Journal of Theoretical and Applied Information Technology
15th November 2017. Vol.95. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPLEMENTATION OF A VEHICLE DETECTION SYSTEM IN THE FPGA EMBEDDED PLATFORM

[1]ATIBI MOHAMED, [2]BENRABH MOHAMED, [3]ATOUF ISSAM, [4]BOUSSAA MOHAMED, [5]BENNIS ABDELLATIF

[1,2,3,4,5] LTI Lab. Faculty of Science Ben M'sik, Hassan II University of Casablanca, Driss El Harti B.P 7955, Sidi Othmane, Casablanca, Morocco

E-mail: [1] atibi.simo@gmail.com, [2]benrabh@yahoo.fr, [3] issamatouf@yahoo.fr, [4]md.boussaa@gmail.com, [5]al_bennis@yahoo.fr

ABSTRACT

This document presents the implementation of a vehicle detection system in the FPGA platform. This system is based on two algorithms, an image processing algorithm that combines an algorithm for detecting areas of interest through the shadow of the vehicle, and an image descriptor (suspected zones) haar like features type, And another classification algorithm named artificial neural network which aims to detect the presence of the vehicles in these zones. To evaluate the results obtained, which showed that the proposed system is a fast and robust vehicle detector, a hardware implementation was performed in the FPGA embedded platform based on the NIOS II microprocessor and its input-output devices. The results of this implementation have shown that the FPGA platform has been able to maintain the performance of this system in terms of computational efficiency and speed, which confers on this system being a real-time vehicle detection system.

Keywords: *Vehicle Detection, Haar Like Features, Artificial Neural Network, FPGA, NIOS II, Real Time*

1. INTRODUCTION

Recently, the world has witnessed an evolution in the field of road safety, which has led to the emergence of driver assistance systems. These so-called intelligent vehicle systems integrate devices and electronic equipment able to detect the state of the traffic, monitor driving rules and calculate inter vehicles distances [1] [2] to avoid road accidents or mitigate their consequences. In doing so, the study of intelligent systems requires technical and highly advanced algorithms for image processing to extract useful information to provide valuable driving assistance [3]. Various types of sensors have been used for various detections in cars mentioned in the literature, such as ultrasonic sensors, laser scanners and radar [4]. However, these sensors are unable to function in the multi-object detection and distinguish between the different obstacles (car, pedestrian .....).

For this reason, research in the field of intelligent vehicles has shifted to detection technologies based on camera vision [6] because cameras are similar to human perception system and provide rich and useful information for the application of shape recognition techniques [5].

The goal of vehicle detection systems is to detect, in a complex digital image from a CCD camera, the presence of vehicles, which is such a difficult task given the complexity of exterior scenes (environment) and their constraints [7].

Generally, the majority of methods used to detect vehicles follow two main phases: hypothesis generation and hypothesis verification [8].

Several approaches for hypothesis generation have been used, such as the detection of information representing the shadow of the vehicle [9], the symmetry of a vehicle in the horizontal and vertical directions [10], texture and the lamp to the back of the vehicle [11]. As for hypothesis verification, there are such approaches, among others, as transformed HOG [12] and the pseudo-Haar [13] features, which have been used for the detection of vehicles and are used in the vision computer to detect objects in digital images.

This document provides a robust and rapid vehicle detection system that detects vehicles in two phases: the first phase is a generation phase which aims to reduce areas of the search for vehicles; this phase is to identify zones of interest that contain the shadow of the vehicle using a segmentation method

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

by thresholding. The second phase is a hypothesis testing phase which consists in verifying the presence of vehicles in suspected areas. This phase brings together two algorithms; the first is an image processing algorithm, while the second is an algorithm of artificial intelligence. The image processing algorithm aims to extract the features of a vehicle using the haar descriptor, and the algorithm of artificial intelligence uses artificial neural networks to detect these vehicles.

However, the properties of vehicle detection systems based on the vision and the complexity of their processing chains require powerful hardware implementation circuit [24], since most of these systems are based on software. The most suitable platform for these properties is the FPGA.

It is a digital programmable circuit that offers several opportunities for intelligent transport systems. Among these advantages, we find parallel processing of data to optimize the algorithms and hardware resources, and therefore the execution of complex processes is done in a short and fast time [25].

