Computer Vision 1: Assignment 3
(Due date: 06.12.2021)

Submission instructions:

- For each programming task, submit a `.py` source code file or a Jupyter/Colab notebook file. Do not include any images or data files you used.
- For each pen & paper task, submit a `.pdf` file. Type your solution in LaTeX, Word, or another text editor of your choice and convert it to PDF. Do not submit photographs or scans of handwritten solutions!
- In all submissions, include at the top names of all students in the group.
- Choose one person in your group that submits the solution for your entire group.

Task 1: Circle detection with Hough transform (programming)

Figure 1 shows a selection of the last Finnish pre-euro coins. The diameters of the coins measured in millimetres are listed in Table 1. We use the Hough transform to detect the coins.

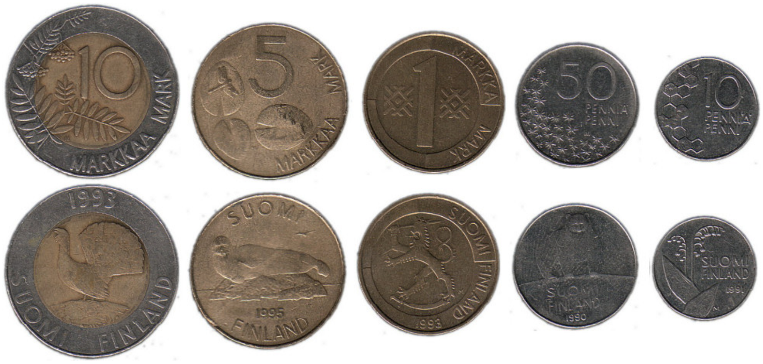

Figure 1: A selection of Finnish coins. From left to right: 10, 5, and 1 marks; and 50 and 10 pennis.

Table 1: Coin diameters specified by the mint.

Coin	10 marks	5 marks	1 mark	50 penni	10 penni
Diameter (millimetre)	27.25	24.50	22.25	19.70	16.30

- Download the image `coins.jpg` from Moodle. Read it and convert to grayscale.
- Calculate the radius r of each coin measured in pixels, by using the data in Table 1 and the fact that the resolution of the image is known to be approximately 0.12 mm/pixel.
- Apply the Canny edge detector to find edges in the grayscale image. Use the built-in function `skimage.feature.canny`. Visualize the edges and check that the outlines of the coins are detected.
- Use `skimage.transform.hough_circle` to calculate the Hough transform of the edge detection result. Use the list of radii you calculated above as the input `radius`. Use default values for the other parameters. Draw the result for each radius. For example, for the radius correspoding to the 5 marks coin you should obtain a result similar to Figure 2 that peaks strongly around the center of the 5 mark coin.

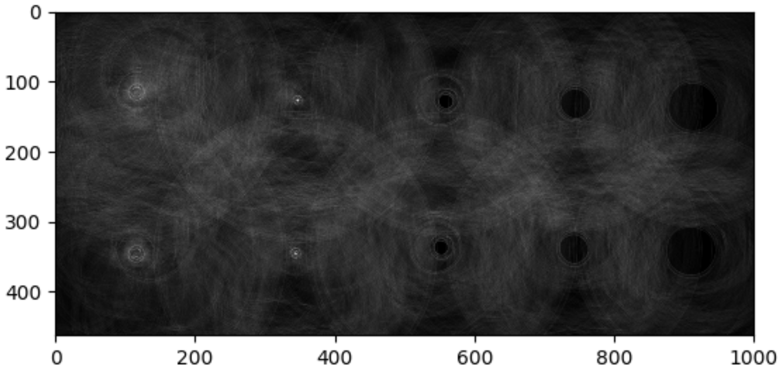

Figure 2: Hough transform for the radius of the 5 mark coin (compare to Figure 1).

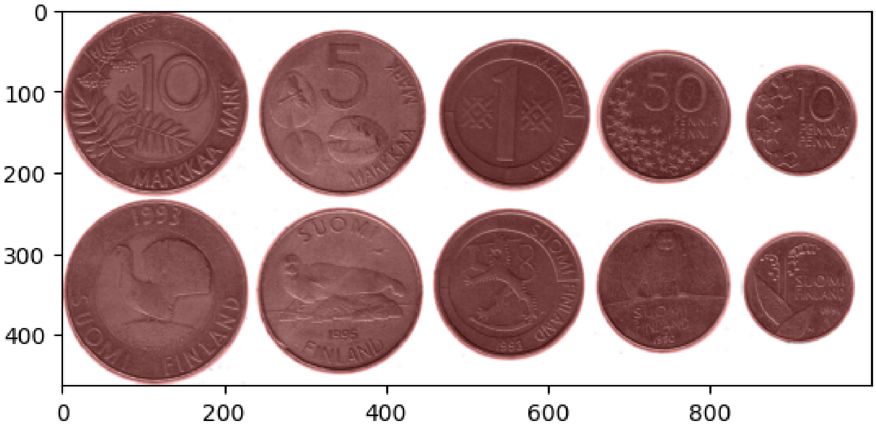

Figure 3: Coin detections shown with colored transparent circles.

