Lane detection system for day vision using altera DE2

Amjad J. Humaidi*[1] , Mohammed Abdulraheem Fadhel[2] , Ahmed R. Ajel[3]
[1]Control and System Engineering, University of Technology, Baghdad, Iraq.
[2]College of Business Informatics, University of Information Technology and Communication, Baghdad, Iraq.
[3]Department of Control and Automation, Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
*Corresponding author, e-mail: 601116@uotechnology.edu.iq[1], Mohammed.a.fadhel@uoitc.edu.iq[2], dr_ahmed.R@eetc.mtu.edu.iq[3]

Abstract

The active safety systems used in automotive field are largely exploiting lane detection technique for warning the vehicle drivers to correct any unintended road departure and to reach fully autonomous vehicles. Due to its ability, to be programmed, to perform complex mathematical functions and its characterization of high speed processing, Field Programmable Gate Array (FPGA) could cope with the requirement of lane detection implementation and application. In the present work, lane detection is implemented using FPGA for day vision. This necessitates utilization of image processing techniques like filtering, edge detection and thresholding. The lane detection is performed by firstly capturing the image from a video camera and converted to gray scale. Then, a noise filtering process for gray image is performed using Gaussian and average filter. Methods from first and second order edge detection techniques have been selected for the purpose of lane edge detection. The effect of manually changing the threshold level on image enhancement has been examined. The results showed that raising threshold level would better enhance the image. The type of FPGA device used in the present work is Altera DE2. Firstly, the version DE2 Cyclone II start with (11xxxxxx-xxxx) together with Genx camera has been used. This camera supports both formats NTSC and PAL, while the above version of FPGA backups only NTSC format. The software of lane detection is designed and coded using Verilog language.

Keywords*: altera DE2, FPGA, LAN detection, noise filtering*

1. Introduction

Presently, road accidents are at the top leading causes of human being death, injury and disability. The loss of economic cost is also mainly attributed to road traffic accidents. It has been reported that the main reason behind most accidents are due to driver behavior. Falling asleep at the vehicle wheel, inattention and distraction (looking but cannot see) are the main identified causes of accidents [1]. To avoid disorders of driver behavior, the vehicle motion is either controlled automatically or supplied by such means which can alarm the driver for any unexpected or potential collision. Therefore, different technologies and systems have been developed to reduce the road accidents such as road monitoring system, Global Positioning System (GPS), collision avoidance system, wireless communication, in a step avoid road accidents [2].

The lane detection technique can locate the lane boundaries of road or path geometry without a prior knowledge. The edge-based techniques are able to convert the detected edges into significant structures; even so they can find a suitable model for these edges. Many edge-based algorithms employ straight lines to formulate the lane boundaries, while others utilize more complicated models like parabola, B-Splines, hyperbola, etc. [3]. Vision-based lane detection and departure warning system has been investigated over two decades. During this period, many different problems related to lane detection and departure warning have been addressed [3, 4]. Deep learning based lane boundary detection method that can accurately detect lane boundaries under complex weather conditions and traffic scenarios in real time [5]. In this work, we present the implementation of lane boundary detection on hardware based on parallel processing to overcome time consuming.

Received May 28, 2018; Revised September 12, 2018; Accepted October 15, 2018

2. Edge Detection Process

In the present work, the considered edge detection process comprises three stages; Noise filtering, edge detection techniques and thresholding. In what follows, a detail explanation will be presented [6].

2.1. Noise Filtering

Digital images are exposed to different types of noise. Noise occurs due to errors in the acquisition process of the image and its adverse effect is that the image pixel values do not represent the true intensities of the real scene. Various sources may cause noise in images. Noise may come from high temperature or transmission such as electronic circuit noise. Another source of noise is referred as quantization error which results from quantizing the pixels of sensed image into discrete levels. The noise can be introduced through brightening shadows or through color-balance processing [7, 8]. Many types of filters are considered in the image processing literature for removing the noise from an image; such as median filter, mean filter, high pass filter, average filter, Gaussian filter, inverse filter and Wiener filtering. In the present work, two famous filters have been only employed for real time edge detection; namely average and Gaussian filters [9].

2.1.1. Averaging Operator

Noise reduction is the main advantageous effect of averaging. On the other hand, the averaging could cause blurring which results in reducing the details in an image. Moreover, it is a low-pass filter; as it permits retention of low spatial frequencies and suppression of high frequency components. Larger template, say 3×3 or 5×5, helps to get rid more noise (high frequencies), however it leads to reduce the level of details [8], [10]. The template weighting functions of averaging operator are unity (or 1/9 such as to guarantee that the outcome of averaging nine white pixels is white) as indicated in Figure 1.

1/9	1/9	1/9	1/9	1/9
1/9	1/9	1/9	1/9	1/9
1/9	1/9	1/9	1/9	1/9
1/9	1/9	1/9	1/9	1/9
1/9	1/9	1/9	1/9	1/9

Figure 1. 5×5 averaging operator template [3]

2.1.2. Gaussian Averaging Operator

The Gaussian averaging operator is one of the best smoothing technique for noise removal of the image. The Gaussian relationship sets up the values of Gaussian operator template. As shown in (1), the formula describes the Gaussian function at coordinates (x, y) and it is controlled by the variance σ^2

$$G(x, y, \sigma) = \frac{1}{2\pi\,\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

To find the coefficients for Gaussian template using (1) which is then convolved with the image. The 3×3 convolution template for Gaussian with standard deviation $\sigma = 1$ is given in (2) [9], [11].

$$\mathrm{G}=\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2}$$

2.2. Edge Detection Techniques

The techniques of edge detection are based on mathematical methods which are capable of recognizing and detecting points in a digital image at which the brightness of the image is sharply changed. These critical points are typically arranged into groups of curved line segments called edges which are represented as binary image. The main objectives of edge detection are [9], [12]:

a. Producing line-based drawings of scenes derived from their images.
b. Extracting the important features from image edges such as corners, lines and curves. The algorithms of computer vision (e.g., recognition) are mostly utilized from such features.
c. Intensity changes at different image locations are physically caused by various events such as [8], [13];
d. discontinuity in surface orientation and/or surface color and texture (surface boundary),
e. discontinuity in depth and/or surface color and texture (object boundary),
f. geometric and non-geometric events,
g. direct reflection of light, such as a mirror (specularity),
h. inter-reflections and shadows (from the same object or other objects).

