IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

REALIZATION OF CANNY EDGE DETECTION ALGORITHM USING FPGA

S.R. Dixit*, Dr. A.Y.Deshmukh

* Research scholar Department of Electronics & Engineering,G.H.Raisoni College of Engineering,
Nagpur-440016, India.
Department of Electronics Engineering, G.H.Raisoni College of Engineering,
Nagpur-440016, India.

ABSTRACT

Edge detection is basic operation of image in digital image processing and it is to be detected edges of an image. Various methods are used for image edge detection like Sobel, Log Operator, Laplacian, prewitt, Canny etc. However other edge detection methods perform poor as compared to the Canny edge detection algorithm. This edge detection can be used to detect the disease like Tonsilities. Therefore in this paper we have proposed a canny edge detection algorithm which is based on VHDL. Generally images are affected by various types of noise; to reduce the effect of such noise we have used Gaussian filtering. It also performs image smoothing. The aim of this paper is to develop a Canny Edge Detection by using XILINX System Generator (XSG) which automatically detects edges of digital image. By using this architecture we can detect the disease and can comment that weather the patient is affected by disease or not. The complete architecture is to be designed by using Xilinx Blocksets. The complete architecture design combines MATLAB, Simulink and XSG. The VHDL code is to be generated by using Xilinx system generator (XSG) and this generated VHDL code is to be synthesized in Xilinx ISE Design Suit 13.1.

KEYWORDS: Canny Edge Detection, Gaussian Filtering, Xilinx System Generator (XSG), VHDL.

INTRODUCTION

Digital image processing is field which deals with manipulation of digital images through a digital computer. It is a sub field of signals and systems but focus particularly on images. DIP is used for developing a computer system that is able to perform processing on an image. The input and output of that system are digital images, the system uses various algorithms to process an image. There is an well-known image processing algorithm is edge detection. Edge detection is a mathematical method used to identify points in a digital image where the image brightness changes sharply or has discontinuities in intensity of image. The set of points at which image brightness changes sharply are typically arranged into a set of curved line segments known as edges. In the same manner finding discontinuities in 1D signal is known as step detection and finding signal discontinuities over the discrete time is known as change detection. Edge detection is a very basic tool in digital image processing. It is useful in machine vision, computer vision, basically in the areas of feature detection and feature extraction.

Generally, the result of applying an edge detector to an image may give a set of curves that shows the boundaries of objects. Hence we can say that, applying an edge detection algorithm to an image may reduce the amount of data to be processed and preserves the important structural properties of an image. This paper uses a Canny Edge Detection algorithm which is developed y John F. Canny in 1986[1]. There are many methods have been proposed for canny edge detection. The method given by [2] uses fast multilevel fuzzy edge detection to increase the performance of edge detection. Mohamed Nasir Bin MohamadShukor& Lo HaiHiung [3] had given a real time implementation of edge detection which uses cyclone II FPGA pair with 1.3 megapixel CMOS camera. QianXu, ChaitaliCharabarti and Jina J. Karam [4] implemented distributed canny edge detector on FPGA which uses Xilinx Vertex – 5 FPGA. It reduces memory requirement and latency. For the images corrupted by Gaussian noise Wang Xiao, XueHui [5] proposed an improved canny edge detector which is based on predisposal method; by this proposed method better edge images were obtained. Alba M. Sanchez et al [6] developed an architecture which is based on Xilinx System Generator for image filtering. [7] Suggest a method for MRI images which is also based on Xilinx System Generator. It gives efficient FPGA implementation of MRI image filtering and tumor characterization. Dr. D. Selvathi and J. Dharani[8] proposed a Beamlet transform edge detection algorithm using FPGA, develop an efficient Beamlet edge detection. An implementation proposed by Dr. MohdFauzi Bin Othman et al [9] gives overview of MRI Brain classification using FPGA.

Edge detection of medical images is a very important work for object recognition of the various human organs like kidneys, brain, tonsils, heart etc. and it is also an essential pre-processing step in medical image segmentation. The architecture design is proposed to detect the tonsils weather it is affected by disease or not.

PROPOSED METHOD

Canny edge detection using Xilinx System Generator contains various processing steps. First step is image need to be filtered to remove existing noise. Next processing steps are to detect edges of an image by using Xilinx System Generator block sets, and then further steps are used to generate the VHDL code. The block diagram for the proposed method is shown as follows.

