

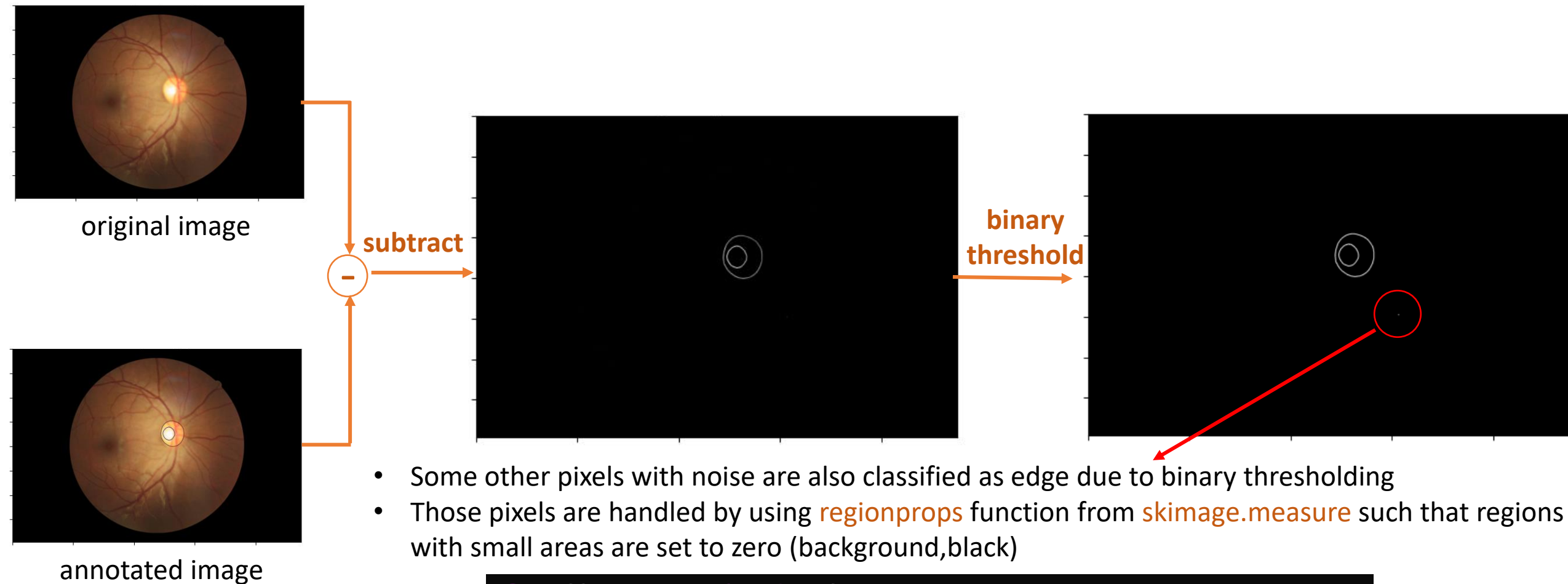
Inner Edge Detection



```
import cv2
from skimage.morphology import closing, square

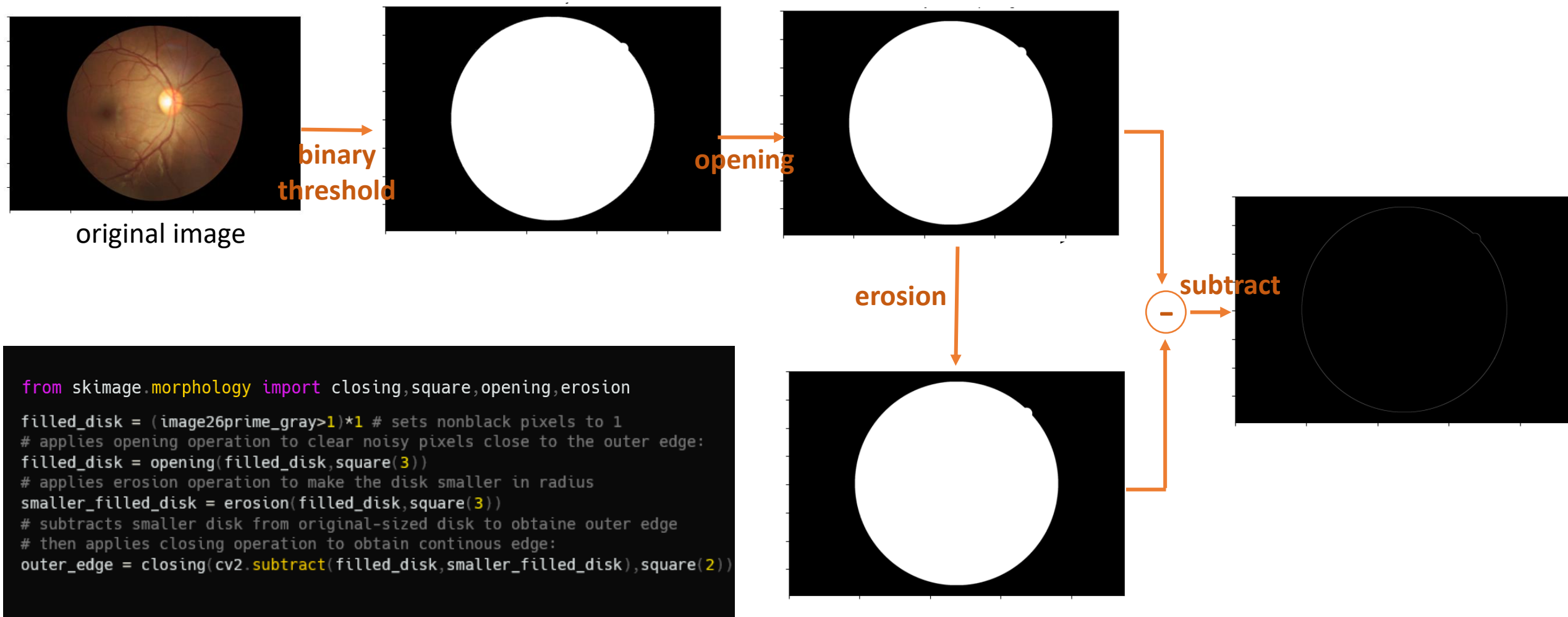
difference = cv2.subtract(image26prime_gray,image26-2_gray) # subtracts two gray scale images
threshold = 30 # threshold pixel gray value
# sets pixels with gray value > threshold to 1, else to 0:
_,inner_edge = cv2.threshold(difference, threshold, 1, cv2.THRESH_BINARY)
inner_edge = skimage.morphology.closing(inner_edge,square(3)) # closing operation to have a continuous edge
```

