Real-Time Drowsiness Detection System for Intelligent Vehicles

Ying-Che Kuo
Department of Electrical Engineering
National Chin-Yi University of Technology
kuoyc@ncut.edu.tw

Wen-Ling Hsu
Department of Electrical Engineering
National Chin-Yi University of Technology
machi791@hotmail.com

Abstract

In this paper, in order to implement a computer vision-based recognition system of driving fatigue. In addition to detecting human face in different light sources and the background conditions, and tracking eyes state combined with fuzzy logic to determine whether the driver of the physiological phenomenon of fatigue from face of detection. Driving fatigue recognition has been valued highly in recent years by many scholars and used extensively in various fields, for example, driver activity tracking, driver visual attention monitoring, and in-car camera systems.

In this paper, we use the Linux operating system as the development environment, and utilize PC as the hardware platform. First, the system uses a camera to obtain the frame with a human face to detect, and then uses the frame to set the appropriate skin color scope to find face. Next, we find and mark out the eyes and the lips from the selected face area. Finally, we combine the image processing of eyes features with fuzzy logic to determine the driver's fatigue level, and make the graphical man-machine interface with MiniGUI for users to operate. The results of experiment show that we achieve this system on PC platform successfully.

Keyword: drowsiness detection, driver fatigue, face detection, fuzzy logic.

1. INTRODUCTION

In recent years, because of the need for public transportation, cars and motorcycles grow at a rapid rate. The reasons of traffic accidents become much more complex, general transport system has been inadequate. Therefore, researches [1] and [2] discuss intelligent transport system (ITS) which combine advanced communicated technologies with information systems. It has become a popular topic of many research units in recent years. The driver's fatigue recognition system is a part of ITS vehicles active safety system.

There are a number of safety devices used in vehicles to protect the driver at present, for examples, seat belts, airbags, brake systems and hard sheet metal, etc. However, these devices always act after the accident happened. There are less of equipments can warn drivers before the accidents happened. Nevertheless, some signs usually exist before many accidents occurring. Driver fatigue recognition system hopes to warn driver when they are fatigued, and avoid traffic accidents caused by fatigue.

In driving fatigue recognition system, we mainly identify the driver’s fatigued level by blinking period, blinking interval or head swing. In [3] Paul Smith presents a system for analyzing driver visual attention, the system relies on estimation of global motion and color statistics to robustly track a person’s head and facial features. The system classifies rotation in all viewing directions, in addition, it is able to track both through occlusion due to eye blinking, and eye closure, large mouth movement, even when the face is fully occluded due to rotation. The system does not break down and it can be used for more advanced driver visual attention monitoring.

The paper [4] presents a hardware system using an active IR illuminator and through software for monitoring some visual behaviors to define a driver’s level of vigilance. This system using a fuzzy classifier to infer the level of inattentiveness of the driver, and fusion these multiple visual parameters yield a more robust and accurate result than by using a single parameter. The system has been tested with different driving habits recorded in night and day driving conditions in the freeway and with different users.

The literature [5] describes a real-time driver fatigue monitor system, it uses coupled cameras equipped with active infrared illuminators to acquire video images of the driver, variously visual cues that typically the level of person need warn in real time and systematically combined to infer the fatigue level of the driver. The visual cues employed eyelid movement, head movement, and facial expression to describe, can acquire probabilistic of fatigue based on the visual cues.

This paper presents a driver's fatigue recognition system combining with the fuzzy logic approach. In different light sources and backgrounds, it can effectively determine whether the current driving situation of fatigue and falling asleep, and then give warning.

In section 2, we will explain the details of fatigue recognition algorithm; the section 3 is to introduce using hardware and software architecture of the system. The section 4 is methods and results of the experiment, and the final section of this paper is conclusions.

2. FATIGUE RECOGNITION ALGORITHM

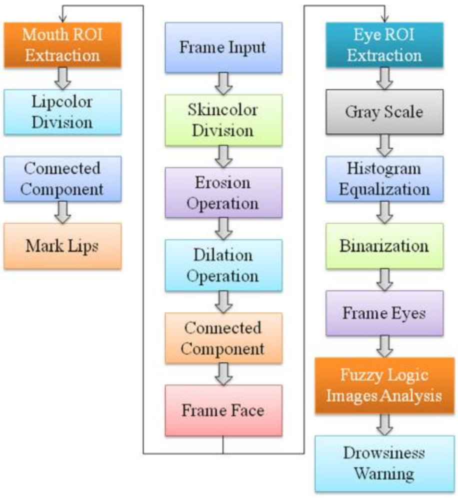


Fig. 1 The flow diagram of fatigue recognition algorithm

In figure 1, we proposed method of driver fatigue recognition using a multi-level image processing to filter out noises, and capture driver facial features in the image frame. Then, we used the eye's feature tracking combining with fuzzy logic approach to recognize the level of driver fatigue as well as give warning according to the degree of intensity.

