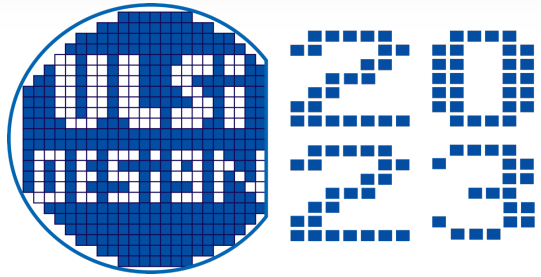
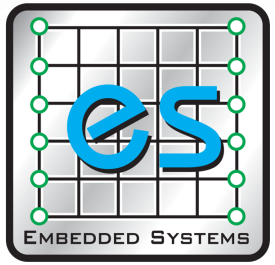




Hardware-software co-design to accelerate Brain Tumour Segmentation

VINAY RAYAPATI

GOPALA KRISHNA REDDY SANAMPUDI

AJAY KUMAR GANDI

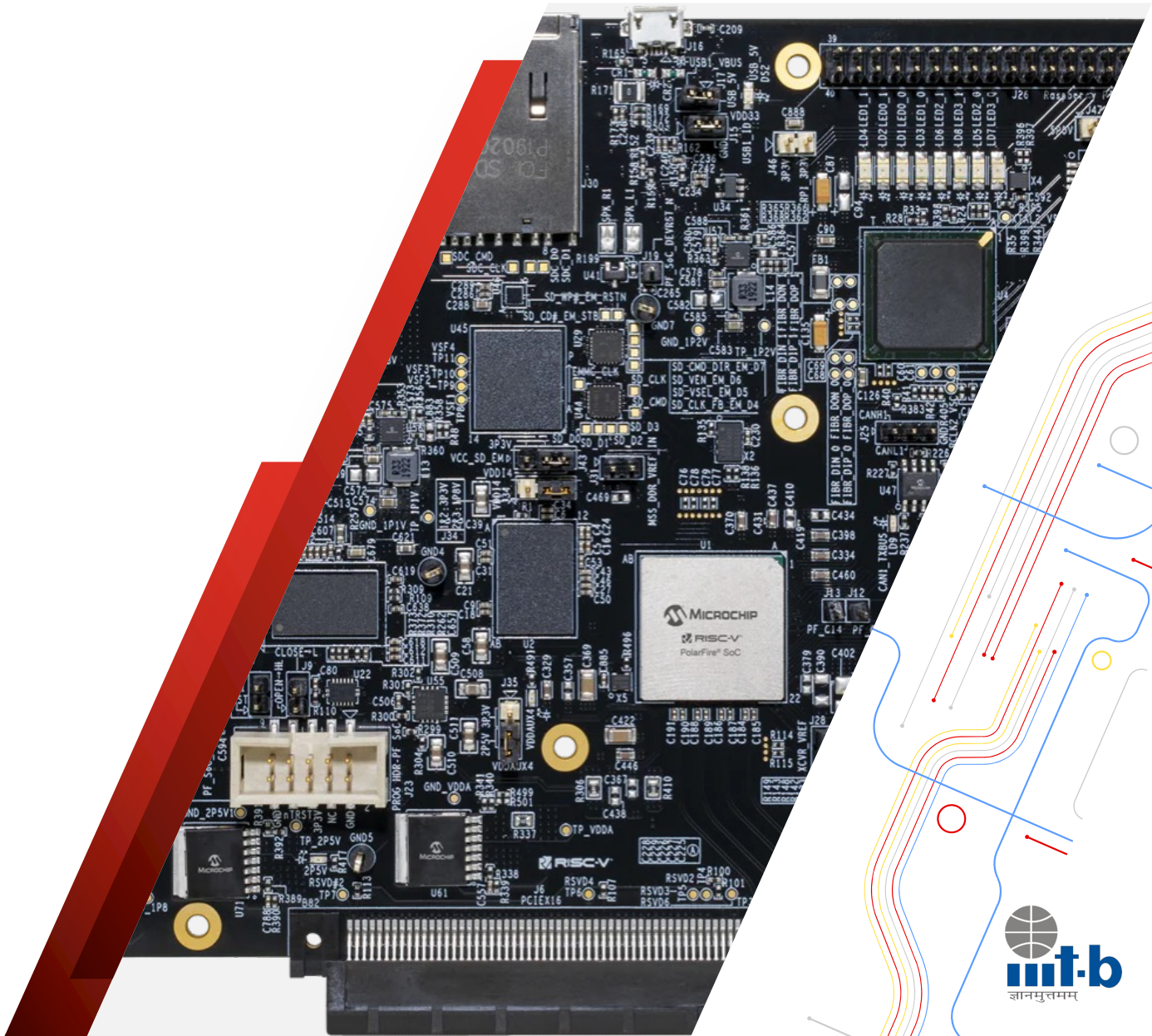
RAVI KIRAN REDDY GOGIREDDY

Prof. NANDITHA RAO

Jan 11, 2023

Agenda

- Objective
- Algorithms
- CPU implementation
- FPGA Implementation
- Results
- Future work
- Acknowledgements
- References



Objective

- To implement a Hardware-Software co-design to **accelerate brain tumour segmentation** using a Polar-Fire Soc Icicle kit MPFS250T_ESFCVG484.
- The image segmentation techniques to be implemented are **Water-shed technique** and **Otsu-thresholding technique.**
- **Hardware Software co-design:** The water-shed technique is to be implemented in software (RISC-V CPU) and the otsu-thresholding technique in the hardware (FPGA fabric)

Algorithms

1. Watershed Segmentation Algorithm

Watershed algorithm is based on extracting sure background and foreground and then using markers will make watershed run and detect the exact boundaries. This algorithm generally helps in detecting touching and overlapping objects in image

2. Otsu Thresholding Algorithm

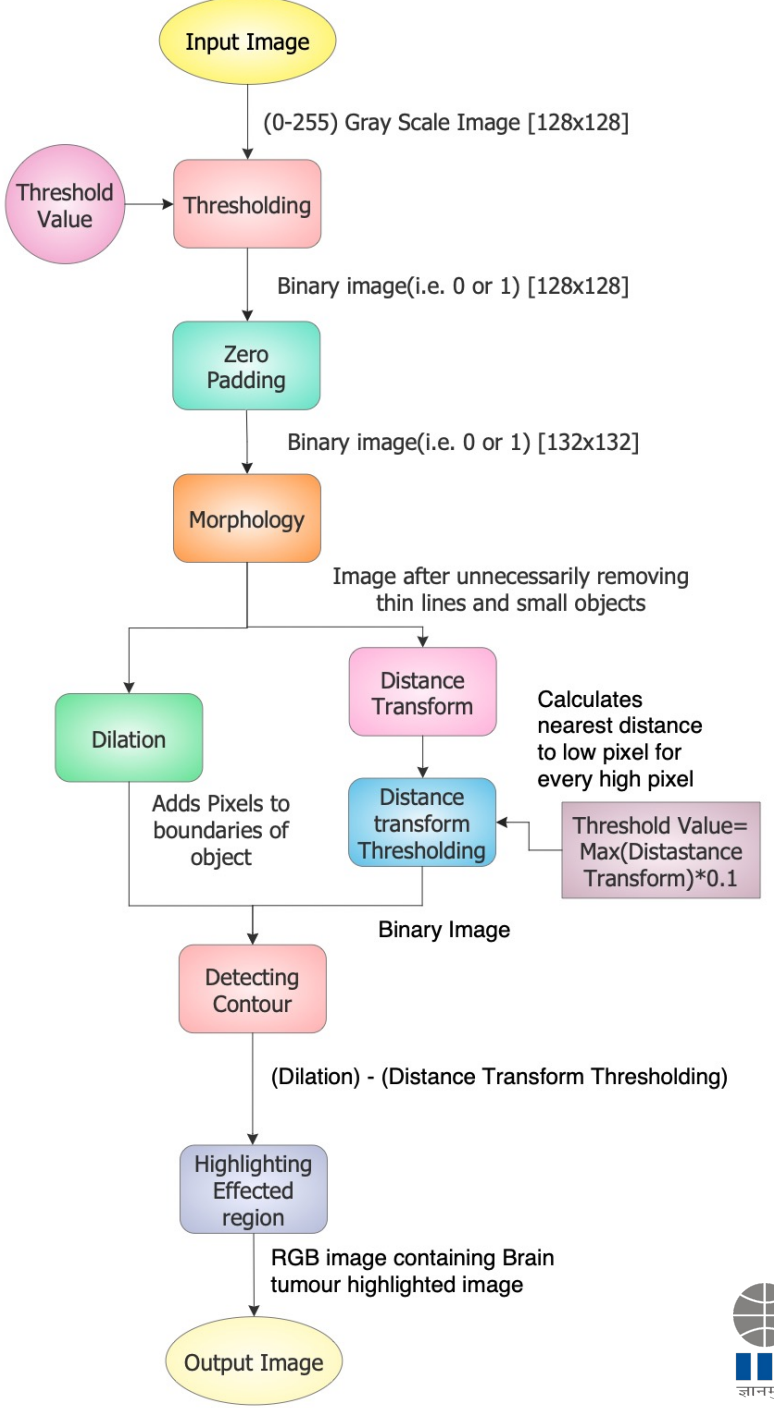
A global adaptive binarization threshold image segmentation algorithm