Inner Edge Detection

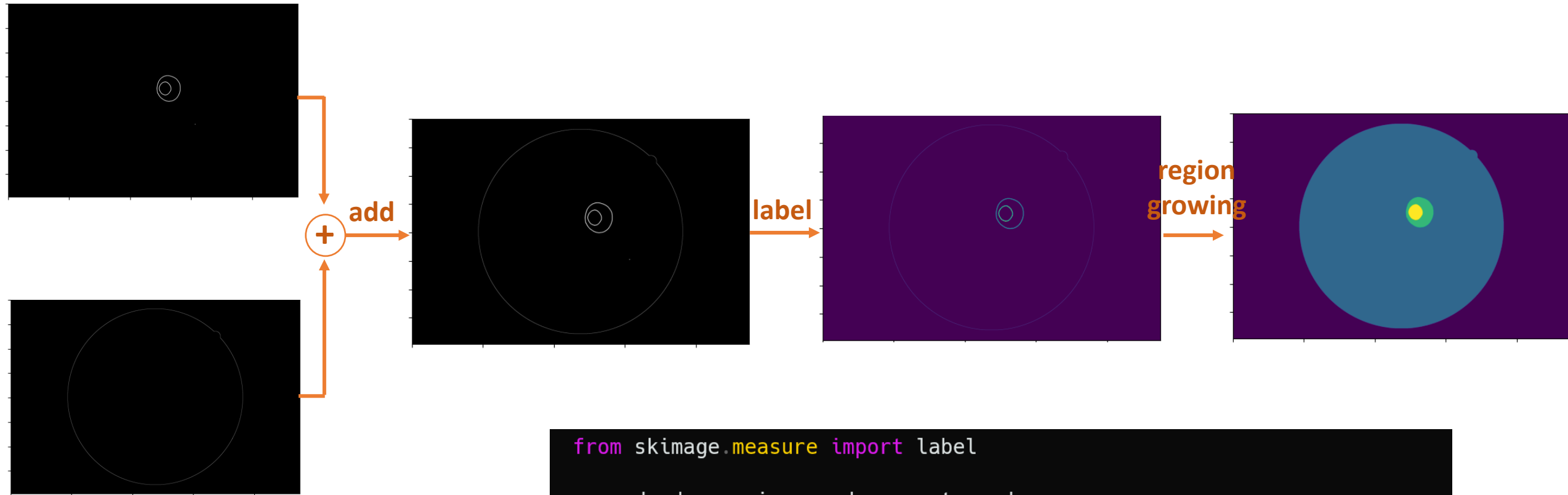


```
from skimage.measure import regionprops
# pixel values in regions with area less than 500 are set to 0 (black,background)
for prop in regionprops(labeled_image):
    if (prop.area < 500):
        for coord in prop.coords:
            merged_edges[coord[0], coord[1]] = 0
```

