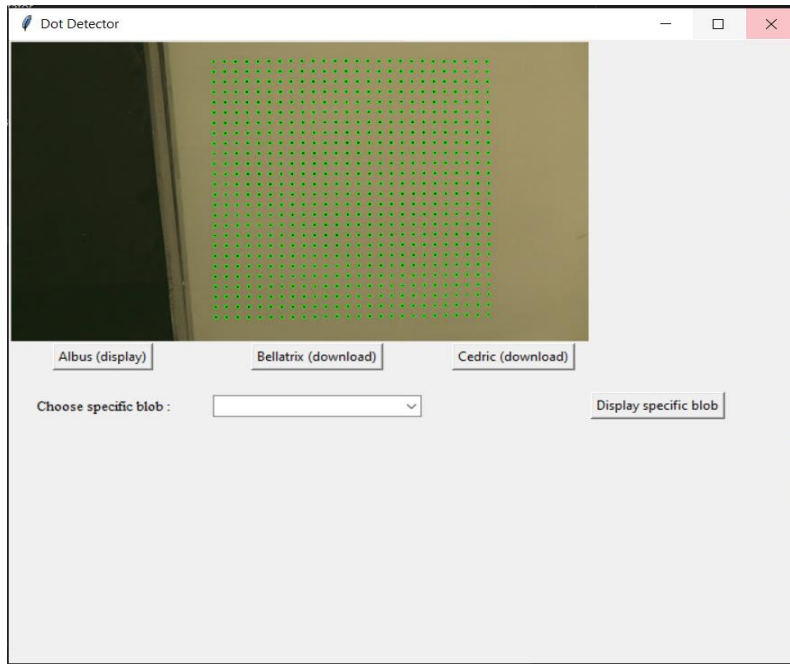


# Dot Detector

By Pritisman Kar



**Requisite:** cv2 and numpy

Displays processed image with dot's highlighted as shown in the left image irrespective of their radius.

**Albus button:** to display if any dot is 2 pixels wide with 0.02 pixels tolerance. If there is any, the same dot will get highlighted with green enclosing contour.

**Bellatrix button:** Download enclosing circle center (X, Y from top left corner) and radius in mils of all dots in CSV file. Tells if a specific blob/dot is 50 pixels wide or not.

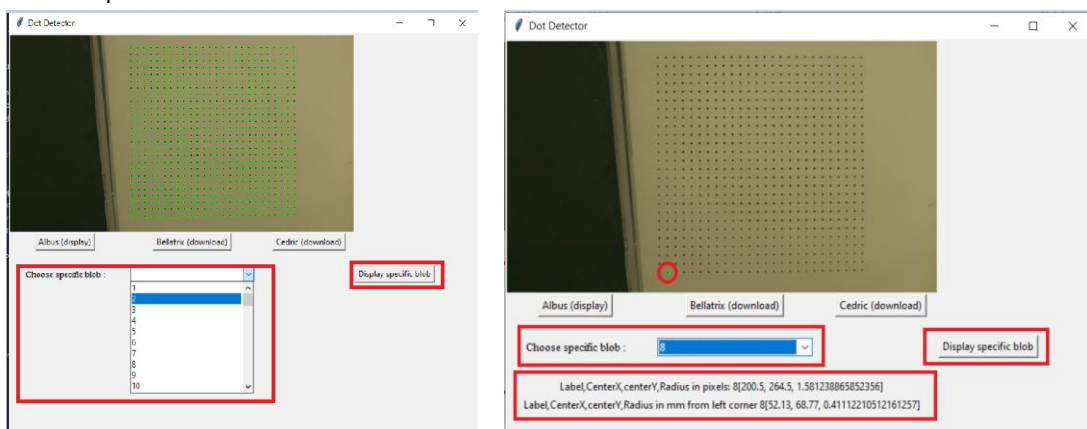
**Cedric button:** Download each dot's contour pixel in text file (each coordinate is TAB delimited)

```
filename = 'test1.jpg'
# Load image
input_image = cv2.imread(filename)
# get circle info and center details
cir_info, xcnts = blob_detector(input_image)

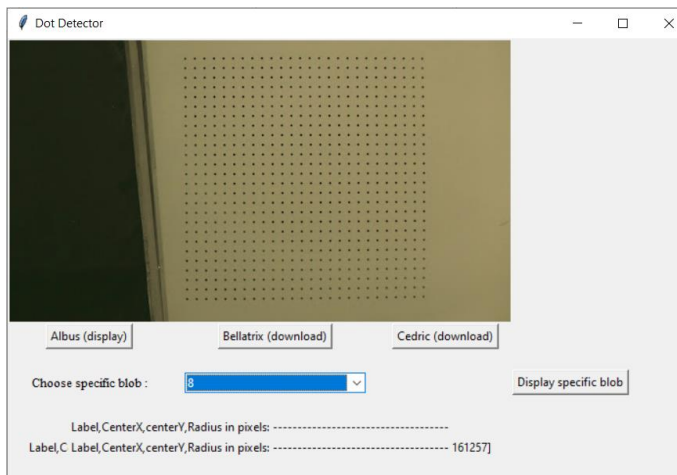
# draw contours based on the extract blob corners
temp_image = input_image
temp_xcnts = xcnts
```

The processed image is displayed when the program – 'detect\_dot\_app.py' is run. The input image is hardcoded and needs to be changed directly in the program. Please refer the screenshot below.