Software/ CPU implementation

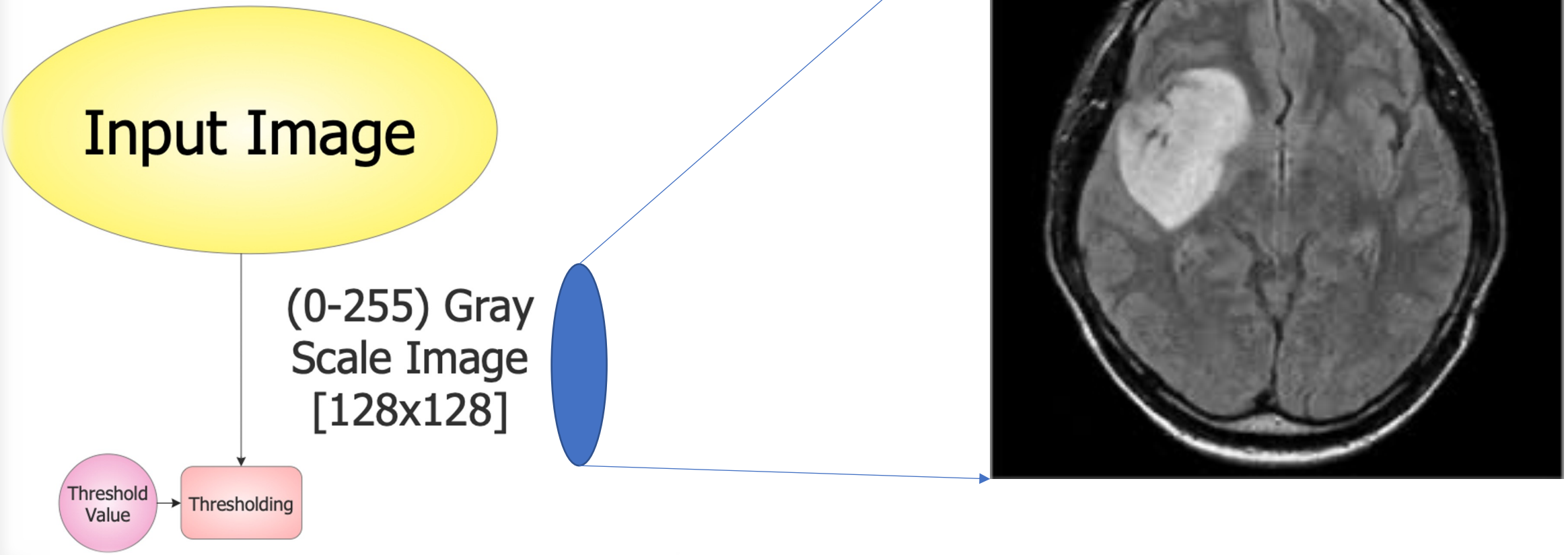
Steps Involved in watershed segmentation:

Watershed segmentation is a region-based segmentation technique that involves the following steps.

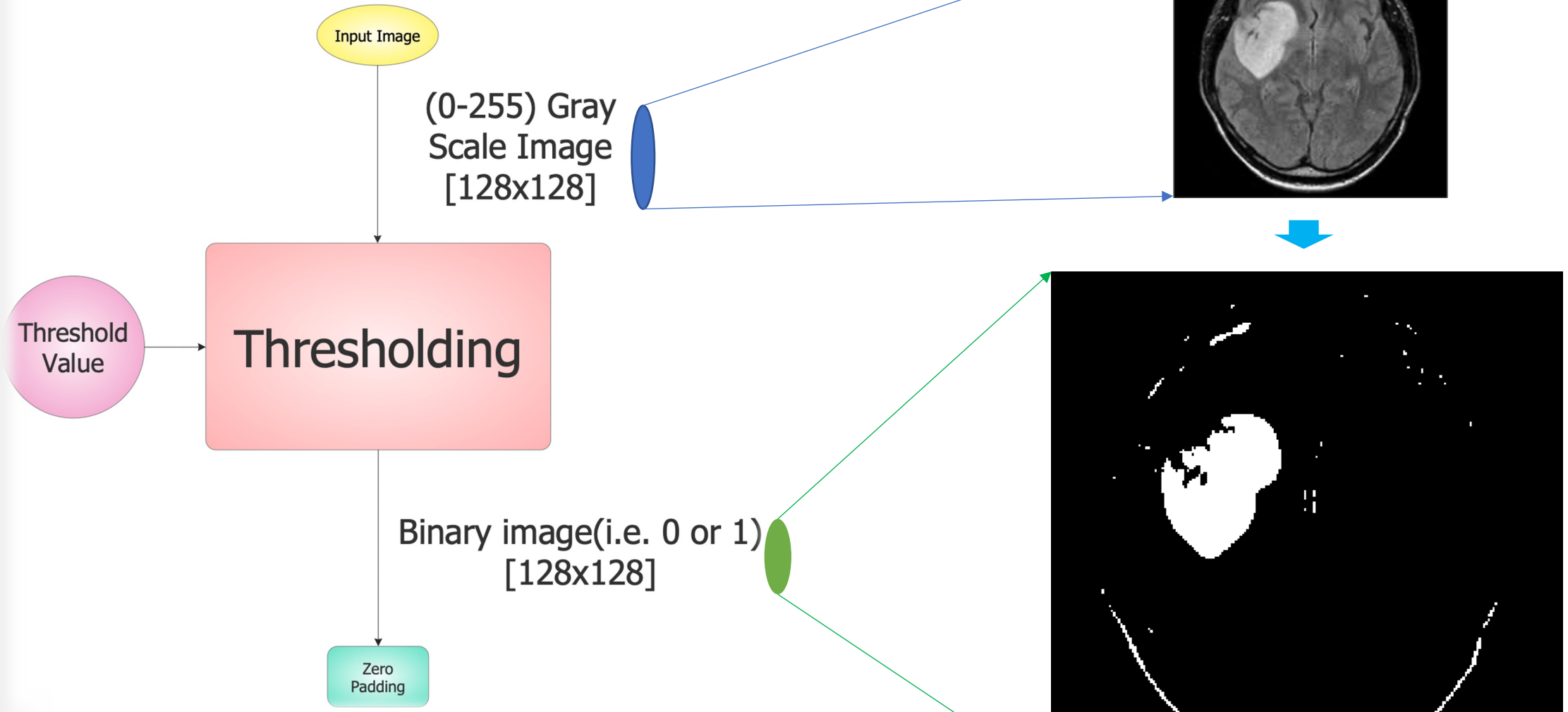
- Threshold Input image
- Morphology
- Dilation
- Distance Transform
- Detecting Contour
- Highlighting Effected region in input image