This document provides a range of contributions in the field of intelligent detection of vehicles. The first is the determination of zones of interest in an image. The second is the use of the concept of the integral image to accelerate and facilitate the calculation of pseudo-haar features. The third is the use of artificial neural networks as a presence detector or not of a car in the image of the scene. The fourth is the implementation of the vehicle detection system in the FPGA platform based on the NIOS II microprocessor.

The paper is divided as follows: The following section details the proposed vehicle detection system. Various experimental results are presented and analyzed in Section 3. The last section is devoted to the conclusion.

2. PROPOSED VEHICLE DETECTION SYSTEM

This section is dedicated to the description of the proposed vehicle detection system. This system is composed of two stages: the generation stage and the hypothesis verification phase.

2.1 Hypotheses Generation HG

This phase first presents a vehicle detection method based on its shadow to detect regions of interest (spaces containing vehicles) in the image and through the calculation of the entropy of the rows and columns which will be useful to eliminate or keep a set of image regions based on the value of the entropy calculated.

The vehicle detection algorithm based on shadow is generally applied to extract the vehicles of regions of interest in the set of images in order to reduce the computational complexity [8]. The basic principle is that regions below vehicles are much darker than other areas on a paved road [6]. This explains why the pixel gray levels in the gray areas are much lower than the other regions in the same image (figure1).

The thresholding selected in this work is to compare an image gray levels with an adaptive threshold to the lower part of the gray-level histogram, which indicates the shadow area below the vehicle.

Its aim is to reduce the quantity of information present in the image and keep only relevant data. The threshold is given by the following expression (1) [16]:

Fig.1. Vehicle detection based on shadow

$$\begin{cases} b_i = 255 \text{ if } x_i \geq T \\ b_i = 0 \quad \text{if } x_i < T \end{cases} \qquad (1)$$

With:

T is the global threshold.

x_i is the gray level of each pixel in the image.

b_i is the new value of the pixel.

The next step is to estimate the areas of interest that contain vehicles, specifying the x, y coordinates of these zones. This method is based on the calculation of the entropy of rows and columns in order to extract the coordinates of the areas that contain more entropy in rows and columns. These maximum entropy areas containing the vehicle

Journal of Theoretical and Applied Information Technology
15th November 2017. Vol.95. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

differ from other areas of minimum entropy (other phenomenon).

The calculation of entropy provides an effective measure of the information conveyed in a gray level channel or color of a given region [17] (row or column). The entropy is calculated as follows (2):

$$E = -\sum_{k=1}^{L} p_k \log_2 p_k \quad (2)$$

With a color value or gray level data (index k = 1, ..., L) that occurs with a probability pk.

After calculating the entropy of rows and columns, a threshold will be selected in such a way that the entropy of the areas of interest is maximized to extract the ordinate and x and y from these areas.

At the end of this phase the areas of interest are extracted to represent the input of the hypothesis verification phase.

2.2 Hypotheses Verification HV

Before starting the hypotheses verification HV phase, the proposed system has to go through a training phase (learning) to prepare the ANN for the vehicle detection task.

This learning phase is done in two steps:

The extraction of features for learning:

During this step, feature extraction is done using the Haar features algorithm [13]. The objective of this phase is to transform the basis of positive images (figure 2) (with vehicle) and negative (figure 3) (without vehicle) in gray scale to feature vectors. The size of these vectors varies depending on window sizes employed during the feature extraction [12].

The Learning:

At the end of this stage a training base was formed. This database contains all the feature vectors to train artificial neural network type multilayer perceptron to adjust its synaptic weights. The Multilayer Perceptron Architecture is an architecture that has shown robust and fast results in the field of intelligent transportation.

Fig.2. positive images

Fig.3. negative images

Eventually a phase of the hypotheses Verification HV follows the training phase of the ANN; it also contains two steps:

The feature extraction for detection:

In this step, a sliding window subdivides each zone of interest detected in the first phase HG in thumbnails. This window which is the same size as that used in the feature extraction phase for learning ,to form a test vector database of the initial image, is extracted using the Haar descriptor.

The detection:

An artificial neural network prepared and trained previously will classify all areas suspected in areas which exhibit the vehicle or in areas that do not represented.

2.2.1 The haar like features

The haar like features is an object and face detection algorithm, introduced by Viola and Jones [13] to resolve a detection problem of multiple objects on a single image regardless of their size (figure 4) [26].

