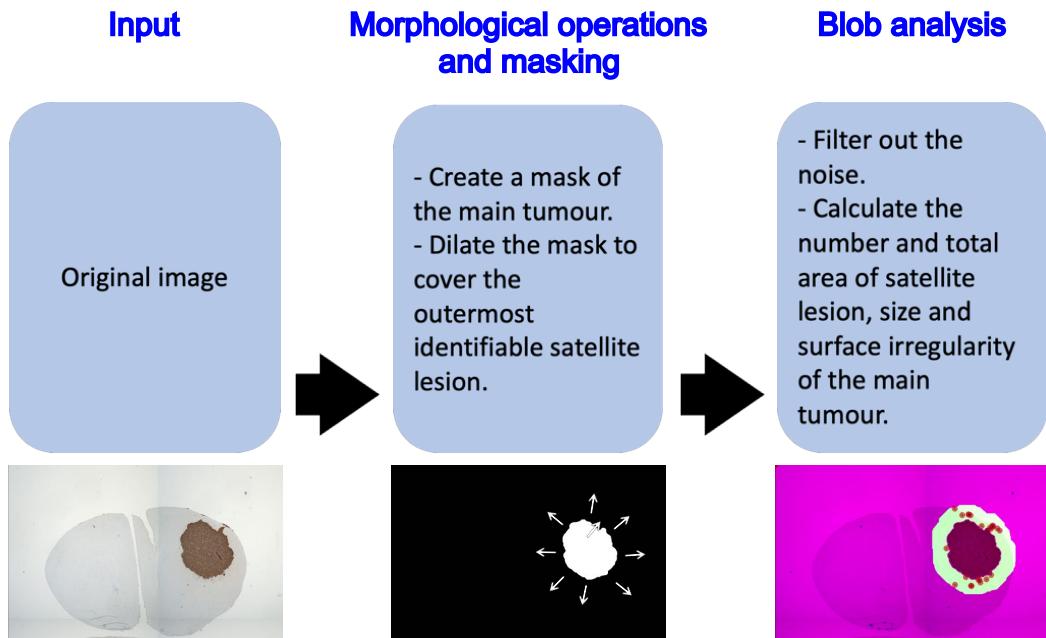


Computer Vision Pipeline



Final output of the tumour analysis algorithm

A satellite area (blue dotted line) was defined by the expended main tumour mask. The computer identified all the spreading tumour lesions (red cross) within the satellite lesion.



References about the algorithm

Main Idea

Isolate/Segment lesions detached to a main tumor, and extract statistics to compare differences between using (DOX + Meth) and (DOX + Saline). Due to the fact that we don't have enough data to train a Supervised Semantic Segmentation Supervised Learning Model, we choose to create the algorithm from computer Vision Base principles:

- Morphological Operations
- Masking
- Blob Analysis

The input of the system will be a image of a Mouse Brain with tumor treated using different methods.

Algorithm Short Description

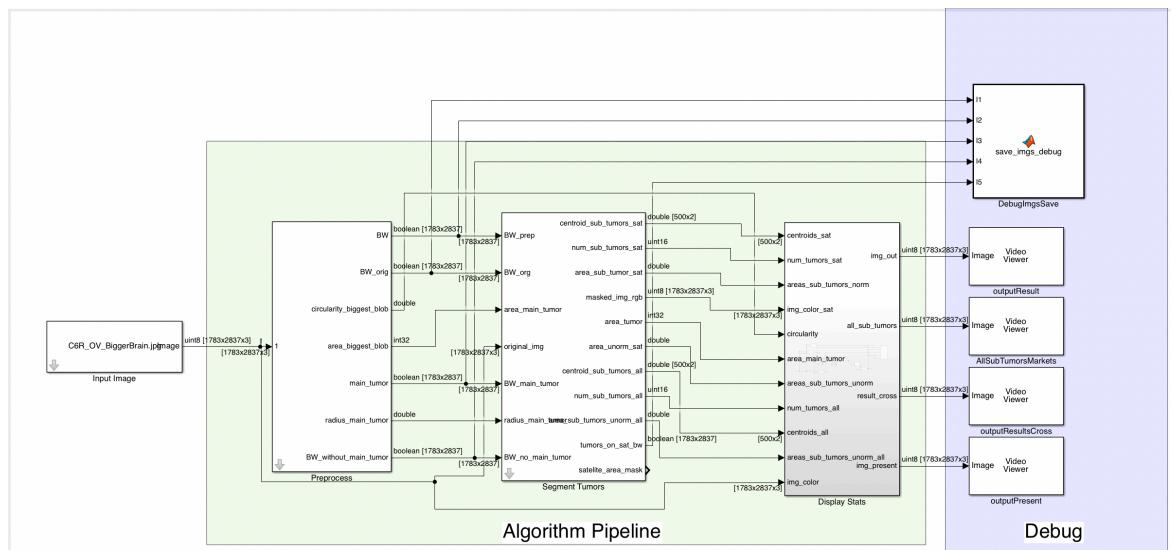
The idea is to select the biggest object of the image, then with this information, do the following

1. Segment out from the image the main-tumor
2. Use a Open morphological operation to separate the "HORNS"
3. Get the main-tumor surface regularity
4. Expand the main tumor borders to calculate the satellite lesion region

After this step the algorithm do the following:

1. Segment all blobs that fall inside the satellite lesion region
2. Filter out blobs with size smaller than 2x2 pixels

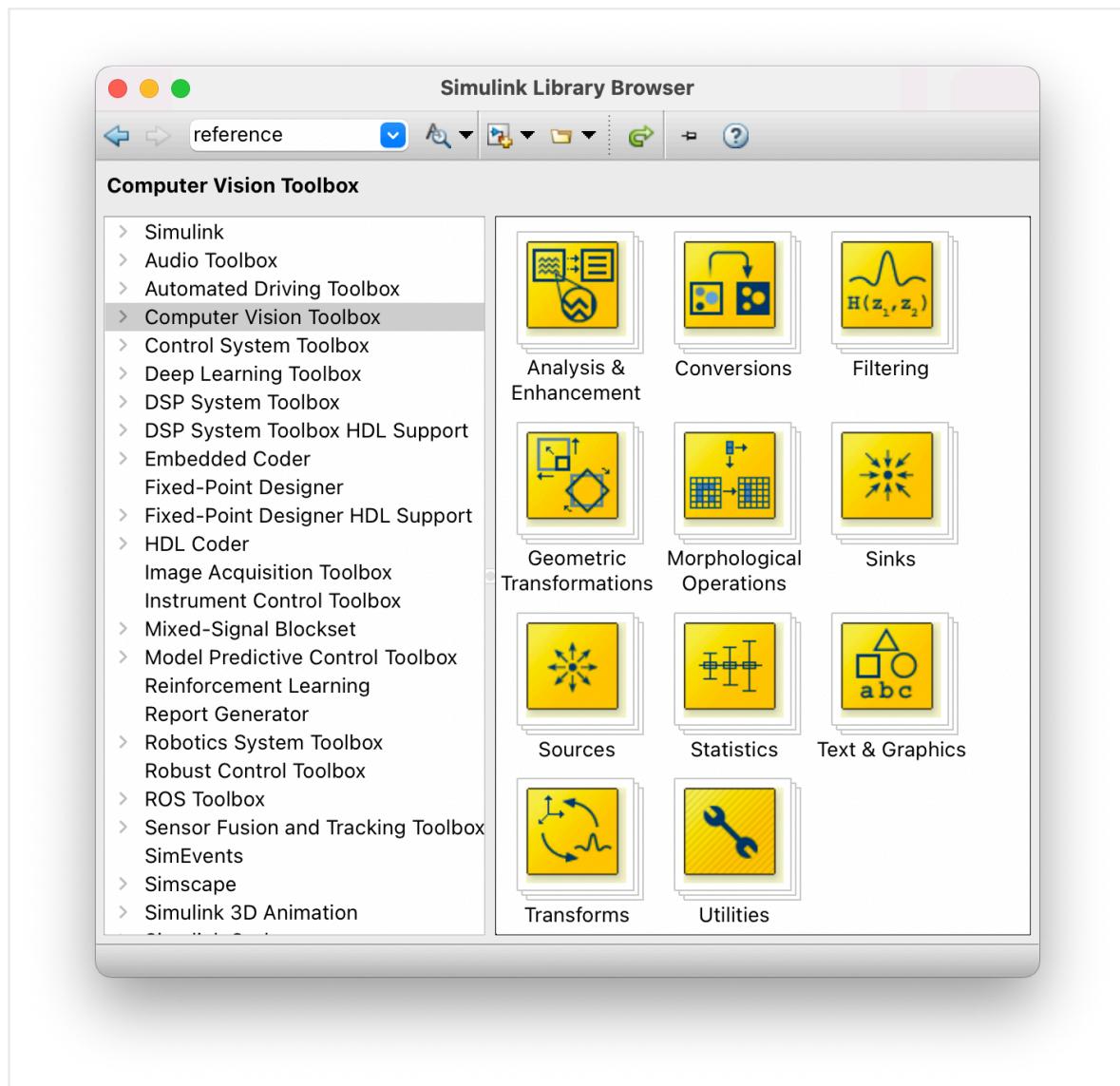
The last step is calculate the stats



Design Methodology

In order to have a fast iteration between algorithm development and data

Analysis, we chose Matlab/Simulink which allows to graphically describe the computer vision pipeline using the Computer Vision Toolbox.

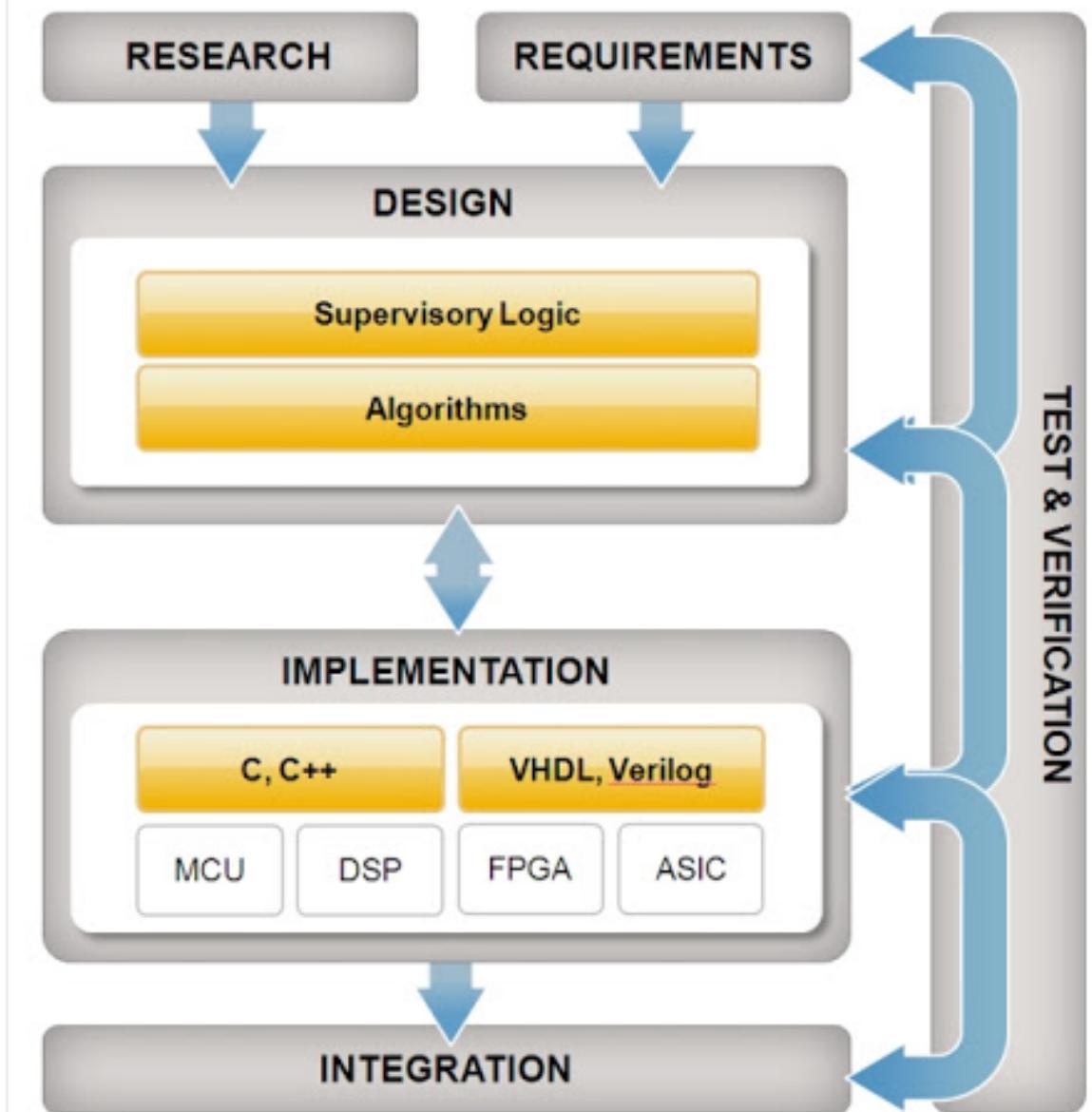


By using these tools we don't need to spend time verifying common operations implementation.