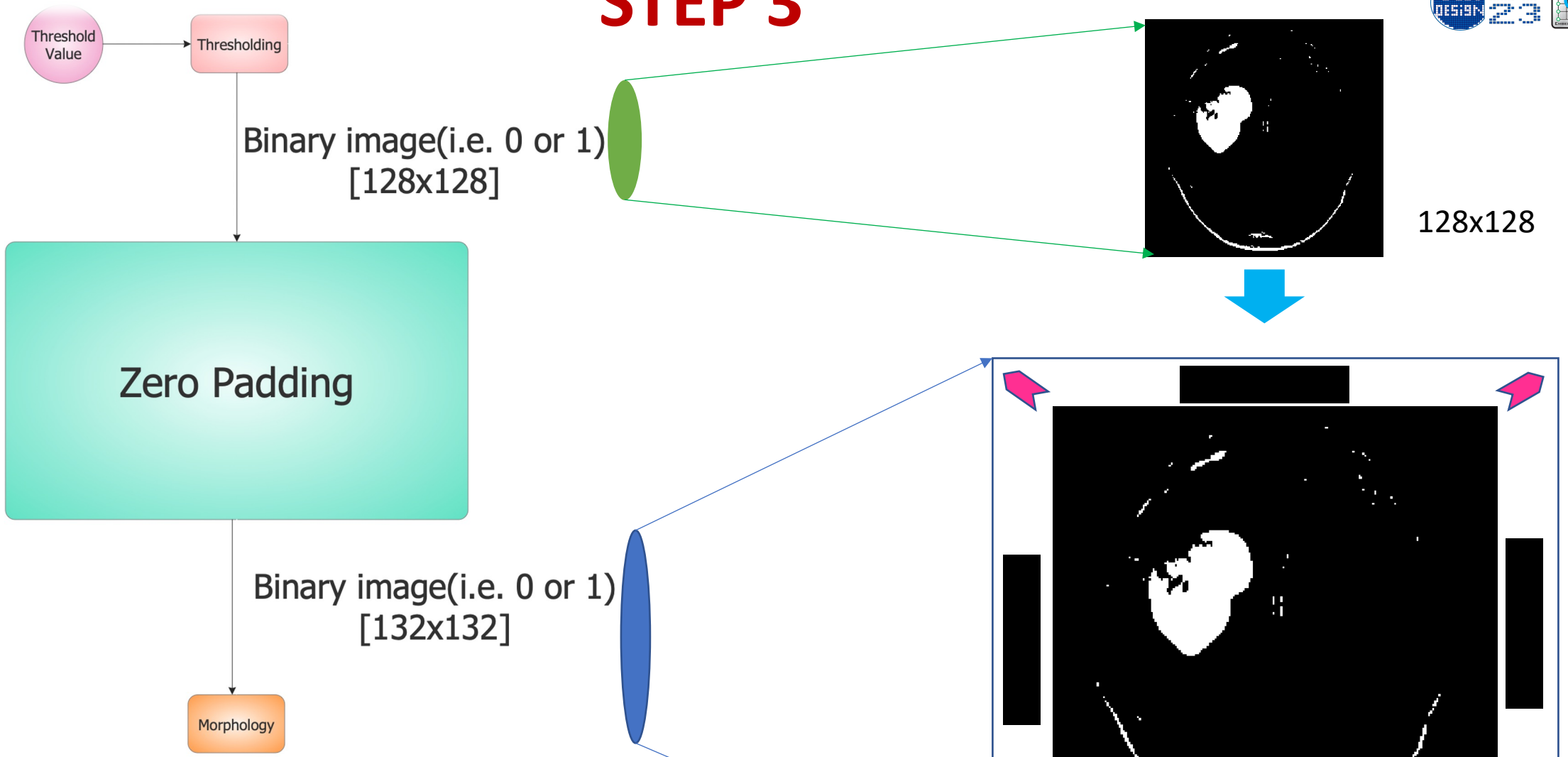
STEP 1


Input Image
(0-255) Gray Scale Image [128x128]
Threshold Value
Thresholding

STEP 2


Input Image
(0-255) Gray Scale Image [128x128]
Threshold Value
Thresholding
Binary image(i.e. 0 or 1) [128x128]
Zero Padding

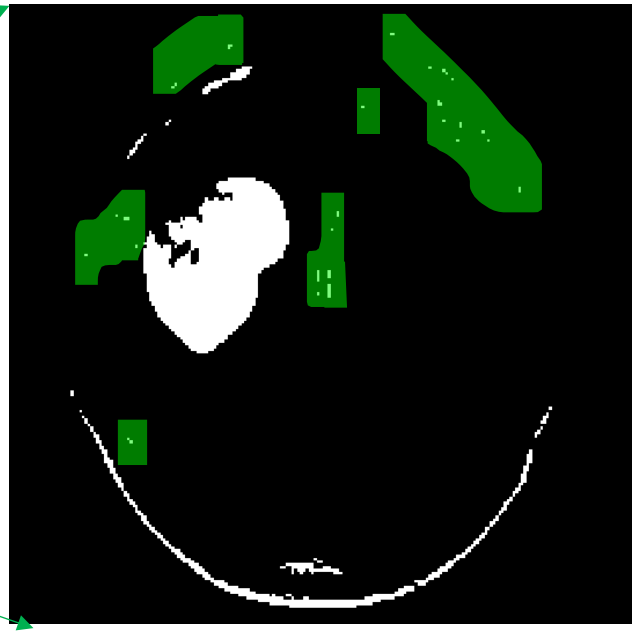
STEP 3


Threshold Value
Thresholding
Binary image(i.e. 0 or 1)
[128x128]
Zero Padding
Binary image(i.e. 0 or 1)
[132x132]
Morphology
128x128
132x132

STEP 4

Zero Padding

Binary image(i.e. 0 or 1) [132x132]