The haar like features is an algorithm which provides for each image the information on the distribution of gray levels in two adjacent regions [18].

Fig.4. Examples of the haar-like features

The haar like algorithm is divided into two steps:

The first step is to structure the representation of the initial image called integral image (3), which was introduced by Viola and Jones in order to reduce the computation time of the haar like features.

The integral image is calculated by summing the pixels in multiple rectangles, which makes the second step of the calculation of Haar like features faster, by calculating the difference between two

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

adjacent rectangles with only 6 references and 8 references for 3 rectangles [19].

$$ii(x,y) = \sum_{x \le x', y \le y'} i(x',y') \quad (3)$$

With:

$ii(x,y)$ is the integral Image.

$i(x,y)$ is the original image

The following two equations in (4) show the calculation of the Haar windows at various scales.

$$s(x,y) = s(x,y-1) + i(x,y)$$
$$ii(x,y) = ii(x-1,y) + s(x,y) \quad (4)$$

With $s(x,y)$ means the cumulative sum of the line and $s(x,-1) = ii(-1,y) = 0$.

The haar like features algorithm in its second part is based on several rectangles and scales, which makes them very slow calculations and the size of the feature vectors very large for the classification phase.

For this reason, our approach consists in reducing the number of filters and using 2 rectangular filters (horizontal and vertical) to a single scale: 24x24.

The resulting set of Haar like features using the two rectangular filters is shown in an image 40x50 pixels is 832 (Figure 5).

Fig.5. Extraction of haar features using the 2 rectangular filters (horizontal and vertical)

These sizes vectors 832 will be the vectors input in the learning process and detection for the artificial neural network, whether for learning size image 40x50 or for the thumbnails extracted from the test regions of interest.

2.2.2 Artificial neuron network

The artificial neural network is an artificial intelligence technology, which is to process information taking inspiration from the processing way of human brain [19].

The artificial neural network has the capacity to solve nonlinear applications and complex real world problems, which makes it the most efficient algorithm for prediction and classification applications [27].

The artificial neural network follows a set of steps to make it work for any application in the field of artificial intelligence.

The choice of architecture:

The artificial neural network is based on a set of processing unit named neuron; these neurons are based on an activation function to guarantee non-linearity in the network, which is in our system the sigmoid function [28].

The choice of architecture consists of selecting how to interconnect these neurons; the architecture used in our case is the multilayer perceptron (figure 6), which uses in addition to the input and output layer, an intermediate layer called hidden layer [20].

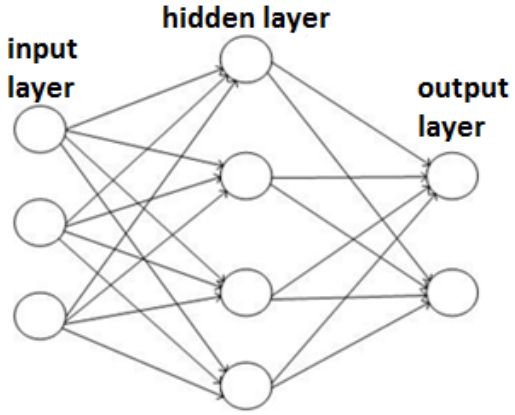

Fig.6. architecture of the MLP

The choice of learning algorithm:

Like all algorithms of artificial intelligence, the artificial neural network needs a learning algorithm, it is a process that consists in adjusting the connections between neurons called synaptic weights. The most appropriate algorithm with MLP architecture is back pro. This is an algorithm that simulates the phenomenon of learning by error.

The back pro algorithm follows a set of steps [28]:

- ✓ Initialize the values of synaptic weights randomly.
- ✓ Present the elements of the learning base successively.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- ✓ Evaluate for each element the error observed (5) at the output for each neuron.
- ✓ Adjust the synaptic weights to minimize the overall error by the gradient descent method [21].

$$E(n) = \frac{1}{2}\sum_{j \in C}(d_j(n) - y_j(n))^2 \qquad (5)$$

The choice of the learning and testing database:

For the artificial neural network works for any application, it requires a learning and test database. These databases must be rich and representative in order that the network converges towards satisfactory results for classification or recognition.

In our vehicle detection system, the multilayer perceptron is used for learning and vehicle detection. The network receives a learning database, which contains various scenes (not vehicles or vehicles), then generalize their learning to detect vehicles in the test areas of interest.

