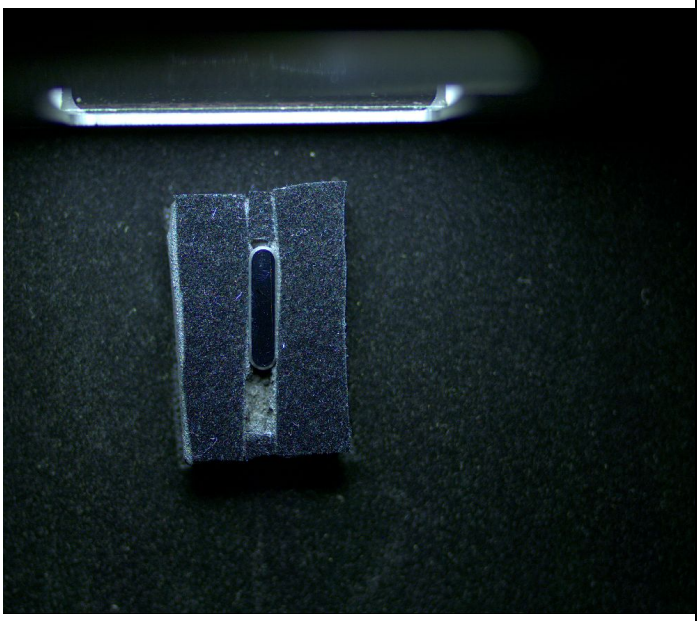
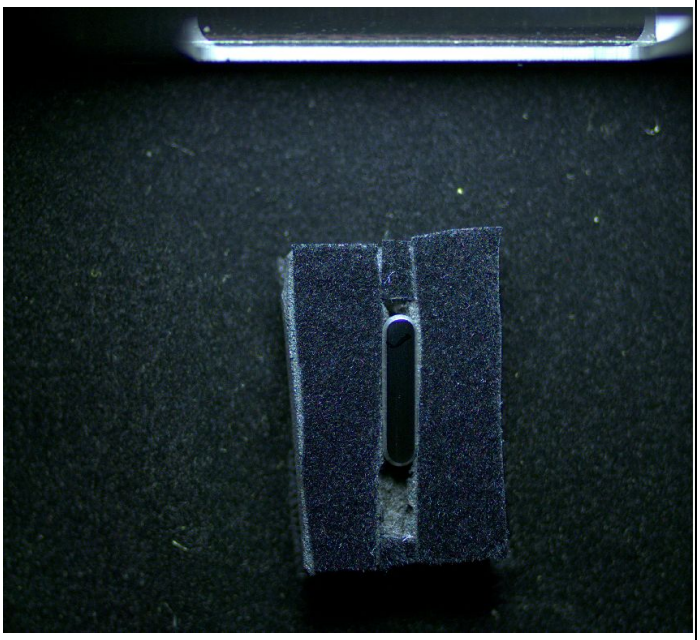
Tittle:

[defect detection automatically on phone button]