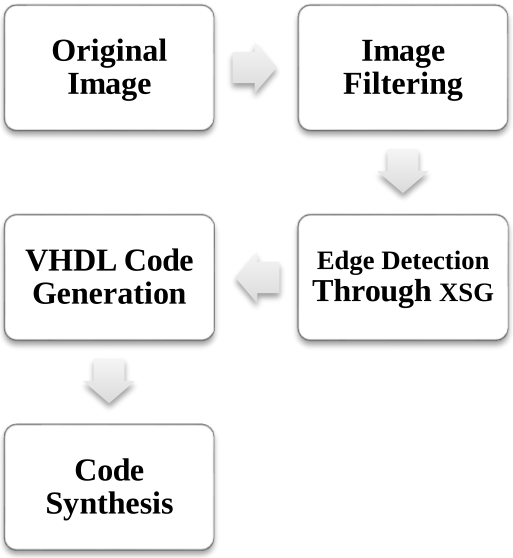


Fig 1.Block Diagram of Canny Edge Detection based on VHDL

Image Filtering

The images obtained from the digital source often contains noise elements, therefore detection of edges in such images gives very poor results. To obtain good results image is need to be filtered before applying edge detection. In proposed method Gaussian filtering is applied before Canny edge detection. This filtering is non-causal system it performs image smoothing by calculating weighted averages.

Canny Edge Detection

The edge detection is used to reduce the amount of data present in an image but preserves the structural properties of an image. The method in proposed architecture is canny edge detection. It is developed to produce optimal edge detection results. The canny edge detection first performs blurring to the image to reduce effect of noise. If we want develop canny edge detection we have to follow some criteria to improve existing methods of edge detection. The very first criterion is that it should not miss any edge that available in image i.e. it should gain lowest error rate. The second criteria is should perfectly localize edge points. The next criterion is that it responds to every single edge of image. The third criterion was necessary because there is a possibility of multiple edges in image.

To implement canny edge detector algorithm there are some processing steps. The initial step is to remove noise elements present in an image before applying edge detection. For removing noise we have used Gaussian filtering. The smoothing is performed using standard convolution after obtaining the suitable mask. After Gaussian filtering the result is a blurred image which is free from noise. Here shows an kernel of Gaussian with 5×5 mask having $\sigma = 1.4$.

$$\mathbf{B} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

The next processing step is to find gradients. the canny edge detection finds the edges with the grayscale intensity. It finds the area where the grayscale intensity changes the most and such types of area is found by gradients. The gradient at each pixel is determined by Sobel-operator. Initially it detects the gradients in X and Y directions. It uses two 3×3 convolution masks to find gradient in X and Y directions.. One is used to find gradient in rows i.e. in y-direction and other in column i.e. x-direction.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Then gradients are detected by using following the formula

$|\mathbf{G}| = |\mathbf{G}_X| + |\mathbf{G}_Y|$

The formula for edge direction is given as

$\Theta = \mathrm{invtan}(G_Y/G_X)$

Once the edge direction is located, the next processing step is to compare the edge direction to a direction that can be traced in an image.

Once the edge directions are known, the next task is called non-maximum suppression which is an edge thinning technique. Non-maximum suppression is used to convert the blurred edges of an image to sharp edges. This produces a thin line across the edges in an output image. The last step is to trace out edges from the image and perform hysteresis throsholding. Up to this step the edges are thin lines across the image. But it contains some streaking. Therefore hysteresis is used for remove such streaking from the output. The reson of streaking is fluctuation of operator above and below the threshold. It is having two types of thresholds – high and low level thresholds

.

VHDL Synthesis:

The implemented design is synthesize in ISE Design Suit, which produces device utilization summary for the targeted device.

PROCESSING STEPS

Xilinx System Generator (XSG) is developed by Xilinx Inc and the industry's leading high level tool. It is using FPGAs for designing high level DSP systems. For the developing the XSG design Simulink environment is used which is an model base design. By using XSG bit stream (*.bit) can be directly created which is necessary for programming FPGA. The Xilinx System Generator directly moves Simulink design directly into Bit Stream (*.bit) file. It provides system modeling and automatic code generation from Simulink and MATLAB. The system generator design is made up of bolcks called as "XILINX BLOCKS". These blocks contains of signs, ports, memory, Rom blocks attributes etc. These blocks are used to produce synthesis in FPGAs or HDL simulation..

