# **BUILT-IN-SELF-TEST SYSTEM FOR FPGA-BASED VEHICLE VIDEO-DETECTION**

#### AND DISTANCE MEASUREMENT

By

#### Md Forhad Ebn Anwar

Bachelor of Science, Electrical and Electronics Engineering

Ahsanullah University of Science and Technology, 2004

A Project Report

presented to Ryerson University

In partial fulfilment of the

requirements for the degree of

Master of Engineering

in the program of

**Electrical and Computer Engineering** 

Toronto, Ontario, Canada, 2018

© Md Forhad Ebn Anwar, 2018

# Author's Declaration

I hereby declare that I am the sole author of this project report. This is a true copy of the project report, including any required final revisions.

I authorize Ryerson University to lend this project report to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this project report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my project work may be made electronically available to the public.

# BUILT-IN-SELF-TEST SYSTEM FOR FPGA-BASED VEHICLE VIDEO-DETECTION AND DISTANCE MEASUREMENT

Md Forhad Ebn Anwar

Master of Engineering, Electrical and Computer Engineering 2018

#### Abstract

Collision of vehicles in highways are very frequent. Because of high speed (more than 100 km/hour), the momentum of collision is too high that leads sever casualty. Automatic Driving Assistance system can assist the vehicle operators to take decision based on realistic practical calculation on safety measures. It is always better to have third eye working parallel with human to avoid road accident. There are several technologies used to develop perfect driving assistance system to achieve higher accuracy in detection, identification and distance measurement of obstacles where vision based system is one of them. Mono-vision system provides cheap and fast solution rather stereo vision. This project work conducted with objective to comprehend computational complexity in implementation of mono-vison camera based object detection where system will generate warning if the detected object has a motion towards target. Processing and analyzing of captured video image is the focused mechanism of implementation and used internal image generator module to mimic actual video camera. Appeared size of the shape of object considered for the decision making. The simulated image pattern can change it's dimension to represent vehicle movement in one direction (Back and forth).

In this work the on-chip car image generation sub-system was proposed designed and partially implemented on the base of the FPGA where Xilinx Zynq-7010 (ZYNQ XC7Z010-1CLG400C) FPGA development board used.

**Keyword**: Computer Vision, mono vision, image processing on FPGA, Automatic Driving Assistance, Vehicle Detection.

iii

# **Table of Contents**

Author's Declarationii
Abstractiii
List of Figuresv
Chapter I: Introduction1
a. Objective
Chapter II: Related Works
Chapter III: Specification and Top Level Design7
Chapter IV: Implementation Plan8
a. Video Clock Generator8
b. VGA Controller10
c. Image Pattern Generator11
d. RGB to Grayscale Converter12
e. Edge Detector and Object pixel counter12
f. Comparator14
Chapter V: Hardware Requirement15
Chapter VI: Implementation16
Chapter VII: Result18
Chapter VIII: Conclusion27
References

# List of Figures

Figure A: Top Level Block Design	7
Figure 1: A frequency divider using D Flip flop	9
Figure 2: Pixel Clock Generator	9
Figure 3: Video timing specifications to display image on Monitor	10
Figure 4: VGA Controller Module	11
Figure 5: Image pattern Generator	12
Figure 6: Sobel Mask in X direction	13
Figure 7: Sobel Mask in Y direction	13
Figure 8: Sobel Mask Convolution	13
Figure 9: Zybo Project Board	15
Figure 10: Video timing specifications to display image on Monitor	16
Figure11: Components Block Diagram	17
Figure 12: Output of the Video frame generator	18
Figure 13a: Schematic of RTL design	20
Figure 13b: Implemented Design View	21
Figure 14: Power consumption of the circuit on chip	22
Figure 15: Power consumption of the circuit summary	23
Figure 16: Utilization of the hardware resources	23
Figure 17: Utilization of the hardware resources in table	24
Figure 18: Utilization of post synthesis in a graph	24
Figure 19: Utilization post synthesis in a table	25
Figure 20: Output of ILA Data for various signals	26