The edge detection techniques comprise two groups; first order and second order. Examples of the first group are Robert, Sobel, Priwitt, kirsch and Robinson technique [14]. Sobel is the most famous algorithm. Robinson technique is characterized by having many masks; as two of these masks are similar to those of Sobel technique. In the present work, Sobel and Robinson have been selected as representatives of first order group. On the other hand, the second order group includes mainly Laplacian, Laplacian of gaussain and Marr–Hildreth algorithm [15]. The first two techniques are chosen to implement real time edge detection.

2.2.1. First Order Edge Detection

In image processing application, the calculation of the first derivative of a signal is approximated by finite differences as indicated in (3):

$$f'(x) = \lim_{d\to 0} \frac{f(x+d)-f(x)}{d} \tag{3}$$

if d=1, then the discrete derivative version is written as

$$f'(x) \cong f(x+d) - f(x) \tag{4}$$

the local minima or maxima of first derivative are utilized to detect the points lying on an edge as illustrated in Figure 2 [8], [16].

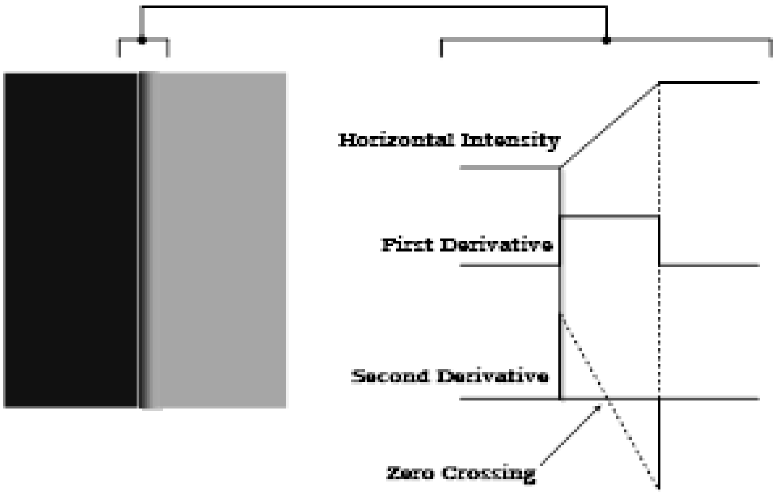


Figure 2. Edge detection derivatives [5]

a. Sobel Edge Detection Technique

The Sobel operator is based on central difference; however, it assigns higher weight value to the central pixels when averaging. The following equation gives 3×3 convolution of Sobel operators.

$$G_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{5}$$

The two masks are individually applied on the input image to give gradient component in horizontal orientation $\mathrm{G_x}$ and other gradient component in vertical orientation $\mathrm{G_y}$ [3], [17]. The magnitude gradient of is usually computed using (6):

$$G\,[f(x,y)] = |G_x| + |G_y| \tag{6}$$

the gradient direction can be represented as:

$$\theta(x,y) = tan^{-1}(G_y/G_x) \tag{7}$$

b. Robinson Edge Detection Technique

The masks of Robinson edge detection masks are called compass masks since they are defined by rotating a single mask to eight main compass orientations: north, west, south, east, north-west, north-east, south-west, and south-east. The masks are depicted in Equation below:

$$\begin{gathered} R_0 = \begin{bmatrix} 1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad R_1 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad R_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad R_3 \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\ R_4 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad R_5 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \quad R_6 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad R_7 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \end{gathered} \tag{8}$$

The maximum value resulting from the convolution of each of the masks with the image is defined as the edge magnitude. The edge detection is found by the mask that produces the maximum magnitude. It is worth to note that the masks $\mathrm{R_2}$ and $\mathrm{R_4}$ are identical to the Sobel masks. Extraction of explicit information about edge in any direction can be found by rotating any of the edge detection masks in a manner like compass masks [4], [18].

2.2.2. Second-order Edge Detection

Points which lie on an edge can be detected by detecting the zero-crossing of the second derivative as shown in Figure 2. The second derivative is approximated by finite differences as depicted in (9) [19]:

$$\begin{aligned} f''(x) &= \lim_{d\to 0} \frac{f'(x+d)-f'(x)}{d} \cong f'(x+1) - f'(x) \\ &= f(x+2) - f(x+1) - [f(x+1) - f(x)] \\ &= f(x+2) - 2f(x+1) + f(x) \end{aligned} \tag{9}$$

where $d = 1$

a. Laplacian Edge Detection

The Laplacian definition of an image defined by $\mathrm{f(x,y)}$ is a second order derivative given by (10):

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{10}$$

The approximation version of above equation in digital image is given to 4 and 8-connectivities as indicated (11) and (12), respectively [11], [20]:

4-connectivity: $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

$$\nabla^2 f = 4\,z_5 - (z_2 + z_4 + z_6 + z_8) \tag{11}$$

8-connectivity: $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

$$\nabla^2 f = 8\,z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9) \tag{12}$$

where z_i is the value of i-th position of matrix element; i.e., for 3×3 mask, the following mask can be represented in terms of z_i [9], [21].