Outer Edge Detection

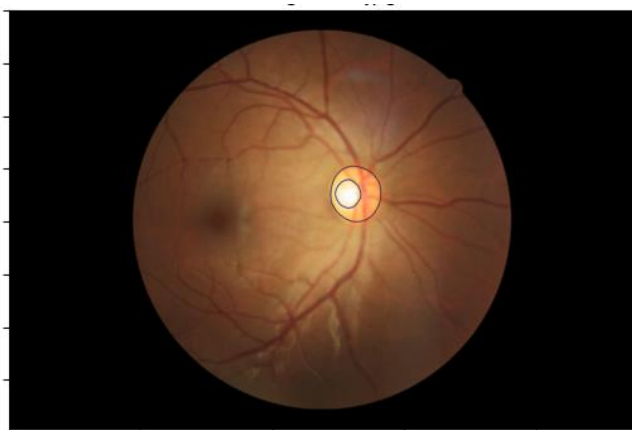


Labeling Regions



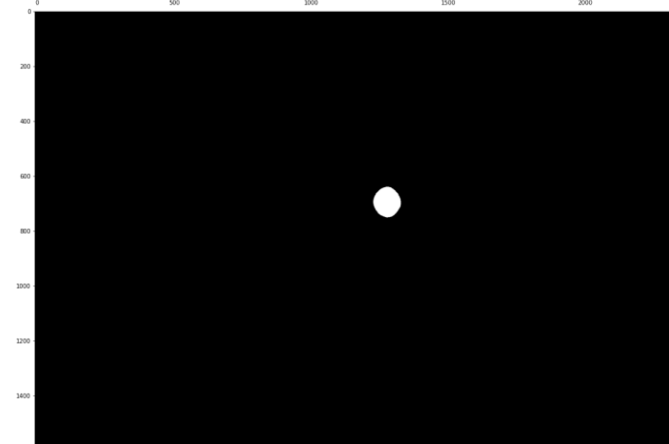
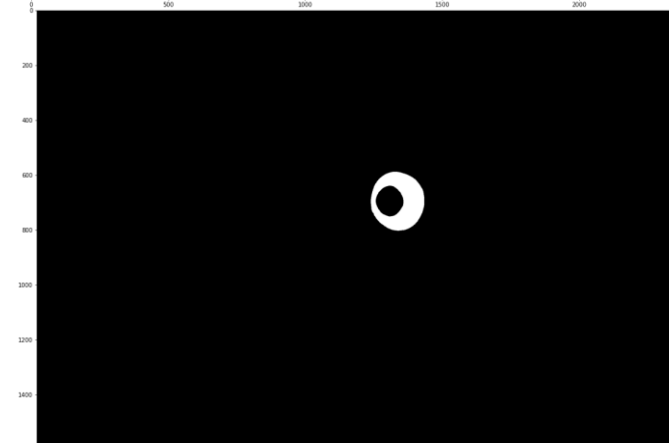
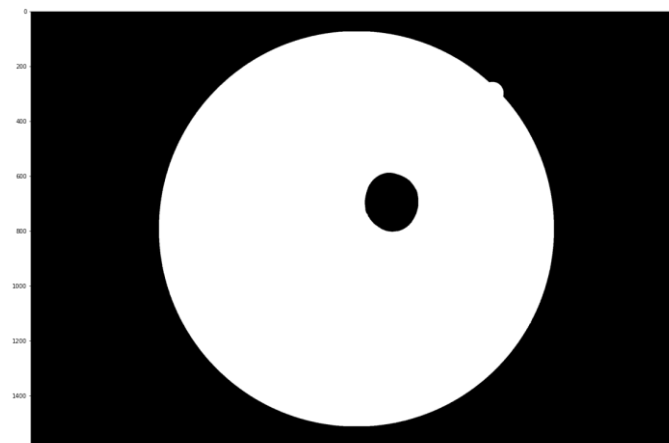
```
from skimage.measure import label

merged_edges = inner_edge + outer_edge
labeled_image, _ = label(merged_edges, return_num=True, connectivity=2)
segmented_image = regionGrowingAlgorithm(labeled_image)
```

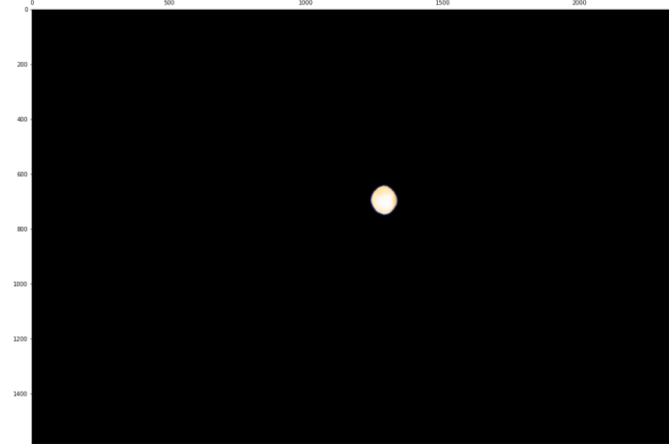
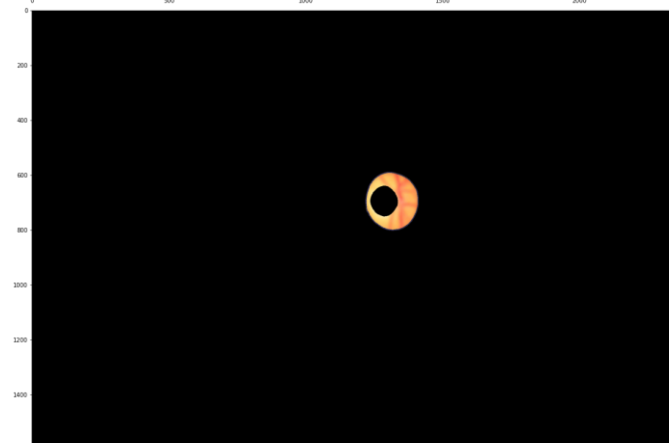
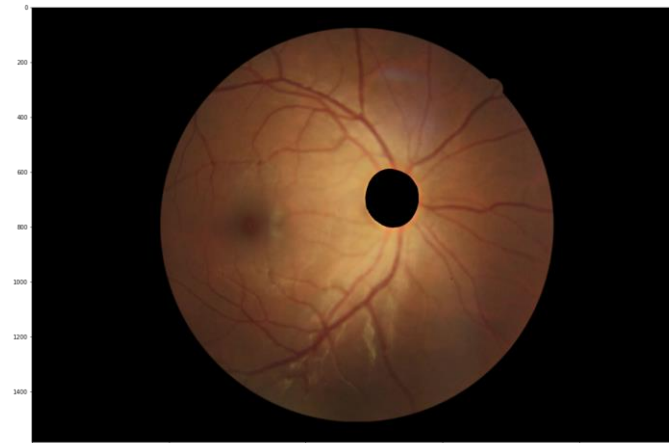