2.1 Face Detection

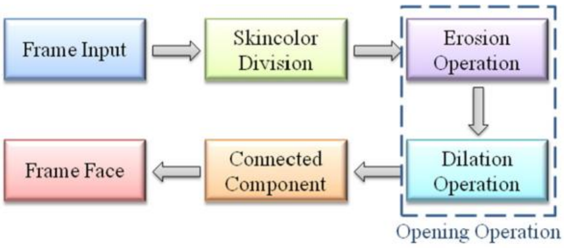


Fig. 2 The flow diagram of face detection

In figure 2, before capturing the facial features, we must find driver's face position. Therefore, this system uses a multi-level image processing to filter out noises, and uses the method of connecting component to detect the driver's face location and mark it.

2.1.1 Skin Color Segmentation

At the beginning, due to the different brightness, there are variations of skin color in RGB color space. Before finding the driver's face location, we convert entire image to the YC_bC_r color space in order to reduce the impact of brightness. We use equation (1) [6] to transfer input image from RGB color space into JPEG YC_bC_r color space, and then use equation (2) to isolate skin color image.

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.1687R - 0.3313G + 0.5B + 128 \\ Cr &= 0.5R - 0.4187G + 0.0813B + 128 \end{aligned} \qquad (1)$$

$$\begin{aligned} 60 &\le Y \le 255 \\ 100 &\le Cb \le 125 \\ 135 &\le Cr \le 170 \end{aligned} \qquad (2)$$

The results of above-mentioned actions are shown in figure 3.

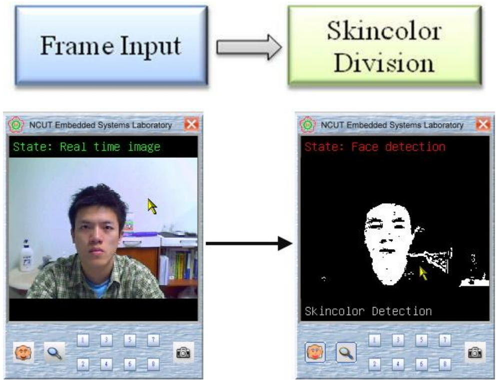


Fig. 3 Skin color segmentation results

2.1.2 Color Enhancing and Noise Filtering

After the skin color segmentation method processed, there are many noises of skin color in the screen. In order to avoid noise interferences, we use erosion operation of morphology to filter small skin color noises. And then use dilation operation of morphology to enhance the strength of the remaining skin color. That is opening operation of morphology image processing; results are shown in figure 4.

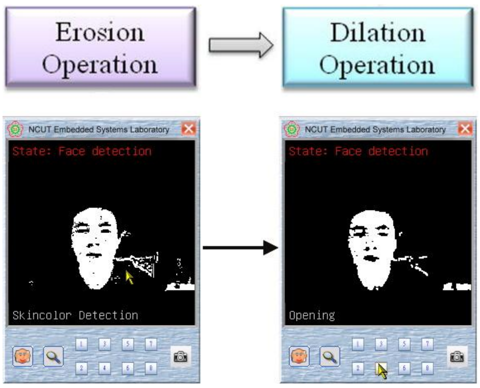


Fig. 4 Result of opening operation

2.1.3 Skin Color Information Processing

In this procedure, we separately process each skin color region in the image, and use connected-component labeling to number each different skin color regions as well as segment. The neighbor relationship between pixel and pixel can be classified as follows: 1) 4-neighbor, 2) 8-neighbor, and shown in figure 5. In this paper, we utilize 8-neighbor to do connected-component labeling.

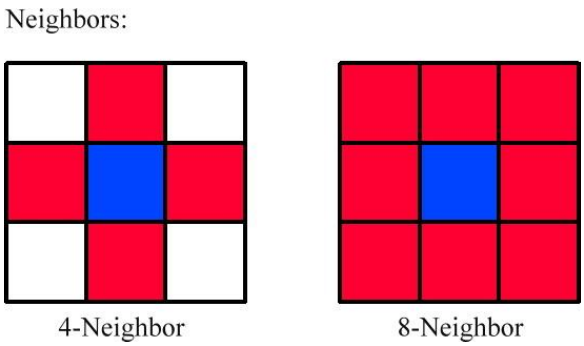


Fig. 5 Relation of 4-neighbor and 8-neighbor

The procedures of Connected-component labeling are described as follows:

1) The system scans image from top to bottom and from left to right. If the pixel has not been labeled, it will be marked a new number.
2) By recursion, pixel has been searched and decided whether it has neighbor pixel. If it does, we label the same number to neighbor pixel; otherwise, we exit recursion.
3) We repeat actions from 1) to 2) until finish scanning image.