All of the steps are done **automatically**

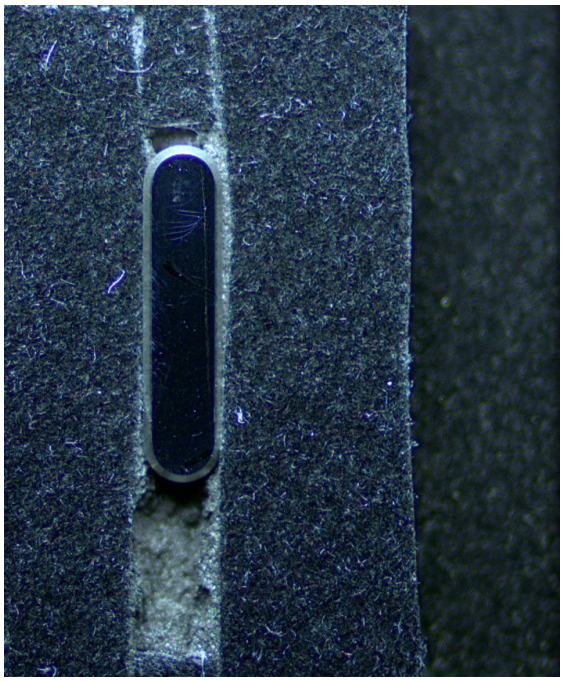
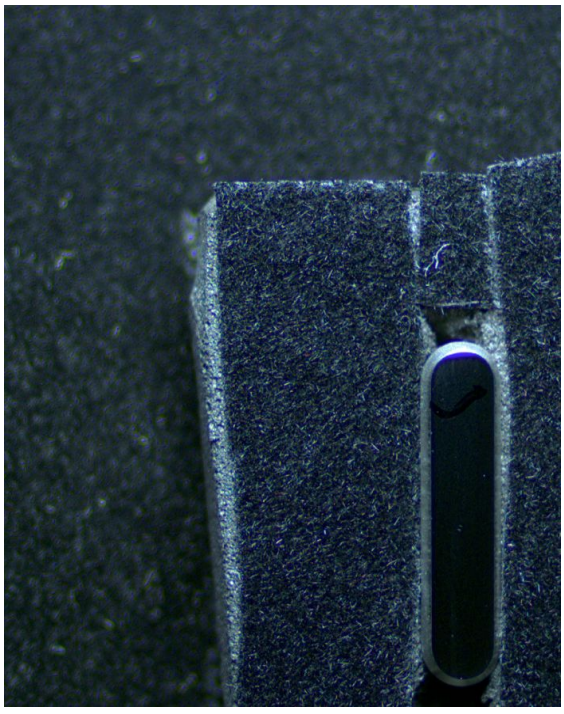
I. PRE-PROCESSING DATA

Step 1: (raw, hand) Crop center of image

Image	**Input:** Big size image	**Output:** After crop
1		
2		

Step 2: Crop eclipse area

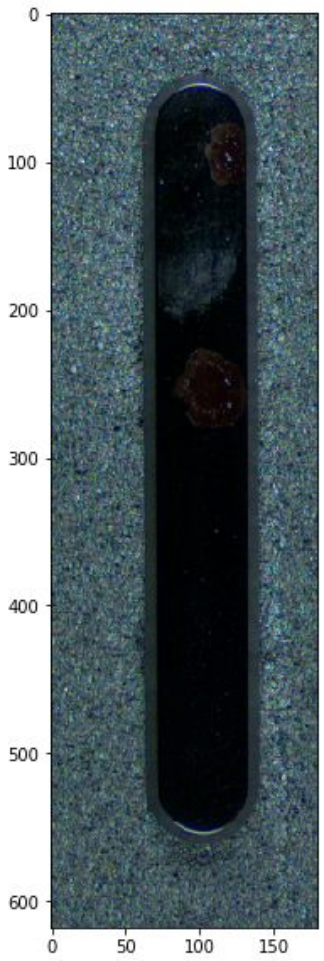
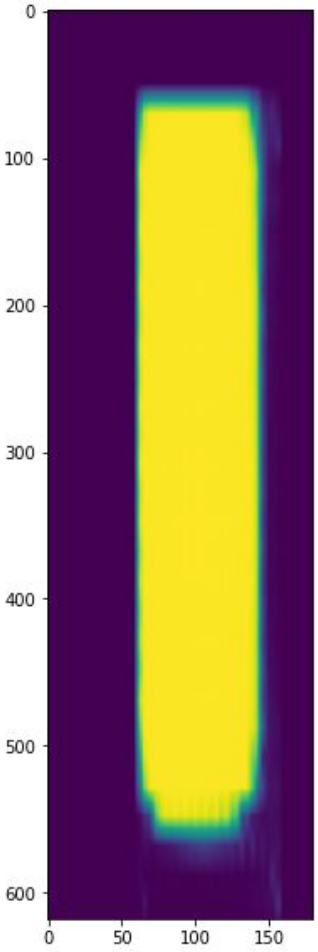
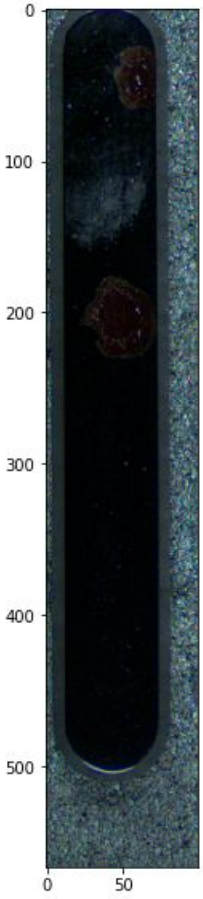
Technology: Dilation, Erosion, Threshold