Morphology

Dilation

Distance Transform

Image after removing unnecessarily thin lines and small objects

STEP 5

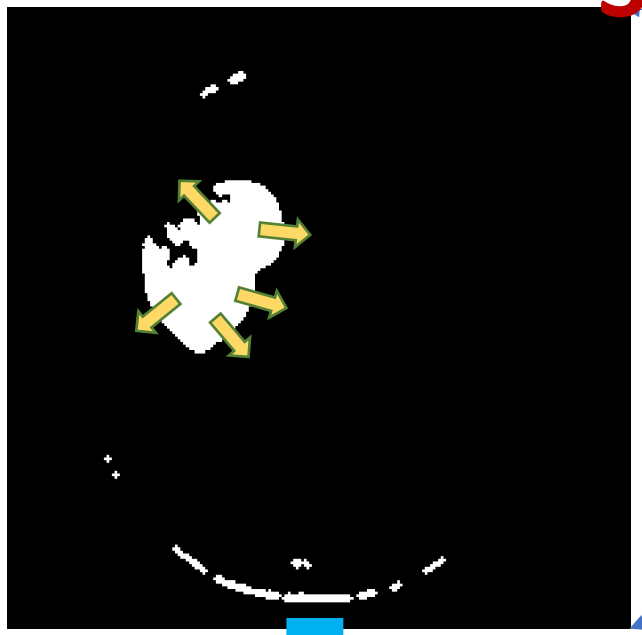


Image after removing unnecessarily thin lines and small objects

Morphology

Dilation

Adds Pixels to boundaries of object

Detecting Contour

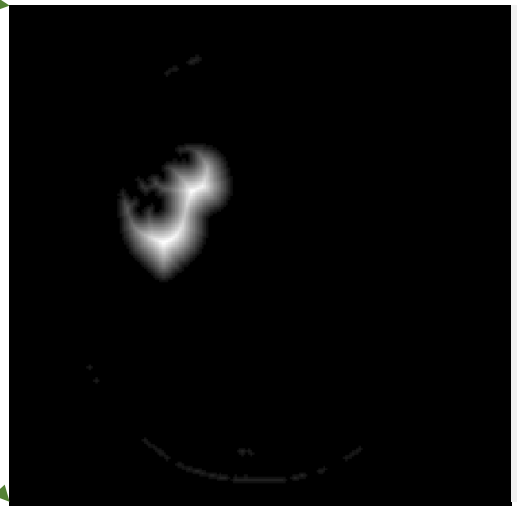
STEP 6

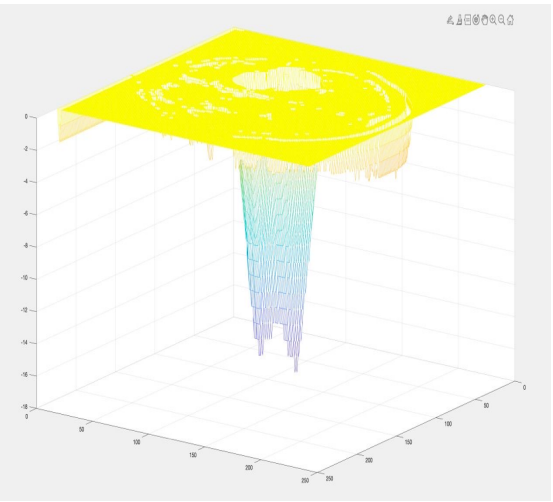
Morphology

Image after unnecessarily removing thin lines and small objects

Distance Transform

Calculates nearest distance to low pixel for every high pixel

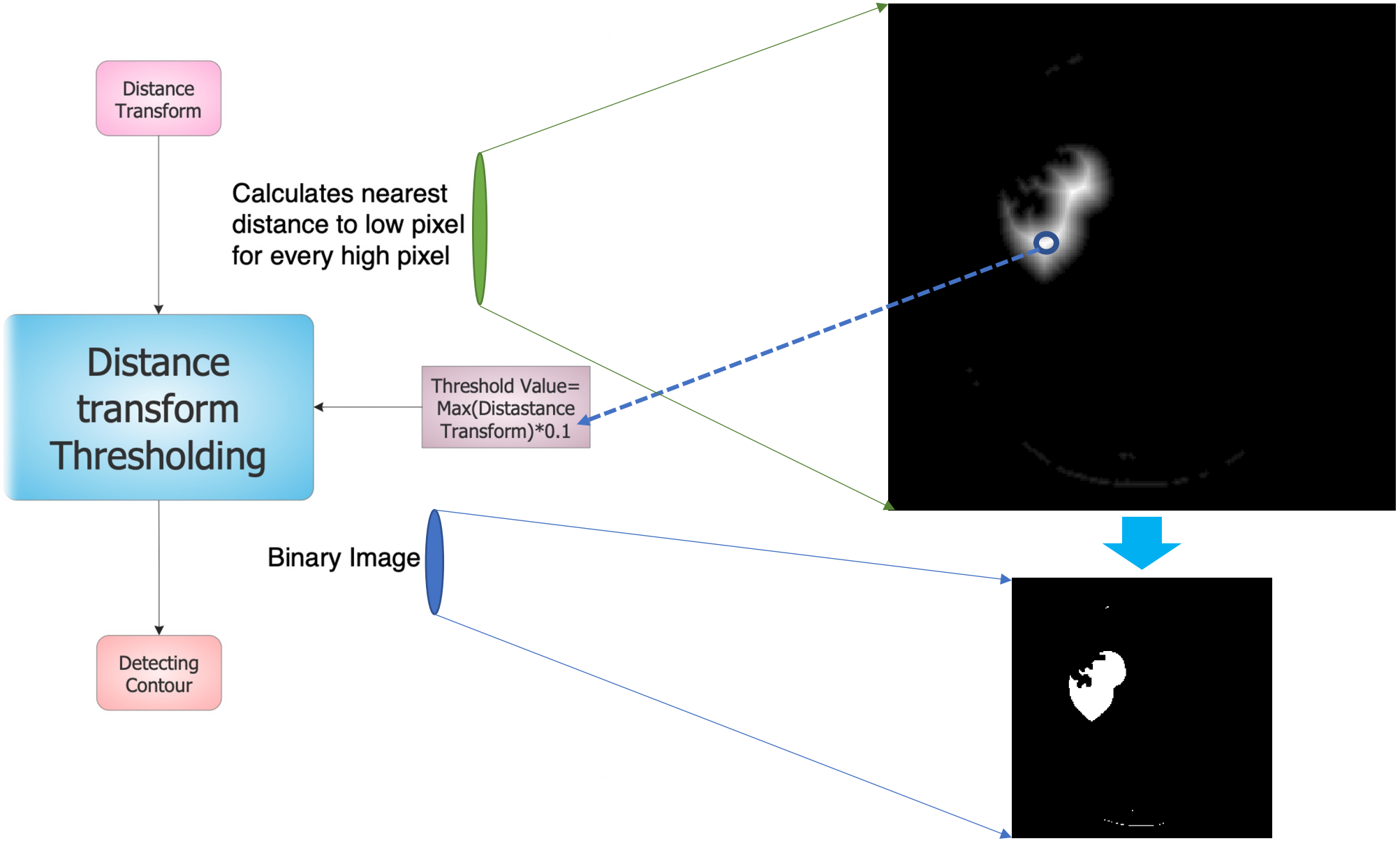




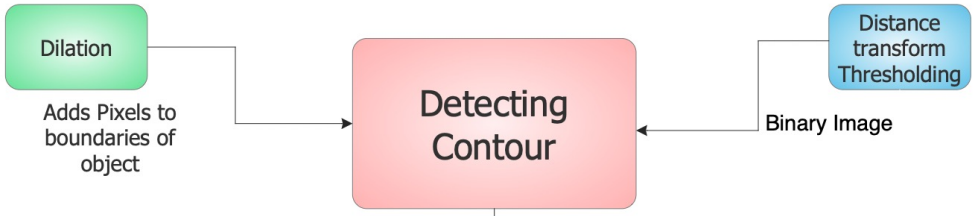
Distance transform Thresholding

Threshold Value= Max(Distastance Transform)*0.1

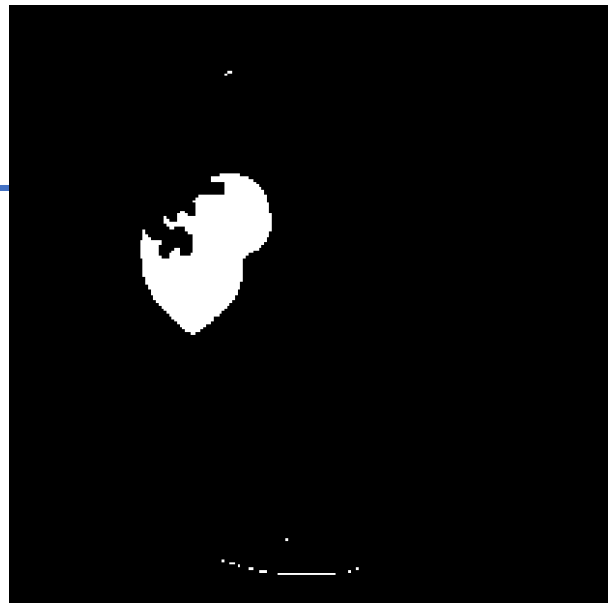
STEP 7


Distance Transform
Calculates nearest distance to low pixel for every high pixel
Distance transform Thresholding
Threshold Value= Max(Distastance Transform)*0.1
Binary Image
Detecting Contour