When connected-component labeling has been done, we use regional skin color ways to calculate the total numbers of pixels for the largest area of skin color image. Then, we find top and bottom, left and right positions in this skin color region, and mark out the location of human face. The above-mentioned results are shown in figure 6。

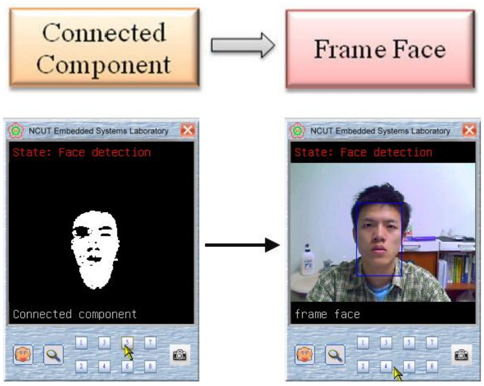


Fig. 6 Face location detection

After finding the location of human face, we join the width and height of human face for the following features' marking algorithms. The information of face is shown in figure 7.

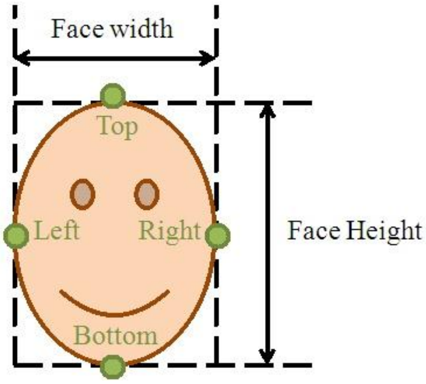


Fig. 7 Face of the information

After the above-mentioned image processing by the Statistics of the results can detect regional high-wide ratio of face reached maximum of 3. Generally speaking, face's high-wide ratio is approximately 1.2:1. It will appear smaller ratio is due to some of the samples would be influenced by darker glasses. Moreover, it also leads skin color region disconnected, and face area on the top and bottom boundaries became unclear. Next, methods of feature's marking are based on the location information of face, so we have to let face region boundary be more clear, our approach, such as equation (3).

$$face\_top = \begin{cases} face\_bottom - 1.2 \times Width & ,\text{if } \frac{Height}{Width} < 1.2 \\ face\_top & ,\text{if } \frac{Height}{Width} \geq 1.2 \end{cases} \qquad (3)$$

Using above-mentioned conditional equation can avoid the influence of glasses effectively.

2.2 Eyes Feature Marking

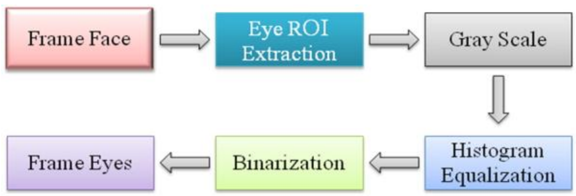


Fig. 8 The flow diagram of eyes feature marking

Driver fatigue recognition system can select accurately face position frame out; then, using a series of image processing actions to find the eye position and mark it, as in figure 8.

2.2.1 Region of Interest (Eyes)

After doing the previous image processing actions, we can get the initial location of a person's face, as well as the high-wide information of face.

Then, we will capture and quantify the eyes' feature by feature marking method.

Before searching for eye features, we define the region of interest, and leave the area that the eyes may show up. In this way, we can reduce the computational complexity and increase system performance. First, we move 8 pixels inward both left and right border of human face location to set eye searching left boundary eye_sl and right boundary eye_sr of ROI region.

Second we define eye searching top boundary eye_st and bottom boundary eye_sb, such as equation (4). According to equation (4) as well as left and right boundaries, we can calculate the search region of eye feature. Then, by equation (1), we can do RGB switching to grayscale images in this ROI image area, shown in figure 9.

$$\begin{aligned} eye\_st &= (face\_top + 0.2 \times Height) \\ eye\_sb &= (face\_bottom - 0.5 \times Height) \end{aligned} \qquad (4)$$

In equation (4), eye_st and eye_sb represent the top and bottom boundaries of the eyes of the search area, respectively, and the face_top and face_bottom denote the top and bottom boundaries of the face location. Finally, the "Height" is the height of the face.