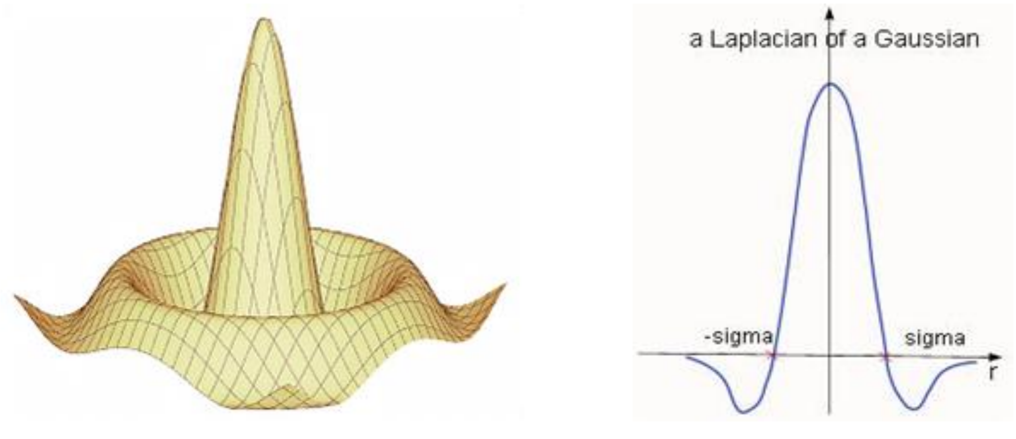
$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

b. Techniq Laplacian of Gaussian (LoG) Operator

Laplacian operator is noise susceptible. To decrease the noise sensitivity, Laplacian of Gaussian (LoG) operator can be used instead. LoG is a two-step algorithm; it firstly performs the Gaussian smoothing followed by Laplacian operation. Low susceptibility of this method to noise is due to noise reduction caused by Gaussian function and probability of detection of false edges is considerably minimized by resultant Laplacian mask. The LoG function for convolution is defined as follows [14], [22]:

$$LOG\ (x, y) = \frac{1}{\pi\,\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{13}$$

Figure 3 (a) shows the three-dimension plot of LoG function which looks like a Mexican hat. The above equation tells that a wider convolution mask is required for higher value of σ. It is easy to find out that the first zero-crossing of the LoG function occurs at $\sqrt[n]{2\sigma}$. Two-dimensional cross section of the zero-crossing of the Mexican hat is illustrated in Figure 3 (b) [17], [23]. In (14) utilizes 5×5 convolution mask to carry out LoG edge detector.



(a) (b)

Figure 3. Laplacian of Gaussian technique (a) plot of LoG function in 3D (b) zero crossing

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{14}$$

2.3. Thresholding (Gray Scale Thresholding)

Thresholding is a popular technique used for segmentation of monochrome image. The image intensity defined by $f(x, y)$ comprises light object and dark background. A threshold level T is designated to extract the objects from background and the thresholded image $g(x, y)$ can be described using (15). The (15) is only suitable when the image has 2 dominant gray level ranges [3], [24], [25].

$$g(x,y) = \begin{cases} 1 & if \quad f\,(x,y) > T \\ 0 & if \quad f\,(x,y) \le T \end{cases} \tag{15}$$

3. Video Signal Formatting for Lane Detection

In what follows, video signal formatting will be elaborated for lane detection. The data read directly from camera and acquired by composite cable is available in the ITU656 format as illustrated in Figure 4 [3]. These data have to be firstly converted into standard format YUV 4:2:2, which is commonly referred to as YCbCr. This format, YCbCr, is a color display format and it is composed of three components; the luma component (Y), blue-difference component (Cb) and red-difference chroma component (Cr). This format is a way of encrypting RGB data and it does not represent an actual color space of its own [4].

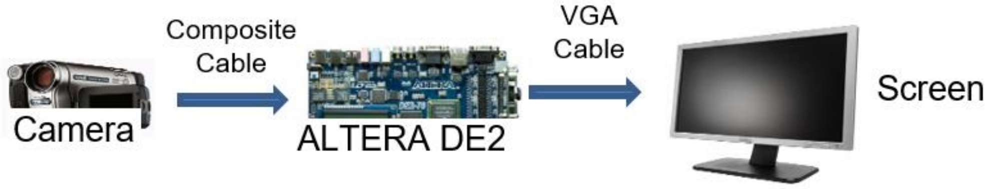


Figure 4. Hardware setup

As soon as the data is changed to this format, the system will sample down the signal from 720 to 640 horizontal pixels. Then the converted data is fed into an SDRAM FIFO which works as a frame buffer. Another conversion process is performed for the output of the FIFO, which transform the data from the format YUV 4:2:2 to the format YUV 4:4:4. At last, newly formatted YCbCr data is once more converted to the standard 10-bit RGB format.

The RGB data is directly fed into VGA controller to be next displayed on standard VGA monitor. However, instead of going directly to the VGA controller, the RGB data are admitted to go through other modules; filtering and edge detection modules [13]. The above discussion is illustrated in Figure 5 and the functionality description of different modules in this figure is summarized in Table 1. The extension (.v) indicates that the Verilog is the programming Languages that has written the code.

Table 1. Summary of Modules

Module name	Description
DE2_TV.v	This file contains the top-level module.
TD_Detect.v	Detect a valid video input.
ITU_656_Decoder.v	Convert the video from ITU-656 format to YUV 4:2:2 formats.
DIV.v	Used by previous module to reduce the horizontal resolution from 720 to 640 lines per frame.
Sdram_Control_4port.v	A buffer used for combining the odd and even field of a frame.
YUV422_to_444.v	Convert the video from 4:2:2 to format 4:4:4 formats.
YCbCr2RGB.v	Convert the video word from 8 to 10 bits to match the video output DAC.
VGA_Ctrl.v	Controls the output to the LCD monitor.
Line_Buffer.v	One line delay buffer used for de-interlacing the image.
I2C_AV_Config.v	Module controlling the video decoder.
Average.v	Modules of average filter.
Gaussain.v	Modules of gaussain filter.
Sobel.v	Sobel module.
Robinson.v	Robinson module.
LoG.v	Laplacian of gaussain module.
Laplacian.v	Laplacian modules.

It is interesting for edge detection to vary the threshold level to display different amounts of detail on the VGA screen. There are eighteen switches resident on the DE2 board. Table 2 illustrates how they can be utilized for resetting, switching filter and for manually adjusting threshold level of edge detection.