Gaussian filtering using XSG

The Gaussian filtering model base design uses Xilinx blocksets like vertex5 line buffer, 5*5 filter Gateway In/Gateway Out etc .Gaussian filtering uses 5 line buffer to store the data in a sequential stream of pixels which constructs 5 lines in output of buffer. Each line is delayed by N samples, where N is the length of line. These outputs are inputs to the 5×5 filter which uses 5 MAC filters to filter the data. It used to filter grayscale images. The implemented Gaussian filtering using Xilinx blocksets is shown in Fig 2.

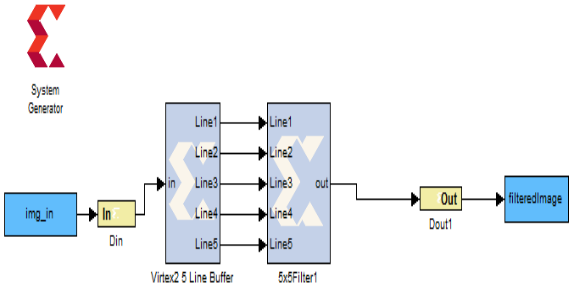


Fig 2. Gaussian Filtering using XSG

Canny Edge Detection using XSG

This design is also implemented by using Xilinx blocksets. Gateway In and Gateway out blocks converts inputs of type Simulink integer, double and fix point to Xilinx fixed point type and back conversion i.e. fixed point input into outputs of type Simulink, double, fix point integer. All the Xilinx blocksets are used between these two blocks only. The Canny Edge Detection using Xilinx blockets in simulink shown in Fig 3.

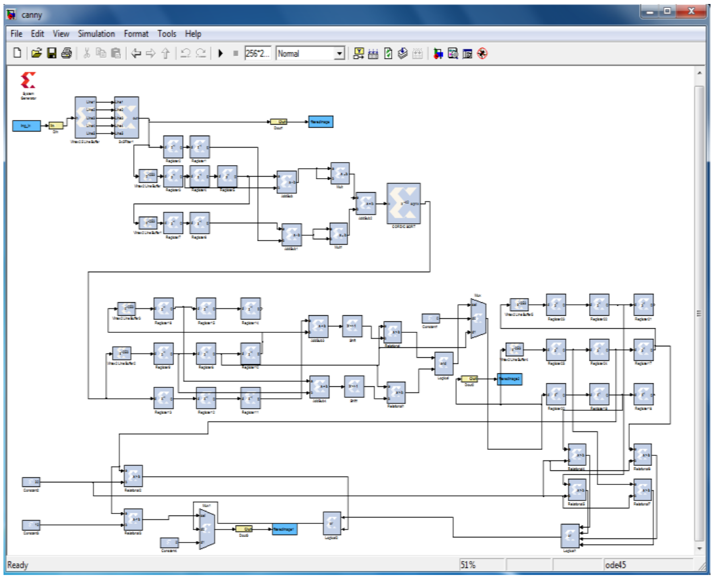
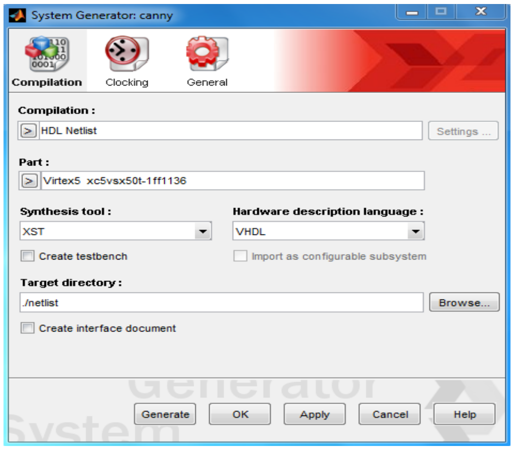


Fig 3.Canny Edge Detection using XSG

VHDL Code Generation

The Xilinx System Generator Token which is used as a control panel for controlling system and simulation parameters block called System Generator block. Any System Generator design which contains Xilinx blocksets must contain at least one or more System Generator token.