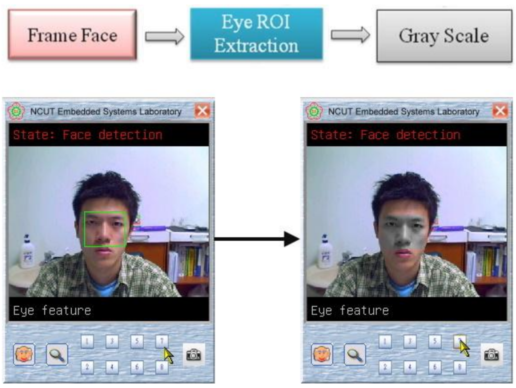


Fig. 9 ROI definition of eyes search and RGB transfer gray-scale

2.2.2 Histogram Equalization

When we capture images, the illumination of light source would affect the brightness level of face, so gray value distribution will much concentrate in certain gray sections. We process this step in gray-scale image of ROI. Histogram is the base for spatial processing technology[7], so the image processing methods of histogram equalization can make images of gray-scale distribution maps to be well-distributed (image enhancement). Therefore, the influence of brightness degree for the input image can be reduced and thus improve the accuracy of recognition. Results are shown in figure 10.

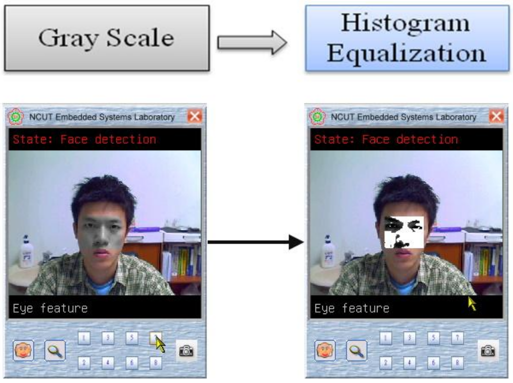


Fig. 10 Histogram equalization of ROI

2.2.3 Binarization and Eyes Marking

We can clearly observe the color of the pupil darker than the skin color around eyes in the input images. In the forecast eye area, we sort the gray-scale value of this area that is equalized by histogram. Next, the color of pupil and eye area will be the blackest within 4% in this region, and we use this grayscale value as the threshold for binarization. If the pixel value is less than the critical value in this region, so we set it to 1, otherwise 0.

Next, we divide it into two parts- left and right to process, and take right eye as an illustration. The image is scanned from top to bottom and from left to right, when the first pixel is scanned, its gray value is 1, and then we take out the x, y coordinate values of the pixel. Thus, we define x value as left boundary of right eye position reye_pl as well as y value as median value of right eye's height reye_pm. Furthermore, we use equation (5) to identify right boundary, top, and bottom boundary of right eye.

$$\begin{aligned} reye\_pr &= reye\_pl + 15 \\ reye\_pt &= reye\_pm - 3 \\ reye\_pb &= reye\_pm + 3 \end{aligned} \qquad (5)$$

The reason what we do is to avoid noises, such as hair, eyebrows or nostrils which would cause errors of judgment.

Similarly, as the same practice, the left eye is scanned from top to bottom and from right to left, then we can define right boundary of left eye position leye_pr and the median value of left eye height leye_pm, and we use equation (6) to identify the left, top, and bottom boundary of the left eye.

$$\begin{aligned} leye\_pl &= leye\_pr - 15 \\ leye\_pt &= leye\_pm - 3 \\ leye\_pb &= leye\_pm + 3 \end{aligned} \qquad (6)$$

After binarization processing and boundary scanning, the result is shown in figure 11.

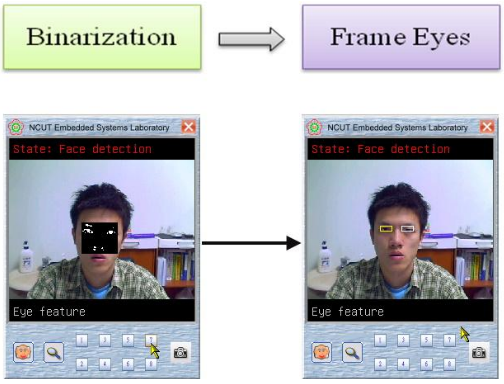


Fig. 11 Binarization and marked eyes

2.3 Lips Feature Marking

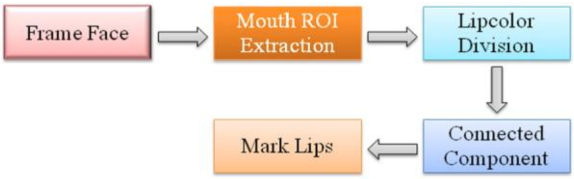


Fig. 12 The flow diagram of lips feature marked

Due to the characteristics of lip's color, it is much easier than eyes, when we deal with skin color segmentation and filtering noises. We identify and mark the location of lips by segmentation of lip color and connected component method. The processing is shown in figure 12.

