Name : Ng Yen Aeng

IC No. : 941229-14-5457

ID No. : 15WAU09632

**Title:**

Image recognition and tracking of an FPGA real-time driver drowsiness detection system.

**Abstract:**

This project focuses on the image recognition and tracking of a Driver Drowsiness Detection System. The proposed architecture mainly consists of two parts which is the image capture and processing, as well as the image recognition and tracking sections. The initial steps for image recognition is to initialize the value of weight with a random number and later construct the hidden layers, input layers and output layers with the format of 10x10 matrix. Backpropagation is used to optimize the weight value so that the hidden layers have better performance and generates a more accurate output for driver drowsiness detection. To ensure that the weights generated is correct, numerical gradient checking is used to check the gradient hidden layers mathematical formula. If no error were found during checking, the trained results will be stored into the training block and it will be ready for recognition. After recognition, the results will be fed into Smart Decision Block for a more accurate decision. This block will store the decision that is made by the training block. If the driver is sleeping, the alarm system will be triggered when two consecutive sleeping decisions are made. The aim of this paper is to develop an Image Recognition and tracking system for driver drowsiness detection using a modified Feed-Forward Neural Network (FFNN) that will resolve issues faced by conventional blocks particularly in terms of performance and accuracy. This proposed FFNN recognition block introduces Smart Decision Block for a more accurate result by making final decision based on two consecutive detections. This system will be modeled, designed and simulated in Verilog using ModelSim software of Mentor Graphics environment. Lastly the designed Verilog codes will be integrated and implemented onto the FPGA board for functional verification purposes. This proposed research contributes to the car safety field and is most likely applied in a smart car system that is used to reduce accidents due to a sleepy or fatigue driver.

**Objectives:**

* To investigate several different facial recognition techniques and algorithm including the FeedForawrd Nueral Network (FFNN) and smart decision techniques for drowsiness detection from past literatures.
* To model, design and verify all sub-modules of the FFNN and smart decision system using HDL based testbench in Modelsim of Mentor Graphics environment.
* To analyze, integrate and implement all the integrated sub-modules of the FFNN based drowsiness detection with the smart decision system into the Cyclone IV FPGA for hardware verification purposes.

**Introduction:**

According to the National Sleep Foundation’s research (2006), 60% of adult drivers had driven a vehicle while feeling drowsy in the past year and more than one-third have actually fallen asleep at the wheel. In fact those who have nodded off, 13% say they have done so at least once a month furthermore, 4% admit they have had an accident or near accident because they dozed off or were too tired to drive.

Sleep related crashes are most common in young people, especially men, adults with children and shift workers. Drowsy driving crashes can result in high personal and economic costs. For example, several drowsy driving incidents have resulted in jail sentences for the driver. Besides that, multi-million dollar settlements have been awarded to families of crash victims as a result of lawsuits filed against individuals as well as businesses whose employees were involved in drowsy driving crashes. To solve the problem stated above, driver drowsiness detection system was introduced to warn drivers when they are fatigue and avoid traffic accidents caused by fatigue or lack of concentration.

The drowsiness detection system that will be implemented in this experiment is separated into two blocks which is the image capture and processing, as well as the image recognition and tracking sections. The main blocks that will be discussed here is the recognition and tracking block. The algorithm being used to solve the recognition and tracking problem is the FFNN. Various journals and literatures were studied to compare different recognition techniques, the literatures that is referred by this study is tabulated in Table 1.

Real-time drowsiness detection system for intelligent vehicles (Kuo and Hsu 2010) was implemented by fuzzy logic. Fuzzy logic require a fuzzy rule set which is difficult to implement using Verilog. Algorithm of the system that is designed by (Cho *etal.* 2009) were Viola-Jones and AdaBoost learning algorithm. By using both algorithms, the system can operate extremely fast however the accuracy drops tragically. PERCLOS algorithm (Vitabile *etal.* 2011) however, was an analogue based algorithm, which is very difficult to implement. Self Organizing Map (SOM) with voting system (Lee 2015) is a type of Artificial Neural Network (ANN) which had better recognition accuracy, speed and minimal resource requirement.