Compilation : HDL Netlist

Part : Virtex5 xc5vsx50t-1ff1136

Synthesis tool : XST

Hardware description language : VHDL

□ Create testbench

□ Import as configurable subsystem

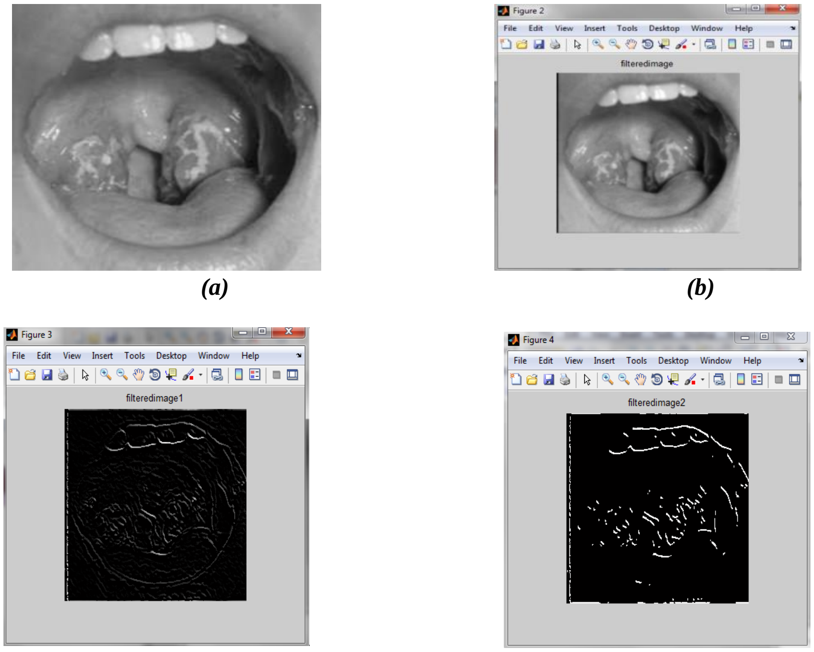
Target directory : ./netlist

□ Create interface document

Fig 4.System Generator Block

RESULTS

The proposed work is implemented by using MATLAB Simulink and XSG (Xilinx System Generator) with Xilinx blocksets. The method was tested on standard test image like ‘cameraman’. The output results show an image filtered from Gaussian filtering, another output is hysteresis throsholdingoutput which removes streaking form an image. Final image is of Canny Edge Detected image. The generated system is targeted for VIRTEX 5 starter kit. Further the VHDL code is generated by using System Generator token, this code is perfectly synthesized in ISE 13.1 Design Suit. After synthesis the device resource usage summary was produced for the targeted device.



(a) *(b)*

(c) *(d)*

Fig 5. (a) Original Image (b) Gaussian Filtering Result
(c) Hysteresis Throsholding Result (d) Canny Edge Detection Result

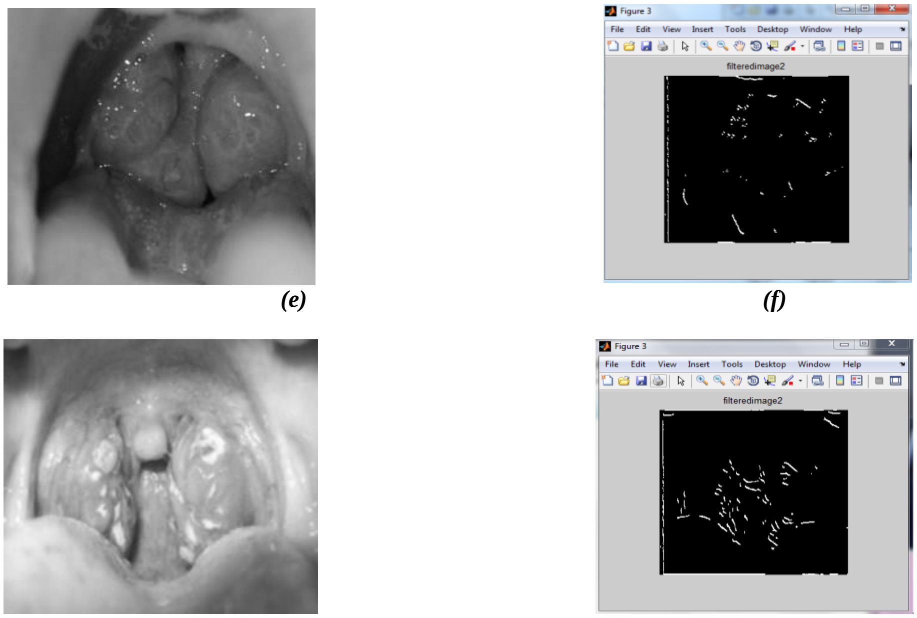


Figure 3
File Edit View Insert Tools Desktop Window Help
filteredimage2
Figure 3
File Edit View Insert Tools Desktop Window Help
filteredimage2