2.3.1 Region of Interest (Lips)

When searching for the features of lip, we define the ROI of lips as the same as the eye, next we remain the area that the lip may present to reduce the operation amount and improve efficiency of the system. First, we inside move 4 pixels of the left boundary and right boundary of the face location, thus we set lip's left boundary lip_sl and right boundary lip_sr of ROI.

Next step is to define the lip's top boundary lip_st and bottom boundary lip_sb, such as equation (7). According to equation (7) as well as left and right boundaries, we can calculate the searching region of lip features, and compare the part of lip with the other region of face by the difference of red and green components. We calculate RGB values of ROI through equation (1) and utilize equation (8) to set out the lip color image from segmentation threshold.

$$lip\_st = (face\_top + Height \div 3)$$
$$lip\_sb = (face\_bottom + 8) \qquad (7)$$

Among, lip_st and lip_sb represent the lip's top and bottom boundaries of the search area respectively. The face_top and face_bottom mean top and bottom boundaries of the face location. The "Height" is face height.

$$1.6 \le (R \div G) \le 2 \qquad (8)$$

The results of image processing are shown in figure 13.

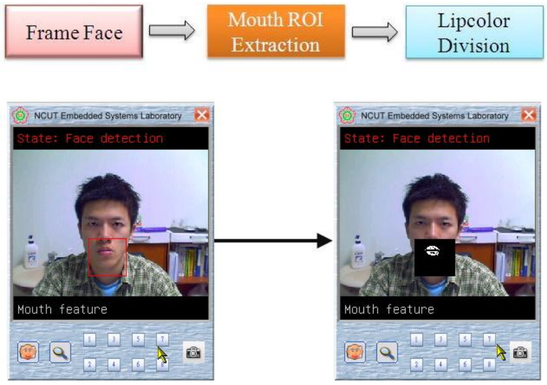


Fig. 13 ROI definition of lips searching and lips color division

2.3.2 Lips Color Information Processing

Then, we deal with each lip color area in the image separately; therefore, we give numbers to each disconnected region of lip color by connected-component labeling method described in section 2.1.3 and segment them. Next, we calculate the largest area of lip color by calculating the total number of pixels in the image, and then find out top and bottom as well as left and right positions of this lips color region. In this way, we can mark out the position of lips. The results are shown in figure 14.

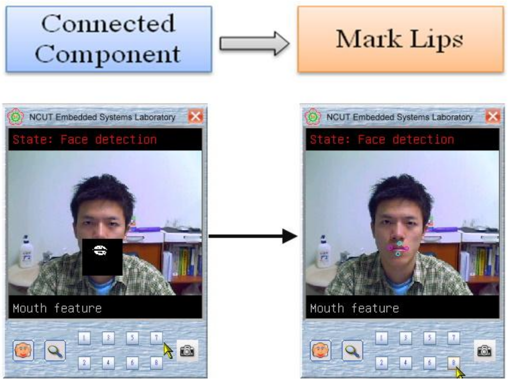


Fig. 14 Result of lips marked

2.4 Use Fuzzy Logic to Computing Fatigue Level

Fatigue is physiology phenomenon of fuzziness, and it is not objective and quantitative; moreover, each person's feelings are different. Therefore we employ fuzzy logic [8] to make computer to determine whether people are fatigued. The fuzzy input variations of fatigue detection system are constituted by blinking period (eyes closed frames) and blinking interval (eyes open frames). We combine the above-mentioned input variations with fuzzy logic; it will issue warning when the driver is fatigued, as shown in figure 15.

Until now, we can select out the eye feature precisely, so we set appropriate threshold based on the results of experiments. When we compute the white spot of ROI image which is higher than a certain value, it is frame of opening eyes; on the contrary, when it is lower than the certain value, it means closing eyes. On the basis of results, when we set the threshold value at about 30, the system can distinguish the eye which is opening or closing effectively.