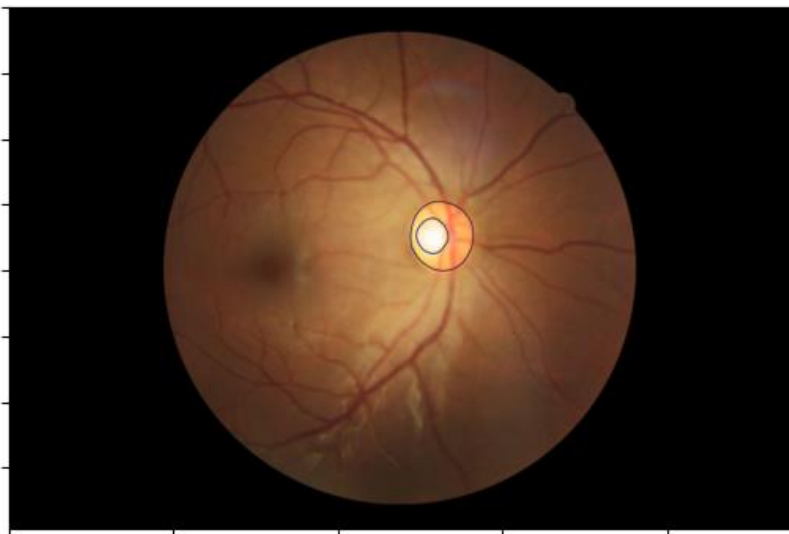
annotated image

Mask



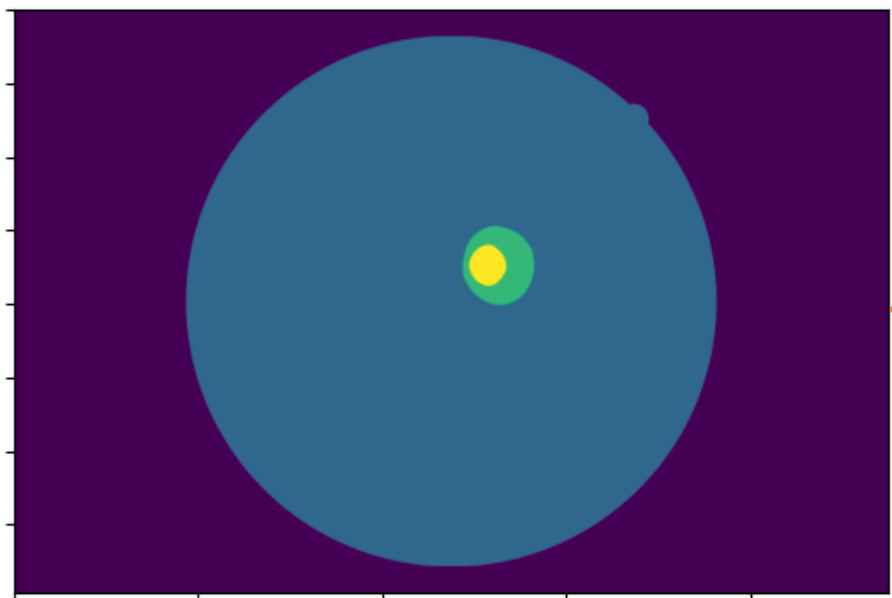
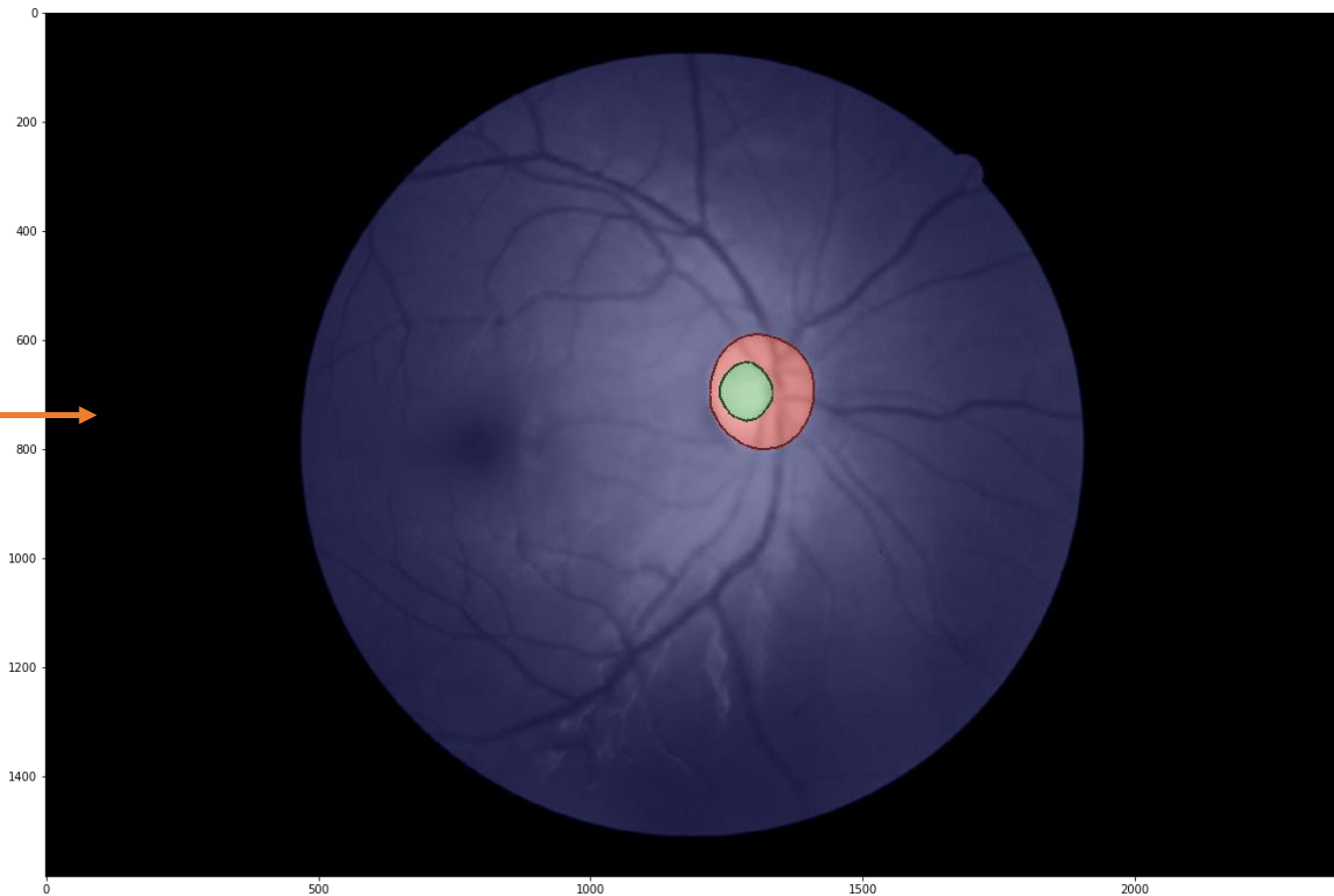
Masked