The ANN architecture of this project is mainly based on the architecture that is used in (Lee 2015) and (Singh and Banga 2013) as both of these systems solved their problems by using ANN. For the system that is designed by (Lee 2015), the SOM that is being used is applied to iris recognition while this project uses FFNN to classify the drowsiness level of the driver.

Table 1: Past Literatures on Drowsiness Detection and Image Recognition System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Researcher(Year) | System | Algorithm | Advantage | Disadvantage |
| 1 | Kuo and Hsu (2010) | Real-Time Drowsiness Detection System for Intelligent Vehicles | Fuzzy Logic | Rule base or fuzzy sets easily modified, can achieve steady state rapidly | Require more fine tuning and simulation before operational |
| 2 | Cho *etal* (2009) | Fpga-based face detection system using haar classifiers | Viola–Jones object detection framework: AdaBoost learning algorithm using Haar features | Extremely fast feature computation | It can hardly cope with 45o face rotation both around the vertical and horizontal axis |
| 3 | Vitabile *etal* (2011) | A real-time non-intrusive FPGA-based drowsiness detection system | Percentage of Eye Closure (PERCLOS) algorithm | Can be used in low light condition | Easily making wrong decision when face are not facing camera. |
| 4 | Wang *etal*(2005) | Real-Time Driver Drowsiness Tracking System | Image processing algorithm using self-defined logic. | Higher accuracy during night time. | Low accuracy when user wearing spectacle. |
| 5 | Lee (2015) | FPGA based Iris Recognition System using Self Organizing Map | Self-Organizing Map (SOM) with voting system | Better recognition accuracy, speed and resource usages by using a voting system | Difficult to determine what input weights to use. |
| 6 | Singh and Banga (2013) | Drowsiness Warning System Using Neural Network | Artificial Neural Network (ANN) | Can handle large amount of data sets. | Slow learner. |

**Problem Statement:**

The aim of this research is to reduce the traffic accidents that is caused by the fatigue drivers which did not focus on the road conditions or even accidentally fallen asleep. There are two main path in this drowsiness detection system which includes both training and recognition respectively. The training section is used to train the system to detect the driver drowsiness – drowsy or not drowsy. While the recognition path is used to analyse whether the driver is now in drowsy mode or awaken mode.

The Voting system of the conventional block that is shown in Figure 1 is used to increase the accuracy was not useful in this project, because the voting system will take more time to recognize the iris pattern, thus the recognition speed will be reduced. In this project, the recognition process is in real time, therefore the recognition accuracy needs to be improved using other method (rather than voting system). Hence, the proposed solution is to classify the drowsiness level that had been constructed by adding a Smart Decision Block to ensure that a balance of both accurate recognition decisions and a fast recognition speed is obtained. Besides that an Alarm System Block were implemented to alert the sleepy driver.