3. RESULT AND EXPERIENCES

3.1 Test and validation

The vehicle detection system developed is executed firstly on environment PC type characterized: (Intel® Core™ i3 CPU (2,40GHZ), RAM 4,00 Go, Linux Mint), using C++ programming language.

The data set used to quantify the performance of the proposed system of vehicle detection is composed of 1156 images extracted from the set of Caltech data 2001 [22]. This image database contains positive (figure 7) and negative images (figure 8). The Positive examples consist of different automobiles and rear views. The increased number of examples diminishes the overall error when learning the artificial neural network and increases the detection rate. The negative examples are random examples without vehicles. All data used contains 526 test images; the test database (figure 9) contains rear view of vehicles on a different scale.

The positive and negative images have the same dimensions 40x50 to extract the same size vectors that will be the input of the neural network during the learning phase. As for the test images are images with 96x144 sizes.

Fig.7. the positive database

Fig.8. the negative database

Fig.9. the test database

As already detailed in Section III, the detection phase is divided into 2 parts:

Hypotheses generation: the system extract the areas of interest to estimate the presence of the vehicle using the thresholding method and the calculation of the entropy of rows and columns to detect the areas that can present the shadow of a car. Thereafter these suspected areas will be classified (validated or rejected) at the Hypothesis verification.

After the learning phase, in which the artificial neural network receives the 1156 positive and negative examples to its input; each example is characterized by a vector of size 832 Haar like descriptor. During the learning of the artificial neural network, two essential criteria are used to indicate the end of this process, decreasing the overall error and the maximum number of iterations reached. The system passes to hypotheses verification in which the network is ready to detect vehicles contained in the test images. The haar descriptor extract the feature vectors for all thumbnails 40x50 size contained in each region of interest extracted from the test image. The artificial neural network processes each vector that's the size is 832 and takes a decision at the end of the presence or absence of the vehicle in the region of the image corresponding to this vector (figure 10).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

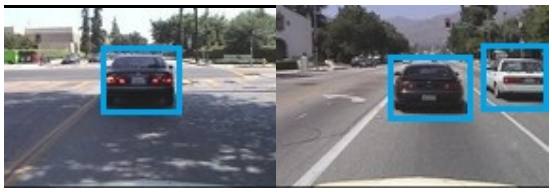

Fig.10. the detection results

To measure the performance of the proposed vehicle detection system, two criteria were selected to evaluate its performance: the detection rate and vehicle detection time; the rate of correct detection is the percentage of vehicles correctly detected on all vehicles present in the images of the test database [12]. These indicators are used to quantify the performance of a detection module [23], and to better illustrate these performances and make a comparison according to different parameters; we varied one of the most interesting parameters of the artificial neural network that corresponds to the number of neurons in the hidden layer.

The following figures (figure 11) and (figure 12) show the results of tests performed:

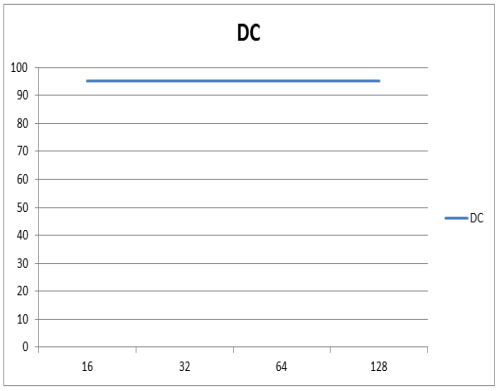

Fig.11. graph of detection rate

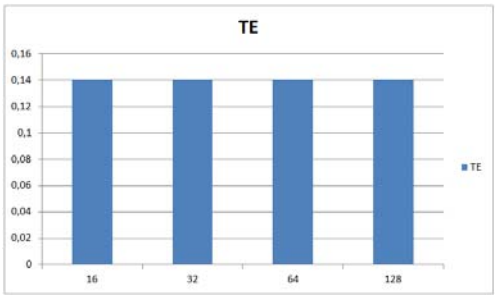

Fig.12. graph of the execution time

The vehicle detection methods based on haar like features have better detection results; their major drawback remains the detection time which is a little high and is a key parameter in driver assistance systems in real time.

The experimental results show that our system has not only a better detection accuracy rate (95,05%), but also faster than other methods, the calculation speed is due to the reduction of search areas made during the phase of hypotheses generating.

