2. During *identifying top edge* step:

   The minimum number of pixels in top edge: $Th_{min.cc.top} = \varepsilon *$ E[width of nanorod]. [Suggesion: Select the value of $\varepsilon \geq 1/2$]. ε is a constant to be chosen by user.

   Dilation of top edge: Length of dilation $= \frac{1}{2}(1-\varepsilon)*$ Number of pixels in top edge

3. During *estimating side edge* step:

   The minimum number of pixels in side edge, $Th_{min.cc.side.fit} \approx 20$ pixels. The minimum number of pixels in a side edge data to consider as input for fitting line is governed by requirement for fitting regression line.

   Merging side edge segment: The difference between slope of two line, $Th_{\Delta slope.side}$: 5°

   Dilation of side edge: $\delta*$ The number of pixels in each line segment [Suggestion: $\delta \leq 1/4$]

All parameters used in the algorithm are listed in Table 3.2. The parameters (termed as essential parameters) should be tuned with proper care, and with appropriate understanding of the image nature and the expectation from the algorithm.

3.4 Illustrating Example

This section shows output in each steps of the algorithm by running the algorithm in a section of an image. The code to implement the proposed algorithm is written in Matlab.

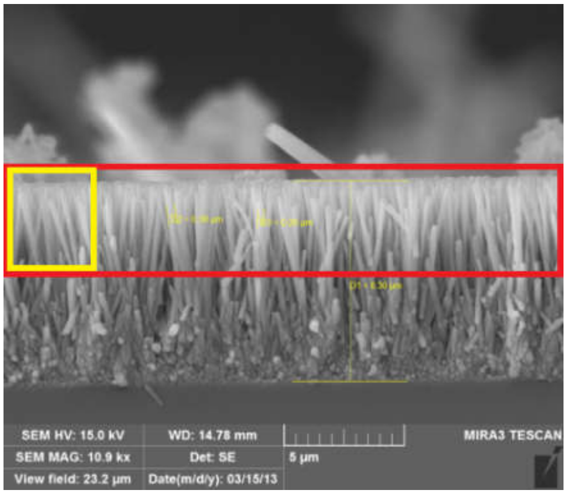
The image to run our proposed algorithm is shown in Figure 3.5 (a). The nanorods are grown vertically, they are NiO-doped p-type Aluminum-doped zinc oxide (AZO) films fabricated by $sol-gel$ method. We run our proposed algorithm to Figure 3.5(b) and the results are displayed, and explained throughout this section. Some applications of AZO are mentioned here. AZO has good conductivity and transparency and relatively low cost compared to other ZnO doped films [59], [8]. AZO nanorod array with hydrogen treatment are likely to be beneficial in different applications (e.g. electronic devices, photoelectrochemical cells, and electrochemical biosensors) [60, 96]. These nanorod arrays can enable a larger surface area and displays an enhanced charge-transport property due to the increased direct electron conduction [44, 10]. Hence, an algorithm that can automatically extract this AZO nanorod from scanning electron micrographs will be beneficial to research purposes.