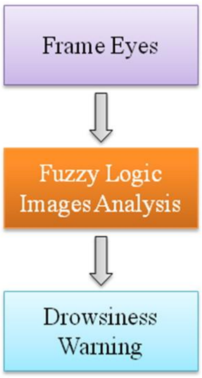


Fig. 15 The flow diagram of fatigue recognition

The system's three main structures of fuzzy control are fuzzification , rule-table and decision , defuzzification, the entire system architecture is shown in figure 16.

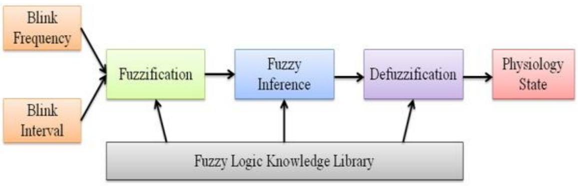


Fig. 16 Architecture of fuzzy system

In order to achieve good results, we must divide input and output fuzzy variables into partitions of appropriate number. We choose appropriate fuzzy variables to cover distribution space of input and output. For two input variables: blinking period and blinking interval and one output variable: physiological state of driver, we define three different levels of fuzzy sets. For each variable, we choose proper membership function. They are as follows:

blinking period ={short、medium、long}
blinking interval ={short、medium、long}
physiological state ={safe、caution、danger}

In this system, we use triangular membership functions, essentially, each different input variable and output variable have different distribution area or number of fuzzy variables with membership functions. This is determined by the designer. In this study, the degrees of input and output variables are defined as:

Degree of blinking period is defined as:
blinking period short = Triangular(0,0,6)
blinking period medium = Triangular(5,10,15)
blinking period long = Triangular(10,20,20)

Degree of blinking interval is defined as:
blinking interval short = Triangular (0,0,7)
blinking interval medium = Triangular (4,8,16)
blinking interval long = Triangular (11,20,20)

Degree of physiological state is defined as:
Safe = Triangular (0,0,4)
Caution = Triangular (3,5,8)
Danger = Triangular (7,10,10)

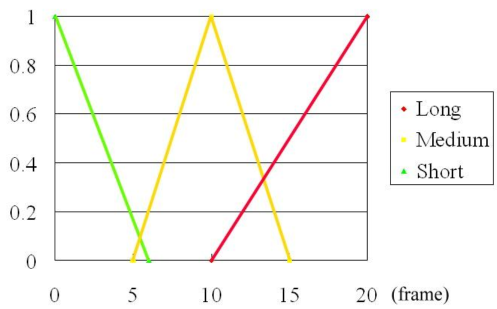


Fig. 17 blinking period

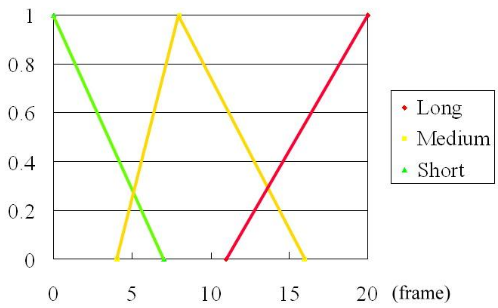


Fig. 18 blinking interval

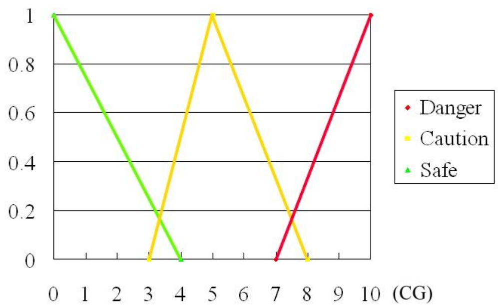


Fig. 19 physiological state

Figure 17 and figure 18 are input variables of blinking period and interval attribution of membership functions. (Green, yellow and red lines' covering areas of the input variables are short, medium and long respectively.) Figure 19 is the output variation of physiological condition attribution of membership functions. (Green, yellow and red lines' covering areas are safety, caution and danger.)

Then, by using Fuzzification approach, we can get accurate frame of opening and closing eyes by image processing. After that, we convert it to blinking period short, blinking period medium, blinking period long, blink interval long, blink interval medium, blink interval short, such fuzzy statements, then we use fuzzy statement to establish fuzzy rule data sheet

This rule is based on items written in spoken language, most of the general control rules are shown as "IF… ,THEN… ". This method makes the experts experience and knowledge to be combined with control rules easily. We select three levels of fuzzy sets for blinking period and interval, so we can get up nine control rules. Nine control rules are as following Table 1.