These results also show the robustness and speed of our method for detection of vehicles using the haar windows as image descriptor and artificial neural networks Multilayer Perceptron such as classifier and detector.

3.2 Hardware implementation

3.2.1 The implementation

Beyond all the prior study done, the completion of this work is marked by the implementation of the vehicle detection system in an FPGA platform to provide the functional autonomy of an embedded system. This implementation is based on using NIOS II softcore microprocessor.

This section is dedicated to the description of the various stages and modules needed to implement our detection system.

The field programmable gate array is integrated circuits that offer cheaper solutions, easy and flexible [29]. The FPGA is used for the implementation and design of complex circuits and complete digital systems on a single chip [30]. The FPGA is composed of three basic blocks which are configurable logic blocks, the input-output blocks and interconnections [31].

The FPGA offers several advantages of parallel processing, speed and flexibility [21], which will be useful in signal and image processing applications as the case of our vehicle detection system that requires speed and parallelism.

The FPGA programming language used as a hardware description language the VHSIC hardware description language VHDL or Verilog in a specific software called Quartus [21].

The vehicle detection system is designed in a Cyclone II 2C70 FPGA fabricated by Altera, which is connected to two 32 Mbyte SDRAM, 2 Mbyte SSRAM and 8 Mbyte Flash memory. The DE2 board include an adequate number of robust interfaces, such as two TV decoder, VGA (10-bit DAC), 10/100 Ethernet, a (IrDA) infrared port.

The design of our vehicle detection system is based on the use of softocore NIOS II processor. This design requires in addition to the NIOS II, other input-output peripherals that help the

Journal of Theoretical and Applied Information Technology
15th November 2017. Vol.95. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

functioning of the system. Among these peripherals: 7-segment display, SD card controller, SDRAM, an LCD display, LEDs and Switches. The following diagram (figure 13) illustrates the implementation of the detection system.

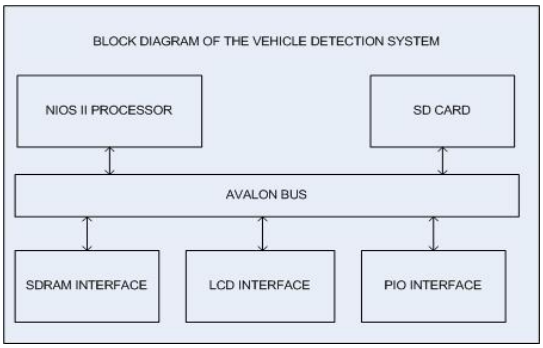

Fig.13. the block diagram of the vehicle detection system

The figure shows the hardware implementation of the vehicle detection system. The system is based on the SD CARD interface; it allows connecting the SD CARD with the NIOS II to transform images into a stream of data that will be used later. The SD CARD interface is provided by connecting pin 4 PIO to SD CARD. Then this stream of data will be stored in a SDRAM memory area.

In this conception, the system performs two tasks. The first task is to read the data from the SD CARD, these data are images in bmp format.

And the second task is to process these images using the respective algorithms image processing (like haar features) and artificial intelligence (Artificial Neural Network) to detect the presence or absence of vehicles.

The LCD controller is controlled by the LCD interface to display the detection result.

The following figure (figure 14) shows the overall system generated by the SOPC Builder.

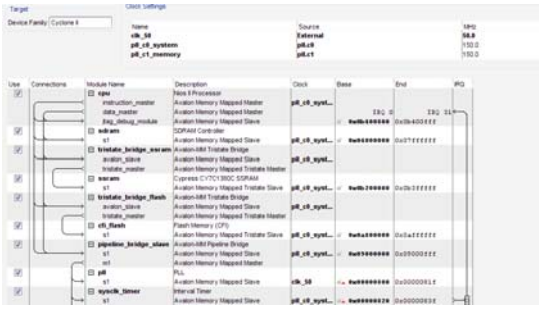

Fig.14. the SOPC builder system

The following figure (figure 15) shows the module generated after compilation by the SOPC Builder. This module will be added to the component library of the FPGA. The microcontroller composed by the NIOS II and peripherals will be programmed later to detect vehicles using a programming language such as C language.