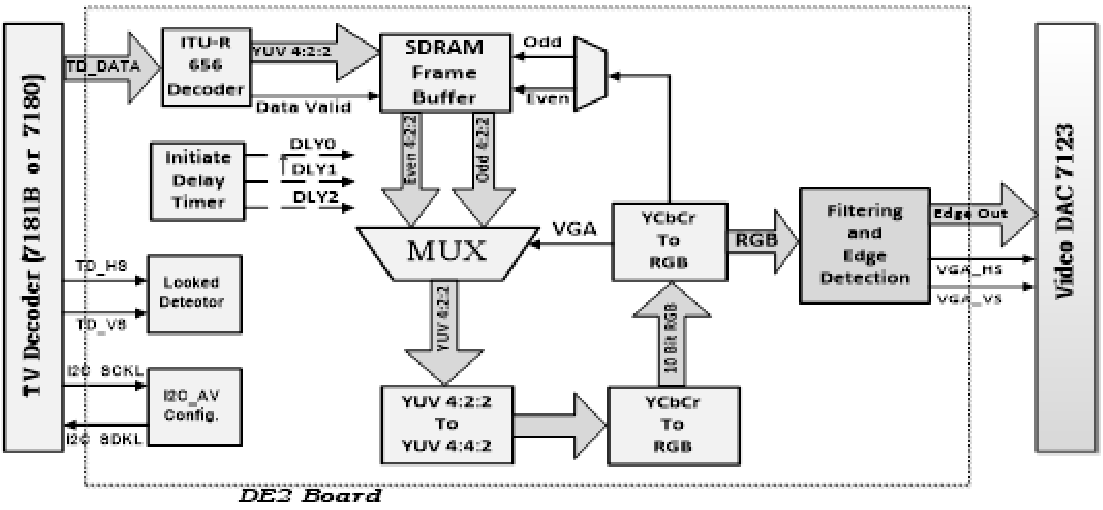


Figure 5. Block diagram of a lane detection

Table 2. Control Edge Detections in DE2 Cyclone II

Input key/switch	Action
Key 0	Reset
Switch 0	Toggle between gray scale image and edge detected image.
Switch 1	Turn on filtering.
Switch 2 to Switch 17	Adjust edge detection threshold.

4. Result

In this section the lane edge detction will be performed based on the the used version of DE2 board and the camera, the results can be devided into either day or night lane detection. As mentioned earlier, the camera Genx (GDV 720 ultra) has two standards; PAL and NTSC, meanwhile Altera DE2 starting with (11xx...) supports only NTSC standard. Also, this type of camera does not support night vision; and hence, the day vision is only considered in the present scenario. Original and gray scale images are displayed in Figure 6 and Figure 7, respectively.

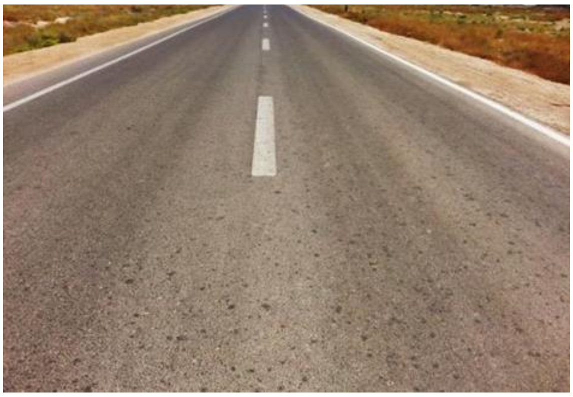


Figure 6. Original color image

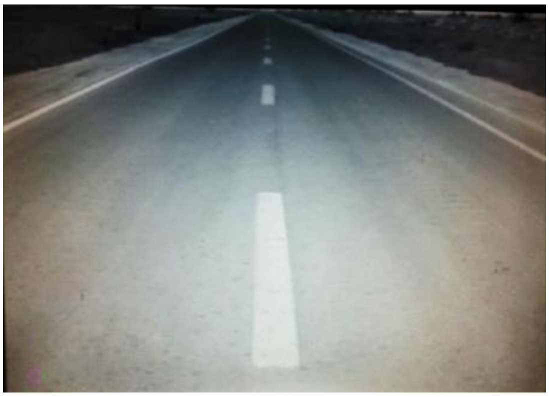


Figure 7. Gray scale image

4.1. Filtering

To reduce the noise; the gray scale image is then fed to two filters, gaussain and avarage filters. Figure 8 and Figure 9 shows the images resulting from avarage and gaussain filters, respectively. It is clear from these figures that the average filter has better filtering charcteristics than Gaussian filter. Therefore, the latter filter will be excluded and average filter will be mainly relied on for filtering purposes in all next simulations. In what follows, the filtered image, based on average filter, is allowed to be edge detected. Two types of edge detection techniques has been suggested; first and second order techniques.

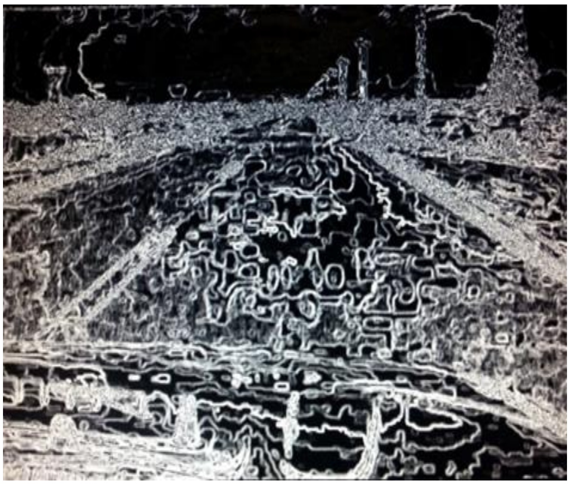


Figure 8. Gaussian filter with standard deviation ($\sigma = 1$)