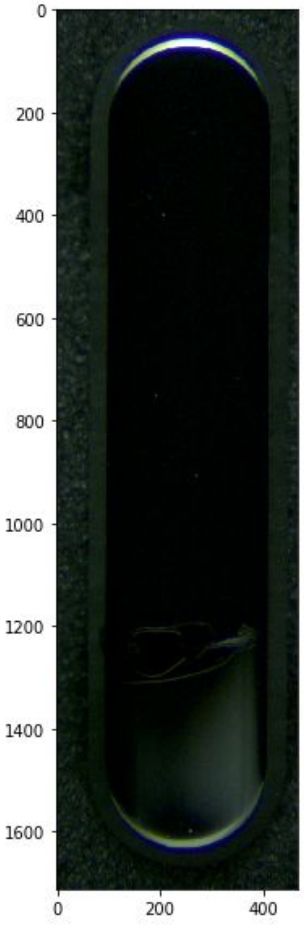
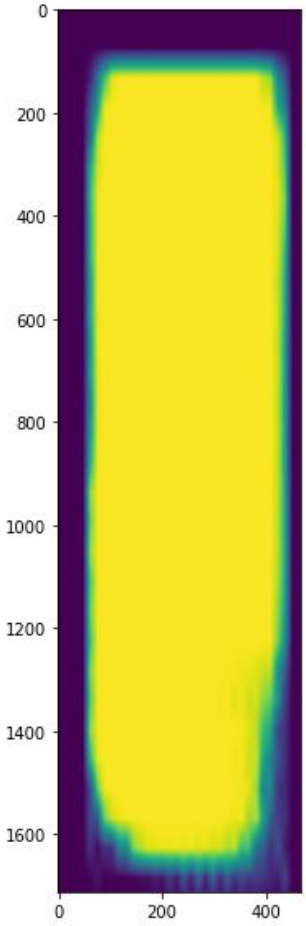
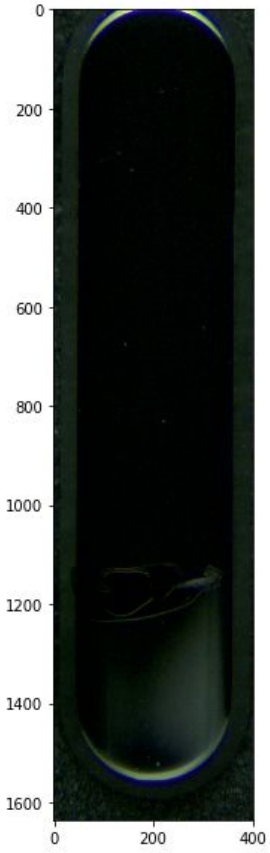
Img	Input:	Output:
1		
2		