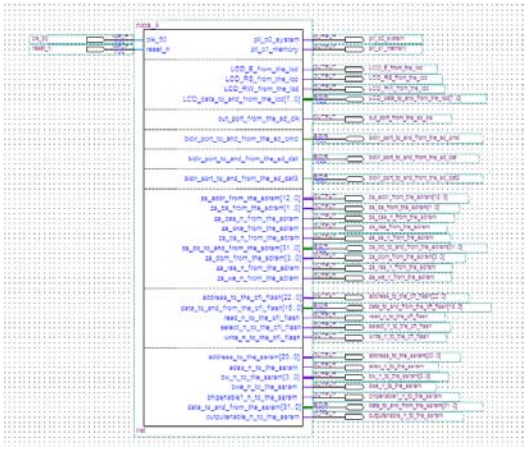

Fig.15. the block design of the proposed system

After designing the hardware parts of our system, the system is connected to the FPGA board DE2_70 via a JTAG_UART interface [32].

Then the system will be developed by NIOS IDE software, which is a tool of Altera. This tool uses a compiler and assembler for the Nios II processor. The development languages used by the NIOS IDE are C, C ++ and assembler [21].

The system implements multiple blocks as shown in the figure, among these blocks:

- ✓ SD card driver implements the drivers for the SD card to detect the SD card and reading data.
- ✓ The FAT 16 block implements the FAT16 file system for reading the image files from the SD card.
- ✓ The bmp block that can read data in bmp format.

Journal of Theoretical and Applied Information Technology
15th November 2017. Vol.95. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

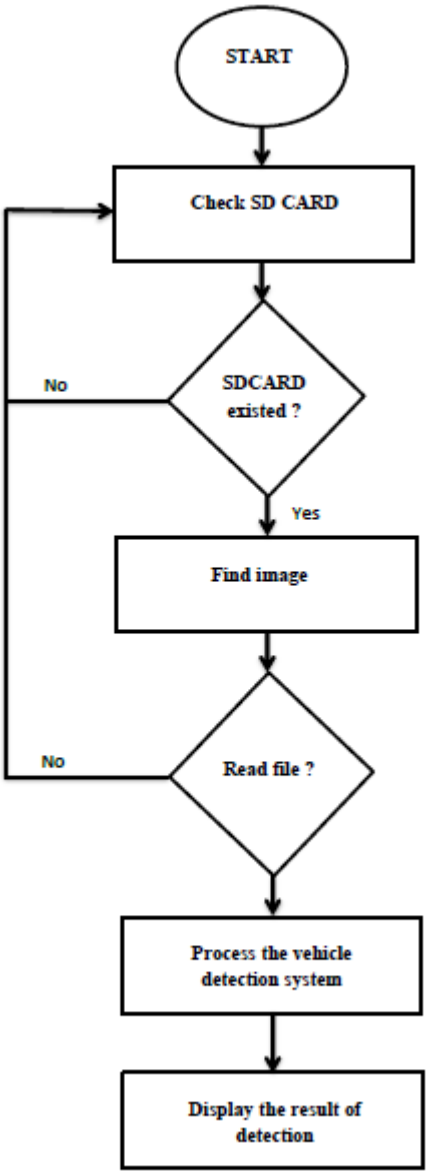

Fig.16. the global vehicle detection system

The figure (figure 16) shows the operation of our vehicle detection system based on the NIOS II processor; the first step is to detect the presence of the SD CARD. After the detection, SD CARD will be activated and the files will be listed in FAT 16 format. The image file types used in our system are to bmp extension file.

The following step is a critical step in our system. Once the image to be processed is selected, the system combining the thresholding algorithm, such as haar features and artificial neuron networks, will detect the presence or absence of a vehicle in the image.

3.2.2 Hardware results

In this section, the implementation of our vehicle detection system will be simulated and tested in the FPGA card DE2_70 EP2C70F896C6 release. This paragraph will be dedicated to the collection of the execution time and material resources used by the system based on the Nios II processor in order to show the performance of our vehicle detection system.

The test of our system was able to make the collection of hardware resources and execution times. The results are shown in the following figure (figure 17):

Revision Name	DE2_70
Top-level Entity Name	DE2_70
Family	Cyclone II
Device	EP2C70F896C6
Timing Models	Final
Met timing requirements	No
Total logic elements	5,214 / 68,416 (8 %)
Total combinational functions	4,458 / 68,416 (7 %)
Dedicated logic registers	2,885 / 68,416 (4 %)
Total registers	3220
Total pins	383 / 622 (62 %)
Total virtual pins	0
Total memory bits	69,568 / 1,152,000 (6 %)
Embedded Multiplier 9-bit elements	4 / 300 (1 %)
Total PLLs	1 / 4 (25 %)

Fig.17. the hardware result

The results obtained during the implementation of the vehicle detection system in FPGA-based on NIOS II microprocessor can show the performance of the system in terms of hardware capacity and execution time.