#### I. Introduction

People have used highways, a fast and convenient form of road transportation. By using highways, people can save transportation time and live a more convenient life. However, drivers face greater risks to their life in situations where an instant reaction time is necessary, especially when driving at high speed on highways, which may cause more severe injuries than accidents on general roads such as urban roads. Damage causes often more severe since all vehicles are driving at high speeds. The causes of accidents are mainly related to drivers who did not keep a safe distance from the front vehicle. Sensor based and vision based technologies are used to develop Assistive Driving System [1]. Sensor based system use sensors like RADAR, LIDAR while vision based system use computer vision technologies using camera sensors to detect vehicle and measure distance. RADAR- and LIDAR-based systems may have difficulty distinguishing the objects directly ahead. However, the vision-based system can detect and recognize front objects by analyzing images from cameras mounted inside vehicles [1]. Vision based system is more realistic to identify objects than other sensor based system. Visionbased systems are usually split into two classes: the monocular vision-based system and the stereo vision-based system. Stereo vision based system can provide distance measurement using 3D vision while mono vision based system also can provide distance using pre- defined scale. However, mono vison system is low cost and high-speed solution compared to stereo vision. Vehicle detection for a vision-based driver assistance system requires the analysis of video images using image processing algorithms to isolate and track moving vehicles in the video sequence. Monocular vision based vehicle detection systems are particularly interesting for their low cost and for the high-fidelity information they give about the driver environment. Monocular vision-based systems provide a more efficient performance than stereo vision-based systems because it can tolerate a complicated algorithm and thus provide better results in less time compared to the stereo vision-based systems with the same algorithm. It is also easier to obtain and embed the vision-based system using a monocular camera compared to a stereo camera.

1

A mimic of real-time mono vision based object detection and behavior observation system is implemented in this project. Objective is to obtain perfect decision applying various image processing methods on the captured image. To focus on image processing and to reduce complexity, a image sequence generator is used in place of real camera. The image sequence is the replica of actual captured image of vehicles by a single camera. All the image processing mechanism is applied on the generated image to predict the object behaviour.

Three major sections will be found next, "Related Works", "Implementation plan" and "Implementation" which are demonstrated in sub sections.

# I (a) Objective

This is to implement a simulation of real time video capture. To generate video sequence with vehicle alike pattern and mimic some actual behavior of vehicle movement. Measuring distance of front vehicle from the appeared dimension of the vehicle which similar to using a mono vision camera. Because of mono vision, 3D vision is not available and only possible method of measuring distance is dimension of object and compare with predefined scale.

# l (b) Task

Task consists of following functions

- Pixel Clock Generation: This is to display the 640X480 pixel @ 60 frames per second. Including blanking time (includes sync pulse, front porch, back porch), it is 800X525 Pixels @ 60 fps and for this we need pixel clock frequency of 25.2 MHz. The FPGA board can provide 50MHz clock from which it need to derive the required frequency clock.
- 2. Display control signal generation: Based on pixel clock, vertical and horizontal signal need to generate which will operate the display monitor. Also need generate row, column and display enable signals for the image pattern generator.
- 3. Image pattern generation: This is to generate specific pattern as replacement of real vehicle image. Also included image control to make effect of depth in one direction, only back and forth.

#### **II Related Works**

Histogram Oriented Gradients (HOG) feature and Support Vector Machine (SVM) method proposed by Xing Li *et al.* to detect vehicle using single camera. Shadow underneath the vehicle during daytime helps determining vehicle existence on the road which utilized to detect forward vehicle [2]. Initial candidate of preceding vehicles is determined by computing edges of shadow underneath the vehicles. Non- vehicle objects like trees, fence, pavements etc. are filtered out using HOG and SVM methods where candidates were compared with sample objects. To make system adaptive to different weather and light condition, statistic histogram analysis is used to segment the shadow underneath the vehicle [2]. The threshold of histogram analysis to segment the shadow is adaptive to weather as intensity of darkness under vehicle varies with surround illumination.