STEP 8



(Dilation) - (Distance Transform Thresholding)

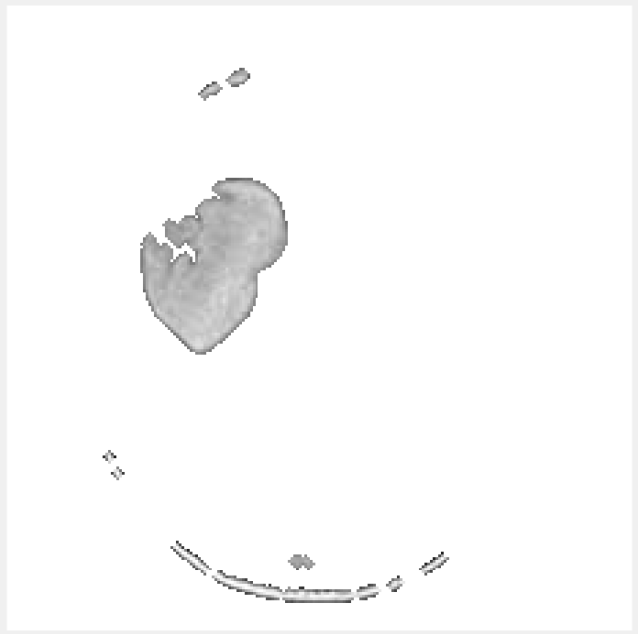


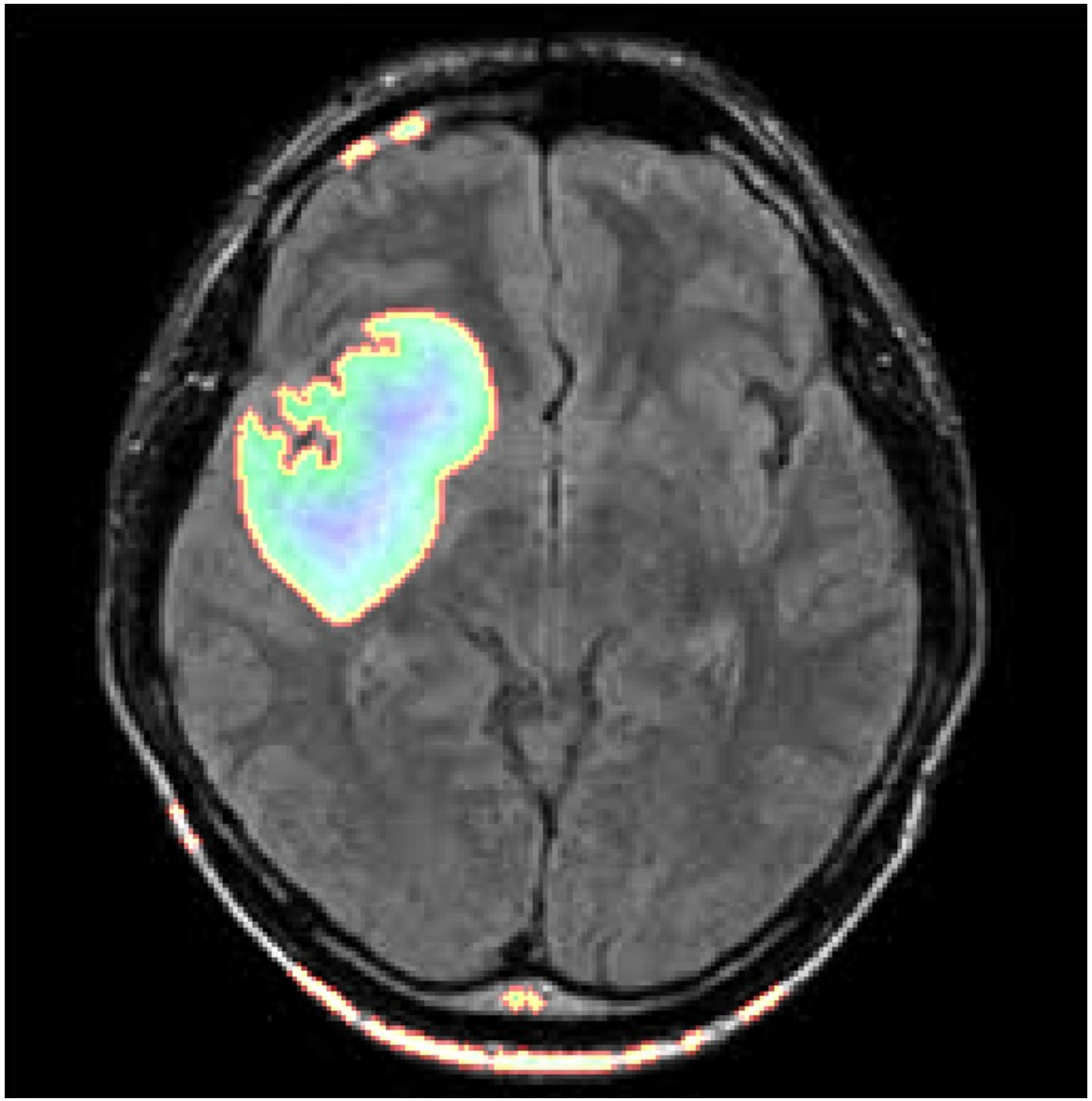
STEP 9

Highlighting Effected region

RGB image containing Brain tumour highlighted image

Output Image



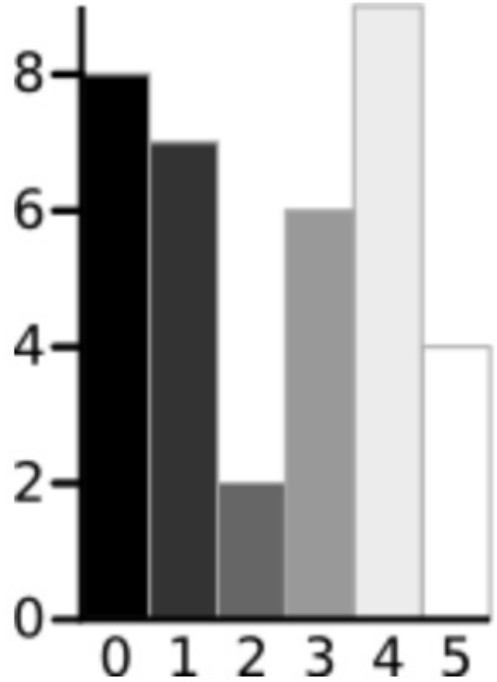


Hardware/FPGA Implementation:

Otsu Thresholding

To perform automatic image thresholding

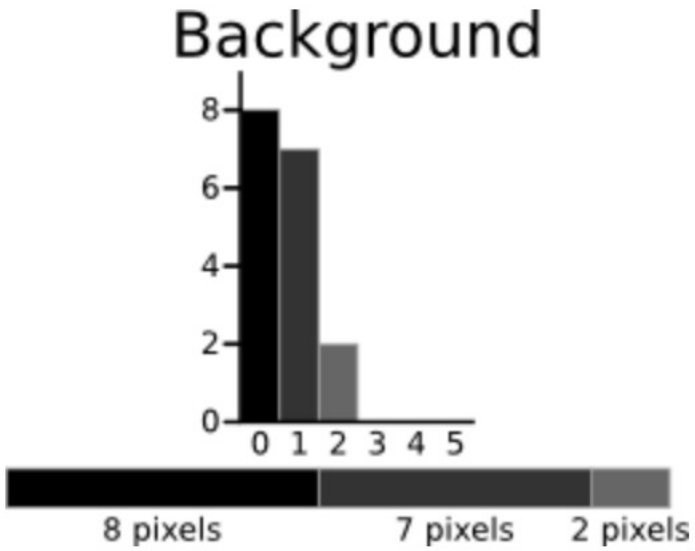
6X6 Image



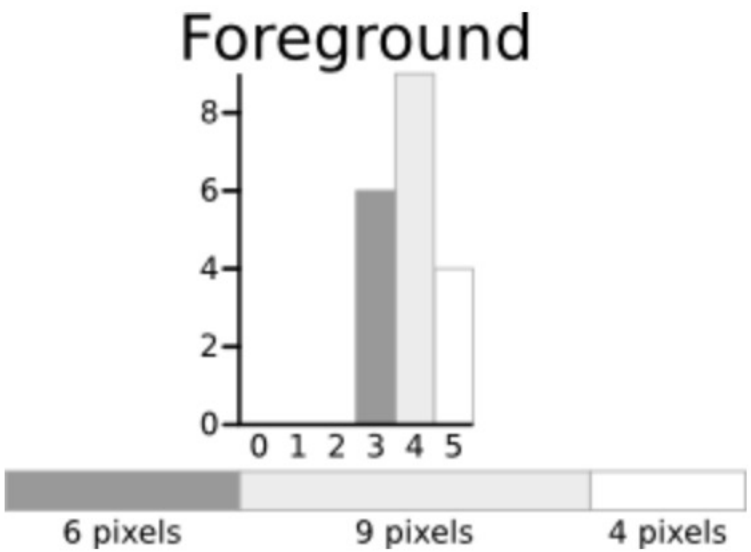
Histogram

Taken from: http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

We separate Image as background and Foreground Based on Threshold value



Histogram of Background Image

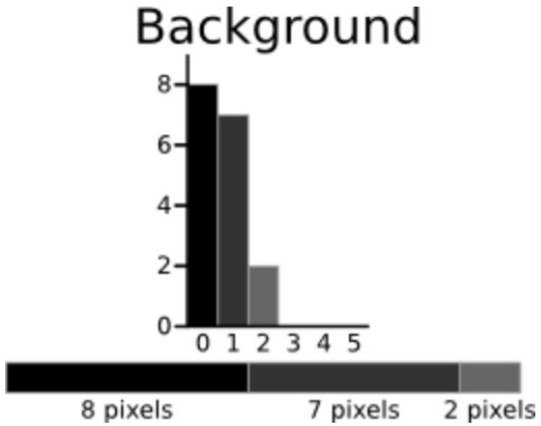


Histogram of Foreground Image

Let Threshold Value=3

Taken from: http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

Calculation



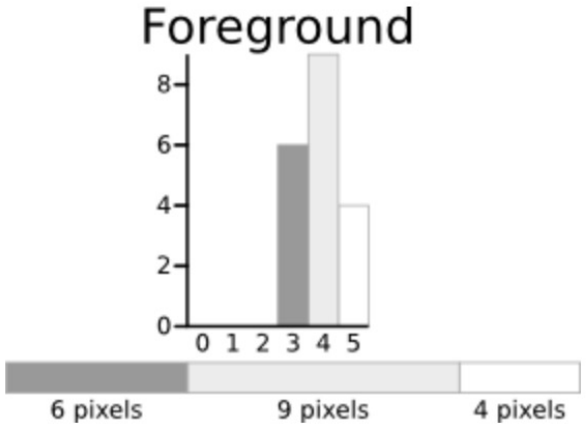
$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

$$\text{Mean} \quad \mu_b = \frac{(0\times 8)+(1\times 7)+(2\times 2)}{17} = 0.6471$$

$$\text{Variance} \quad \sigma_b^2 = \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17}$$

$$= \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17}$$

$$= 0.4637$$



$$\text{Weight} \quad W_f = \frac{6+9+4}{36} = 0.5278$$

$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

$$\text{Variance} \quad \sigma_f^2 = \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19}$$