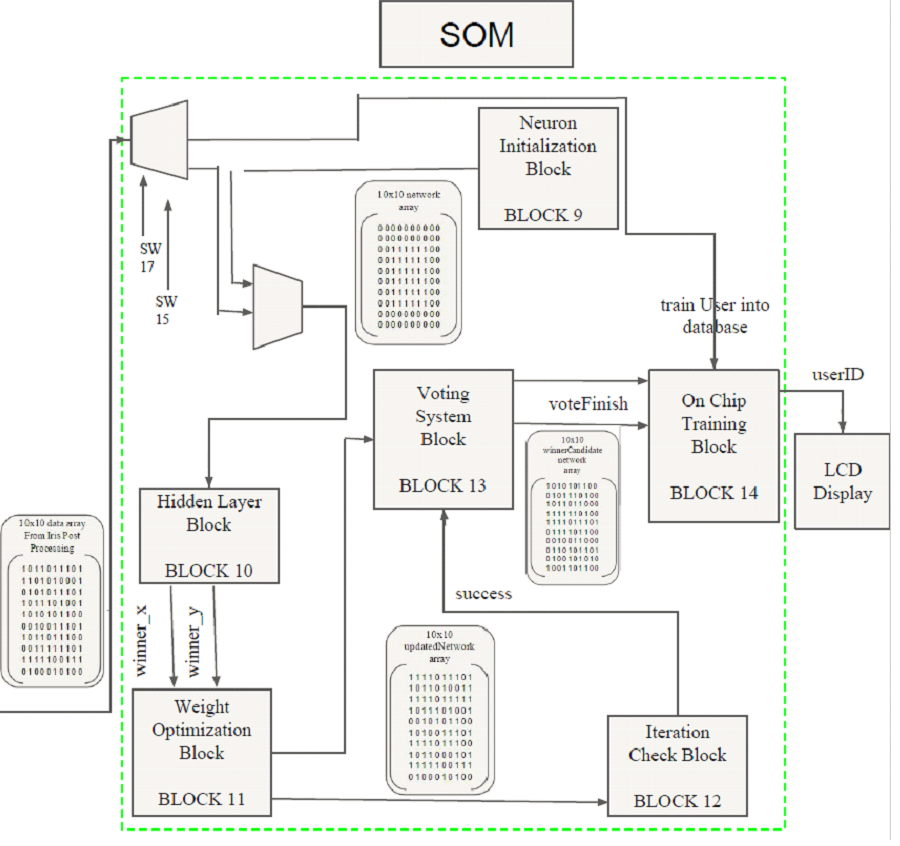


Figure 1: Image recognition conventional block using SOM (Lee, 2015)

**Proposed Block:**

As shown in Figure 2, the proposed drowsiness detection system block contains two main blocks which is the Image Capture and Processing, and Image Recognition and Tracking. The first main block (Image Capture and Processing) uses a CMOS camera to capture the image of the driver and process the image by segmenting and separating the eye part then image compression is performed. The compressed image information in the format of 10x10 matrix is then passed to the second main block (Image Recognition and Tracking). This block is the focus of this research, it will be responsible to train the system to recognize the drowsiness of the driver and analyze the drowsiness level of the driver. All seven sub-blocks will be discussed in the subsequent paragraphs.