Ritesh *et al.* proposed a three-step vehicle detection framework to detect and track the target vehicle within an image. Hypothesis Generation, Hypothesis Verification and Algorithm Optimization are the three steps to detect and track a leading vehicle [3]. Brightness control and pattern matching is used to select preliminary candidate for target vehicles from the captured image by single camera on board. The normalised cross correlation (NCC) method used to match with pre- defined template of vehicles to filter out non- vehicle objects in Hypothesis Generation step [3]. In Hypothesis Verification, a Region of Interest is set using lane detection method where candidates discarded which are not within same lane. Adaptive Image Cropping (AIC) is implemented as third step Algorithm Optimization. The ROI shrunk to focus on only target found in previous Hypothesis which helped increase image processing frequency. Perimeter of ROI depends on target size and localization [3].

A vision based technique is applied to develop Forward Collision Avoidance Assist System (FCAAS) by Liang-Chien Liu *et al.* using a monocular camera [4]. A warning signal generates based on the distance measured between own and forward vehicle. System consists of two major process on captured image frame from video sequence; Lane marking detection, Vehicle detection and tracking. Lane marking used as reference

4

to set region of interest to select potential candidate of vehicle. Vehicles are tracked and distance measured which are detected within the lane marking. Adaboost classifier used to determine detected object as vehicles [4]. Lane marking based detection system only applicable in Highways where marking available. So, this system not suitable for local roads.

Manuel Ibarra Arenado *et al* demonstrated method of vehicle identification and relative distance measurement methods using single camera. Cost efficiency is the major key factor of implementation of mono vision system where stereo vision can provide higher accuracy [7]. Vehicle identification made easy by tracing number plate. Considered standard format of number plate [7].

Detection of preceding vehicle by locating taillight during night time is the objective of the paper published by Ying and Hsuan. It also measures the relative distance between test car and the target vehicle to generate warning as collision avoidance process. This method can detect preceding vehicles during night time by identifying two red taillights [8]. Identifying individual vehicle based on tail light only bit challenging and was done using "Taillight Clustering" method [8].

Chao-Ho Chen *et al* determination of Region of Interest is achieved using a different method in the paper while vehicle detection done based on "Shadow under vehicle". "Vanishing point" is the objective point in the image which sets the maximum limit of processing and a reference point f distance measurement. 78% of front vehicle can be detected using this method as per result of practical experiment [9].

Detection of vehicle and realize the behaviour is the crucial part. Several methods applied in all above papers to determine vehicle identity, differentiate target vehicle by defining region of interest, tracking vehicles to realize safety factors based on distance between frontal tracked vehicle and own vehicle.

Vehicle maintains some mandatory standards like shape of body, color of lighting, size of the number plate, dimension of specific group of vehicles. These standard parameters help detecting object as vehicle and measure distance based on behavioral change of these physical elements. Like, if the on the sequence of captured image of vehicle shows that the tail lights getting closer, it can be concluded that the distance between the cars decreasing.

To simulate specific behavior related to depth between two cars, it is generated sequence image pattern which changes its dimension to represent real vehicle movement at front of test vehicle.

# III Specification and Top-Level Design

- 1. Object is small vehicle
- 2. Detect front vehicle only based on dimension, in this case dimension of the image pattern
- **3.** The pattern will change its size from small to large and large to small which will represent the distance. Smaller size means higher distance.

The following block diagram in Figure A shows the required external inputs and outputs of the image generator. The width of the color output is based on Zybo board which have 16-bit VGA port. Following are the functionality of the pins,

Sys\_clk: (Input) takes system clock signal from the board as input.

Rst: (Input) Hard rest works at active "0".

Image\_ctl: (Input) controls the dimension of the image object to represent vehicle distance.

V\_sync, H\_sync: (Output) Synchronization signal for display monitor

Red, Green, Blue: (Output) Digital color output signal for the monitor to display the generated image.



Figure A: Top level Block Design – Image generator

## **IV. Implementation Plan**

The system consists of Video Clock Generator, VGA Controller, Image pattern generator, RGB to Binary Converter, Memory Block, Object Edge Detector, Warning Generator and VGA Monitor. Input clock speed is 50 MHz which will be used for all computation in all modules except for video output and video generator. It is expected to accomplish all task in between generation of two consecutive video frame to generate real time results.

Design and implementation of preliminary components and combining them to achieve full functional circuit, all done on the Xilinx platform using VIVADO 2016.3 tools.

