



Draw Week Challenge

The grayscale image file *'assets/fillings.tif'* is a dental X-ray corrupted by noise. Find how many fills this patient has and their sizes in number of pixels.

Goal: Find number of fillings + their sizes (pixel area)

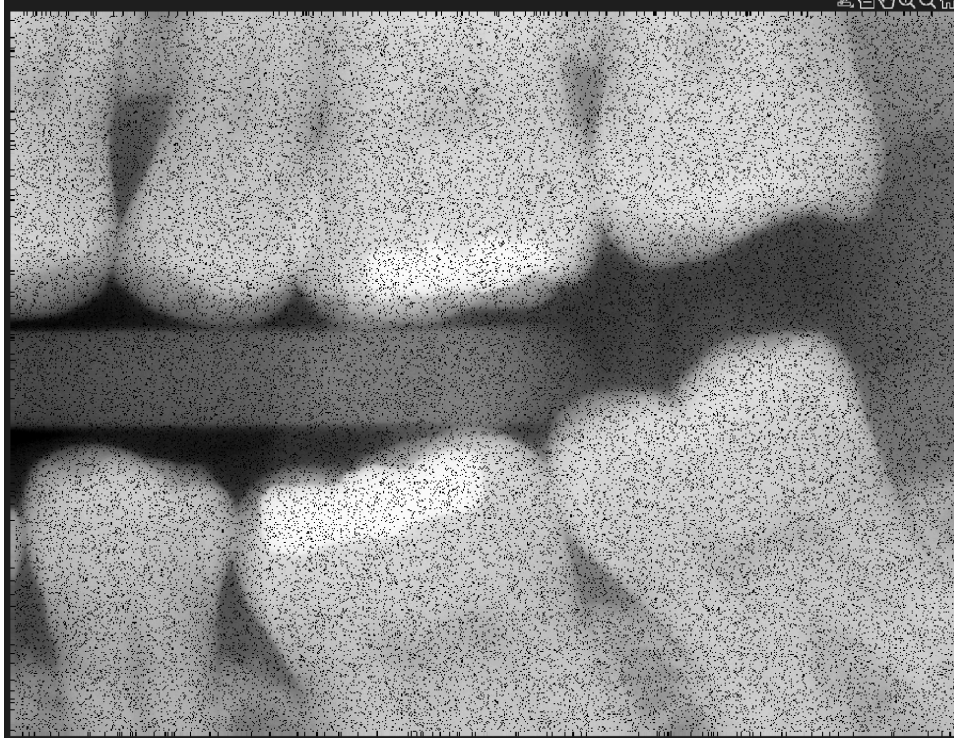
Solution:

Step 1: Read the original image

Code:

```
clear all;  
close all;  
f = imread('assets/fillings.tif'); % read the image  
imshow(f);
```

Output:

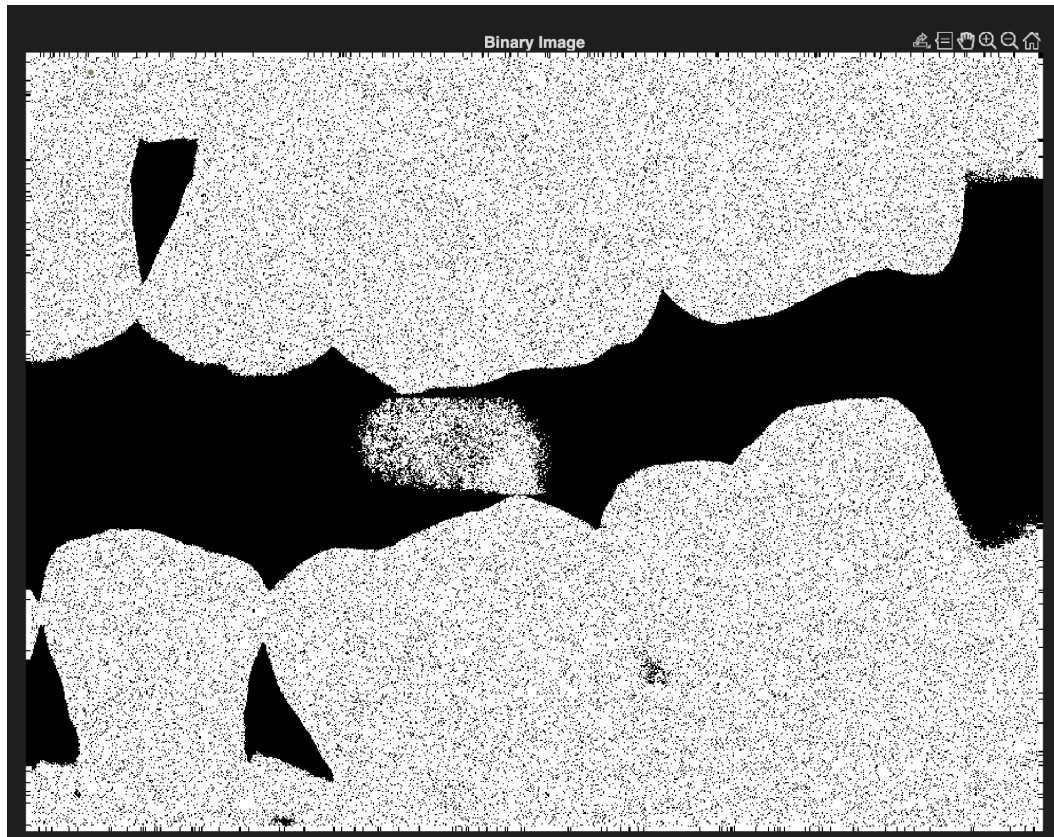


Step 2: Convert the original image to binary

Code:

```
level = graythresh(f);  
BW = imbinarize(f, level);  
imshow(BW)  
title('Binary Image')
```

Output:



Step 3: Remove small noises, count fillings, and measure sizes + Combine all outputs in one montage

Code:

```
clear all; close all; clc

% Step 1
f = imread('assets/fillings.tif');

% Step 2: Threshold
level = graythresh(f);
BW = imbinarize(f, level);

% If fillings appear black, invert
BW = ~BW;

% Step 3: Remove small noise
```

```

BW_clean = bwareaopen(BW, 50);

% Step 4: Connected components
CC = bwconncomp(BW_clean);
stats = regionprops(CC, 'Area');

areas = [stats.Area];
num_fillings = length(areas);

% Display results
fprintf('Number of fillings: %d\n', num_fillings)
disp('Sizes (in pixels):')
disp(areas')

% Show montage
montage({f, BW, BW_clean}, 'Size', [1 3])
title('Original | Thresholded | Cleaned & Used for Counting')

```

Output:



Observation:

First, I converted the X-ray into a binary image so that the bright fillings stand out from the rest of the teeth. Then I removed small noisy spots so that only the main bright regions remain. After cleaning the image, I used connected component analysis to count how many separate white regions there are. Each of these regions represents a filling, and I measured their sizes in pixels. The montage

helped me visually check that the detected regions actually match the fillings in the original image.