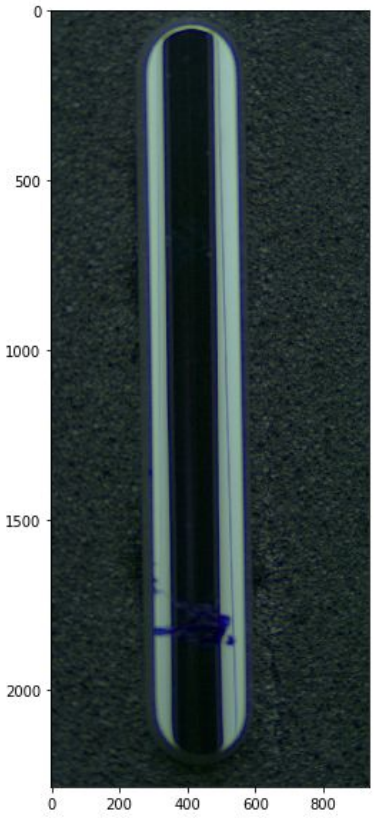
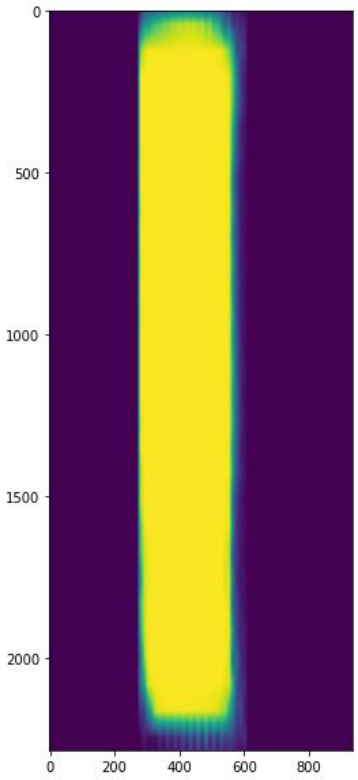
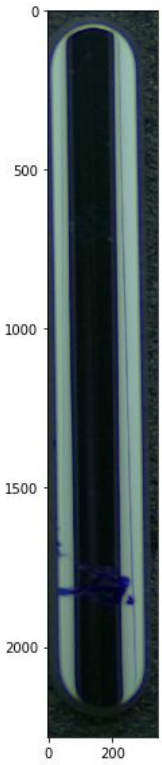
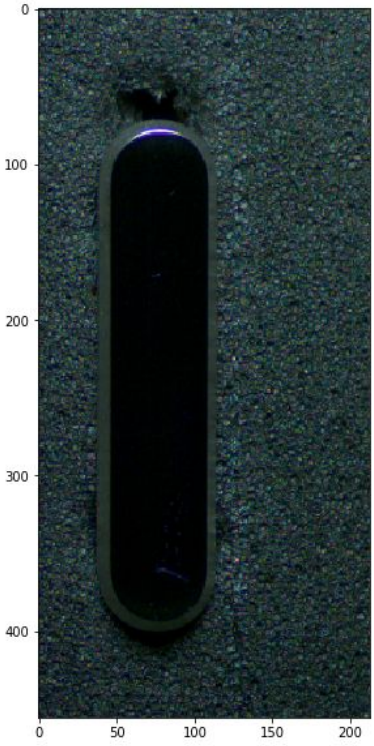
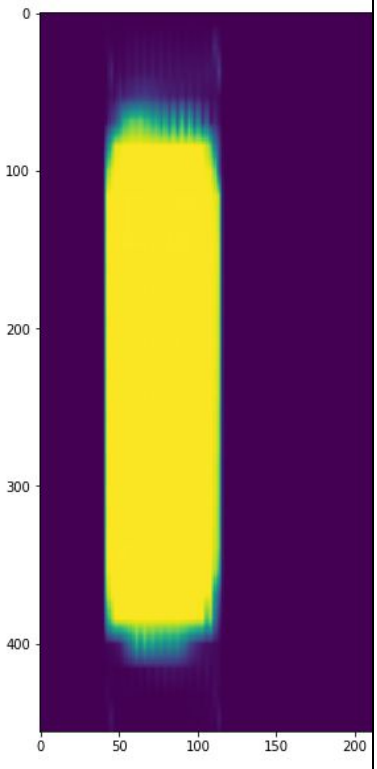
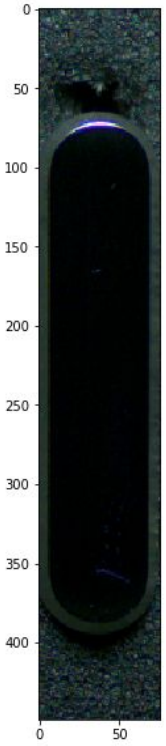
Step 3:

Detect eclipse button (training) → **Mask** (training)→ **Cut eclipse** (inference)

Technology: R-CNN

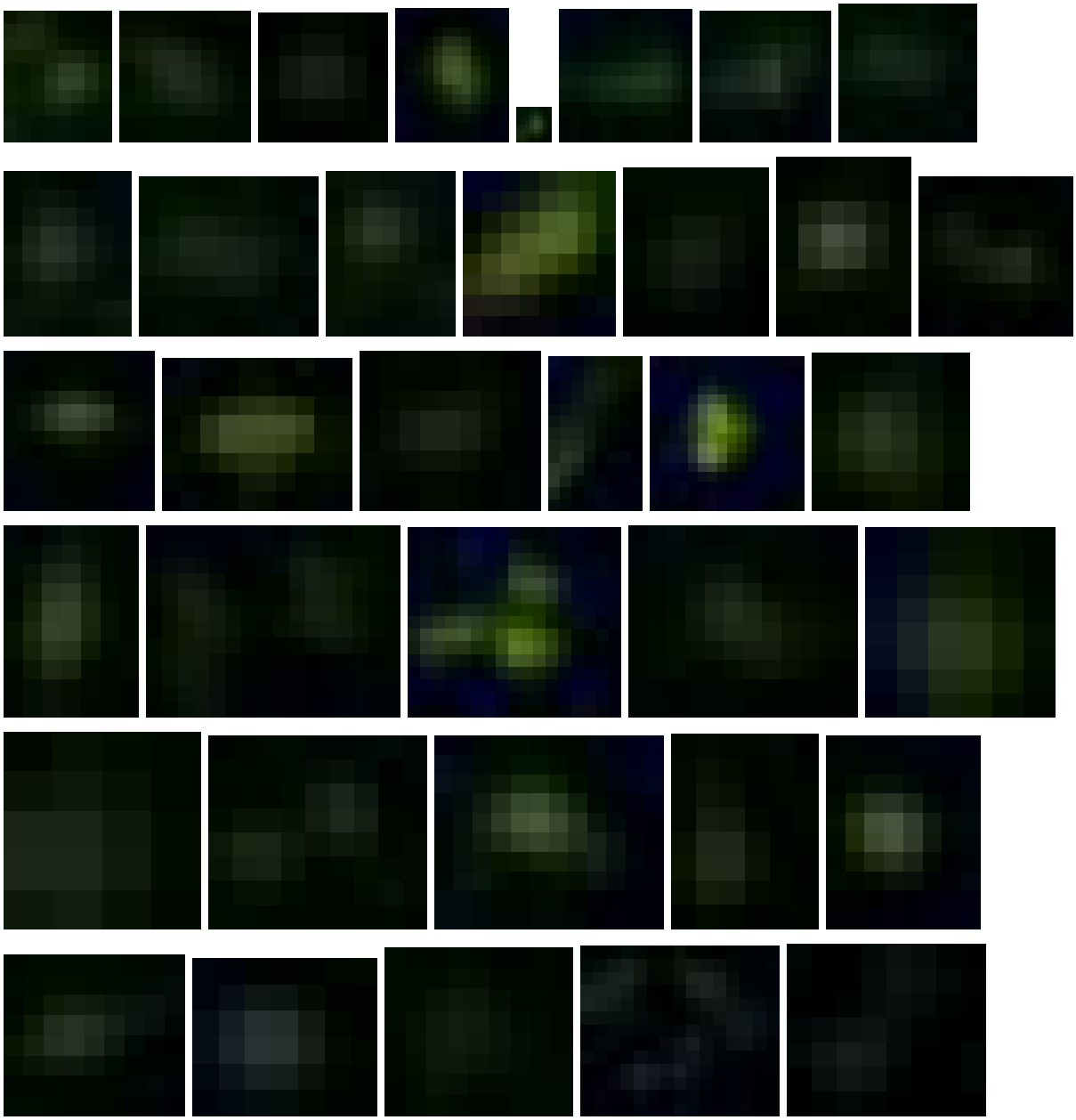
Img	Input	Output: mask	Output: ROI
1			

2	 0 200 400 600 800 1000 1200 1400 1600 0 200 400	 0 200 400 600 800 1000 1200 1400 1600 0 200 400	 0 200 400 600 800 1000 1200 1400 1600 0 200 400