Table 3.2: List of threshold parameters in proposed algorithm

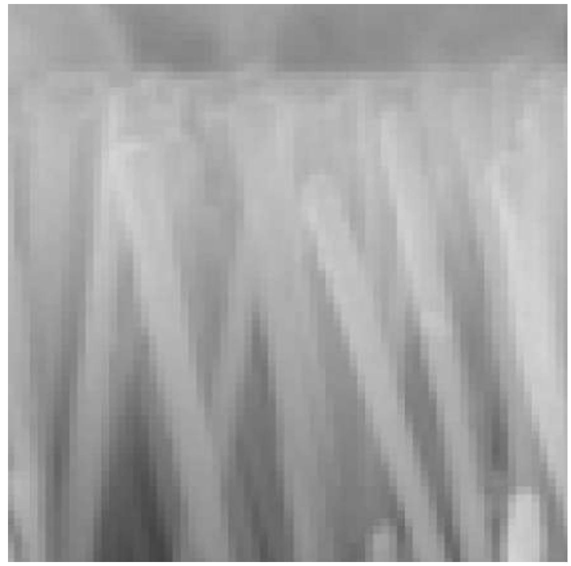
Threshold	**In which step**	**Description**
$Th_{min.cc.top}$	Identifying top edge	The minimum number of connected components in each top edge. If any top edge contains data point less than this threshold, that top edge will be discarded.
$Th_{dilate.top}$	Identifying top edge	The value by which each top edge will be extended/ dilated. After obtaining top edges, we need to dilate those top edges as we need to find the intersections of top edges with side edges.
$Th_{min.cc.side.fit}$	Estimating side edge	The minimum number of connected components in each side edge. If any side edge contains data point less than this threshold, that side edge will be discarded.
$Th_{theta.side}$	Estimating side edge	After robust regression fit, we get lines for side edges. But, from the nature of our nanorod image, all side edges are almost vertical, hence we consider the side edges whose gradient angle falls between 45 to 135
$Th_{\Delta slope.side}$	Estimating side edge	The maximum difference between slopes of two lines to be connected. During connecting each fitted line with corresponding line, this threshold will prevent lines to connect whose slope difference is higher than this threshold.