Another advantage if this method is that one could also generate C/C++ code or Hardware.

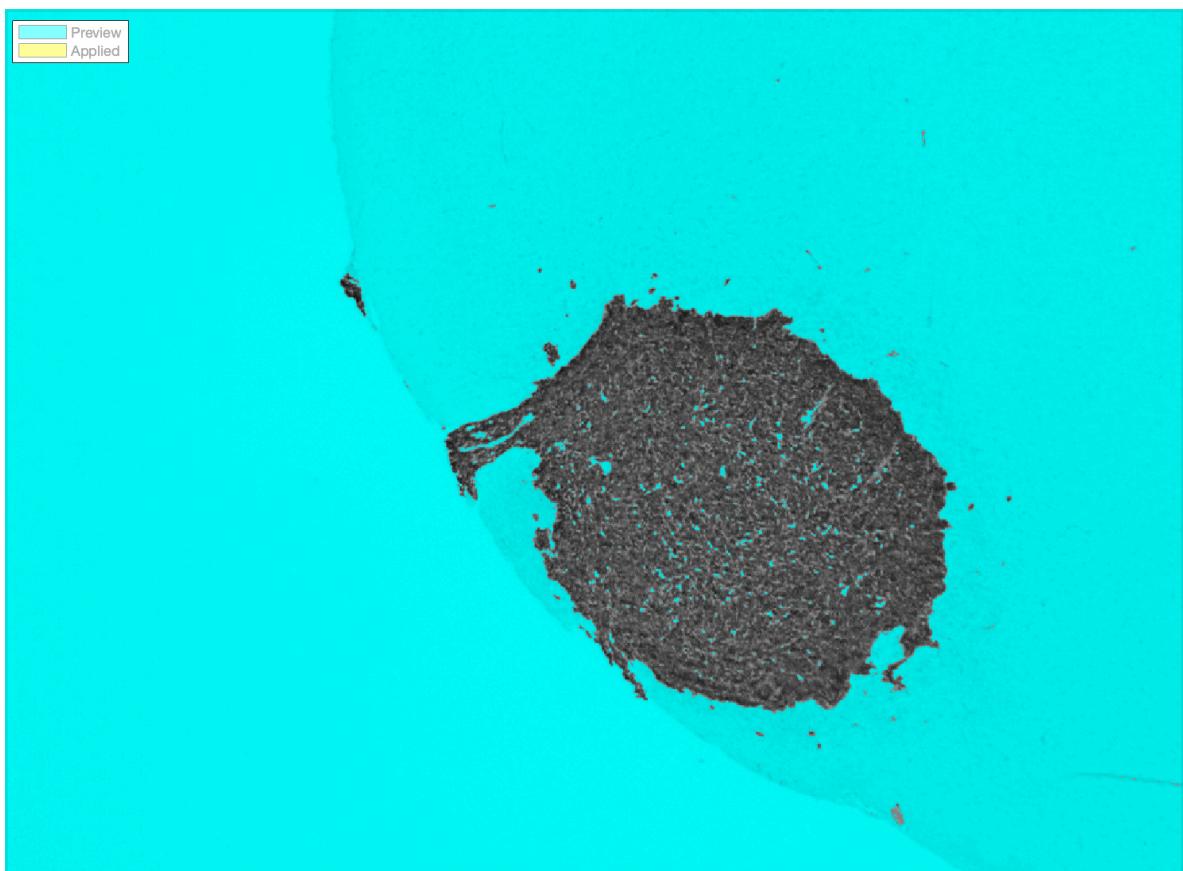
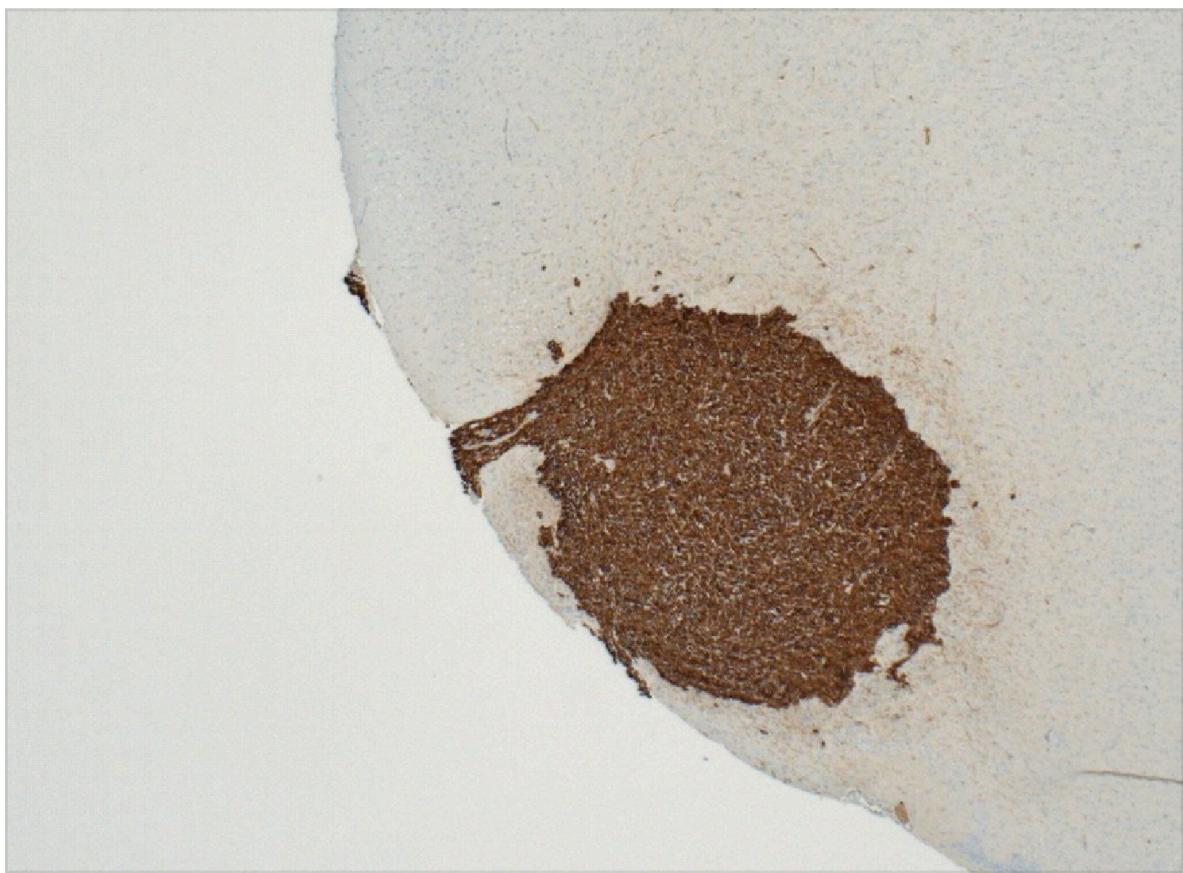
This methodology of analysing data and developing algorithms and generating Source code is called Model Based Design, and is a methodology widely used on Aerospace, Automotive and Pharmaceutical Industry.