Objective is to implement the circuit completely on hardware platform to achieve low latency and low power solution.

## IV (a) Video Clock Generator

The VGA standard displays video of 640x480 resolution. To display the captured image sequence on the monitor, need define specific clock frequency for this video standard. This is called "Pixel Frequency". Pixel frequency is calculated from number of pixels per frame and number of frames per second. Number of pixels includes color signals and synchronization signal [5]. A frequency divider is required to obtain 25.175 MHz pixel clock from the 50 MHz system clock to drive 640x480 standard @ 60 frame per second [5]. Frequency divider is generated using counter which will cut the system frequency in half. Frequency divider made using a D Flip flop by connecting the  $\overline{Q}$  output to the D input as feedback. And it holds clock input for one clock cycle and thus output Q. Figure 1 shows a frequency divider using D Flip Flop.



Figure 1: A frequency divider using D Flip flop

The component "Pixel\_clk" designed to provide VGA Controller the required clock signal. Figure 2 shows the pixel\_clk component block.



Figure 2: Pixel Clock Generator

# IV (b) VGA Controller

This is the module which generates all signals required for video display and to generate video frame sequence with specific pattern.

Figure 3 contains the table of video timing specifications based on which VGA controller designed.

Resolutio	on (pixels)	640x480
Refresh	Rate (Hz)	60
Pixel Clo	ock (MHz)	25.2
	Display	640
Harizontal (nivel clocks)	Front Porch	16
	Sync Pulse	96
	Back Porch	48
Тс	otal	800
	Display	480
Vertical (rows)	Front Porch	10
venical (10w3)	Sync Pulse	2
	Back Porch	33
Тс	otal	525
h_sync	Polarity	n
v_sync	Polarity	n

Figure 3: Video timing specifications to display image on Monitor



The block module of the VGA controller showing in the figure 4

Figure 4 : VGA Controller Module

### IV (c) Image Pattern Generator

To mimic a real camera, it is used a pattern generator which will generate sequence of image frame with a defined pattern and will deliver color signals to the VGA output. Based on received signal from VGA controller it generates RGB color signals. It will generate a red rectangle on black background with the signal of row-column input. "disp\_ena" will allow to start generating the pattern. The Zybo board have a 16-bit VGA out port. Red and Blue 5 bits and Green have 6 bits. So, had to design the RGB out put signal for 16 bits. To change the size of the pattern, an external switch is used. When switch set to "0", size of image will increase in consequent frames up to maximum size. This represents approaching object and vice versa. The block module of image generator is shown in the Figure 5.



Figure 5: Image pattern Generator

# IV (d) RGB to Grayscale Converter

This is always easy to process monochrome image than color image in terms of computational complexity, time requirement and storing. This module is dedicated to convert 16-bit RGB image to 8-bit Grayscale image. Grayscale represents each pixel with 8 bits starting from 00 to FF intensity. RGB to grayscale image can be derived using line projection described as [6]

I = arR + agG + abB

Where non-negative co-efficient ar, ag, ab satisfies

$$ar + ag + ab = 1$$

Different choices available for the coefficient value but for this module ( $\alpha r$ ,  $\alpha g$ ,  $\alpha b$ ) = (0.3, 0.59, 0.11) is used.

# IV (e) Edge Detector and Object pixel counter

Sobel Mask algorithm is applied to detect object. Sobel operator calculates the gradient of image for each pixel. It uses a 3x3 kernel to roll over the image matrix to find sharp intensity difference of each pixel with neighbour pixels along X and Y direction to find 2-D edge of objects in the image. Function used to calculate gradient as follows

 $G=\sqrt{( [G_X] ^2 + [G_Y] ^2) \dots}$  (i)

Where, G is the magnitude of the gradient,  $G_X$  is the gradient along X direction and  $G_Y$  is the gradient along Y direction.

Figure 6 and Figure 7 shows the Sobel mask matrix in both X and Y direction

 $\begin{array}{cccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array}$ 

## Figure 6: Sobel Mask in X direction

$^{-1}$	0	1
-2	0	2
$^{-1}$	0	1

Figure 7: Sobel Mask in Y direction

Consider a Matrix 4x4 size and sobel mask to be convolution with the shaded part of the matrix.