$Th_{dist.grad}$	Estimating side edge	The maximum distance between the endpoints of two lines along gradient direction
$Th_{dist.grad.normal}$	Estimating side edge	The maximum distance between the endpoints of two lines along normal to gradient direction
$Th_{dilate.side}$	Estimating side edge	The value by which each side edge will be extended/ dilated. After obtaining side edges, we need to dilate those side edges as we need to find the intersections of top edges with side edges.

The outputs of the algorithm to a particular section of the image are displayed here. For the input image Figure 3.5 (a), we run the algorithm in only section marked by *yellow* rectangle. The larger view of yellow inset is shown in Figure 3.5 (b).

- Preprocessing: The input image shown in Figure 3.6(a) is preprocessed as explained in Section 3.2, and the output is shown in Figure 3.6(b).



(a) Original input image



(b) Section of image in (a) used for running the algorithm in the illustration

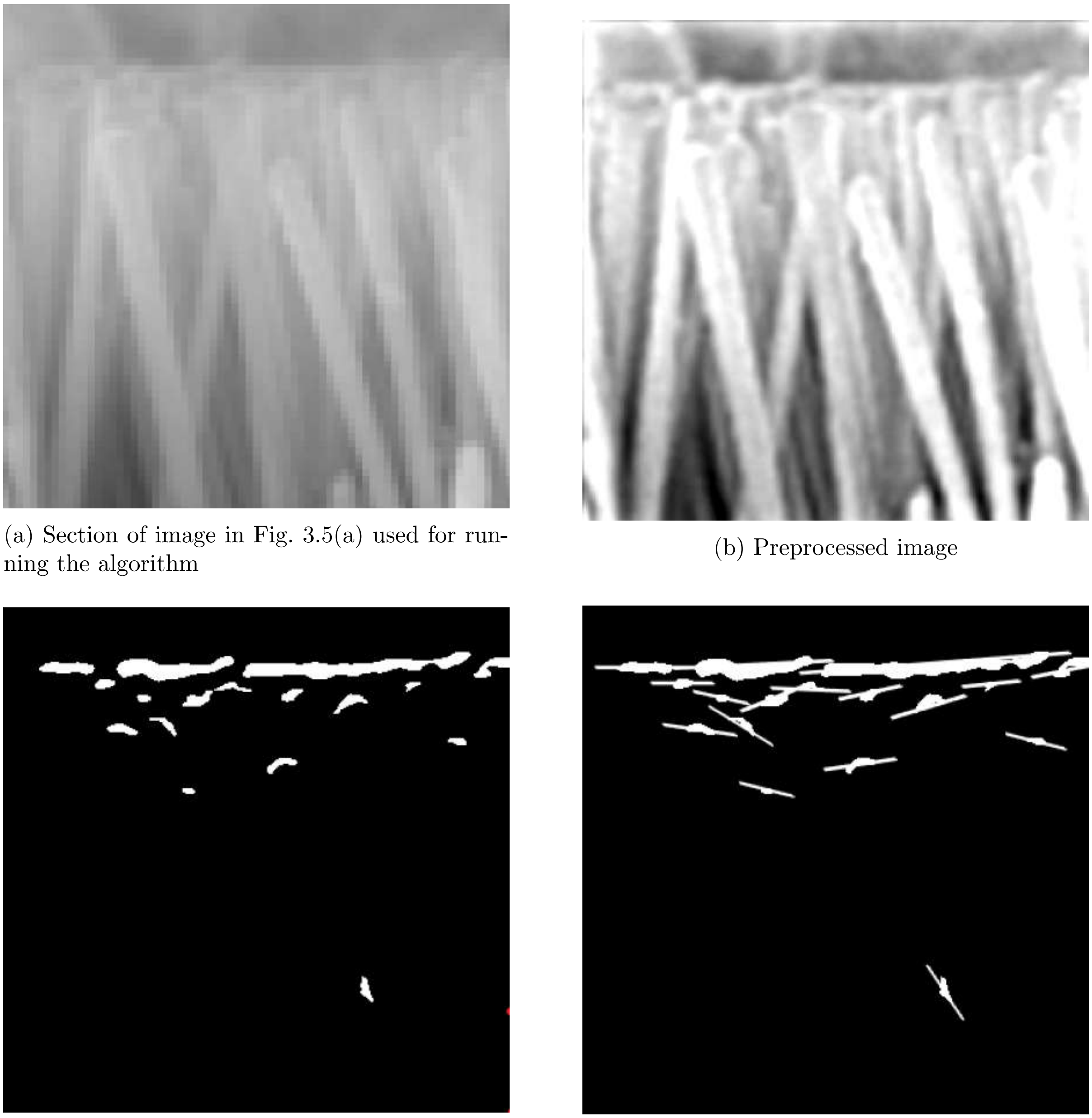
Figure 3.5: Original input image and section of that image