(e) *(f)*

(g) *(h)*

Fig 6. (e) Original Image (f) Gaussian Filtering Result (g)Hysteresis Throsholding Result (h) Canny Edge Detection Result

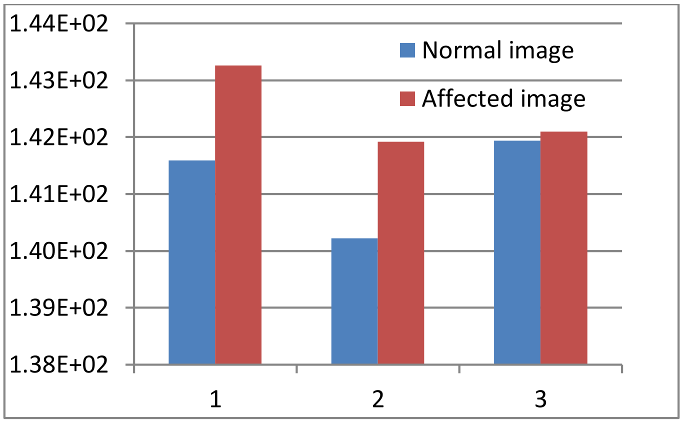
These are two experiments on tonsilities affected images and it is compared with normal images. However only comparing by visualizations is not considered, therefore we sated three parameters mean, Standard deviation and Variance. The analysis for these three parameters is shown in following parameters. It is observed that mean of affected images is always greater than mean of normal images for all the three experiments. The other parameter is Standard Deviation it is having greater value for normal images and lower value for affected images the case is followed in Variance analysis. Thus by comparing these images on these three parameters we can say that weather the tonsilis are affected by disease or not affected with disease.

Comparison of images based on parameters

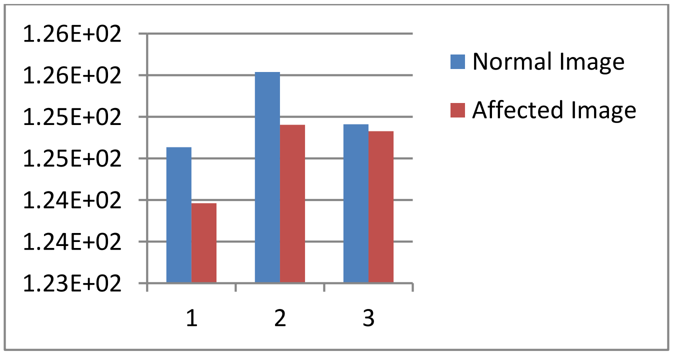
Sr.no	**Images**	**Mean**	**Standard Deviation**	**Variance**
1	Normal Image	1.415e+02	1.246e+02	1.55e+004
	Affected Image	1.432e+02	1.239e+02	1.53e+004
2	Normal Image	1.402e+02	1.255e+02	1.57e+004
	Affected Image	1.41e+02	1.249e+02	1.56e+004
3	Normal Image	1.410e+02	1.249e+02	1.56e+004

	Affected Image	1.4210e+02	1.248e+02	1.55e+004

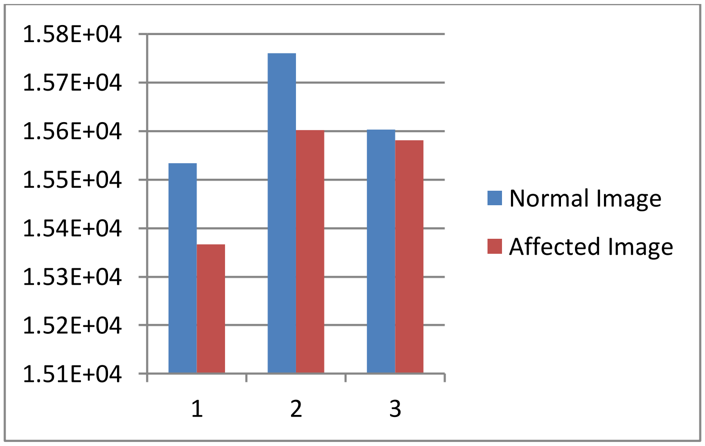
Mean Analysis:



Standard Deviation Analysis:



Variance Analysis:



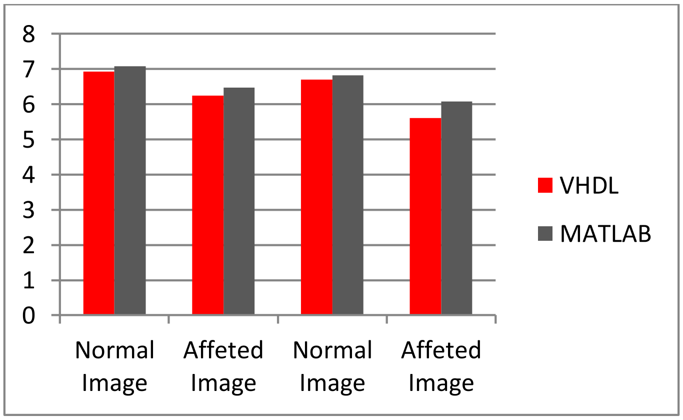
The generated output results are compared with existing MATLAB results. For comparison of edge detected images we sated two parameters i.e. PSNR and Performance Ratio. By comparing PSNR we have achieved almost 90% of results as compared to MATLAB, whereas compared with performance ratio we have achieved almost 50% of MATLAB results.

MATLAB to VHDL Comparison

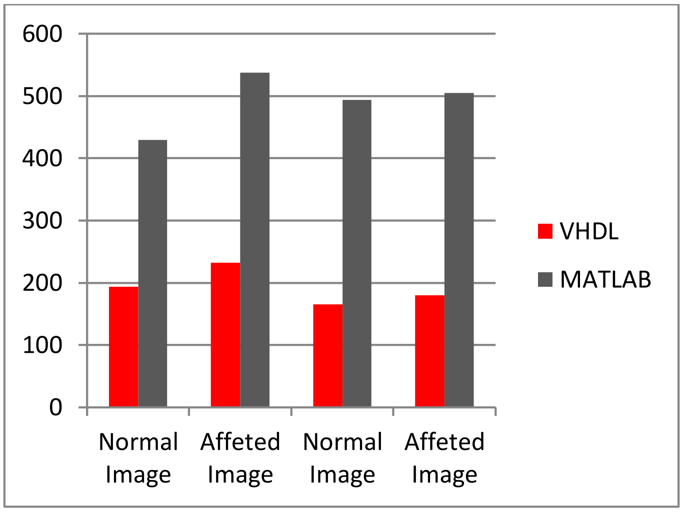
The implemented design results are compared with existing MATLAB results. For comparison of edge detected images we have sated two parameters i.e. PSNR and Performance Ratio. By comparing PSNR we have achieved almost 90% of results as compared to MATLAB, where as compared with performance ratio we have achieved almost 50% of MATLAB results.

Sr. No.	Images (Normal/ Affected)	VHDL		MATLAB	
		PSNR	Performance Ratio	PSNR	Performance Ratio
1.	Normal Image	6.9241	193.2128	7.0814	429.3700
	Affected Image	6.2411	231.8278	6.4755	537.2618
2.	Normal Image	6.6936	165.4569	6.8165	494.1075
	Affected Image	5.6063	180.2121	6.0852	504.5756
3.	Normal Image	6.8997	179.6382	7.0948	487.1349
	Affected Image	5.6371	181.5725	5.7472	321.1555

PSNR Analysis:



Performance Ratio Analysis:



The device utilization summary is calculated for the implemented design. Performance of this design implemented in Vertex5 (xc5vsx50t-1ff1136) and Vertex5 (xc5vsx50t-1ff665) as shown in following Table. The implemented architecture gives lower complexity and improves efficiency in area. It also a good choice for low cost hardware design.

