The Canny Edge Detector:

1. The Canny Edge Detector is designed to be an optimal edge detector algorithm that performs on greyscale images and produces an output image showing the positions of tracked intensity discontinuities.

2. It is a multi-stage algorithm that uses the following stages for detecting edges:

* Noise Reduction
* Gradient Calculation
* Non-maximum suppression
* Double thresholding
* Edge Tracking by Hysteresis

3. **Noise Reduction:**

1. As edge detection is highly sensitive to image noise(which is produced as a result of random variation of brightness and colour information), it must be removed prior to proceeding with edge detection.
2. We use Gaussian blur to smoothen the image which reduces the noise(Gaussian Noise in particular) to remove all the noisy edges in the given image.
3. The following equation is the equation for the 2-D Gaussian kernel of size (2k + 1)x(2K + 1) that is iterated on the image in both the dimensions to reduce the noise.



Here x and y represent coordinates of the kernel where the origin is placed at the centre. σ is the standard deviation of the Gaussian distribution. For larger standard deviations, larger kernels are required in order to accurately perform the Gaussian smoothing.

4. **Gradient Calculation:**

1. This step detects the edge intensity and direction at every single point of the image.
2. The magnitude of the gradient at a point determines if it possibly lies on an edge or not. A high gradient magnitude means the colours are changing rapidly - implying an edge. A low gradient implies no substantial changes.
3. The direction of the gradient shows how the edge is oriented.
4. To calculate these gradients the standard Sobel edge detector is used, which makes it easy to highlight the intensity changes in both the X and Y directions by applying respective horizontal and vertical filters.
5. The following are the standard Sobel kernels for horizontal and vertical directions:  
   
6. The following are used to find the magnitude and direction of gradients at every point:





where  and  are the X and Y derivatives at the point being considered.

VII. Here the image obtained will be having intensity ranging from 0 to 255 which is not uniform. To make this even and remove all the thin edges we go through the following steps.

5. **Non-Maximum Suppression:**

1. In this step, we perform an operation on every pixel in the image such that if it is not maximum then it is suppressed.
2. The pixel values are suppressed by orienting each pixel value into one of the four possible edges direction from that pixel.
3. The four possibilities need to be treated separately to check for non-maximum suppression. If the current pixel value is maximum and its magnitude is greater than the upper threshold, you mark this pixel as an edge.
4. Otherwise, that pixel value is suppressed and no edge is considered to be going through that point.
5. The result of this step is the same image with thinner edges which are broken at some points which will be fixed by another thresholding called Hysteresis.
6. We can also notice some variation regarding the edges' intensity like some pixels seem to be brighter than others which we will try to cover this shortcoming with the next two steps.

6. **Double Thresholding:**

1. In this step, we classify every pixel into three categories: strong, weak and irrelevant pixels.
2. Strong pixels have intensity so high that they definitely are contributing to the final edge. Weak pixels are the one which has their intensity levels smaller than strong pixels and larger than that are considered to be irrelevant. All the other pixels come into irrelevant pixels category.
3. Now we practically define two thresholds which classify the pixel values as above. The higher threshold classes out the higher intensity pixels(higher intensity than the high threshold) and lower threshold classifies the known, relevant pixels(lower than the lower threshold). All the intermediate pixels are considered weak pixels.
4. This results in the image to have only two intensity levels(high threshold and low threshold) and all the relevant pixels are taken care of by the next Hysteresis step.

7. **Hysteresis Edge Tracking:**

1. In this step, we will classify a pixel as strong(which means it is contributing to any of the final edges) if it is having any of the strongly classified pixels are lying in its 8-adjacency.
2. Otherwise, we consider that pixel is not contributing to any edge and we suppress it the minimum value.
3. Finally, the output of this stage is the ultimate edge detected output.

8. The difference between the upper threshold and lower thresholding makes it identify the smaller edges in the image. If the difference is more only the thick edges are produced and if the difference is less then every minute small edge comes into the picture. This selection of higher and lower thresholds varies from image to image depending on the extent of detailing one needs in the final edge detector output.

9. The steps Double thresholding and Hysteresis edge tracking can be combined in some explanations.