- Identifying Top Edges: The top edges $Edge_{top}$ for input image is obtained by applying Gaussian convolution in Y direction shown in Figure 3.6(c), deleting small components from the image (top edges containing less than 6 pixels are deleted). The top edges are dilated, and $Edge_{top.dilated}$ is obtained in Figure 3.6(d).

- Estimating Side Edges: $Edge_{canny}$, edges of the input image is obtained by applying Canny operator shown in Figure 3.7(a). Now, the pixels of $Edge_{canny}$ that are not intersected with $Edge_{top.dilated}$ are considered for side edge, robust regression is fitted, and $Edge_{side.fitted}$ is obtained shown in Figure 3.7(c).

  A few fitted lines are merged based on their slope difference, distance between endpoints of lines along slope and perpendicular to slope of the lines shown in *yellow lines* in Figure 3.8(a). All possible side edges are dilated shown in *green lines* in Figure 3.8(b).

- Edge-to-nanorod Association: Now, based on the intersection between top edges and side edges, candidate nanorods are extracted. The candidate nanorods might have an inappropriate number of edges. The intersections between top edges and side edges are shown by *blue circle* in Figure 3.9(a), and final nanorod's top edge centroids are shown by circles with *yellow* fill color in Figure 3.9(b).



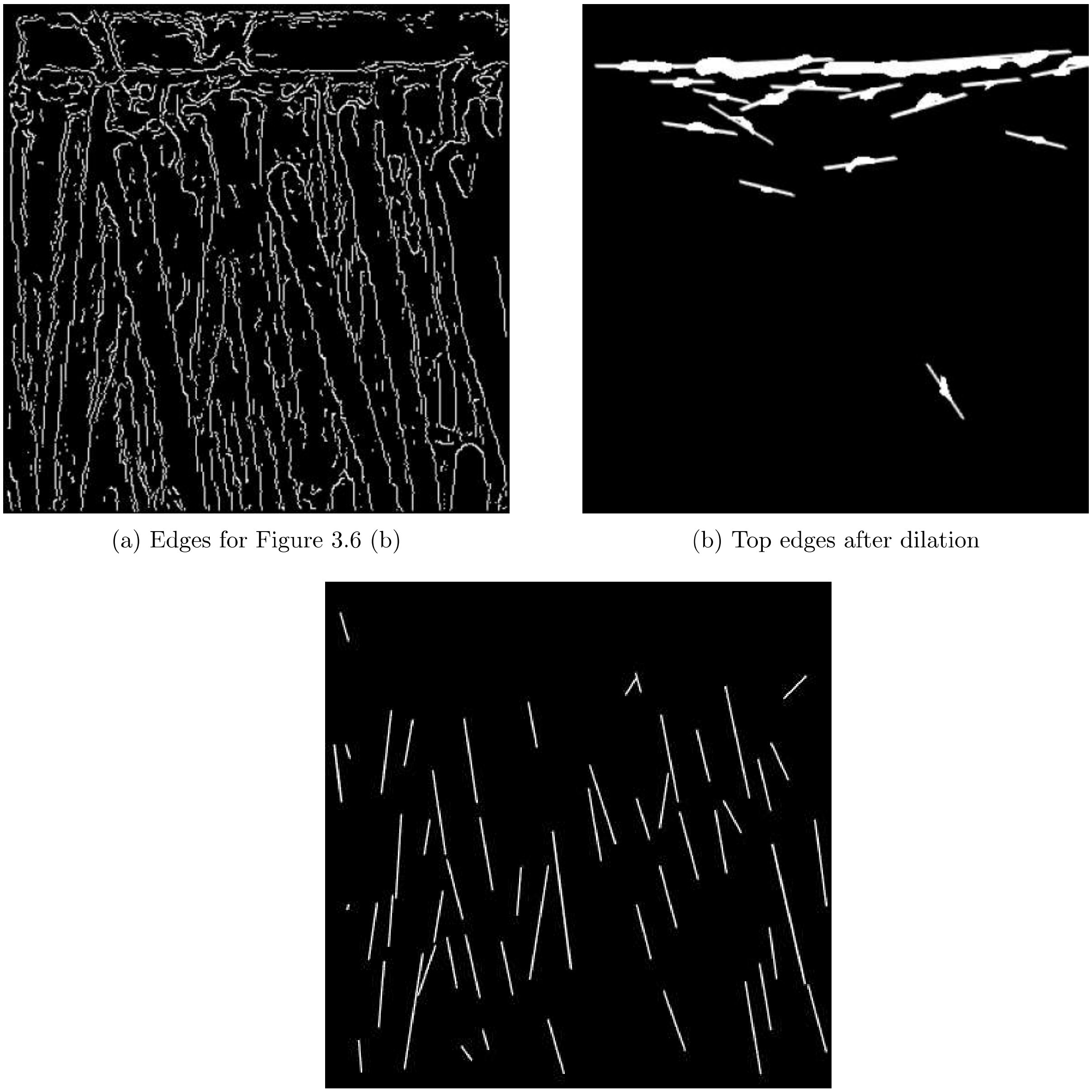
(a) Section of image in Fig. 3.5(a) used for running the algorithm

(b) Preprocessed image

(c) Top edges

(d) Top edges after dilation

Figure 3.6: The outputs in *Preprocessing* and *Identifying Top Edges* step

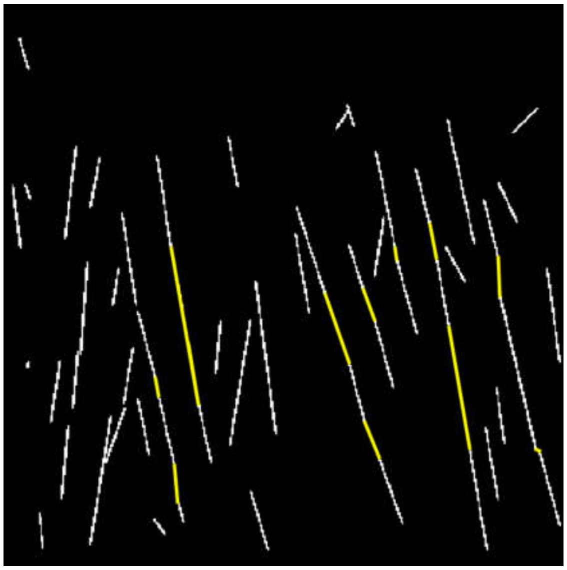


(a) Edges for Figure 3.6 (b)

