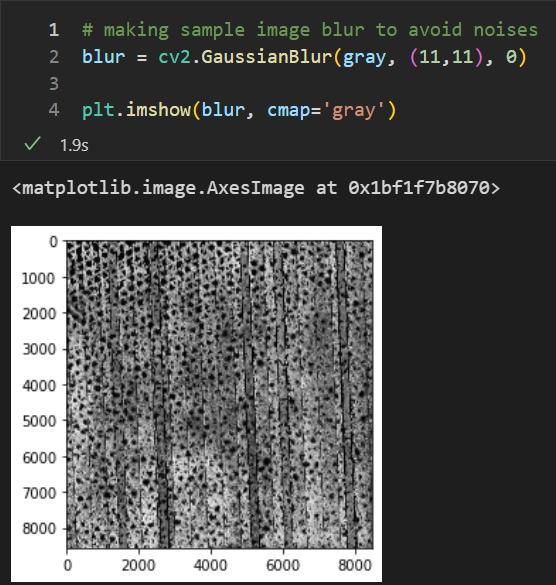
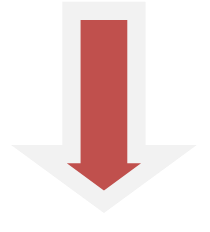
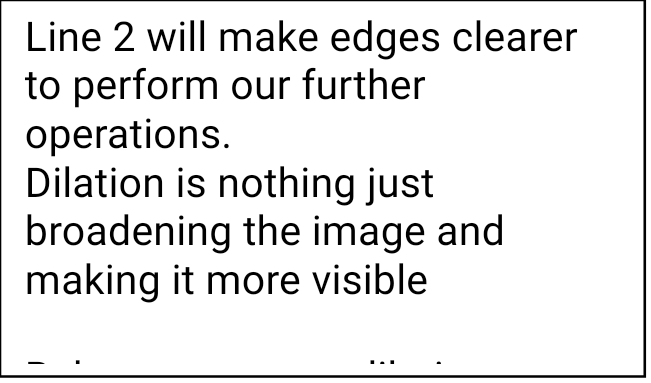
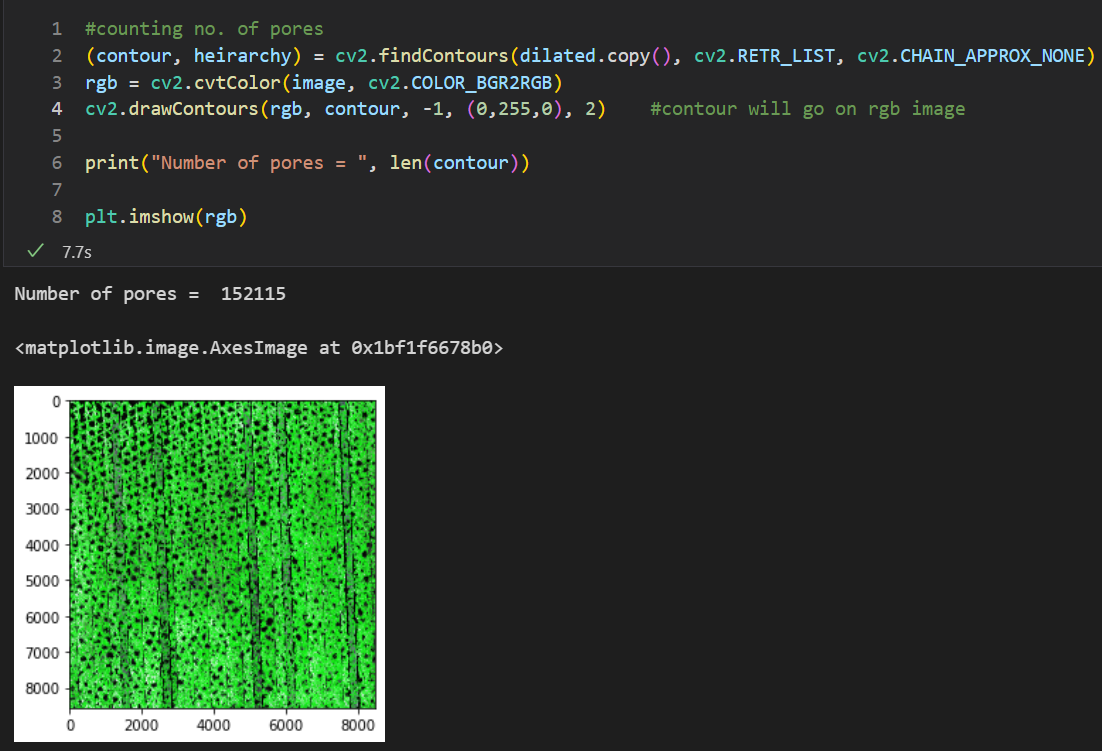
In first 2 lines, I am importing the libraries that is used in making the python script. In order to run the script, user must install the library by writing the command “pip install cv2” in the terminal. Similarly, for matplotlib as well.  
In the following 2 lines, I have read Sample image (jpg format) provided to us and converted the image into grayscale which will make our life easier.  
Line 7 is printing the image by color mapping it to gray