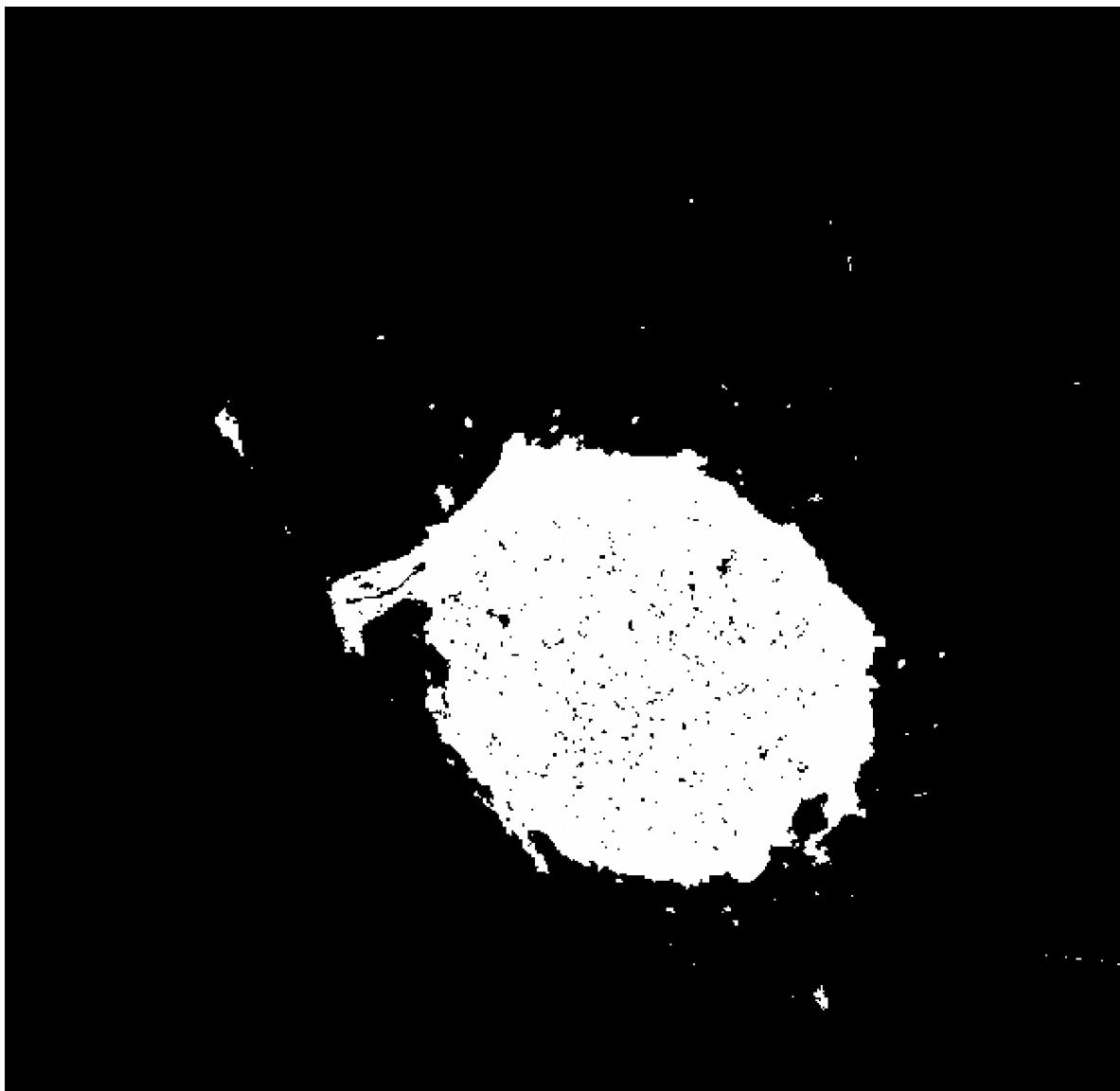
Model-Based Design



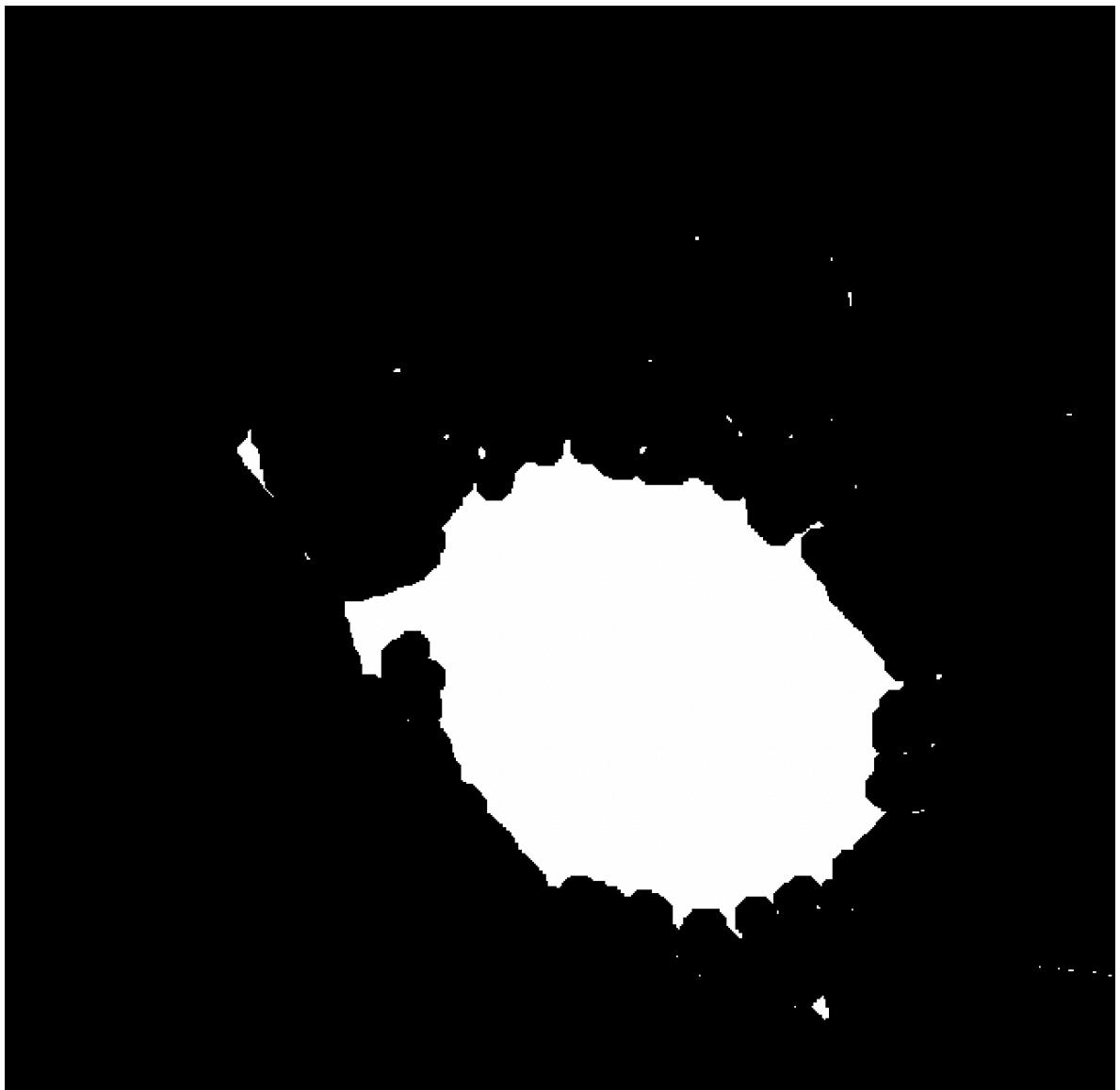
Stages of the algorithm

C3L_Bottom Slide_TR_p00_0_A01f00d4_bottom.jpg