Regarding the material resources, the implementation based on the Nios II processor designed by developers of Altera remains the most optimized architecture compared to hardware's architectures; this is an implementation that does not consume enough material resources.

In addition, the algorithms used during the processing are described in a software and not hardware; this means that they are in the form of a program written in C language, and which does not require additional hardware resources.

For the execution time, the implementation has clear advantages in terms of speed of calculation, which reached 0.072 s; this is due to the optimization of neuronal and hardware architecture of the system and the speed of processing algorithms proposed in our system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. CONCLUSION

In our article, the implementation of a driving assistance system in the FPGA platform was made and has given satisfactory results in terms of resources used and in terms of execution time.

The implemented system has proposed a method of reducing research space in a scene image by using thresholding method based on the calculation of entropy, which has had a great impact on the detection time. It also proposed a method that combines powerful algorithms between the image processing field and the decision-making. This combination of haar like features descriptor and the classifier based on Multilayer Perceptron type artificial neuron networks has given very encouraging experimental results. It entrusts this proposed system with the speed and robustness required for use in real-world applications that require real-time processing.

However, several improvements can be made to this single-object detection system to allow a multi-object robust, fast and usable detection (vehicle, piétons. etc) in real applications of road safety. These improvements will be our future work.

REFRENCES:

[1] Wang, H., & Zhang, H. (2014). A Hybrid Method of Vehicle Detection based on Computer Vision for Intelligent Transportation System. International Journal of Multimedia&Ubiquitous Engineering, 9(6).

[2] Negri, P., Clady, X., & Prevost, L. (2007, January). Benchmarking haar and histograms of oriented gradients features applied to vehicle detection. InICINCO 2007, Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics, Robotics and Automation 1, Angers, France, May 9-12, 2007 (pp. 359-364).

[3] Mioulet, L., Breckon, T. P., Mouton, A., Liang, H., &Morie, T. (2013, February). Gabor features for real-time road environment classification. In Industrial Technology (ICIT), 2013 IEEE International Conference on (pp. 1117-1121).IEEE.

[4] T. Gandhi and M. Trivedi, "Pedestrian protection systems: Issues, survey,and challenges," IEEE Trans. Intell. Transp. Syst., vol. 8, no. 3, pp. 413–430, Sep. 2007.

[5] Ge, J., Luo, Y., &Tei, G. (2009). Real-time pedestrian detection and tracking at nighttime for driver-assistance systems. Intelligent Transportation Systems, IEEE Transactions on, 10(2), 283-298.

[6] Li, W., Liu, P., Wang, Y., & Ni, H. (2014). Multifeature Fusion Vehicle Detection Algorithm Based on ChoquetIntegral.Journal of Applied Mathematics, 2014.

[7] Vitabile, S., Gentile, A., &Sorbello, F. (2002). A neural network based automatic road signs recognizer. In Neural Networks, 2002.IJCNN'02.Proceedings of the 2002 International Joint Conference on (Vol. 3, pp. 2315-2320).IEEE.

[8] Z. H. Sun, G. Bebis and R. Miller, "On-Road Vehicle Detection: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, (2006).

[9] Y. M. Chang, S. S. Huang, L. C. Fu, P. Y. Hsiao and M. F. Lo, "Vehicle detection and tracking under various lighting conditions using a particle filter", IET Intelligent Transport Systems, vol. 6, no. 1, (2012).

[10] J. Arrospide, L. Salgado, M. Nieto, and F. Jaureguizar, "On-board robust vehicle detection and tracking using adaptive quality evaluation" , IEEE International Conference on Image Processing, San Diego, California, USA, (2008) October.

[11] R. O. Malley, E. Jones and M. Glavin, "Rear-Lamp Vehicle Detection and Tracking in Low-Exposure Color Video for Night Conditions", IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 2, (2010).