In line 2, we are blurring the image, since edge detection is susceptible to noise in the image.  
We are not interested in the actual detail of the image. Instead, we want to apply edge detection to find the structure and outline of the objects in the image so we can further process them.

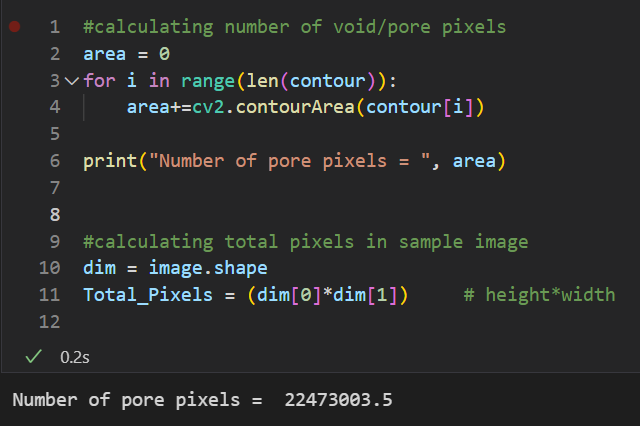
Till now I was setting image from gray scaling to blurring to feed that output to Canny Function  
Here, I am using Canny algorithm/function to detect the edges (sharp change in pixels).  

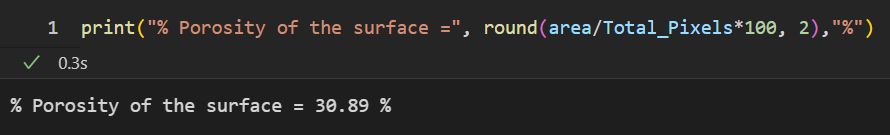



Here, in Line2, we are using findContours function. Wait what is Contour first? Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.  
findContours primarily takes 3 parameters

* Image
* contour retrieval mode (consider what hierarchy you want – inner edge, outer edge count etc.)
* contour compression (manipulates number of vertices or joint point like rectangle contour can be represented by dots and lines both)

Line 3, I have converted image to RGB format as cv2 default format is BGR  
Line4, as function name suggests I have drawn contour on the rgb image and “-1” parameter means all contours are selected, (0,255,0) represents color of the contour (here Green) and 2 shows thickness.  
Line 6, prints no. of voids/pores present in the image.  
Line 8 gives the below image output image shown.



In the first 6 lines, I have calculated number of pore pixels using basic for loop and contourArea function (which takes specified contour as parameter and gives number of pixels used to show that contour). Then, we have summed all contour pixels and stored it in area and printed it.  
  
In Line 10 & 11, I am finding total pixels present in the Sample image using .shape function.  
Or instead you could get it directly through the properties of the image.

Then using simple % porosity formula, final answer is

**% POROSITY (of the Sample Image) = 30.89%**

**Tunable Parameters**

Dilation iterations, retrieval mode of contour detection, kernel size for blurring, threshold levels for canny should be adjusted according to the needs

**References**

**https://pypi.org/project/opencv-python/**