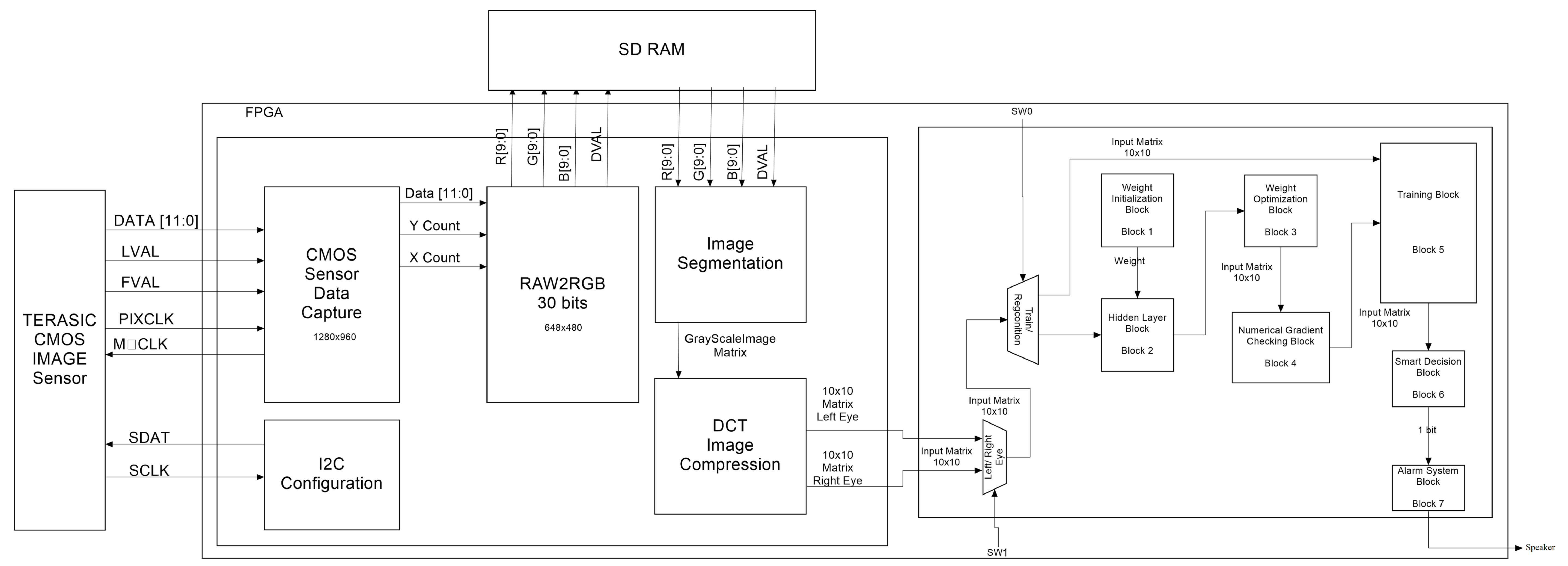


Figure 2: Proposed architecture of the drowsiness detection system

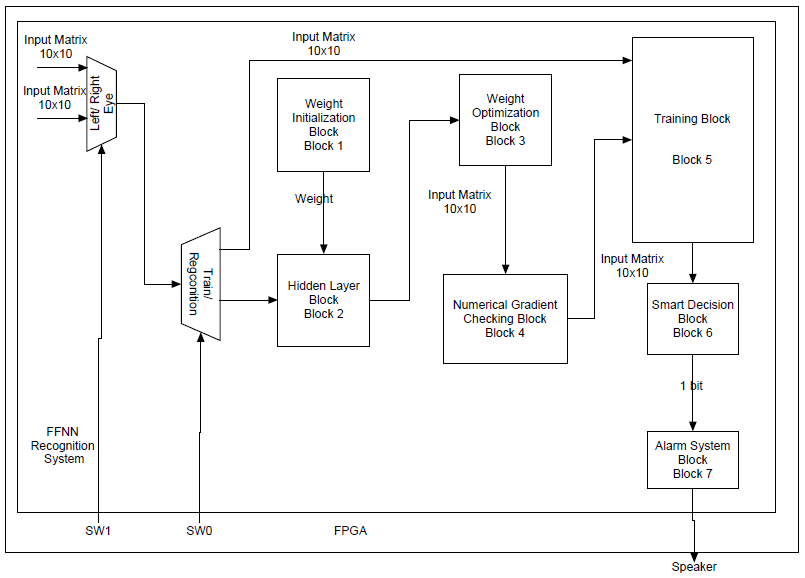


Figure 3: Proposed image recognition and tracking block of drowsiness detection system

Firstly, there are two multiplexer in this recognition and tracking block, the first multiplexer is used to control the input of the eye (left eye or right eye) while the second multiplexer chooses the path of the input to recognize or train. The first sub-block is Weight Initialization Block (block 1), the main function of this block is to initialize random values for the weight. The values of the weights should be different and non-zero so that the hidden unit in the hidden layer gets different and non-zero signals. If all the signals are the same or equal to zero, the neural network might not be functioning as expected.

Secondly, the Hidden Layer Block (Block 2) which defines the relationship between the hidden layers, input layers and output layers by using the weight that is obtained from Block 1. The 10x10 matrix that contains the details of the eyes will be passed into this block and it will go through some mathematical formula (multiply with the weight and add the result together for each neural) and generate two output 10x10 matrix which represent sleeping and awaken state. The two outputs will be compared with the inputs during Recognition to classify the drowsiness status of the driver (sleepy or awake). However, the output layer obtained from Block 2 is not optimized, therefore Weight Optimization Block (Block 3) is needed to optimize the weight by means of back propagation. It is because the non-optimized weight will give a fault result to the system, back propagation used to reduce the result error by optimizing the weight. The weight will now be optimized and the value of the output layer will be changed.