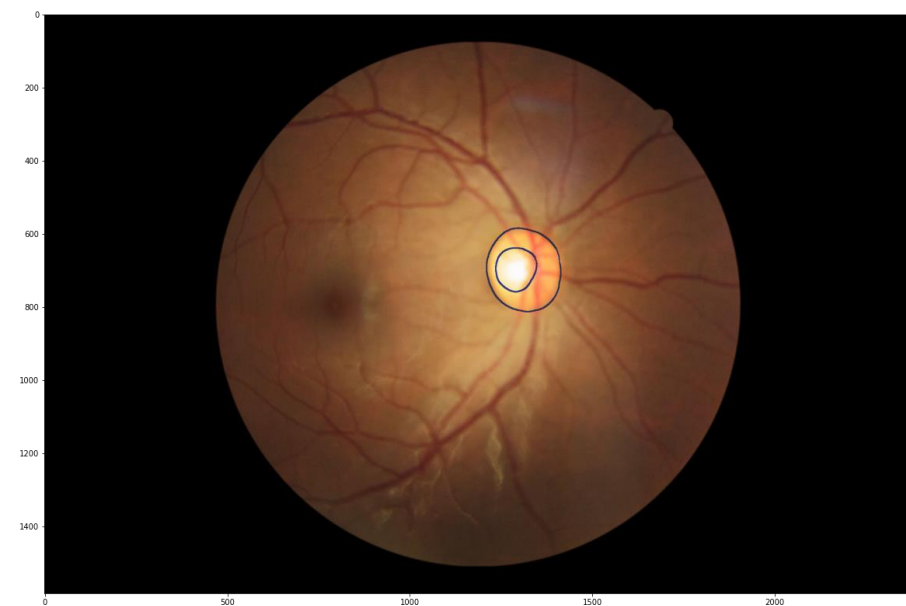


annotated image

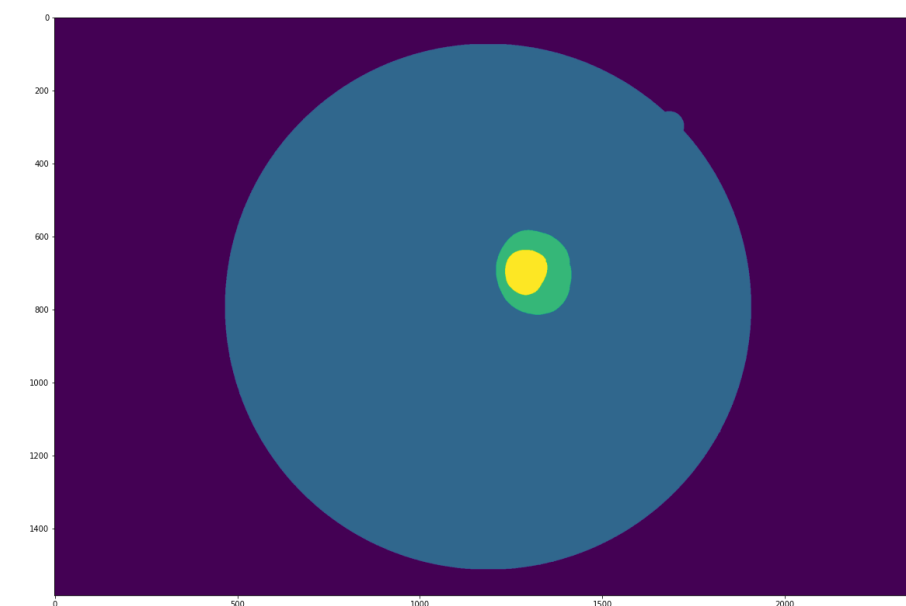
overlay



segmented image

