General Comments:

1. All the programs are in MATLAB script files. So, they can be directly executed!
2. Comments have been inserted at appropriate parts of the code. Therefore the code is self-explanatory.
3. Three models are implemented for this project. Algorithm for each model is explained below along with additional instructions if needed.

A. HOUGH TRANSFORM MODEL:

1. ALGORITHM:

- Image is converted to grayscale and passed through canny edge filter. Location of edge points in form of 0/1 are stored in 'BW'.
- 'PARAMETER SPACE' is created storing the votes of potential candidates.
- Then the program loops through row and column of 'BW' and calculates values of center coordinates by utilizing parametric equation as mentioned in report. For each of the values, a count of one is incremented in the parameter space correspondingly.
- By finding maximum vote count value in the parameter space, corresponding parameter values (center coordinates & radius) for most confident candidate prediction are determined.
- Visualization plots are created.

2. MODEL INSTRUCTIONS:

- If the image size is large (like 1000*1000*3), then use the commented command in the script: `%origIm=imresize(origIm,0.5);` to scale down it appropriately.

- Change the radius range using the following command according to the nature of image:
  `RADIUS=(25:floor(sqrt((rows*rows)+(columns*columns))/2));`

B. LEARNING AUTOMATA (CIRCLE) MODEL:

1. ALGORITHM:

- Image is converted to grayscale and passed through canny edge filter. Location of edge points in form of 0/1 are stored in 'BW'.
- In order to ensure that the algorithm converges in feasible time, the image is scaled down until the number of edge points 'Edge_Pt_Num' are below certain threshold (like 1600).
- Then the program loops through row and column of 'BW' and stores the coordinate values. Then by using 'randperm' command, the program selects randomly only 5 percent of edge points. The command 'nchoosek' is used to form combinations of three points from this 5 percent edge point pool.
- Using mathematical equations, center coordinates and radius are determined from this set of 3 points. Further, any unwanted candidates are removed. Only 10 percent square of area around image center is taken for center coordinate values to reduce computation time.
- Number of iterations are selected and all the initialization values for Learning Automation algorithm are taken.
- For randomly selecting action set, at each iteration a random number from 0-1 is selected. Actions to be considered are taken until sum of these actions exceeds this random number. This way it is ensured that at each iteration, random order and number of actions are selected for evaluation.
- Reinforcement signal and updated probability values for that iteration are computed according to the equations mentioned in the report.
- Maximum value of probability is located in the last iteration column and predicted result is plotted. Peaks in the probability distribution space are also determined for counting number of circles in the image.

2. MODEL INSTRUCTIONS:

- Change the radius range using the following command according to the nature of image:
  `Radius_Range = [2, floor(sqrt((rows*rows)+(columns*columns))/2)];`
- Values of 'K_max' & 'Learning_Rate' can be changed to play around.
- Very rarely an error occurs during random initialization of action selection. Just execute the program again in that case.
- Assign value of 0.0005 as learning rate generally. For edge point threshold, for the command: `while Edge_Pt_Num>2500;` assign value of 1600 generally and 2500 if the image has more than 20 curves in them.

C. LEARNING AUTOMATA (SUPERELLIPSE) MODEL:

1. ALGORITHM:

- General flow of the program is similar to the Learning Automation (circle) model. Minor changes are explained.
- MATLAB function file 'fit_ellipse' is utilized for constructing ellipse from 6 edge point combinations. (This file is taken from Mathworks website)
- Additional Parameter 'n' & 'k' are passed in the program, which are used to get coordinates of superellipse curve.
- The peak values in the probability curve are sorted in descending order and then relevant number of peaks are filtered out (50% of maximum peak values and above are considered) to get number of tree rings present in the image.

3. MODEL INSTRUCTIONS:

- Change the radius range using the following command according to the nature of image:
  `Radius_Range = [2, floor(sqrt((rows*rows)+(columns*columns))/2)];`
- Values of 'K_max' & 'Learning_Rate' can be changed to play around.
- Very rarely an error occurs during random initialization of action selection. Just execute the program again in that case.
- Assign value of 0.0005 as learning rate generally. For edge point threshold, for the command: `while Edge_Pt_Num>2500;` assign value of 1600 generally and 2500 if the image has more than 20 curves in them.