P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P4
P5	P <sub>6</sub>	P7	P٥
P <sub>9</sub>	P <sub>10</sub>	P <sub>11</sub>	P12
P13	P14	P15	P16

Figure 8: Sobel Mask Convolution

It can calculate gradients in X and Y direction using following function.

 $G_X = (P_9 + 2^* P_{10} + P_{11}) - (P_1 + 2^* P_2 + P_3)$  $G_Y = (P_3 + 2^* P_7 + P_{11}) - (P_1 + 2^* P_5 + P_9)$ 

So, objective function of this project is

 $G = \sqrt{[(P_9+2*P_{10}+P_{11}) - (P_1+2*P_2+P_3)]^2 + [(P_3+2*P_7+P_{11}) - (P_1+2*P_5+P_9)]^2]}$ 

# IV (f) Comparator

This is the decision maker, identify approaching object or departing object comparing the object pixels count each image frame. Near object will have bigger appearance than far object and hence pixel count will be higher for nearer position of object. Two output signal denotes objects appearance. It will receive two stored pixel count corresponding to current frame and previous frame from memory stake and compare them. If number of pixels in current frame is higher than the number of pixels from previous frame; system will decide that front object is approaching and will provide result accordingly, in this case a "Red LED".

# V Hardware Requirement

Have used Xilinx Zynq-7010 (ZYNQ XC7Z010-1CLG400C) development board in this project. Zybo Zynq-7000 ARM/FPGA SoC Trainer Board is a feature-rich, ready-to-use, entry-level embedded software and digital circuit development platform built around the smallest member of the Xilinx Zynq-7000 family, the Z-7010. The Z-7010 is based on the Xilinx® All Programmable System-on-Chip (AP SoC) architecture, which tightly integrates a dual-core ARM Cortex-A9 processor with Xilinx 7-series field programmable gate array (FPGA) logic. A computer monitor having VGA input connecting to the zybo board made the system ready to implement. Figure 9 shows the project board with a VGA camera attached.



Figure 9: Zybo Project Board

# **VI Implementation**

# Specifications (Constraints of implementation)

This is a replica of actual captured video frame implementation and hence considered some straight forward specifications for the object behaviour.

Shape of Object is known.

Object is clearly visible

No consideration of environmental factors

No object identification verification

Decision based on only physical dimension of object

Figure 10 contains the table of video timing specifications based on which VGA controller designed.

Resolutio	on (pixels)	640x480
Refresh	Rate (Hz)	60
Pixel Clo	ock (MHz)	25.2
	Display	640
Horizontal (pixel	Front Porch	16
clocks)	Sync Pulse	96
	Back Porch	48
Тс	otal	800
	Display	480
Vertical (rows)	Front Porch	10
vertical (IOWS)	Sync Pulse	2
	Back Porch	33
Тс	otal	525
h_sync	Polarity	n
v_sync	Polarity	n

Figure 10: Video timing specifications to display image on Monitor

Figure 11 Shows the complete block design of the implementation with inter connections.



17

### Result

Implementation done up to Image pattern generation and display on monitor. Chipscope Pro integrated and left in the in the circuit.

Achieved the video signals properly as expected for 640X480 Resulation @ 60Hz and display shows as per in figure 12.

It is a Blue and red combinational pattern.

Because of the Monitor display area, it displayed four of the generated pattern on the display.



Figure 12: Output of the Video frame generator

ILA	Depth	Assigned Signal	Expected Value
Probe			
probe0	1	System Clock	50 Mhz
probe1	1	Reset	"0" when in operation
probe2	1	h_sync	"1" when in Display
probe3	1	v_sync	"1" When in display
probe4	5	Intensity of Red Pixel	"10101"
probe5	6	Intensity of Green Pixel	"101010"
probe6	5	Intensity of Blue Pixel	"10101"
probe7	1	Pixel Clock	25 Mhz
probe8	10	pixel_h	Count between 0 to 799
probe9	9	pixel_v	Count between 0 to 524
probe10	1	disp_ena	"1" when in Display
probe11	8	row	Value between 0 to 479
probe12	9	column	Value between 0 to 639

# ILA Probe Outputs:

# Hardware implementation reports (Clock generator to Image Generator)



Figure 13a: Schematic of RTL design

Figure 13a shows the RTL schematic. Figure 13b Shows the implemented Design view of the board



Figure 13b: Implemented Design View

Figure 14 shows the power usage by the circuit on chip after implementation on Xilinx board.