[12] Negri, P., Clady, X., Hanif, S. M., & Prevost, L. (2008). A cascade of boosted generative and discriminative classifiers for vehicle detection.EURASIP Journal on Advances in Signal Processing, 2008, 136.

[13] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features.In Computer Vision and Pattern Recognition, 2001.CVPR 2001.Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511).IEEE.

[14] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio,"Pedestrian detection using wavelet templates," in Proceedingsof the IEEE Computer Society Conference on Computer Visionand Pattern Recognition (CVPR '97), pp. 193–199, San Juan,Puerto Rico, USA, June 1997.

[15] Liu, J., Wong, C. C., Zhang, Y., Liu, S., Guo, Y., & Zhao, Y. (2015). On-road Vehicle Detection and Tracking Based on Monocular vision.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[16] Gupta, M. R., Jacobson, N. P., & Garcia, E. K. (2007). OCR binarization and image pre-processing for searching historical documents.Pattern Recognition, 40(2), 389-397.

[17] R. C. Gonzalez and R. E. Woods, Digital Image Processing. EnglewoodCliffs, NJ: Prentice-Hall, 2002.

[18] Oualla, M., Sadiq, A., &Mbarki, S. (2014, April). A survey of Haar-Like feature representation. In Multimedia Computing and Systems (ICMCS), 2014 International Conference on (pp. 1101-1106). IEEE.

[19] Nazeer, S. A., Omar, N., Jumari, K. F., & Khalid, M. (2007, March). Face detecting using artificial neural network approach. In Modelling& Simulation, 2007.AMS'07. First Asia International Conference on (pp. 394-399). IEEE.

[20] Shende, S., & Patel, R. (2013). Review of Face Detection using PCA and ANN Techniques. International Journal on Advanced Computer Theory and Engineering (IJACTE ISSN (Print): 2319–2526, Volume-2, Issue-5.

[21] Atibi M, Atouf I, Boussaa M, Bennis A. Parallel and Mixed Hardware Implementation of Artificial Neuron Network on the FPGA Platform.International Journal of Engineering & Technology; 2014; 0975-4024, 6(5).

[22] Online in: http://www.vision.caltech.edu/html-files/archive.html

[23] C. R. Wang and J. J. Lien, "Automatic Vehicle Detection Using Local Features-A Statistical Approach", IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 1, (2008).

[24] Tisan, A., &Cirstea, M. (2013). SOM neural network design–A new Simulink library based approach targeting FPGA implementation. Mathematics and Computers in Simulation, 91, 134-149.

[25] Aguirre-Dobernack, N., Guzmán-Miranda, H., & Aguirre, M. A. (2013, November). Implementation of a machine vision system for real-time traffic sign recognition on FPGA. In Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE (pp. 2285-2290).IEEE.

[26] Zheng, Y., Yang, C., Merkulov, A., &Bandari, M. (2016, May). Early breast cancer detection with digital mammograms using Haar-like features and AdaBoost algorithm. In SPIE Commercial+ Scientific Sensing and Imaging (pp. 98710D-98710D).International Society for Optics and Photonics.

[27] Nath, S., Kotal, S. D., &Kundu, P. K. (2016). Seasonal prediction of tropical cyclone activity over the north Indian Ocean using three artificial neural networks. Meteorology and Atmospheric Physics, 1-12.

[28] Mohamed, A., Issam, A., Mohamed, B., &Abdellatif, B. (2015). Real-time Detection of Vehicles Using the Haar-like Features and Artificial Neuron Networks. Procedia Computer Science, 73, 24-31.

[29] Alçın, M., Pehlivan, İ., &Koyuncu, İ. (2016). Hardware design and implementation of a novel ANN-based chaotic generator in FPGA.Optik-International Journal for Light and Electron Optics, 127(13), 5500-5505.

[30] E.Z. Mohammed, H.K. Ali, Hardware implementation of artificial neural networkusing field programmable gate array, Int. J. Comput. Theory Eng. 5 (5)(2013) 780–783.

[31] Josh, N. N., Dakhole, P. K., &Zode, P. P. (2009, December). Embedded Web Server on Nios II Embedded FPGA Platform. In Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on (pp. 372-377).IEEE.

[32] Hong-wei, L., Jian-ai, L., & Ling-ling, K. (2011, May). Implementation of SD card music player using altera DE2-70. In Multimedia and Signal Processing (CMSP), 2011 International Conference on (Vol. 2, pp. 150-153). IEEE.