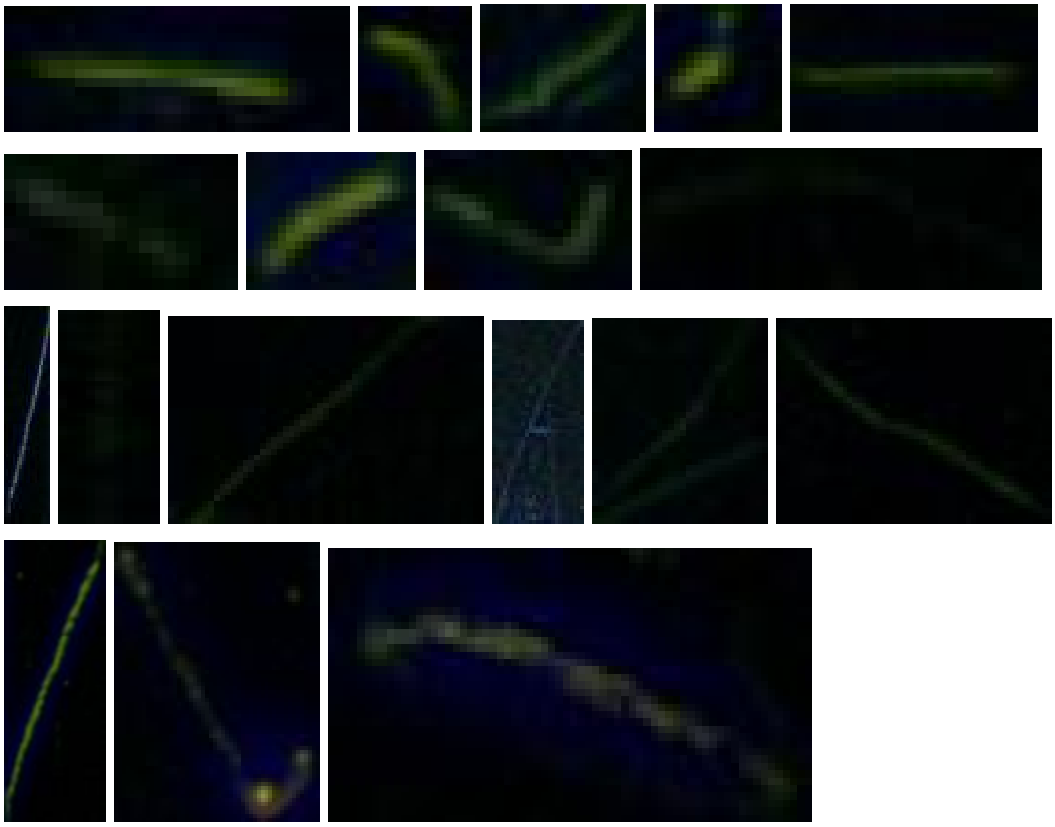
3	 0 500 1000 1500 2000 0 200 400 600 800	 0 500 1000 1500 2000 0 200 400 600 800	 0 500 1000 1500 2000 0 200
4	 0 100 200 300 400 0 50 100 150 200	 0 100 200 300 400 0 50 100 150 200	 0 50 100 150 200 250 300 350 400 0 50