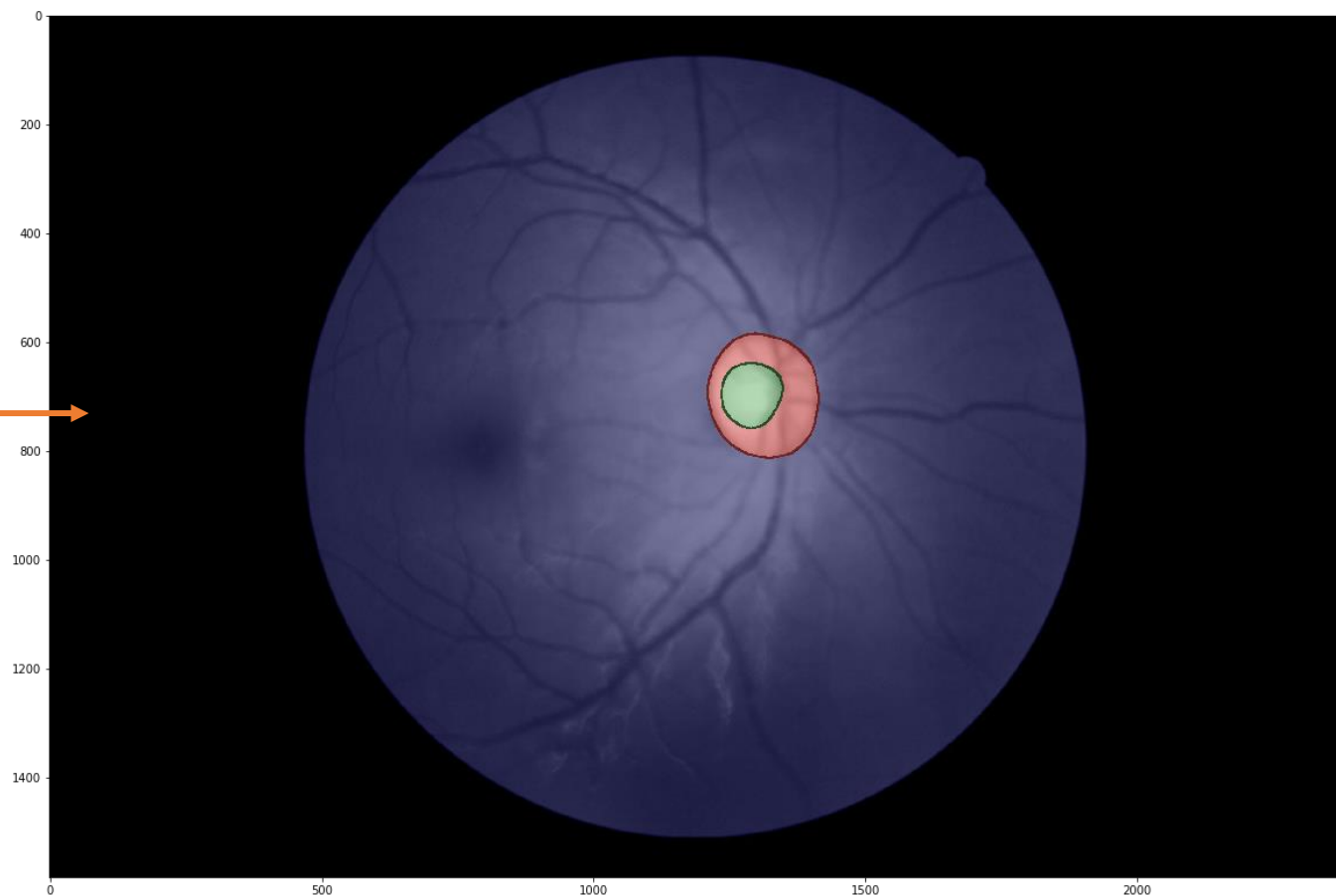


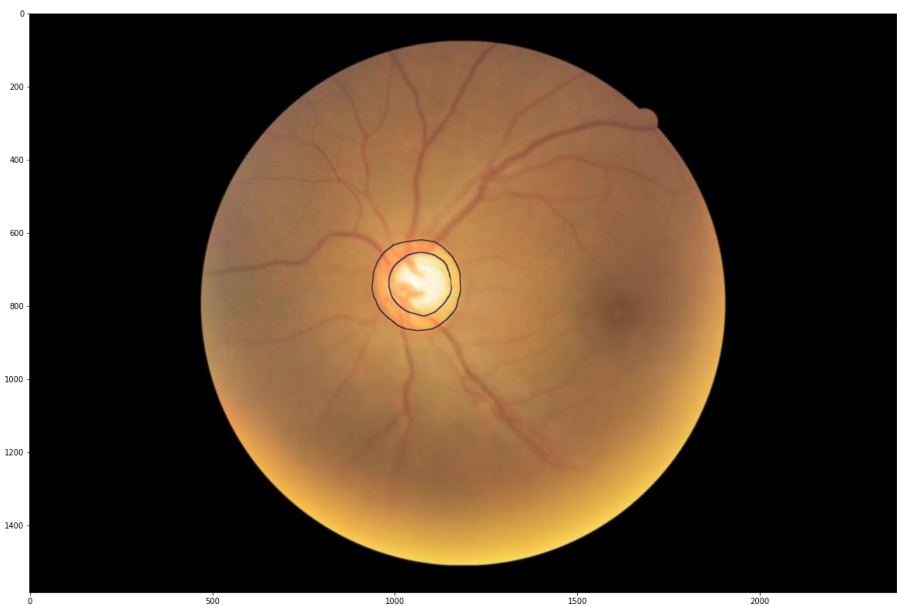
annotated image



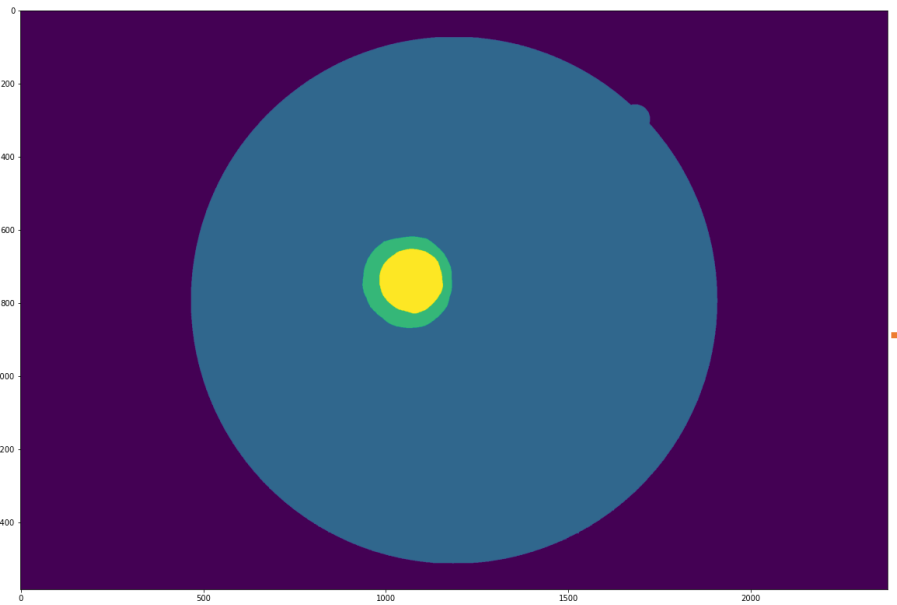