Table 1
Fuzzy rule data sheet

Rule1 : IF blinking period short and blink interval long THEN safe
Rule2 : IF blinking period short and blink interval medium THEN safe
Rule3 : IF blinking period short and blink interval short THEN caution
Rule4 : IF blinking period medium and blink interval long THEN caution
Rule5 : IF blinking period medium and blink interval medium THEN caution
Rule6 : IF blinking period medium and blink interval short THEN danger
Rule7 : IF blinking period long and blink interval long THEN danger
Rule8 : IF blinking period long and blink interval medium THEN danger
Rule9 : IF blinking period long and blink interval short THEN danger

Finally, by using center of gravity Defuzzification, we convert the fuzzy variables which are inferred by fuzzy logic into actual number and present it, and then we acquire the output value of fuzzy logic *CG*. Thus, we can recognize whether the driver is fatigued or not. In equation (9), CG is the output of fuzzy system. K is the number of triggered rules. Yi is the degree of control rule i. Bi is the membership function center of gravity of control rule i $(1 \le i \le K)$.

$$CG = \frac{\sum_{i=1}^{K} Yi \times Bi}{\sum_{i=1}^{K} Yi} \quad (9)$$

3. SYSTEM IMPLEMENTATION

3.1 Hardware Structure

In this paper, we utilize PC based hardware system, the specification data sheet such as Table 2. External installation is Logitech QuickCam E3500 (CMOS) webcam which helps us image capture of basic, and its maximum resolution is VGA (640x480).

Table 2
Hardware specification data sheet

Mother Board
CPU： Intel Pentium4 processor 533/400 MHz FSB
Memory：DDR266 (PC 2100) / DDR200(PC1600) 184 pin (2.5v)
USB Host and Device Port
TFT LCD

3.2 Software Structure

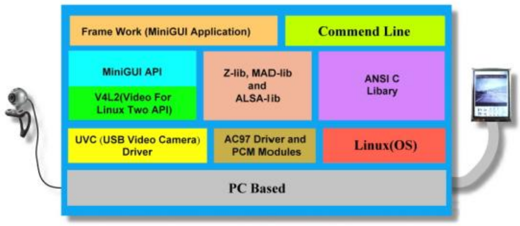


Fig. 20 Software structure

In this paper, the software architecture is shown in figure 20. The UVC (USB video camera driver) driver is a webcam driver of LINUX. It makes the user utilize webcam on LINUX platform by UVC driver. Through USB video camera driver, we store the video streaming (motion JPEG, AVI Format) in the register. Then, we store JPEG (AVI Format) which has not been joined Huffman coding into the temporary block of memory by utilizing MiniGUI function and API of video for LINUX two (V4L2) [9]. Furthermore, we insert Huffman coding table and JFIF information into temporary block of memory and we use MiniGUI [10] to provide windows component and graphic function to display images on the monitor through frame buffer qvfb of Linux.

Table 3
Libraries which described in this paper

Name of libraries	ANSI C	V4L2	MiniGUI
Explanation	Library of standard C language.	Access video-information bit stream of webcam.	Build and construct the GUI interface.

"User application program" is composed of two parts of library and application program. Library includes three components, image capture library, GUI (Graph User Interface) program library and ANSI C library. Table 3 shows the specific functions of these three components. MiniGUI application was written by C language and executed under Linux operation system. The functions of the designed application program include image capture, driver fatigue recognition algorithm and GUI procedure.

3.3 Software Development

There are two essential steps to building and constructing our system except the selection of the hardware platform and operating system at the beginning.

1) Image Capture in Linux: The V4L2 is an API for executing the image capture function under LINUX. It only needs hardware driver to offer input and output function (ioctl) and makes the image capture programming easily. Figure 21 shows the operation procedures of V4L2. We can use timer function of MiniGUI to set up capture frame quantity in every second.

2) Building GUI Framework: At present time, most systems have a graphic user interface. We choose MiniGUI open source code to implement the graphic user interface (GUI) for plentiful user operating interface. The reason is not that MiniGUI has special fortes but its code size is smaller than general GUI open source code. The code size is one of the decisive conditions while we implement a real-time system. Besides, the system designer must also consider the question in many aspects such as the cost, hardware resource of the system and user's habit, etc.