II. DETECT DEFECT & CLASSIFY

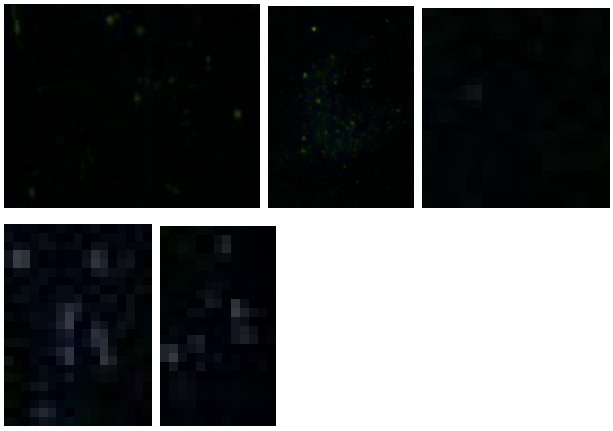
+ **Type of defect: Dot**



+ **Type of defect: Line**



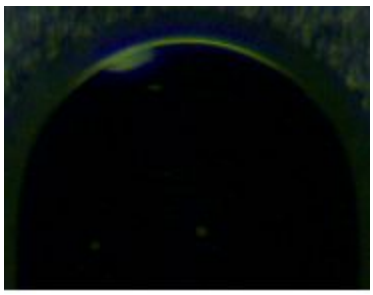
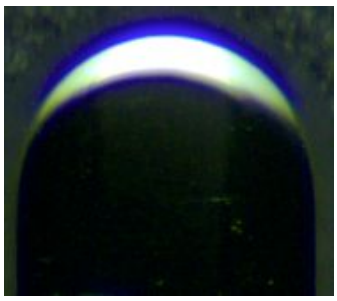
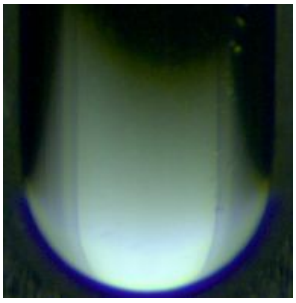
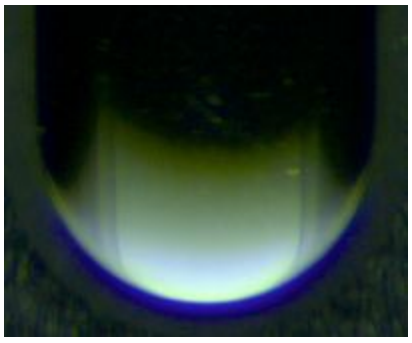
+ **Type of defect: Sparse**

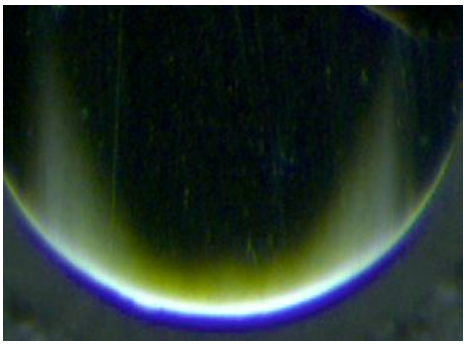


+ **Type of defect: Density**

+ **Type of defect: Whitefog**

+ **Type of defect: Corner**



Marker