segmented image

overlay



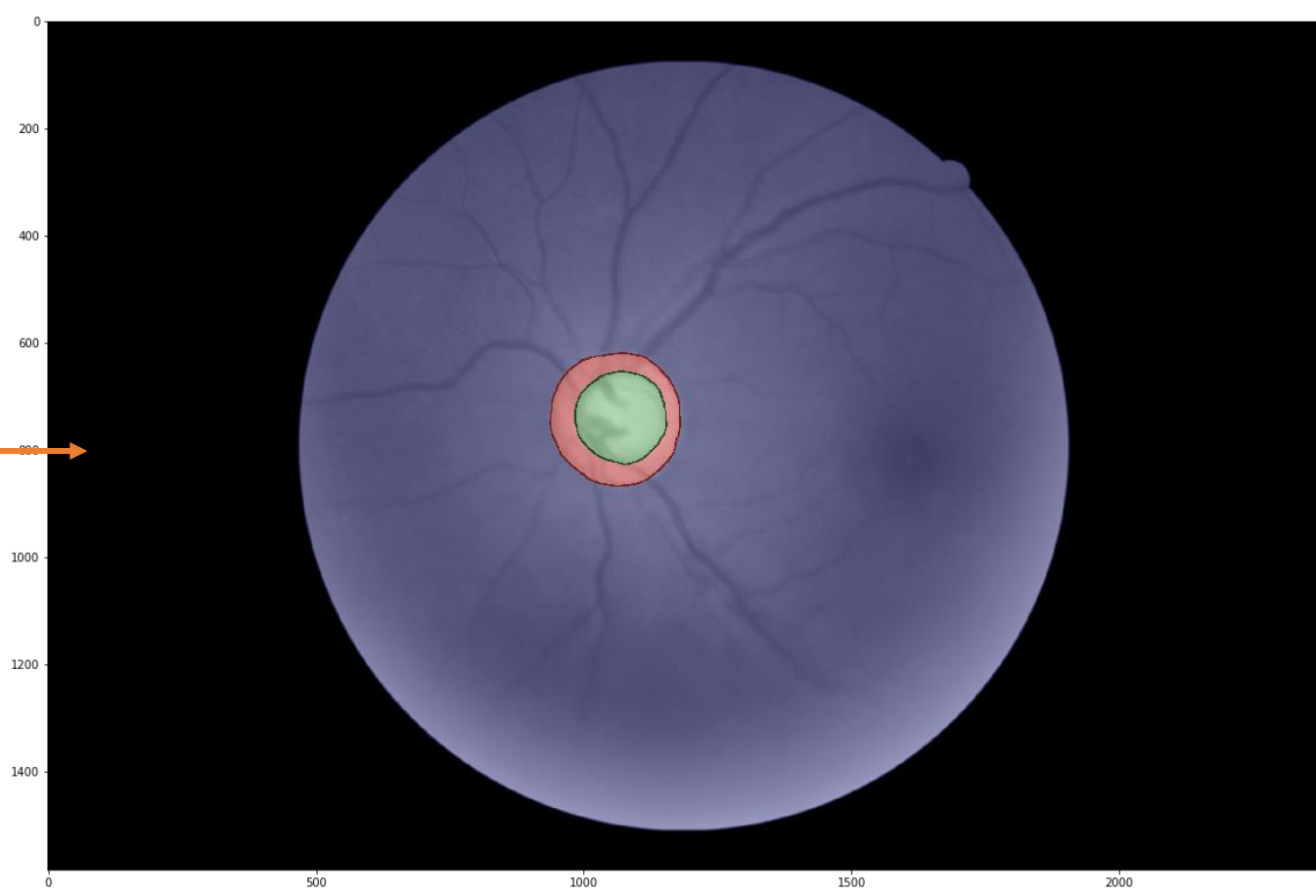


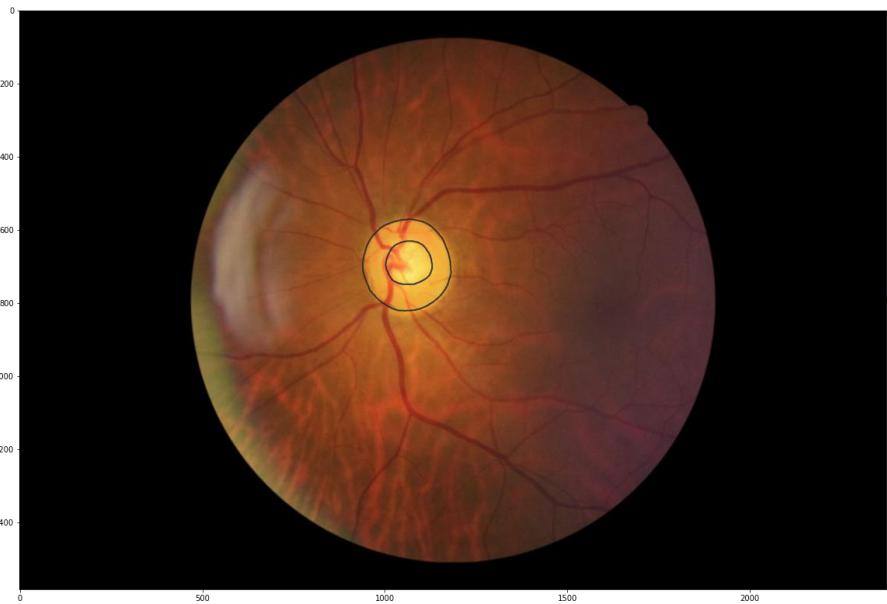