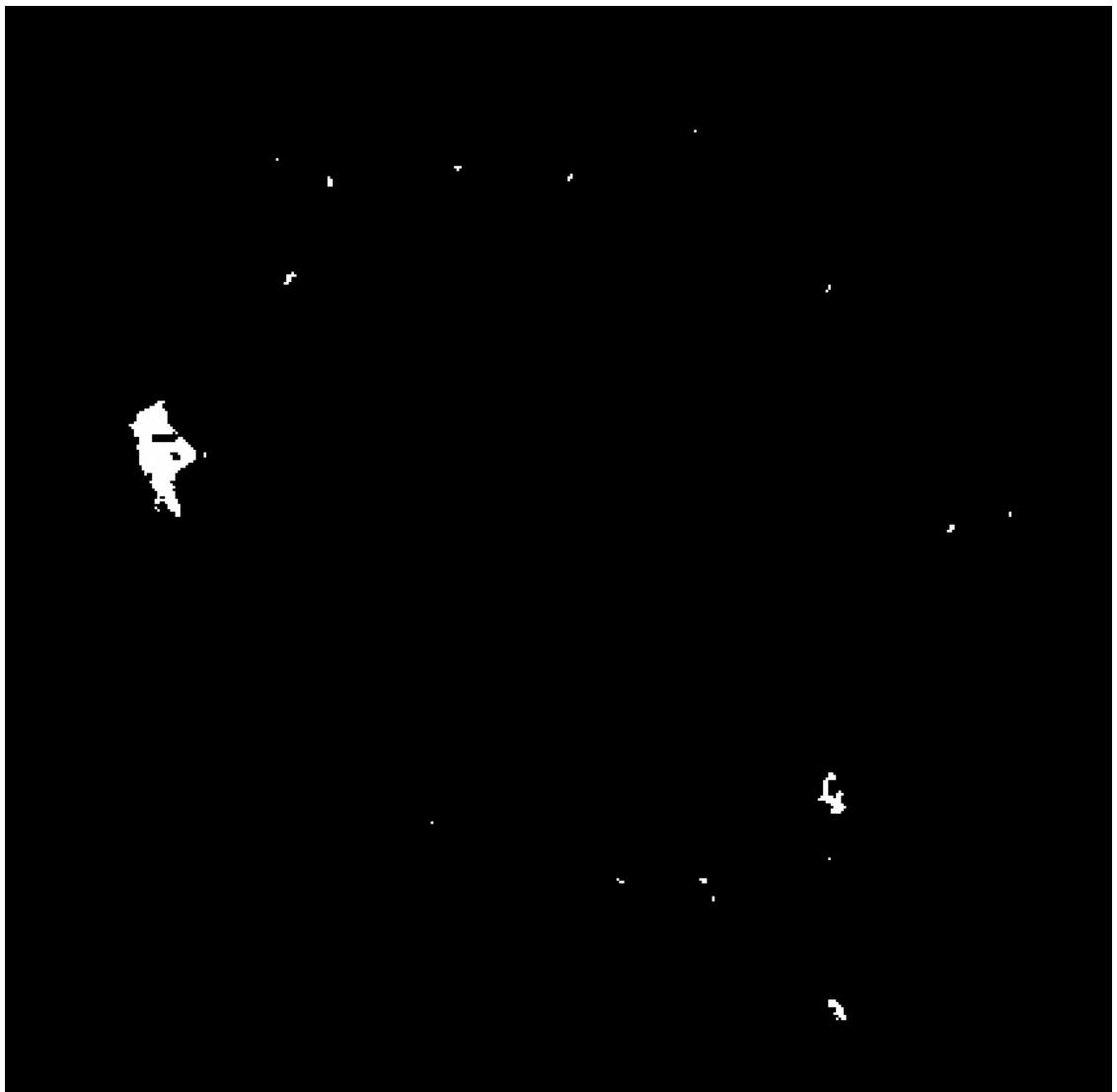




After Close



Optional (Open step to separate the "horns" , which was actually not used)



Then we calculate the satellite area (2x radius of the biggest tumor)