Figure 14: Power consumption of the circuit on chip

Figure 15 shows summary power consumption which is significantly low 12.4mw. Power consumption mainly depends on clock frequency and we used only 50MHz as main clock. Also, temperature very low (near to room temperature). Because it is relatively small system, designed on low frequency (as per requirement from specification), power consumption of the system is negligible.

Junction Temperature:26.4 °CThermal Margin:58.6 °C (5.0 W)Effective & JA:11.5 °C/WPower supplied to off-chip devices0 WConfidence level:MediumImplemented Power Report	Total On-Chip Power:	0.124 W
Thermal Margin:       58.6 °C (5.0 W)         Effective & JA:       11.5 °C/W         Power supplied to off-chip devices:       0 W         Confidence level:       Medium         Implemented Power Report       Keling	Junction Temperature:	26.4 °C
Effective dJA: 11.5 °C/W Power supplied to off-chip devices: 0 W Confidence level: <u>Medium</u> <u>Implemented Power Report</u>	Thermal Margin:	58.6 °C (5.0 W)
Power supplied to off-chip devices: 0 W Confidence level: <u>Medium</u> Implemented Power Report	Effective dJA:	11.5 °C/W
Confidence level: <u>Medium</u> <u>Implemented Power Report</u>	Power supplied to off-chip devices:	0 W 0
Implemented Power Report	Confidence level:	Medium
	Implemented Power Report	

Figure 15: Power consumption of the circuit summary

Figure 16 Shows the utilization of the FPGA hardware resource graphically



Figure 16: Utilization of the hardware resources

Figure 17 Shows the utilization of the FPGA hardware resource in a table

LUT         1566         17600           LUTRAM         197         6000           FF         2178         35200	8.9 3.2
LUTRAM 197 6000 FF 2178 35200	3.2
FF 2178 35200	
	6.1
BRAM 2 60	3.3
IO 20 100	20.0
BUFG 2 32	6.2

Figure 17: Utilization of the hardware resources in table

Figure 18 Shows the utilization post synthesis in a graph



Figure 18: Utilization of post synthesis in a graph

Figure 18 Shows the utilization post synthesis in a table. LUT utilization is less than 1% as per Post Synthesis report.

шт		Available	Utilization %
101	127	17600	0.72
FF	25	35200	0.07
IO	20	100	20.00
BUFG	1	32	3.13

Figure 19: Utilization post synthesis in a table

Figure 20 shows the output dashboard of chipscope. Video signals and the image signals are as per spec except for the color bits.

1570	1520 1525	IS70 IS75 IS80	570  575  580  585	1651 522 085 1530 0651 1650
	\$70      \$75			570 575 580 585 589 589
- ŏ - - -	1 1 1 575	10 1 1 1 575 1 1 1 580 1 1 1 1		589 589 575 575 580 585 585 589 589
	575	575	575	575 580 585 589 575 50 580 585 50 589

Figure 20: Output of ILA Data for various signals

# Conclusion

The image pattern generation is done in this project with the image control which controls the size of the image to represent vehicle distance. Based on the control signal (Approaching or going far), the image changes it's dimension from small to maximum size from which it can determine the depth of the vehicle by comparing with predefined scale in Look up table.

Methods demonstrated in the papers including the implemented one are suitable for slow speed traffic region where vehicle speed remains within 50 Kmh. And mostly preceding vehicle tracking where relative distance remains almost constant for most of the time. Moving object detection is more difficult in urban or sub urban area where roads are not evenly asphalted. On the other hand, in highway, smooth asphalt helps detect vehicle when vehicle under shadow is the method of detection. Due to requirement of high speed processing of image, it is suitable to process low resolution image, however, resolution do not affect much on vehicle detection as system deals with large moving objects.