- Use `skimage.transform.hough_circle_peaks` to extract peaks from the Hough transform. Select two peaks per Hough space (per radius), and a total of 10 peaks (that is, 2 for each coin). Set `normalize` to `True` to not give preference to larger circles – see the function documentation for details. Get all outputs from the function, i.e., your call should look like: `accums, cx, cy, radii = hough_circle_peaks(...)`.

- Apply `matplotlib.patches.Circle` to superimpose the 10 circles extracted on the original image. See `https://matplotlib.org/api/_as_gen/matplotlib.patches.Circle.html` for more help. Use `from matplotlib.patches import Circle` to import the circle tool to your code. The result should look similar to Figure 3.

Task 2: Line fitting with RANSAC (programming)

In this task, we fit a line to noisy data by the RANSAC algorithm. The data depict detections of an edge in an image, but it contains outliers which we wish to discard.

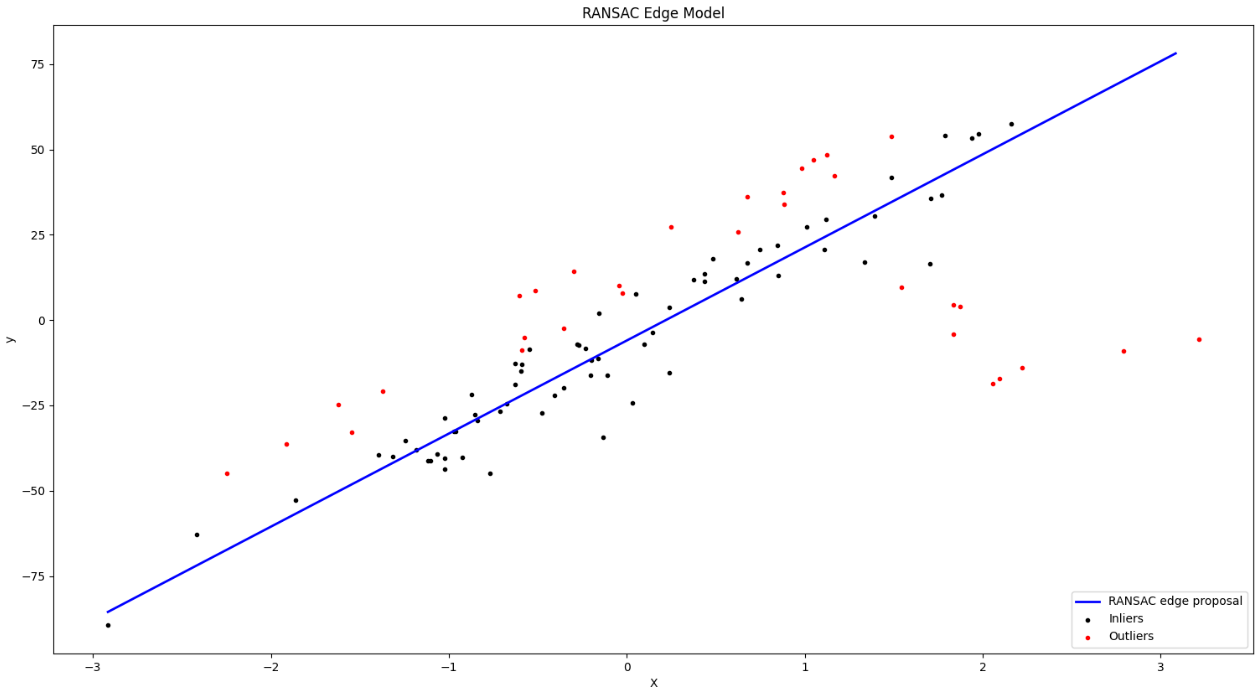

Figure 4: Inliers (black dots), outliers (red dots), and the line fit (blue) found by RANSAC.

- Download the data file `noisyedgepoints.npy` from Moodle. The data can be read by a code snippet such as:

```
with open('noisyedgepoints.npy', 'rb') as f:
    X = np.load(f)
    y = np.load(f)
```

- Find the best fitting line using RANSAC. You can either use the built-in function from `scikit-learn`: `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RANSACRegressor.html`, or, you may implement your own version of RANSAC.

- After fitting the line, plot the outlier and inlier points with different colors, and draw the line fit found by RANSAC. The result should look similar to Figure 4.

Task 3: RANSAC properties (pen & paper)

The fraction of inliers in the data is ϵ, and m points are required to define a single model hypothesis. Prove that

$$k \geq \frac{\log(1-p)}{\log(1-\epsilon^m)}$$

model hypothesis iterations are required for RANSAC to succeed with probability at least p (success: at least one model hypothesis has only inlier points).