$$= \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19}$$

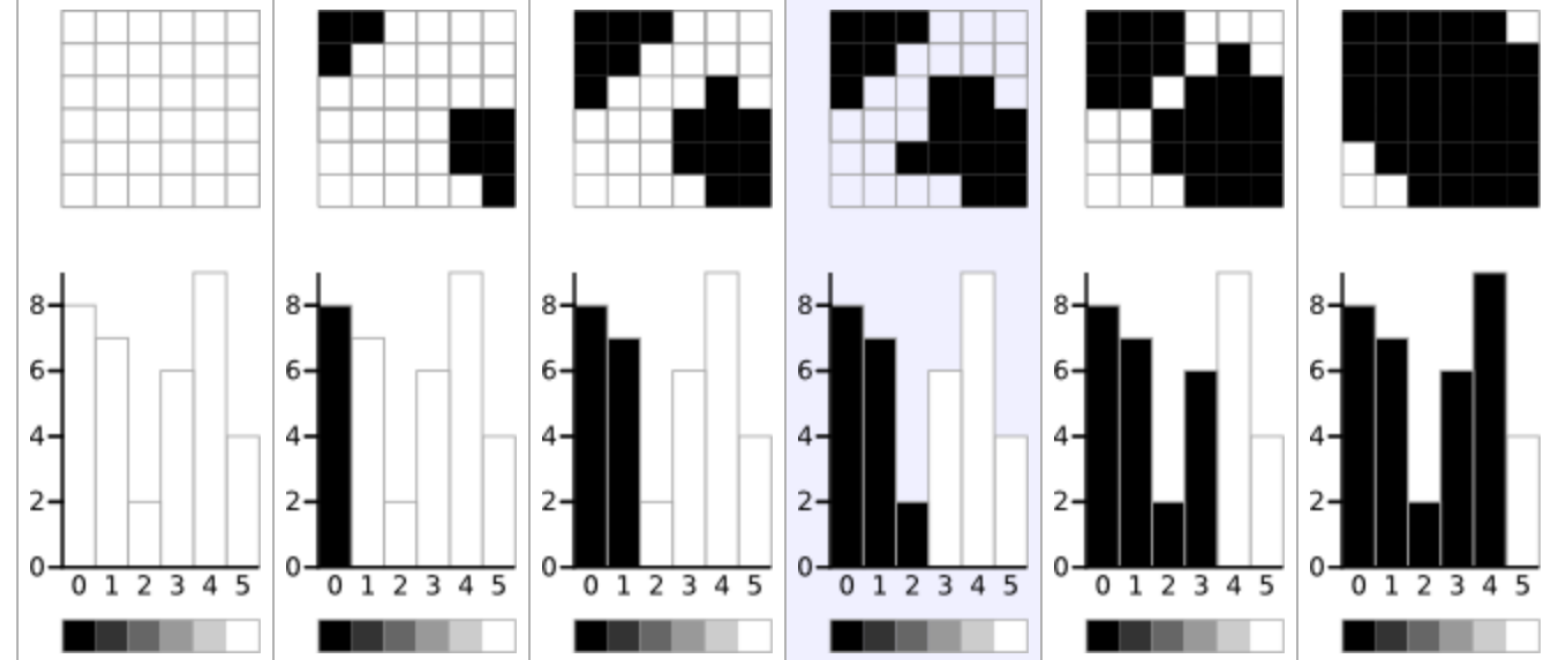
$$= 0.5152$$

$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152$$

$$= 0.4909$$

Taken from: http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

Comparison

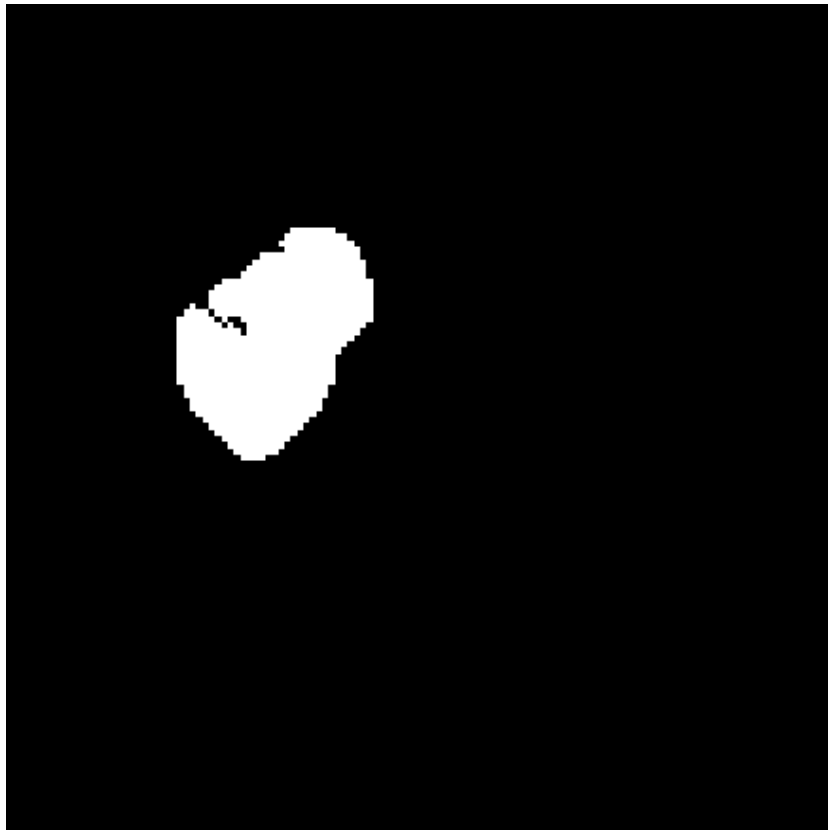
Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2_b = 0$	$\sigma^2_b = 0$	$\sigma^2_b = 0.2489$	$\sigma^2_b = 0.4637$	$\sigma^2_b = 1.4102$	$\sigma^2_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2_f = 3.1196$	$\sigma^2_f = 1.9639$	$\sigma^2_f = 0.7755$	$\sigma^2_f = 0.5152$	$\sigma^2_f = 0.2130$	$\sigma^2_f = 0$
Within Class Variance	$\sigma^2_W = 3.1196$	$\sigma^2_W = 1.5268$	$\sigma^2_W = 0.5561$	$\mathbf{\sigma^2_W = 0.4909}$	$\sigma^2_W = 0.9779$	$\sigma^2_W = 2.2491$

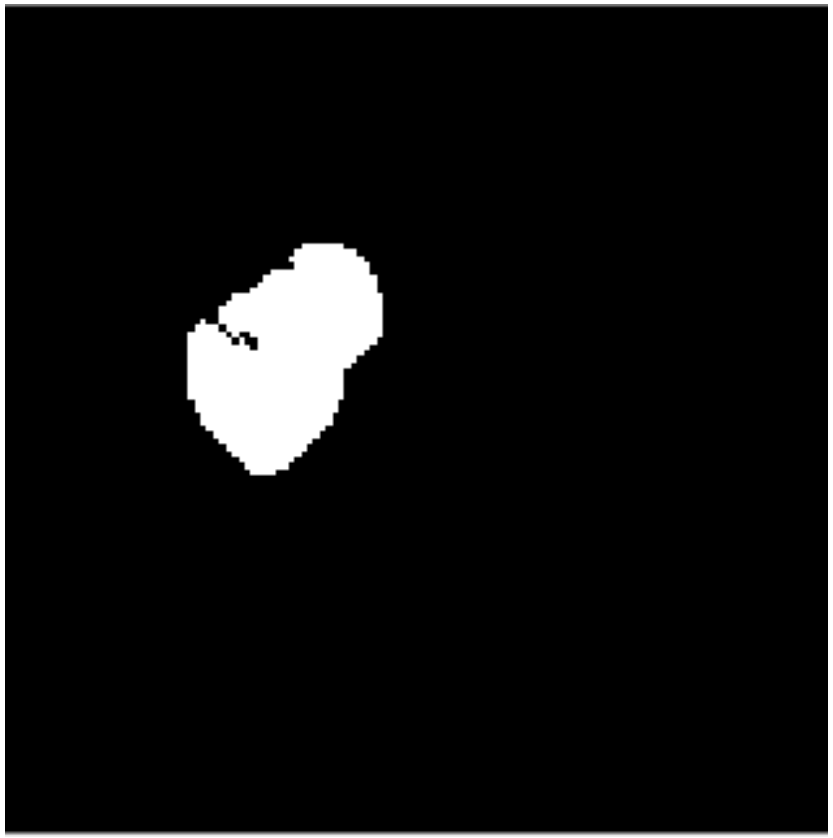
Taken from: http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

Implementation

- The proposed design is implemented on a Polar-Fire Soc Icicle kit for 128*128 pixels sized brain tumor image.
- The water-shed Algorithm is implemented on one of the five 64-bit RISC-V cores available on the kit. It consumes less than 128kB of the scratch-pad L2 cache memory.
- The Otsu-thresholding algorithm is implemented on the FPGA fabric.
- The UART is used to extract the output data from PS to PC.
- The post-processing of the extracted data is done using MATLAB.
- The Cross-verification of the resulted output images from our implementation is done with the MATLAB generated images for the same input image.