annotated image



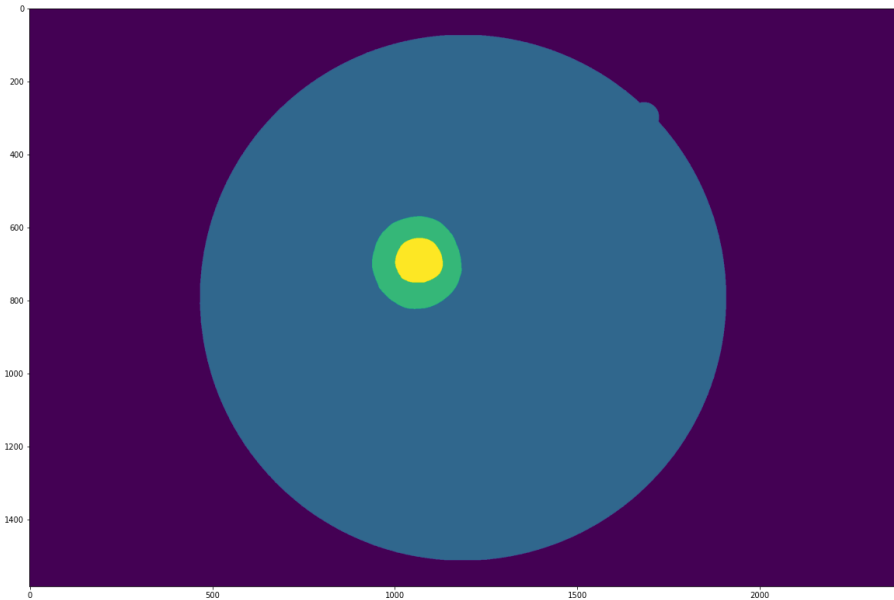
segmented image

overlay





annotated image



segmented image

overlay