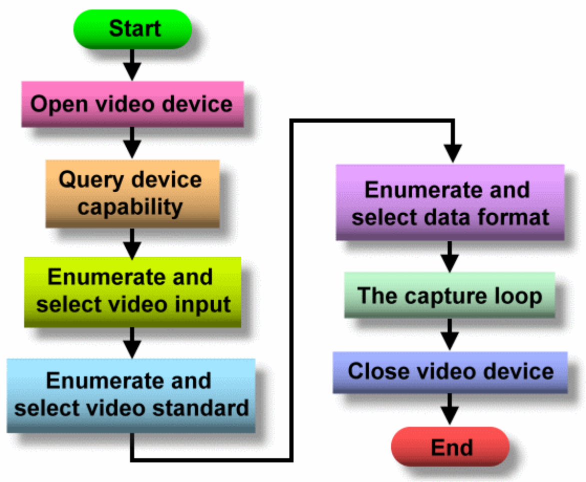


Fig. 21 The operation procedures of V4L2

4. EXPERIMENTATION AND RESULTS

In this paper, the camera is set up on LCD monitor. User being in front of the monitor imitates the situation of driver sitting in front of dashboard when driving. The camera faces to driver to catch images, such as figure 22.

Fig. 22 System development of simulation

Figure 23 is results of driver fatigue recognition. We use rectangular blue box to frame the face and white and yellow box to frame left and right eyes. In this paper, we can recognize driver's physiological state by combining with eye's feature and fuzzy logic and warn driver when they are fatigued.

According to (a) of figure 23, when the driver's physiological condition is normal in the screen, the system will determine that the driver is conscious; moreover, it will show as safety in the graphic user interface. Otherwise, when the driver is slightly fatigued, the system will change safety into caution to remind driver, as shown in (b). Furthermore, the system will display danger to warn the driver to pay much attention or take a rest when the driver is tired seriously or drowsy, as shown in (c).

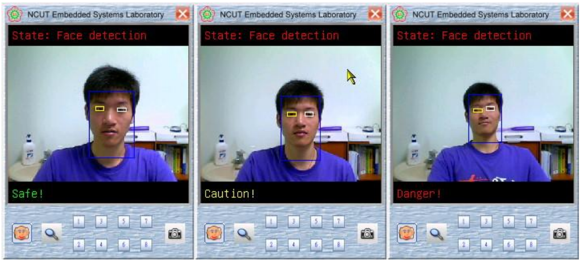
(a) (b) (c)

Fig. 23 Recognition result of driver fatigue

Figure 24, figure 25 and figure 26 are different samples of driver fatigue recognition. The system can work well on different drivers and analyze drivers' physiological condition precisely. It will remind or warn the driver according to his level of fatigue.

(a) (b) (c)

Fig. 24 Recognition result of driver fatigue (Sample 1)



(a) (b) (c)

Fig. 25 Recognition result of driver fatigue (Sample 2)

(a) (b) (c)

Fig. 26 Recognition result of driver fatigue (Sample 3)

5. CONCLUSION

This paper presents a system of drowsiness detection for driving car. Its main functions are face detection, feature extraction, warning of fatigue, and photograph for recording.

The system can find the positions of face and features in different light conditions and backgrounds and further issue warning. In the future, we still have to add up much environmental factors to improve the system and make it nearer to commercialization.

REFERENCE

[1] Pavlidis, I.,& Morellas, V.,& Papanikolopoulos, N., “A vehicle occupant counting system based on near-infared phenomenology and fuzzy neural classification” Intelligent Transportation Systems, IEEE Transactions, June, 2000.

[2] Bird, N.D.,& Masoud, O.,& Papanikolopoulos, N.P.,& Isaacs, A., “Detection of loitering individuals in public transportation areas” Intelligent Transportation Systems, IEEE Transactions, June, 2005.

[3] Smith, P.,& Shah, M.,& da Vitoria Lobo, N. , “Determining driver visual attention with one camera” Intelligent Transportation Systems, IEEE Transactions, Dec, 2003.

[4] Bergasa, L.M.,& Nuevo, J.,& Sotelo, M.A.,& Barea, R.,& Lopez, M.E., “Real-time system for monitoring driver vigilance” Intelligent Transportation Systems, IEEE Transactions, March, 2006.

[5] Qiang Ji,& Zhiwei Zhu,& Lan, P., “Real-time nonintrusive monitoring and prediction of driver fatigue” Vehicular Technology, IEEE Transactions, July, 2004.

[6] E. Hamilton, JPEG File Interchange Format, C-Cube Microsystems, September, 1992.

[7] Gonzalez,W.(2007), Digital Image Processing 3/e. Taipei：Princeton.

[8] Wang, W.G., Realization Fuzzy, chwa books corporation, Taipei, Taiwan, June, 2007.

[9] M. H. Schimek. (2008) V4L2 API Specification. [Online]. Available: http://v4l2spec.bytesex.org/

[10] (2008) Beijing Feynman Software Technology Co.,Ltd.[Online].Available:http://www.minigui.com/