Next is the Numerical Gradient Checking Block (Block 4) which is used to troubleshoot the errors that occur when finding the weights. The main idea here is to check the mathematical formula by derivation and finding the gradient to make a comparison. After making sure that the weights are correct, the 10 x 10 output matrix will be passed into the Training Block (Block 5). Then Block 5 will define the sleeping or awaken category by using the 10 x 10 matrix output from Block 4 during the Training Path while it will categorize the 10 x 10 input matrix as sleeping or awaken during the recognition process.

After categorizing the input matrix, the decision on whether the driver is sleeping or not will be passed into the Smart Decision Block (Block 6) to ensure that a more accurate decision is made. This block will store the decisions for both eyes and will prompt the system to capture the eyes again if the driver is suspected to be in the sleepy category. When this block makes two consecutive “drowsy” decisions, it will then provide instructions to the Alarm System Block (Block 7) to sound the speaker or alarm. Finally, Block 7 is used to activate the alarm when a drowsy driver is detected.

**Methodology:**

Figure 4 shows the design flow chart of the image recognition and tracking of an FPGA real-time driver drowsiness detection system where it will begin with literature review of the driver drowsiness detection system architectures and end with final hardware testing and integration on an FPGA. The following paragraphs discuss each step in detail.

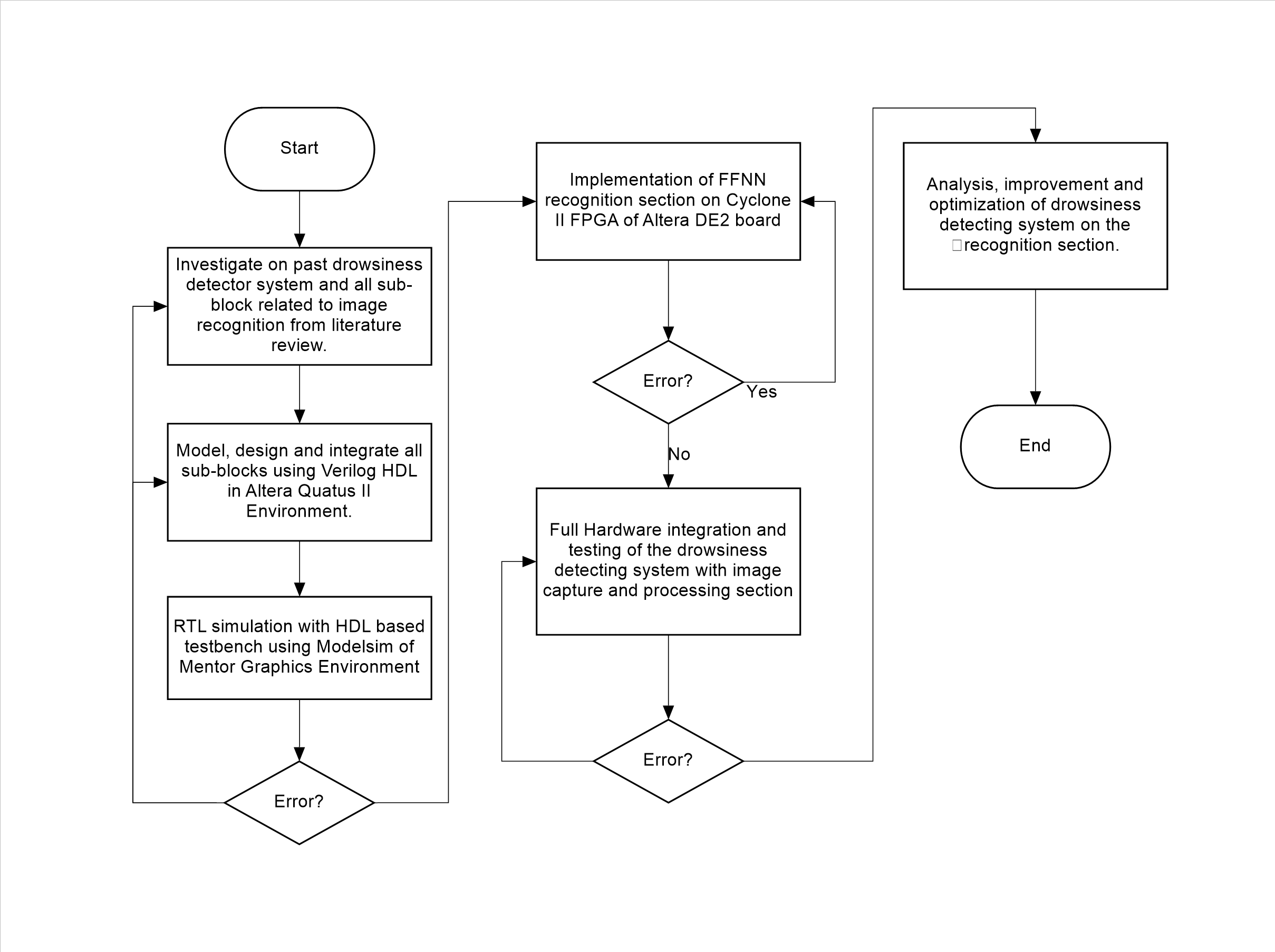


Figure 4: Design flow chart of image recognition and tracking of an FPGA real-time driver drowsiness detection system.

The proposed drowsiness detection system begin with investigating the drowsiness detection system and all sub-blocks related to image recognition from literature review. This project is divided into two main modules which is the image capture and processing as well as the image recognition and tracking sections. In literature review, all the functions, input and output of each sub-block in the image recognition and tracking had been researched and understood. Then the model and design of the entire sub-system is developed using Verilog code in Quartus II. After the code is implemented, a HDL-based testbench is written to verify the designed code using Modelsim of Mentor Graphic environement. If any error occurred or the result is not shown as expected, the past literature will be reviewed and the sub-block will be remodeled. On the other hand, if