 56 cetner=( 90 , 619 )
Overall detected circles count =  5
Money Count in Pounds 24.0
```



In [9]:

```
img = cv2.imread("cases/case4_hard.png")
edge_image, img_lines, img_rectangles, hough_space, peaks, lines, rectangles, areas = hough_windowed_rectangle(img)
img_w_circles, circles = hough_cricles(img, region=20, threshold=15, r_range=(40, 60))
print("Money Count in Pounds", countMoney(circles, areas))
img_w_rect_circles = graph_rectangles_and_circles(img, rectangles, circles)
graph_output(edge_image, img_lines, img_w_rect_circles, hough_space, peaks)
```

```
Detected Rectangle 1 width= 295 height= 646
Corner points=
 [[148 318]
 [715   4]
 [857 263]
 [291 576]]
Detected Rectangle 2 width= 304 height= 603
Corner points=
 [[247 528]
 [850 524]
 [852 829]
 [248 832]]
Overall detected rectangles count =  2
Detected Circle 1 r= 53 cetner=( 994 , 670 )
Detected Circle 2 r= 54 cetner=( 94 , 139 )
Detected Circle 3 r= 54 cetner=( 394 , 139 )
Detected Circle 4 r= 54 cetner=( 843 , 833 )
Detected Circle 5 r= 56 cetner=( 94 , 619 )
Overall detected circles count =  5
Money Count in Pounds 24.0
```


Edge Image from Line Hough Transform

Hough rho and theta Space

Detected Lines

Detected Rectangles and Circles

```
In [ ]:
```