DEVICE UTILZATION SUMMARY
(Vertex5 xc5vsx50t-1ff1136)

Resource	Used	Available	Device usage
Registers	2,931	97,280	3%
LUTs	2,247	97,280	2%
Logic	1,914	97,280	1%
Memory	302	26,240	1%
Number of occupied Slices	1,042	24,320	4%
Number with an unused Flip Flop	375	3,306	11%
Number with an unused LUT	1,059	3,306	32%
fully used LUT-FF pairs	1,872	3,306	56%
slice register sites lost to control set restrictions	92	97,280	1%
IOBs	83	640	12%
BlockRAM/FIFO	14	212	6%
Total Memory used (KB)	342	7,632	4%
BUFG/BUFGCTRLs	1	32	3%
DSP48Es	9	128	7%

DEVICE UTILZATION SUMMARY
Vertex5 (xc5vsx50t-1ff665)

Resource	Used	Available	Device usage
Registers	2,920	32,640	8%
LUTs	2,247	32,640	6%
Logic	1,914	32,640	5%
Memory	302	12,480	2%
Number of occupied Slices	989	8,160	12%
Number with an unused LUT	952	3,199	29%
fully used LUT-FF pairs	1,968	3,199	61%
slice register sites lost to control set restrictions	83	32,640	1%
IOBs	83	360	23%
BlockRAM/FIFO	14	132	10%
Total Memory used (KB)	342	4,752	7%
BUFG/BUFGCTRLs	1	32	3%
DSP48Es	9	288	3%

CONCLUSION

The presented proposed work gives an efficient and robust architecture for Canny Edge Detection algorithm using Xilinx System Generator. The edges are perfectly detected using Xilinx Blocksets in Simulink. Canny Edge Detector provides better results than other operators i.e. Sobel, Laplacian, Log, prewitt etc. The technique uses updated Xilinx System the System Generator is present within ISE 13.1 design suit. The implemented architecture is targeted on Vertex5 xc5vsx50t-1ff1136 and Vertex5 (xc5vsx50t-1ff665) starter kits. The obtained results of tonsilities images perfectly detected the affected tissues. By using the standard parameters the normal and affected image can be compared. It is observed that mean of normal images is always lower than affected images, whereas Standard Deviation and Variance of normal images is always greater than affected images for all the three experiments. The generated VHDL results are compared with MATLAB results on the basis of PSNR and Performance Ratio and it is shown that in PSNR analysis of VHDL results are achieved almost 90% of MATLAB results, Performance ratio is 50% of MATLAB results. Generated VHDL code is perfectly synthesized in ISE design suit, produced device utilization summary which improves efficiency in area. This approach is very useful in medical images to diagnose the diseases.

REFERENCES

[1] John F. Canny, “A computational approach to edge detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 8. 1986.

[2] Jinbo Wu, Zhouping Yin, and YoulunXiong, "The Fast Multilevel Fuzzy Edge detection of Blurry Images", IEEE Signal Processing Letters, Vol. 14, No. 5, May 2007.

[3] Mohamed Nasir Bin Mohamed Shukor, Lo HaiHiung, Patrick Sebastian, “Implementation of Real-Time Simple Edge Detection on FPGA”, International Conference on Intelligent and Advanced Systems 2007

[4] QianXu, ChaitaliChakrabarti and Lina J. Karam, “A Distributed Canny Edge Detector and Its Implementation on FPGA”, 2011 IEEE.

[5] Wang Xiao, XueHui, “An Improved Canny Edge Detection Algorithm Based on Predisposal Method for Image Corrupted By Gaussian Noise”

[6] Alba M. Sanehez G., Rieardo Alvarez G., Sully Sanehez G, "Architecture For Filtering Ximages Using Xilinx System Generator", International Journal of Mathematical Models and Methods in Applied Sciences, Volume 1, 2007.

[7] Mrs. S. AllinChriste, Mr.M. Vignesh, Dr.A.Kandaswamy, "An Efficient FPGA Implementation of MRI Image Filtering and Tumor Characterization Using Xilinx System Generator", International Journal of VLSI Design & Communication Systems (VLSICS) Vo1.2, No.4, December 2011.

[8] Dr.D.Selvathi, J.Dharani, “Realization of Beamlet Transform Edge DetectionAlgorithm Using FPGA”, 2013 International Conference on Signal Processing, Image Processing and Pattern Recognition [ICSIPR].

[9] DrMohdFauzi Bin Othman ,Norarmalina Abdullah, NurAizudin Bin Ahmad Rusli, "An Overview Of MRI Brain Classification Using FPGA Implementation", IEEE Symposium On Industrial Electronics& Applications (ISIEA) Oet201O,Malaysia.