the simulation is successful, the Verilog codes will be implemented on the FPGA board and if any error occurs, the codes will be remodeled before hardware implementation on the FPGA. After the hardware implementation successful, all sub-blocks together with the image processing sections will be integrated for final hardware verification. The system will be modified and optimized until final output reaches the expected performance with no function errors.

**Conclusion:**

The proposed design of Driver Drowsiness Detection for Safety purposes is divided into two main sections, which include image capture and processing as well as image recognition and tracking. This work focuses on the final ANN recognition stage using the Forward Propagation algorithm with a Smart Decision System. This work aims to resolve speed and accuracy issues in conventional expert systems using a smart decision system to improve the accuracy while using Forward Propagation to improve the training speed for ANN. The Forward Propagation Neural Network will be designed, modelled and verified using HDL based entry in Modelsim of Mentor Graphics environment before being integrated and implemented with other main sections into the Cyclone II FPGA. The expected ANN training time is 5s and the expected recognition time is within 1s, while the expected accuracy is 90%. This Forward Propagation based algorithm for Driver Drowsiness Detection architecture can be applied to car safety systems that could potentially reduce the accident occurrences due to sleepy or fatigue drivers

**Reference:**

* YC Kuo, WL Hsu., 2010, Real-Time Drowsiness Detection System for Intelligent Vehicles.
* Cho, J., Mirzaei, S., Oberg, J. and Kastner, R., 2009, February. Fpga-based face detection system using haar classifiers. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*(pp. 103-112). ACM.
* Vitabile, S., De Paola, A. and Sorbello, F., 2011. A real-time non-intrusive FPGA-based drowsiness detection system. *Journal of Ambient Intelligence and Humanized Computing*, *2*(4), pp.251-262.
* SF Lee. , 2015, D Development of an FPGA based Iris Recognition System using Self Organizing Map for Augmented Security.
* F Wang, HY Cheng, XM Guan., 2005, Real-Time Driver Drowsiness Tracking System.
* Singh, I. and Banga, D.R.V.K., 2013. Development of Drowsiness Warning System using Neural Network. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *2*(8), pp.3614-3623.
* National Sleep Foundation, 2006. Sleep in America poll. Online available at : drowsydriving.org/about/facts-and-stats/. [Accessed on 4 August 2016]