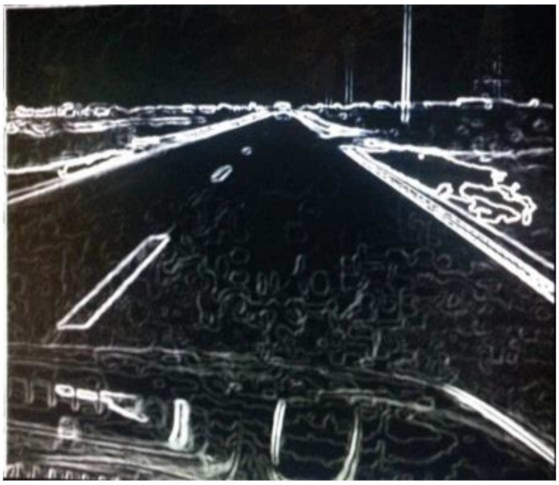


Figure 9. Average filter

4.2. First-order Edge Detection

For the first order type, two techniques has been selected, Sobel and Robinson; while for the other type, Laplacian and Laplacian of Gaussian (LoG) has been chosen. Figure 10 and Figure 11 shows the images coming out from Sobel and Robinson techniques, respectively. One can easily see that more details would appear in image resulting from Robinson than that resulting from Sobel method. This may be attributed to eight mask operator which supports Robinson technique and four mask operator which support the other method.

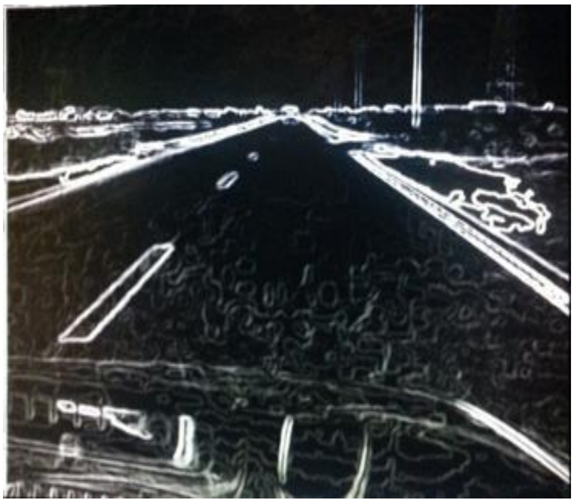


Figure 10. Sobel edge detection

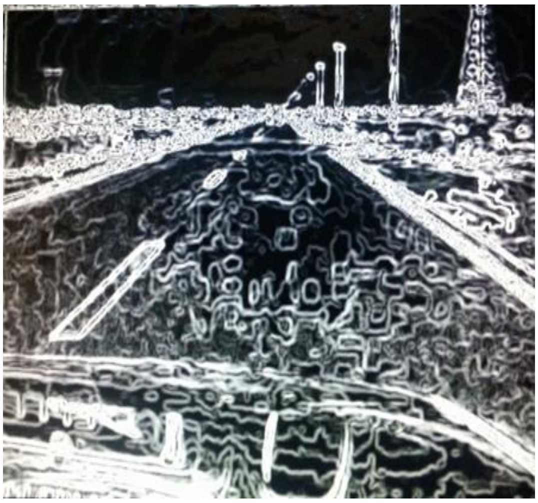


Figure 11. Robinson edge detection

4.3. Second-order Edge Detection

Next, the second order type has been considered. The edge detected images from Laplacian and LoG is illustrated in Figure 12 and Figure 13, respectively. It is well-known that this second order type could not give better results since they are based on detection of edge location from its zero-crossing property and hence such type of edge detction techniques would be sensetive to noise. However, it is clear from these figures that LoG gives better edge detection features than its counterpart. The conclusion drawn from the above argument is that Sobel edge detection technique could give the best edge detection characteristics and it outperforms the others.

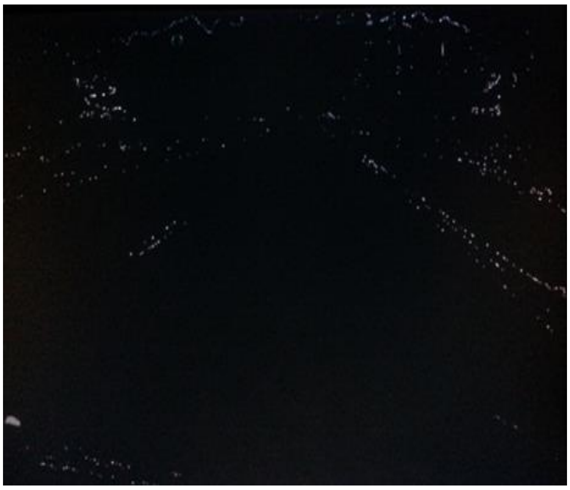


Figure 12. Lapacian edge detection

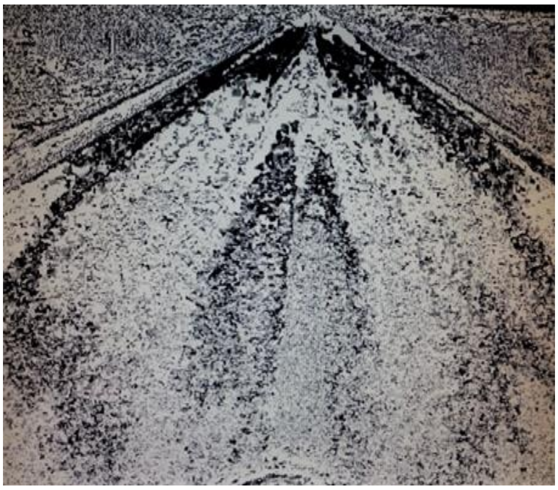


Figure 13. LoG edge detection

4.4. Thresholding

It is interesting to examine the effectiveness of the considered techniques against the change of threshold level. Three levels have been taken into account; low, medium and high. These three levels has been practically set-up and designated using the switches implanted on DE2 board; such that the level selection is performed manually. Table 3 shows the bands of considered levels for both Sobel and Robinson. It is worthy to notify that the level band and value changes according to edge detection technique, environment, type of camera, board version and even on orientation angle of camera. For the present scenario, the low, medium and high level values for Sobel edge detection technique are designated as "SW [3]", "SW [5]" and "SW [7]", respectively. On the other hand, the low, medium and high-level values for Robinson edge detection technique are designated as "SW [5]", "SW [8]" and "SW [10]", respectively. The switch SW [0] is preserved for admitting either edge detection or gray scale image and the switch SW [1] is dedicated for switching the filtering process. Figures 14-25 show the images after detection with both first order techniques and with different three threshold levels.