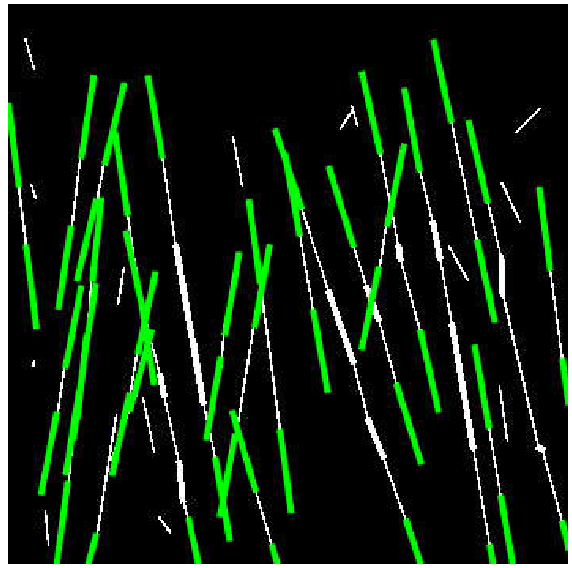
(b) Top edges after dilation

(c) Side Edges in the image after model fit

Figure 3.7: Obtaining side edges

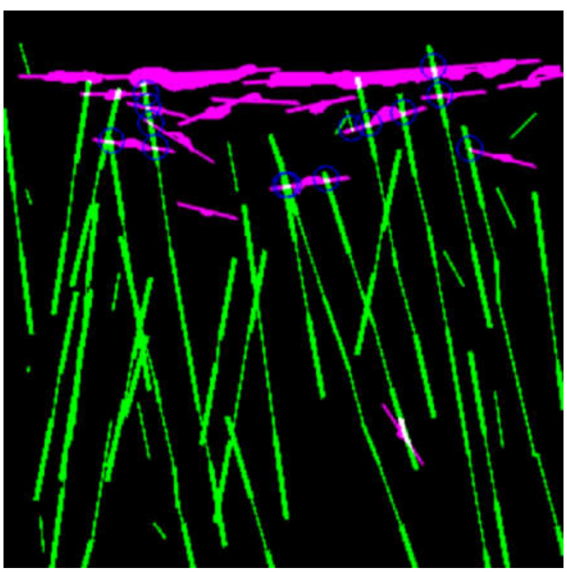


(a) Merged side edges

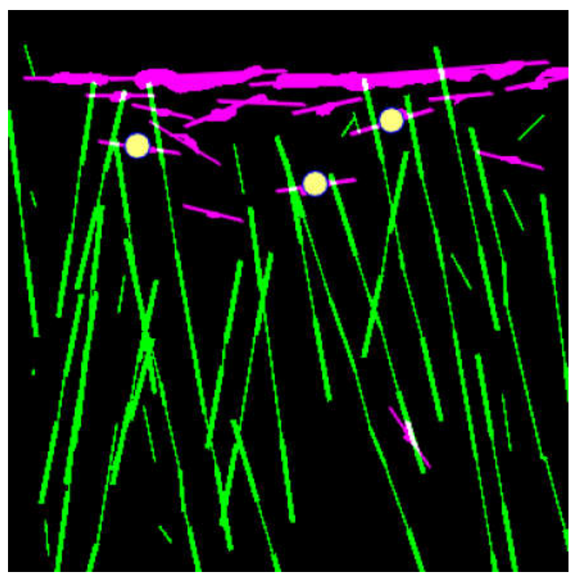


(b) Dilated side edges

Figure 3.8: Outputs after merging, and dilating side edges



(a) Intersections of top and side edges



(b) Final nanorods with top edge; centroids of top edges are marked by solid yellow circles

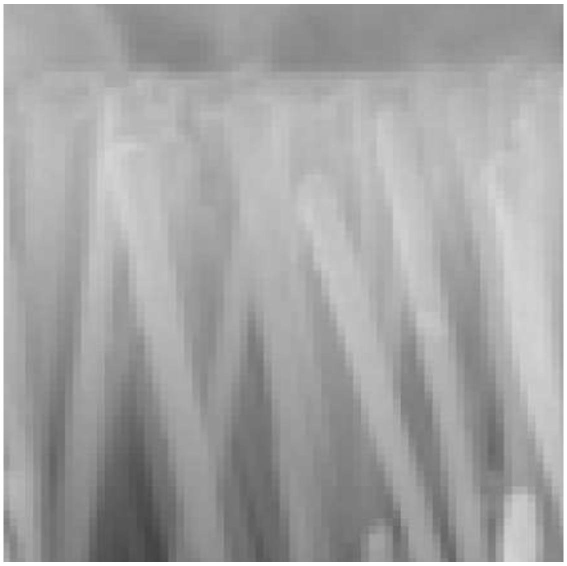
Figure 3.9: Outputs after *Edge-to-nanorod Association* and *Estimating Projection Lengths* steps