This was a simulation of pre- processing of real time vehicle detection and image processing. So, future works includes

- 1. Use real camera in place if image generator
- 2. Detect vehicle using image processing for edge detection
- 3. Determine vehicle position and measure dimension of the pixel area
- 4. Use comparator to compare measured area with predefined scale to achieve actual distance of the vehicle
- 5. Generate signal based on result of comparator as driving assistant
- 6. Consider other aspects of vehicle movement, environmental factors, identify objects other than vehicle and filter out them, set of regions of interest to reduce resource of utilization are all fall in future work which can make efficient driving assistance system.

### References

[1] Sayanan Sivaraman and Mohan Manubhai Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis", *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, VOL. 14, NO. 4, DECEMBER 2013.

[2] Xing Li and Xiaosong Guo, "A HOG Feature and SVM Based Method for Forward Vehicle Detection With Single Camera", *2013 Fifth International Conference on Intelligent Human-Machine Systems and Cybernetics*.

[3] Ritesh Kanjee, Asheer K. Bachoo and Johnson Carroll, "A Three-Step Vehicle Detection Framework for Range Estimation Using a Single Camera", 2015 IEEE Symposium Series on Computational Intelligence.

[4] Liang-Chien Liu, Chiung-Yao Fang, and Sei-Wang Chen, "A Novel Distance Estimation Method Leading a Forward Collision Avoidance Assist System for Vehicles on Highways", *IEEE Transactions on Intelligent Transportation Systems* (Volume: PP, Issue: 99).

[5] Van-Huan Tran and Xuan-Tu Tran, "An Efficient Architecture Design for VGA Monitor Controller", *Consumer Electronics, Communications and Networks* (CECNet), 2011 International Conference on.

[6] Yi Wan and Qisong Xie, "A Novel Framework for Optimal RGB to Grayscale Image Conversion", 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics.

[7] Manuel Ibarra Arenado, Juan Maria Pérez Oria, Carlos Torre-Ferrero, Luciano Alonso Rentería, "Monovision-based vehicle detection, distance and relative speed measurement in urban traffic", *IET Intelligent Transport Systems,* (Volume:8, Issue: 8), Dec 2014

[8] Ying-Che Kuo and Hsuan-Wen Chen, "Vision-based Vehicle Detection in the Nighttime", *International Symposium on Computer, Communication, Control and Automation, 2010.* 

[9] Chao-Ho Chen, Tsong-Yi Chen, Deng-Yuan Huang and Kai-Wei Feng, "Front Vehicle Detection and Distance Estimation Using Single-Lens Video Camera", *Third International Conference on Robot, Vision and Signal Processing, 2015.* 

[10] C. F. Wu, C. J. Lin and C. Y. Lee, "Applying a Functional Neurofuzzy Network to Real-Time Lane Detection and Front-Vehicle Distance Measurement" *IEEE Man, and Cybernetics, Part C: Applications and Reviews,* Vol. 42, No. 2, Oct. 2011, pp.577-589.

[11] C. Y. Fang, J. H. Liang, C. S. Lo, and S. W. Chen, "A Real-Time Visual-Based Front-Mounted Vehicle Collision Warning System", *Proc. of 2013 IEEE Computational Intelligence in Vehicles and Transportation Systems (CIVTS),* Singapore, pp.16-19, April. 2013.

[12] <u>http://lslwww.epfl.ch/pages/teaching/cours\_lsl/ca\_es/VGA.pdf;</u> Date: 20-Apr-2018

[13] Guangtao Cui, Junzheng Wang, Jing Li Robust, "Multilane detection and tracking in urban scenarios based on LIDAR and mono-vision", *IET Image Processing, May* 2013

[14] Hiroaki Iwata and Keiji Saneyoshi, "Forward Obstacle Detection System by Stereo Vision", 2012 IEEE International Conference on Robotics and Biomimetics.

[15] Jian-Gang Wang, Lubing Zhou, Yu Pan, Serin Lee, Zhiwei Song, Boon Siew Han and Vincensius Billy Saputra, "Appearance-Based Brake-Lights Recognition using Deep Learning and Vehicle Detection", *2016 IEEE Intelligent Vehicles Symposium*.