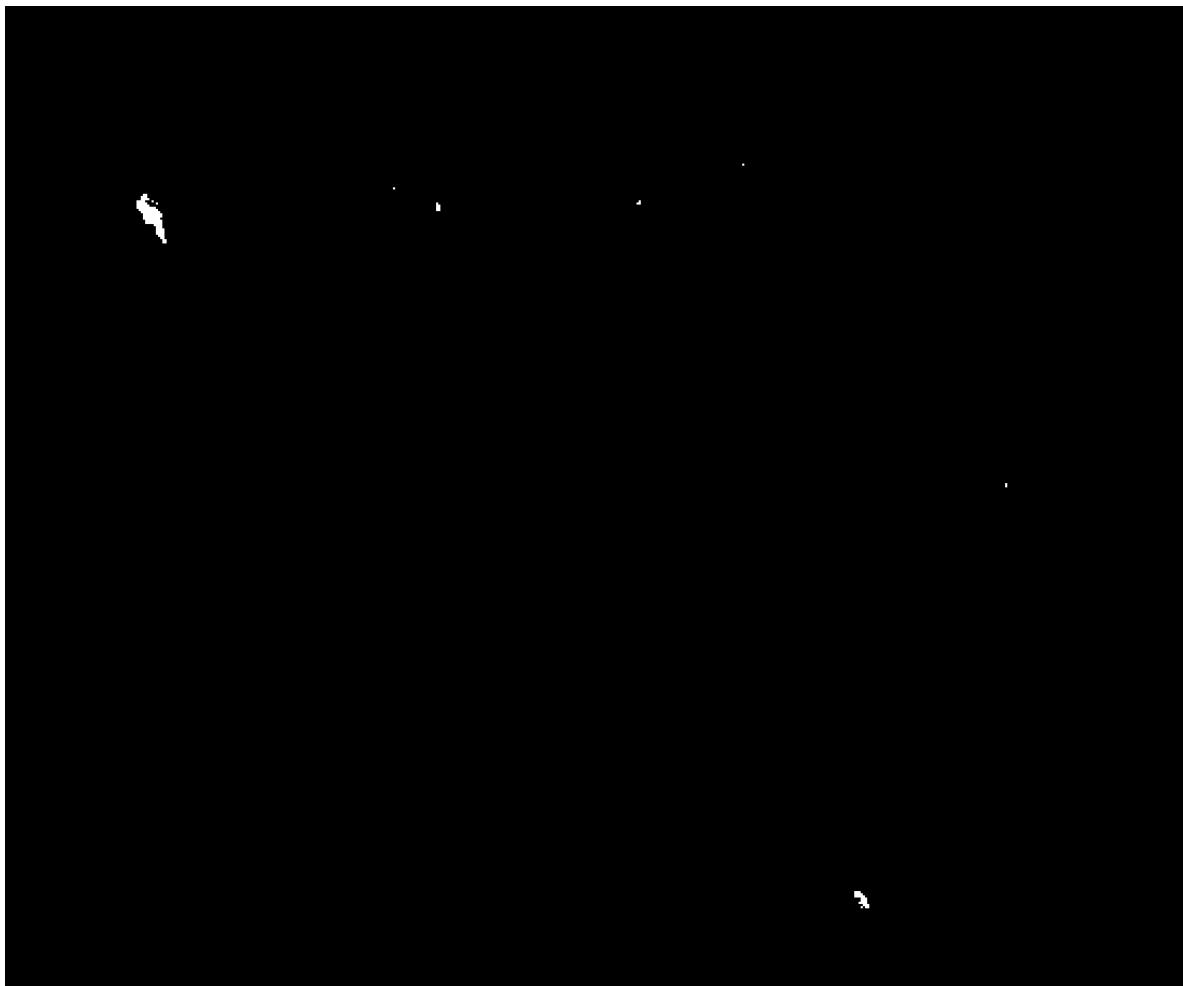


Select all blobs inside the satellite area

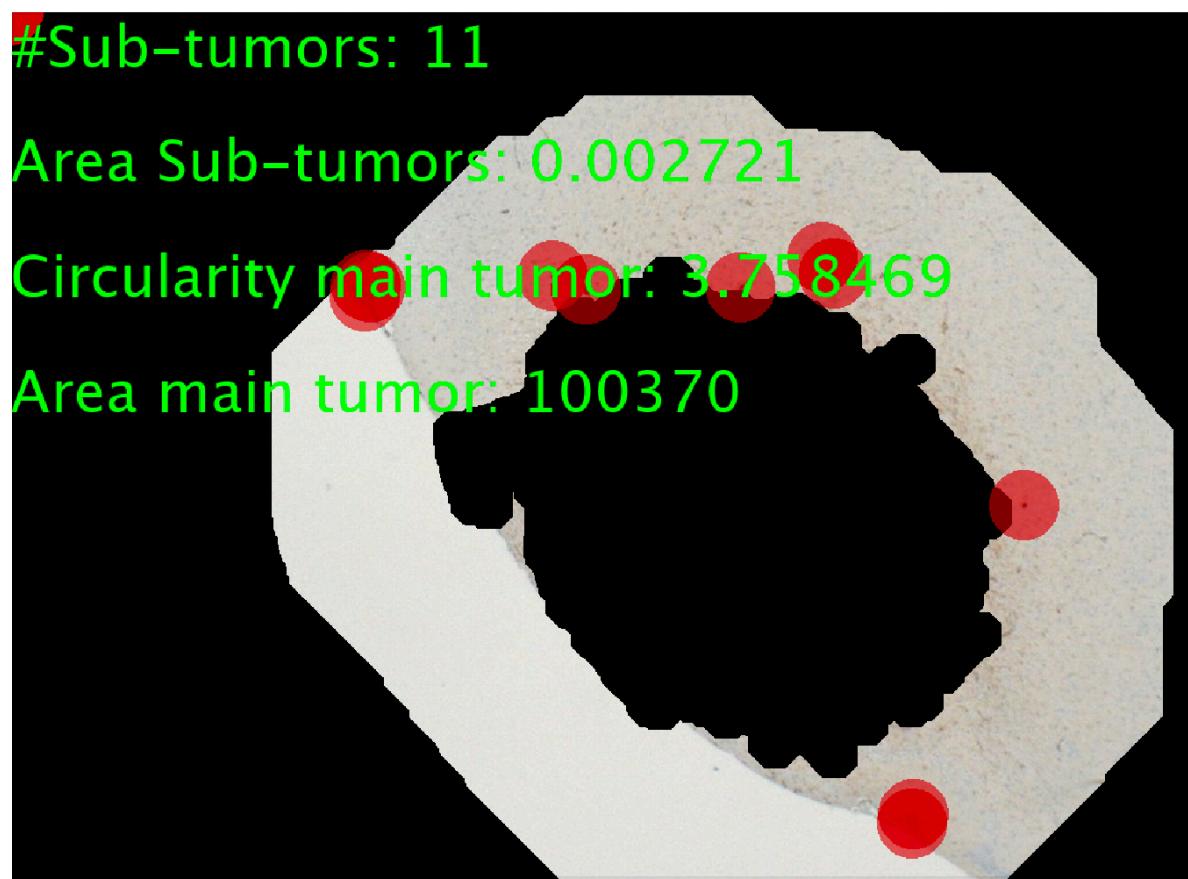


We considered anything smaller than 2x2 pixels an artefact

```
% Erode (Consider anything smaller than 2x2 pixel area artifact)
morph_element_small = strel('disk', 1);
smaller_tumors = logical(imerode(output_masked, morph_element_small));
```