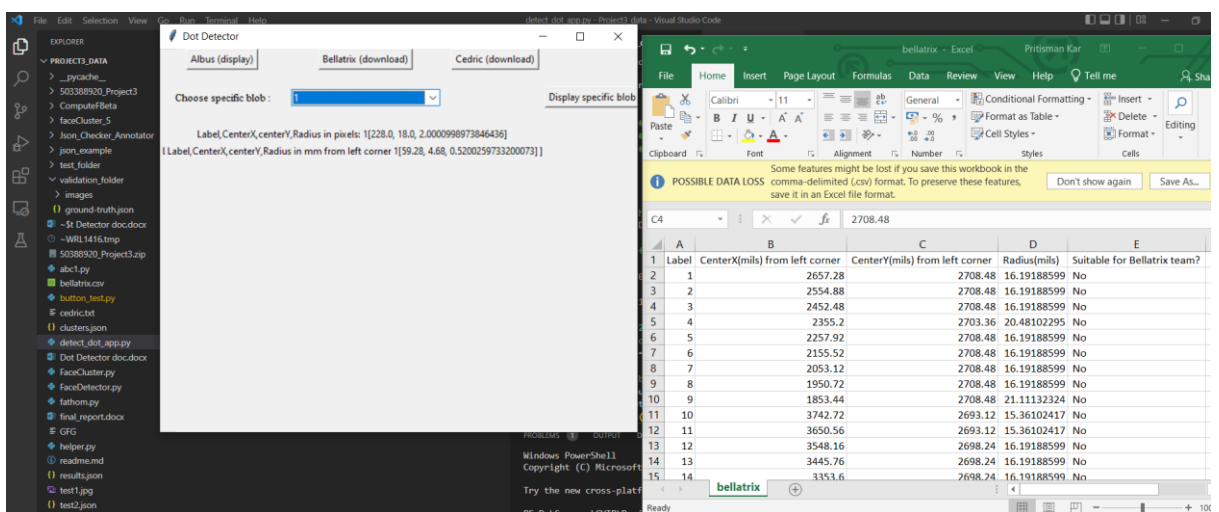
- One can choose the label/counter/number in the dropdown menu ('choose specific blob') and click on 'Display specific blob' to highlight it in the image as well as print its enclosing circle X, Y coordinates, radius in pixel and mm value.



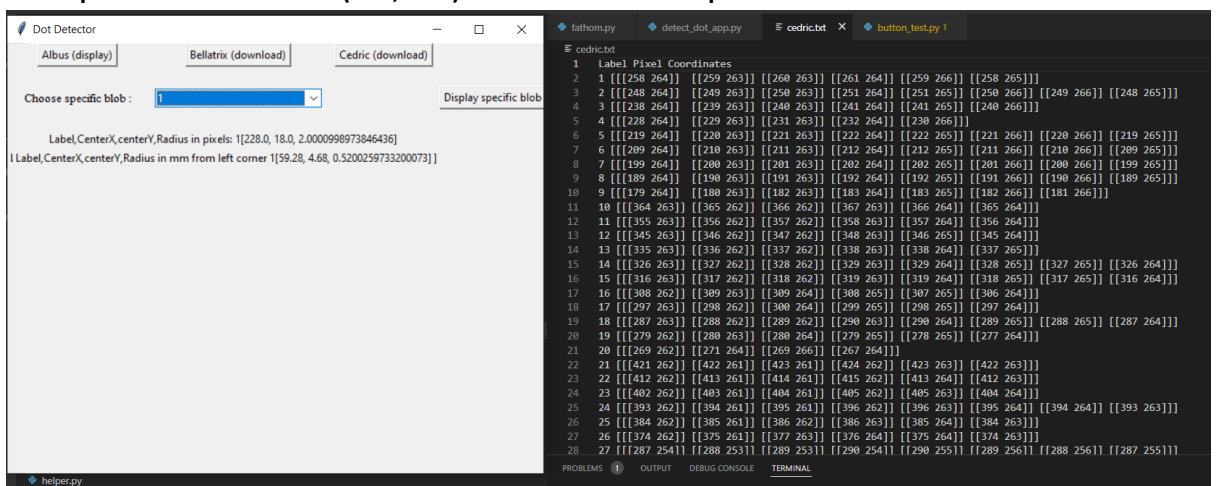
- NOTE – In order to detect all highlight blob's again, select 'A' at the end of the dropdown and click on 'Display specific blob'.
- Results from clicking 'Albus(display)', 'Bellatrix(download)' and 'Cedric(download)' has been shown below.



Since, the filter condition for Albus doesn't matches with any dot in the test image, no dot was highlighted in this case.



- bellatrix.csv got downloaded in the execution folder. The column name and some rows are displayed above.
- **Assumption – All the distance (mm, mils) are measured from top left corner.**



- As you can see in the above image, after clicking on 'Cedric(download)' button, a 'cedric.txt' file gets created with all contour coordinates (vertices) separated by TAB.
- **Logic to detect Dot/blobs in the image:**
  - Normal cv2 function such as 'drawContours' fails to eliminate noises (present in the part's boundary) in the image.

- So, first the input undergoes simple thresholding as adaptive thresholding didn't yield expected result.
- This is followed by erosion to enlarge the black dot's/blob in the images. This helps in merging the noise present in the part's boundary with the boundary itself creating a large black spot.
- Specific area and number of vertices criteria filters are applied to remove these noises from consideration. This is set through trial and error method.
- Then all the enlarged blob that satisfies the above condition, are considered for extracting patches (bounding box) from the original image and determine the contour, enclosing circles (using standard cv2 function) inside each patch.