Table 3. Mechanism of Adjusting the Threshold of Edge Detection

Type edge detection	Low level switches	Medium level switches	High level switches
Sobel	SW [2] to SW [3]	SW [4] to SW [5]	SW [6] to SW [7]
Robinson	SW [2] to SW [5]	SW [6] to SW [8]	SW [9] to SW [10]

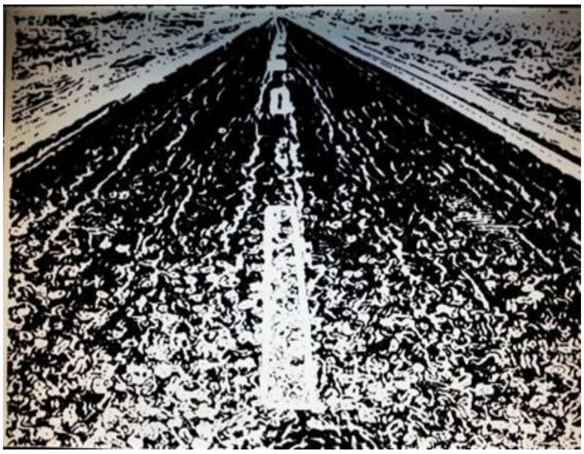


Figure 14. Sobel technique
(low level)

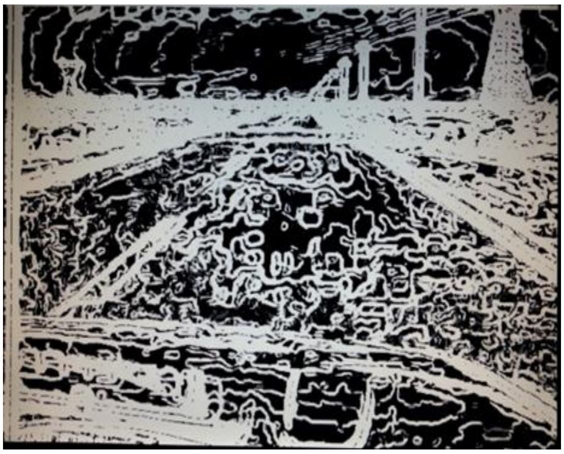


Figure 15. Robinson technique
(low level)

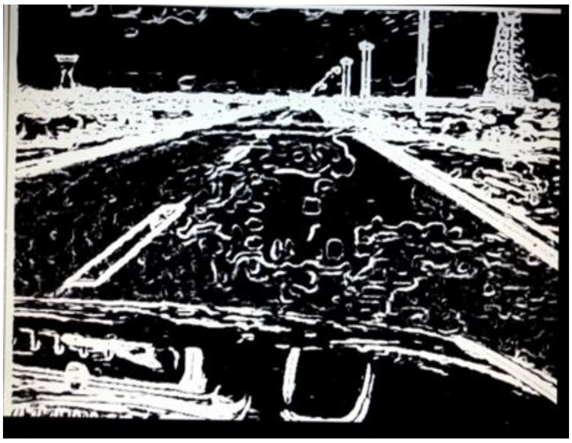


Figure 16. Robinson technique
(medium level)

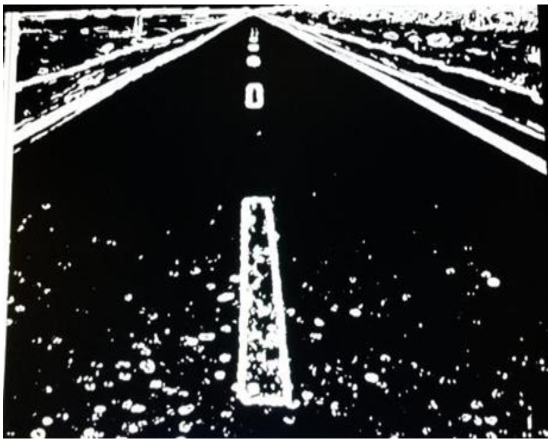


Figure 17. Sobel technique
(medium level)

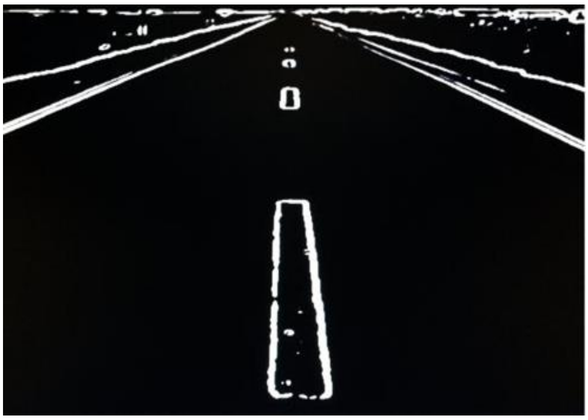


Figure 18. Robinson technique
(high level)

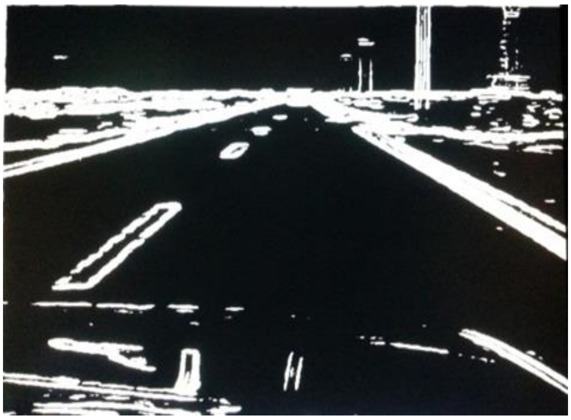


Figure 19. Sobel technique
(high level)