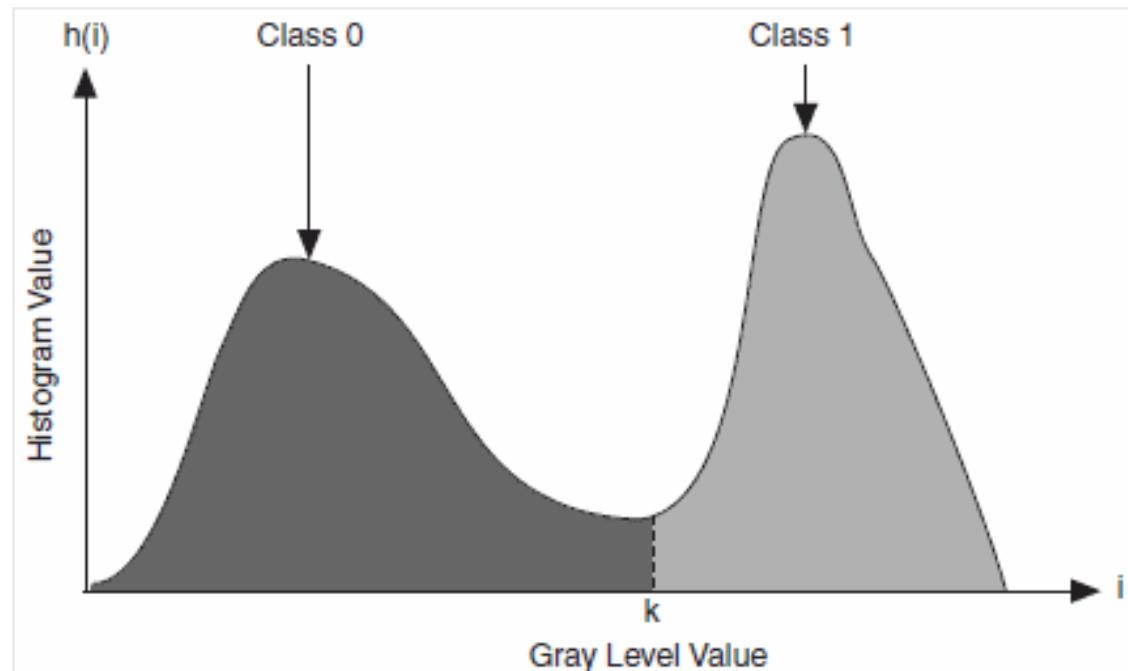


The final step is the calculation of the sub-tumor areas that fall inside the satellite region



Otsu

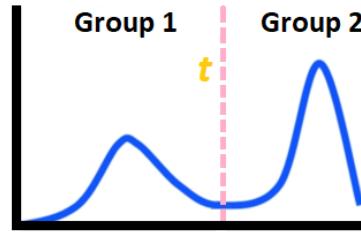
This algorithm is part of the adaptive threshold algorithms
Perfect world



Otsu try do do the perfect world automatically

Automatic Thresholding: Otsu's Method

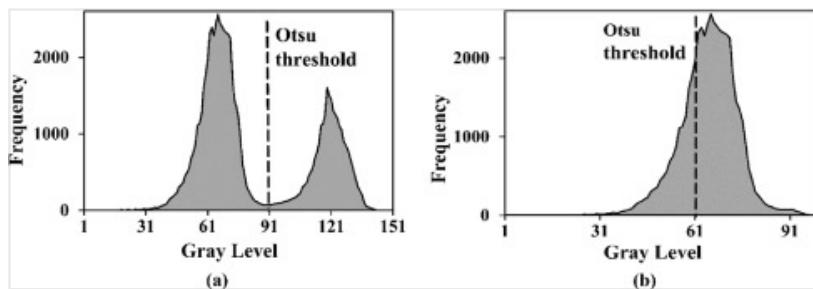
Assumption: The histogram is bimodal



Method: Find the threshold t that minimizes the weighted sum of within-group variances for the two groups that result from separating the gray tones at value t

Where Otsu will fail

If there is no distinct peaks on the histogram Otsu will have difficulties

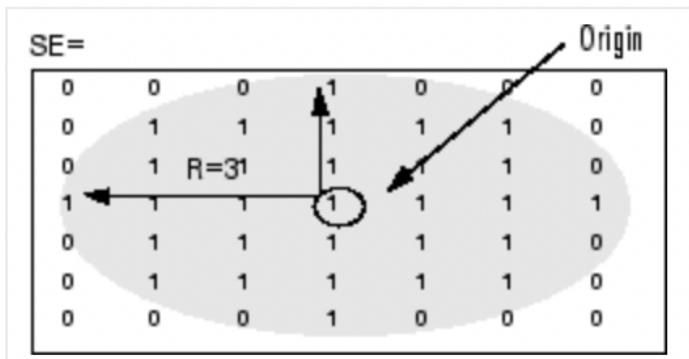


Morphological Operators

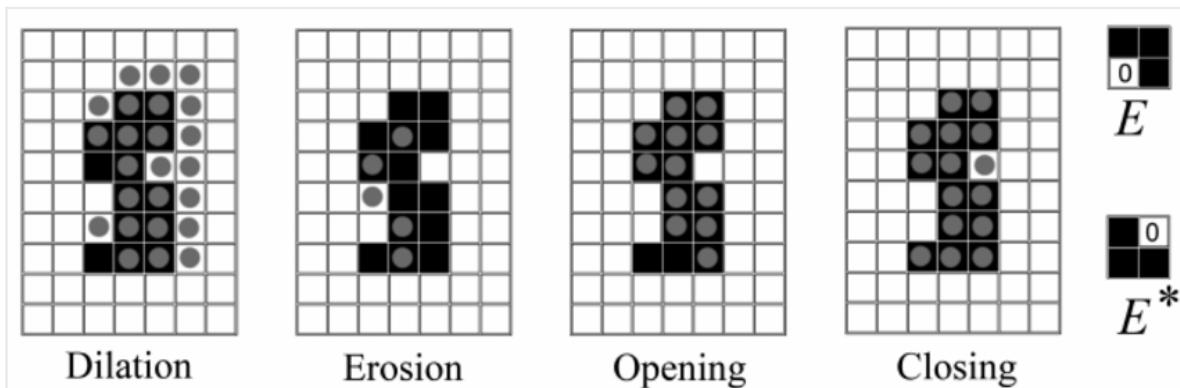
After the image is binarised we rely on the following morphological operations