Results of Watershed: Morphology

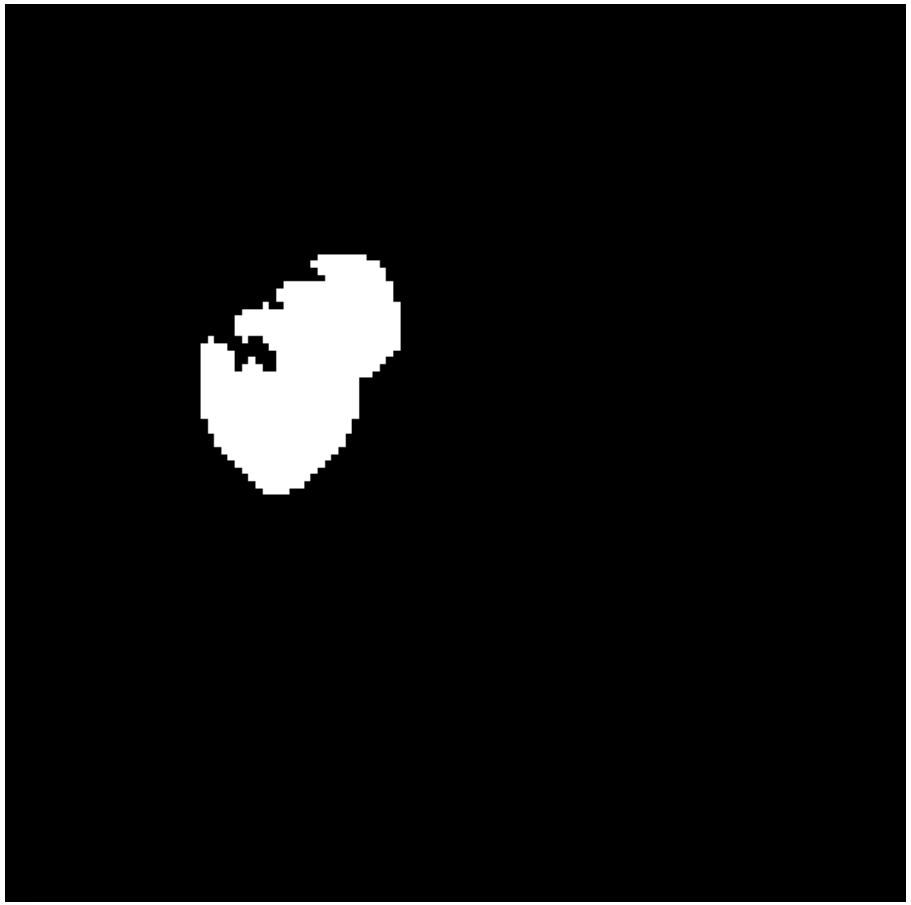




SIMULATED OUTPUT

HARDWARE OUTPUT

Results of Watershed: Dilation

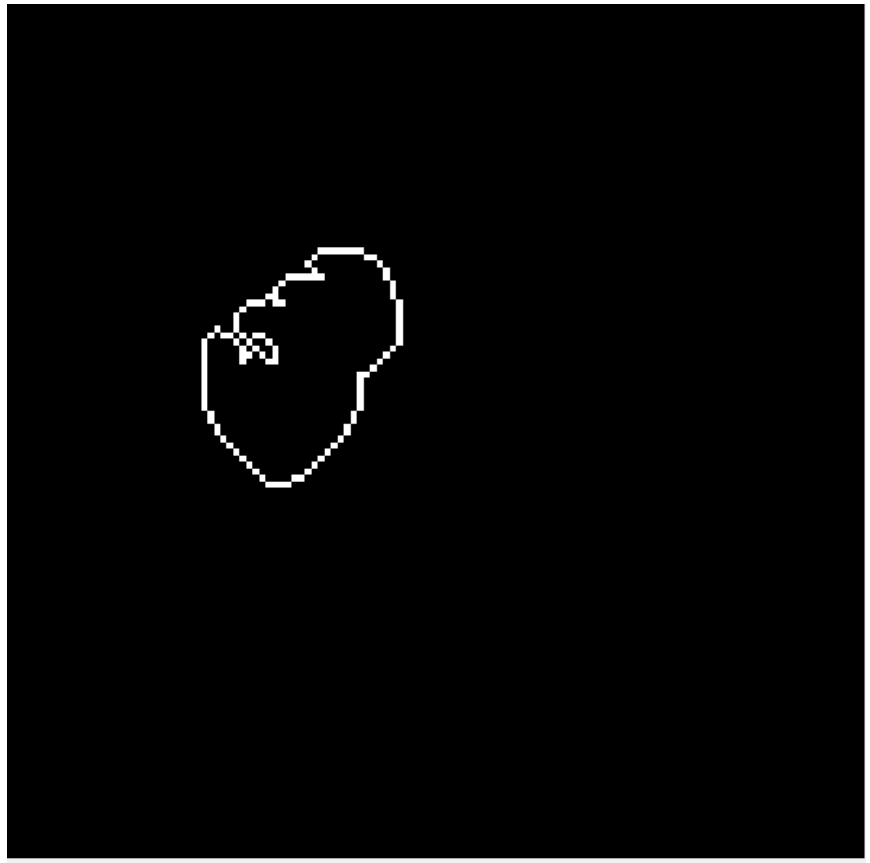


SIMULATED OUTPUT

HARDWARE OUTPUT

Results of Watershed: Contour Detection

SIMULATED OUTPUT



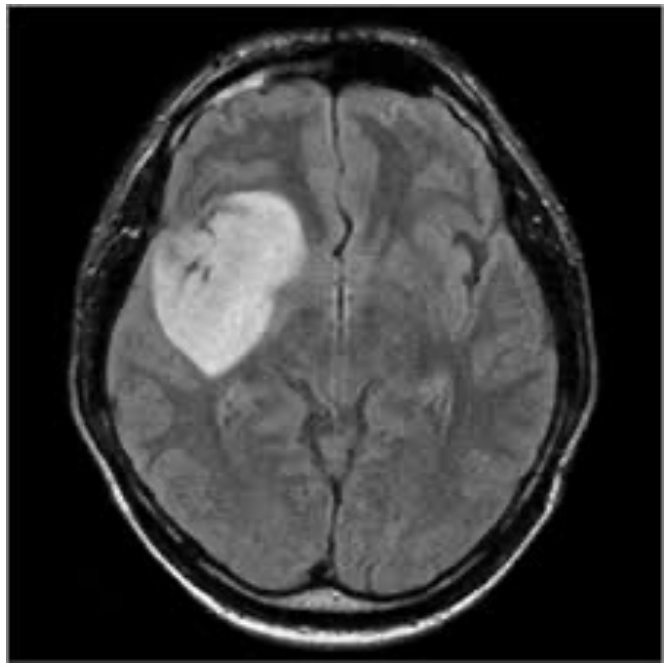
HARDWARE OUTPUT

Results of Watershed: Highlighted region

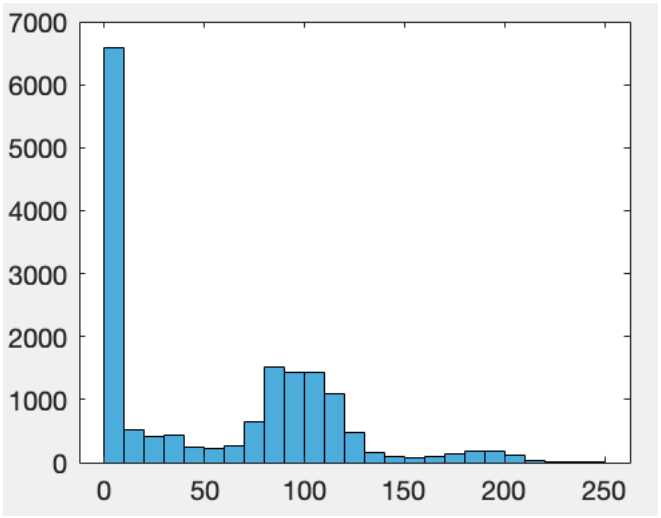
SIMULATED OUTPUT

HARDWARE OUTPUT

Results of Otsu Thresholding:



Input Image



Histogram of Input Image

Thresholded Image using Otsu's Method

Results: Logic Resources

Detailed Logic Resource Usage

Type	4LUT	DFF
Fabric Logic	18549	21201
uSRAM Interface Logic	0	0
LSRAM Interface Logic	216	216
Math Interface Logic	756	756
Total Used	19521	22173

Resource Usage

Type	Used	Total	Percentage
4LUT	19521	254196	7.68
DFF	22173	254196	8.72
I/O Register	0	432	0.00
User I/O	101	144	70.14
-- Single-ended I/O	101	144	70.14
-- Differential I/O Pairs	0	72	0.00
uSRAM	0	2352	0.00
LSRAM	6	812	0.74
Math	21	784	2.68
H-Chip Global	1	48	2.08
PLL	0	8	0.00
DLL	0	8	0.00
Transceiver Lanes	0	4	0.00
Transceiver PCIe	0	2	0.00
MSS	0	1	0.00

Results: Power and Timing

Power Summary

	Power (mW)	Percentage
Total Power	166.330	100.0%
Static Power	155.788	93.7%
Dynamic Power	10.542	6.3%

```
====== 2. Timing Result of HLS-generated IP Core (top-level module: ostu_top) ======

+--------------+---------------+------------+-------------+----------+-------------+
| Clock Domain | Target Period | Target Fmax | Worst Slack | Period   | Fmax        |
+--------------+---------------+------------+-------------+----------+-------------+
| clk          | 10.000 ns     | 100.000 MHz | 3.213 ns    | 6.787 ns | 147.341 MHz |
+--------------+---------------+------------+-------------+----------+-------------+
```

Future Work

Classification based on CNN model can be done to auto-detect and classify the tumour. This can be implemented as a hardware-software co-design with tasks (layers) divided between the CPU and FPGA fabric

Acknowledgements:

- Microchip technical support team for multiple Query meetings

References:

- Microchip Smart HLS Guide
- Microchip Libero Guide
- Microchip Soft console Guide
- Polar fire Icicle kit examples:

https://github.com/polarfire-soc/icicle-kit-reference-design

THANK YOU!