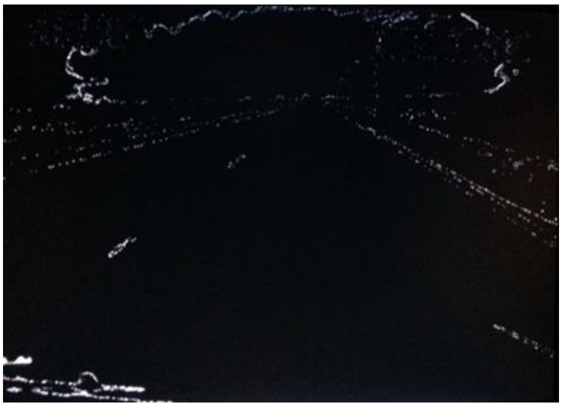


Figure 20. Laplacian technique
(low level)

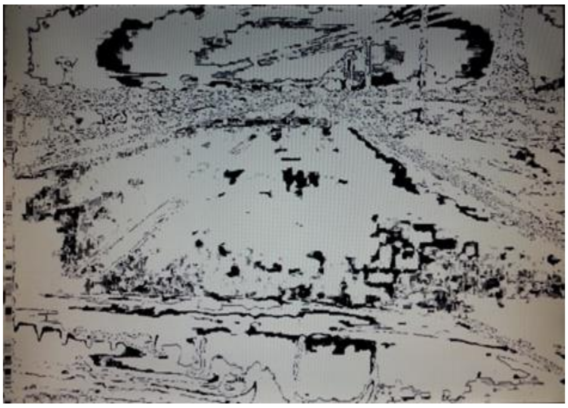


Figure 21. LoG technique
(medium level)

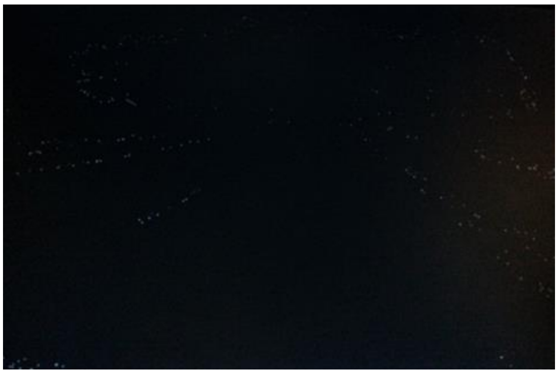


Figure 22. Laplacian technique
(medium level)

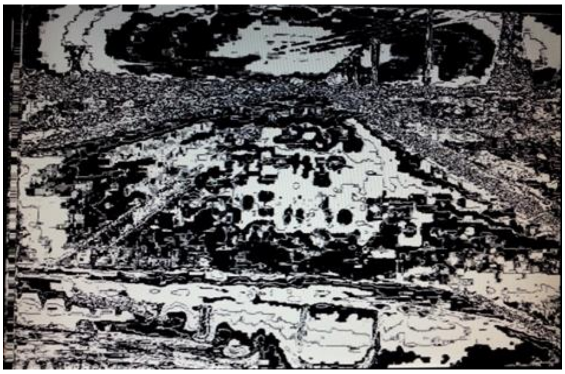


Figure 23. LoG technique
(medium level)

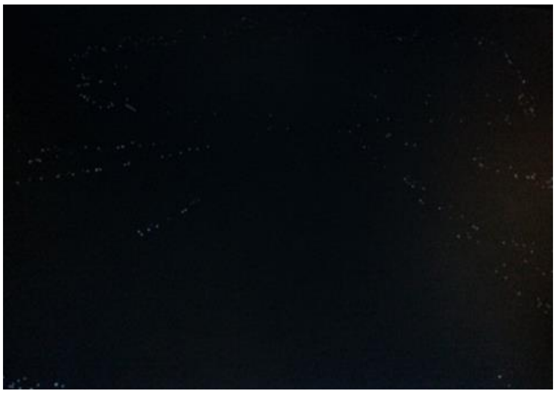


Figure 24. Laplacian technique
(high level)

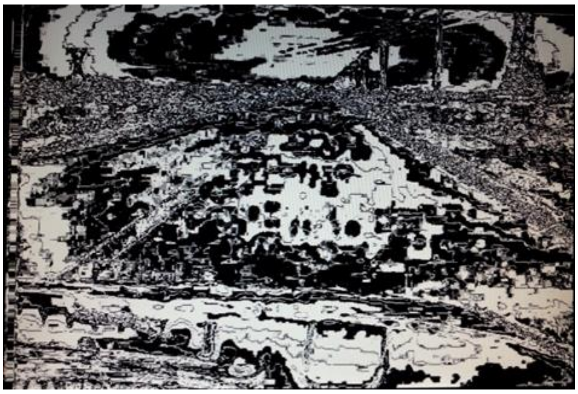


Figure 25. LoG technique
(high level)

The conclusion drawn from this observation is that increasing the threshold level would better enhance the edge detected image. The reason behind this enhancement is that raising the threshold level would discard the weak features of the image with low intensity. Table 4 displays the flow summary report that appeared after Quartus II 32-bit version 11.1 succeeding in compilation.

It is clear that Laplacian technique wins the minimum numbers of logic elements. However, one may deduce from the table that the number of logic elements depends on number of masks, their size and the strategy of edge detection computing; where the Sobel technique has 2 masks with 3×3 size, Robinson has 8 masks with 3×3 size, Laplacian has 1 mask with 3×3 size and LoG has 1 mask with 5×5 size. As such, LoG has maximum number of logic elements due to having maximum number of masks with large mask size which in turn reserve higher number of line buffers. It is worthy to notify that the total number of logic elements fabricated inside FPGA under study is 33,216 total logic elements and Sobel technique, e.g., occupy 9% of the total logic elements. It is interesting to investigate the change of table summary after thresholding process. Table 5 lists the information about number of logic elements, combination functions and dedicated logic registers for all considered edge detection techniques. However, due to deleting and adding instructions at the thresholding process, there is a slight change in the number of logic elements depending on the number of these elements required modified instructions.