The final nanorods (with exactly 3 correct edges for each nanorod) are shown in Figure 3.10(b). The *yellow* circles shown are the centroids of each top edge of final nanorods in Figure 3.10(c).

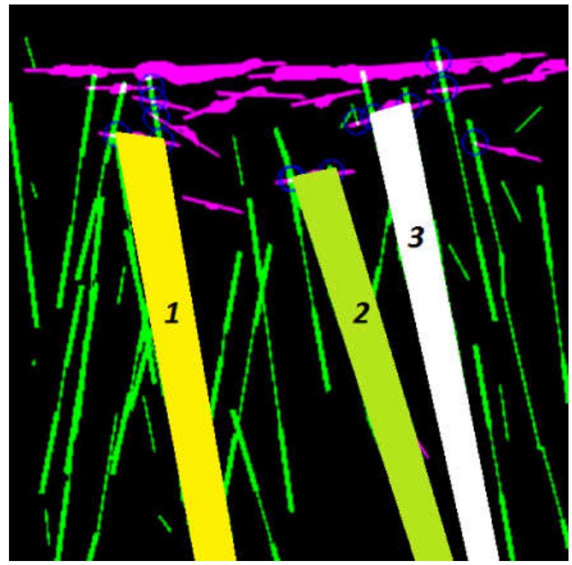
3.5 Discussion

The illustration presented in this chapter shows that the algorithm works well for a small section of an image. The major advantage of the proposed algorithm is it can extract a good number of nanorods from image. In the illustration, there are five nanorods in the input image from which three nanorods are extracted by the algorithm. The performance of the algorithm to entire image is shown in next chapter. Some nanorods are too small that we can ignore them. Hence, we ignore the nanorods whose heads lie near substrate. Any nanorods below selected areas by yellow border in Figure 3.1(b) are not considered.

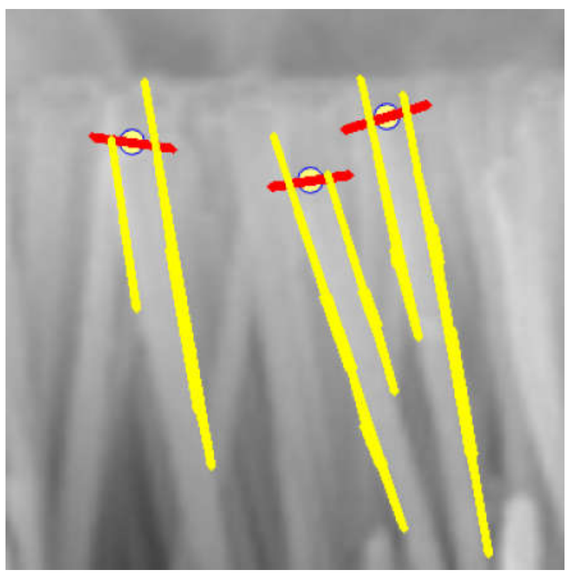
There are several well established segmentation methods that can divide the image into proper segments. But, our research task is not only segment the image, but also extract nanorods. Therefore, we need to associate each possible segmented edges with corresponding nanorods. This task is successfully performed by our proposed algorithm. The proposed algorithm overcomes the limitations of existing algorithms to detect nanorods from scanning electron micrograph. We run the algorithm in several other nanorod images (illustrated in next chapter) to evaluate the performance of our algorithm.



(a) Section of image in Fig. 3.5(a) used for running the algorithm



(b) Extracted nanorods



(c) Final nanorods: left and right edges are marked by yellow color, top edges are marked by red color

Figure 3.10: Input and final output from propsoed algorithm for the illustrating example