- Dilate
- Erode
- Open
- Close
- Skinning (Not used)
- Skeletonization (Not used)

The morphological operator will be guided by a morphological structuring element which will be a "shape" that will be applied for every non-white pixel on the original "blob"

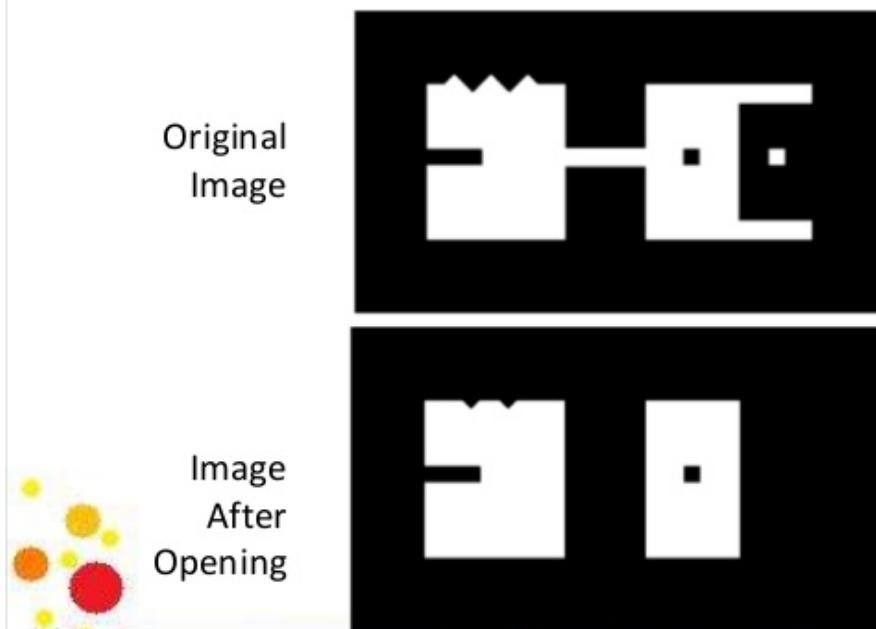


Here are the results of a



We use open and close to separate the main tumor from the "horn" of the tumor

Opening and Closing (Cont...)



Statistics extracted from Biggest Tumor

- Area
- Perimeter
- Centroid (location X,Y on bigger image)

How centroid is calculated, a circularity closer to one

$$\text{circularity} = 4\pi \times \frac{[\text{area}]}{[\text{perimeter}]^2}$$

```
function circularity = get_circularity(area, perimeter)
% References
% https://uk.mathworks.com/matlabcentral/answers/108115-determine-shape-of-a-tumor
[biggest_area, idx_biggest] = max(area);
perimeter_biggest = perimeter(idx_biggest);
% A perfect circle will be 1 (This is normalized)
circularity = (perimeter_biggest^2) / (4*pi*double(biggest_area));
```

Selecting Sub-tumors

After the main tumor is selected we grow an area of same shape (dilate) based on the double of the radius defined on the main tumor, we call this the "Satellite Region"

```
% Close the mask
morph_element = strel('disk', 20);
morph_element_big = strel('disk', double(radius/2));
mask_out_biggest_blob = imclose(uint8(mask_out_biggest_blob), morph_element);
mask_out_biggest_blob = imdilate(uint8(mask_out_biggest_blob), morph_element);
% Select an area around the biggest tumor based on it's radius
mask_big = imdilate(uint8(mask_out_biggest_blob), morph_element_big);
mask_area = mask_big - mask_out_biggest_blob;

% Take out from the original binarized image the biggest blob
%smaller_tumors = (img_orig_bw .* (~mask_out_biggest_blob)) .* ~mask_area;
output_masked = img_orig_bw .* logical(mask_area);
```

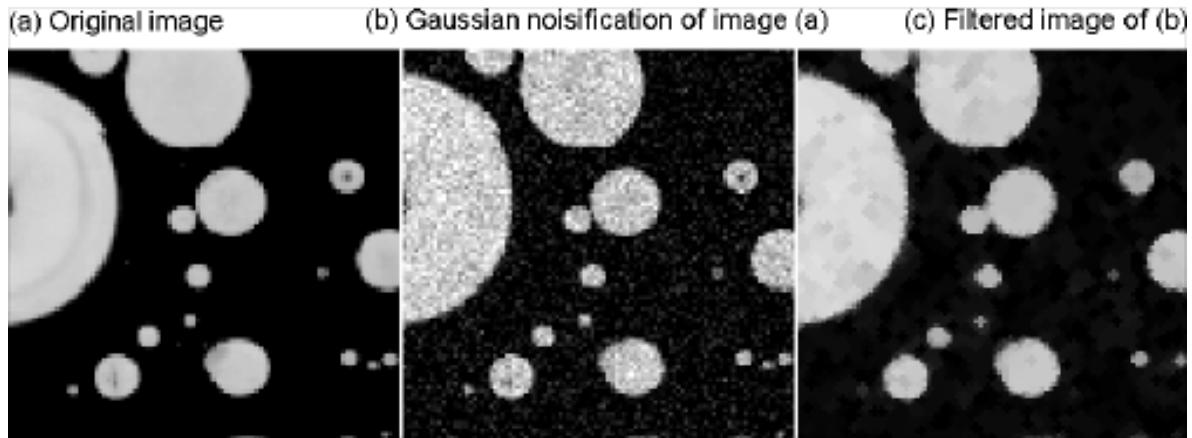
With the "Satellite Region" we crop data from the original image on RGB space

```
% Mask the original image with the mask
masked_out_image = img_rgb;
for channels = 1:3
    masked_out_image(:,:,:,channels) = img_rgb(:,:,:,channels) .* mask_area;
end
```

Future improvements

Normalize all images to have similar size

Use a gaussian filter to make life of Otsu easier to get different peaks



References

- <https://www.sciencedirect.com/science/article/abs/pii/S0167865511000365>
- https://docs.opencv.org/4.5.2/d7/d4d/tutorial_py_thresholding.html
- https://en.wikipedia.org/wiki/Otsu%27s_method
- <https://uk.mathworks.com/help/images/morphological-dilation-and-erosion.html>
- <https://itec tec.com/matlab/matlab-getting-an-error-saying-simulink-cannot-determine-sizes-for-a-matlab-function-block/>
- <https://itec tec.com/matlab/matlab-not-have-enough-information-to-determine-output-sizes/>
- <https://www.mathworks.com/help/images/ref/regionprops.html>
- <https://www.youtube.com/watch?v=3igvjJr20Ws>
-