Table 4. Summary of Resources for DE2 Start with-11

Edge detection methods	Total logic elements	Total combinational functions	Dedicated logic registers
Sobel edge detection	1989	1611	1209
Robinson edge detection	2656	2252	1161
Laplacian edge detection	1947	1583	1221
LoG edge detection	3860	3408	1367

Table 5. Summary of Resources after Thresholding Process

Edge detection methods	Total logic elements	Total combinational functions	Dedicated logic registers
Sobel edge detection	1983	1609	1163
Robinson edge detection	2617	2262	1161
Laplacian edge detection	1974	1628	1209
LoG edge detection	3833	3380	1281

5. Conclusions

Based on the observations of the results, one may draw the following main conclusions. For the sake of clarity, the conclusion points can be classified according to lane detection and lane tracking scenarios. Two filters have been suggested for lane detection part; average filter and Gaussian filter. The results of present work have shown that the average filter has much better filtering characteristics than Gaussian one and for all considered situations. For our case, raising the threshold level successfully enhanced the image resulting from most edge detection techniques. This indicates that all unimportant image features, clutter and noise reside at low level intensity. The results have shown that Sobel edge detector has the best edge detection characteristics among the others. Also, the threshold level raising has better enhancement for the image based on Sobel technique than that resulting from other techniques. For current work, second order edge detection techniques failed for performing edge detection in most scenarios and for different threshold level.The number of logic elements occupied by lane detection process relies on the number of masks and mask size for specified edge detection algorithm. In terms of minimum number of logic elements, Laplacian technique comes the first, while Sobel has the second order. Due to failure of Laplacian detector in most situations, Sobel become the first candidate edge detector.

References

[1] Roberts, Brook, et al. A Dataset for Lane Instance Segmentation in Urban Environments. arXiv preprint arXiv:1807.01347 (2018).

[2] Cao, Jiale, Yanwei Pang, Xuelong Li. Learning multilayer channel features for pedestrian detection. *IEEE transactions on image processing*. 2017; 26(7): 3210-3220.

[3] Mohammed Abdulraheem Fadhel. Design and Implementation of a Lane Detection and Tracking System Using FPGA, M.Sc. Thesis, University of Technology, Baghdad, Iraq, 2015.

[4] Humaidi, Amjad Jaleel, and Mohammed Abdulraheem Fadhel. *Performance comparison for lane detection and tracking with two different techniques*. Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Al-Sadeq International Conference on. IEEE, 2016.

[5] Narote, Sandipann P, et al. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*. 2018; 73: 216-234.

[6] Huang, Yuhao, et al. *Spatial-Temproal Based Lane Detection Using Deep Learning*. IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham, 2018.

[7] Russ John C. The Image Processing Handbook, Sixth Edition, North Carolina State University, CRC Press, 2011.

[8] Nixon Mark S. and Aguado Alberto S. Feature Extraction & Image Processing for Computer Vision. Third edition, Oxford, Newnes, 2012.

[9] Gonzalez Rafael C, Woods Richard E. Digital image processing, Third Edition, Prentice-Hall, 2008.

[10] Acharya Tinku, Ray Ajoy K. Image Processing Principles and Applications. John Wiley & Sons, 2005.

[11] Huiyu Zhou, Jiahua Wu, Jianguo Zhang. Digital image processing–Part I, 2010.

[12] Huiyu Zhou, Jiahua Wu, Jianguo Zhang. Digital image processing–Part II, 2010.

[13] User Manual of Altera DE2 Board, Version 1.5, 2012.

[14] Maini, Raman, Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP.)* 2009; (3)1: 1-11.

[15] Reddy R, Vijaya Kumar, et al. Comparative Analysis of common Edge Detection Algorithms using Pre-processing Technique*. International Journal of Electrical and Computer Engineering*. 2017; 7(5): 2574-2580.

[16] Kardian, Aqwam Rosadi, Sunny Arief Sudiro, Sarifuddin Madenda. Efficient Implementation of Mean, Variance and Skewness Statistic Formula for Image Processing Using FPGA Device. *Bulletin of Electrical Engineering and Informatics.* 2018; 7(3): 386-392.

[17] Israni, Sheetal, Swapnil Jain. *Edge detection of license plate using Sobel operator*. Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016.

[18] Parikh, Parth V, Bhinjan A, Dalwadi, Ghanshyam D, Zambare. Sobel edge detection using FPGA-Artix®-7. *International Journal of Scientific & Technology Research*. 2016; 5(6): 250-253.

[19] El Hajjouji I, et al. *FPGA based real-time lane detection and tracking implementation*. Electrical and Information Technologies (ICEIT), 2016 International Conference on. IEEE, 2016.

[20] Narote, Sandipann P, et al. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*. 2018; 73: 216-234.

[21] Baili, Jamel, et al. *Lane departure detection using image processing techniques*. Anti-Cyber Crimes (ICACC), 2017 2nd International Conference on. IEEE, 2017.

[22] Mukhtar, Amir, Likun Xia, Tong Boon Tang. Vehicle Detection Techniques for Collision Avoidance Systems: A Review. *IEEE Trans. Intelligent Transportation Systems*. 2015; 16(5): 2318-2338.

[23] Wen, Xuezhi, et al. Efficient Feature Selection and Classification for Vehicle Detection. *IEEE Trans. Circuits Syst.* Video Techn. 2015; 5(3): 508-517.

[24] Hadi, Raad Ahmed, Ghazali Sulong, Loay Edwar George. Vehicle detection and tracking techniques: a concise review. arXiv preprint arXiv:1410.5894 (2014).

[25] Al-Zubaidi, Laith. Deep Learning Based Nuclei Detection for Quantitative Histopathology Image Analysis. Diss. University of Missouri-Columbia, 2016.