ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

Abnormal Driver Behavior Detection Using Parallel CPU and GPU Algorithm through Facial Expression, Thermal Imaging and Heart Rate Data Fusion

Dino Isa, Goh Chia Chieh, RoselinaArelhi

Abstract*—Abnormal driving recognition is a complicated task. It poses an even greater challenge if such a system is to be implemented in a limited space. An efficient method to implement such a system with low power consumption while maintaining optimal recognition accuracy is proposed. The task is split into multiple threads which can be processed simultaneously in a parallel CPU (Central Processing Unit) and GPU (Graphic Processing Unit) architecture on a Single Board Computer (SBC). If the abnormal driving behavior is detected, the system will send the facial expression data, together with the current location of the vehicle to a dedicated server via broadband connection. Such data can then be retrieved through any Smartphone device by an authorize user such as personnel in law enforcement and security.

***Index Terms*—CUDA, Facial recognition, Heart rate monitor, Parallel CPU and GPU, Real-time emotion recognition, Thermal imaging.**

I. INTRODUCTION

Road accidents are a global menace in every country. According to the latest statistic compiled in 2012 by the United Nations (UN) [1], over 1.27 million cases of fatalities were reported in 2004. The figures increase at an alarming rate every year. If no action is taken, the number of fatalities will increase to 2.4 million by 2030 [1]. A significant portion of these accidents are caused by drivers being intoxicated above the legal limit. The research described here addresses this issue along with other issues related to driver health and behavior which can be detected through analysis of the data fusion between facial expression, thermal imaging and heart rate information. Analysis of this data is greatly facilitated by using hardware and software which take advantage of parallel processing between an on board CPU and GPU. Due to its high impact, much research has been carried out to prevent or lower the rate of accidents [2] [3] [4] [5] [6] [7]. With the advancement in technology related to memory, processing speed, data transfer bandwidth and task parallelism architecture implementation [8][9] [10], Artificial Intelligence (AI) has found a wide-spread adoption within the smart vehicle research community [11] [12] [13] [14]. A main drawback of all these techniques including those using an inference engine for pattern recognition is the performance of the visual expression recognition system which suffers from unpredictable performance due to lighting conditions in an uncontrolled environment [15] [16] [17]. In this project this issue is addressed by fusing data from three different sources namely a visual camera, a thermal camera and a wireless heart rate monitor. Recent studies have shown that thermal based imaging offers an alternative to normal face recognition [18] [19] [20] [21]. It also offers greater advantage over normal webcams in term of susceptibility to poor lighting conditions. Although the fusion of both thermal and visual face data can be processed to detect emotion [22], the accuracy in facial expression detection is degraded by small changes in a subject's appearance, such as when the eyeglasses are worn. Besides this, it is difficult to distinguish aggressive emotions by solely monitoring the fused visual and thermal data alone. In this research heart rate monitoring is used to obtain an extra dimension of information thus improving overall system accuracy. For the Smart Vehicle System proposed here, the visual, thermal image and heart rate monitoring systems provide a more accurate judgment of a specific aggressive emotion as opposed to results obtained from visual and thermal data fusion alone. By monitoring pulse rate in the time domain, different rates of changes in heart beat can be used to infer different emotions. With the combination of visual, thermal, and heart rate data, a more precise result can be derived in the final output. This research overcomes the challenges of the accuracy of the visual face recognition system acting alone in a moving vehicle by combining the functionalities of the latest hardware and software technology based on parallel processing and parallel analysis of facial expression data and thermal webcam image. The heart rate data need not be processed in parallel due to the fact that the data rate and volume is small compared to the vision and thermal imaging data. Hence, as stated above, only the vision and thermal data were processed in parallel. The core system in the smart vehicle is the sensor and monitoring system which are linked to the centralized embedded system known as the event data recorder (EDR). The two main differences between the traditional event data recorders [23] [24] and the revised version which is installed in the vehicle used in this research are as follows:

1. The driver's face, heart beats and emotions are monitored and recorded in real-time. On other systems, usually only one of the three characteristics above is monitored [25] [26] [27].
2. There is instant broadband or wireless connection access in case of emergency. This connectivity is used to send a continuous visual face expression, thermal and heart rate data to a server for real time processing. In other systems

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

broadband connectivity is only used to SMS or with GSM based system which function to alert authorities in an emergency [28] [29].

Another distinguishing feature is that it is embedded with an AI system which attempts to predict occurrence of a life-threatening incident and alerts the driver accordingly. For example if a driver is intoxicated, the heat dissipated from his/her body will differ from the norm and the heart rate will also be slightly increased. Based on the judgment from the AI, it will then warn the driver to discontinue driving. Another very common problem is that an intoxicated driver easily feels drowsy, and thus has a higher chance of falling asleep during driving. The camera which is fixed in front and facing the driver seat will detect such facial feature changes related to sleepiness and give an audible warning to wake the driver. As mentioned previously, the system is embedded with an AI engine to predict occurrences of dangerous situations and dangerous driving. The AI engine in questioned is implemented using a support vector machine (SVM) [30] [31] [32] [33]. The Support vector machine is a supervised learning method used for classification and regression. A classification algorithm is used in this research. As shown in Fig 1. There are two groups of data, **x (**darker dots) and **o (**lighter dots). The main objective is to separate the two groups of two dimensional data (assigned the label as -1 and +1) in a higher dimensional space such that the nonlinear data which could not be separated in two dimensional space can now be separated in feature space. The aim here is to obtain the maximum margin between the hyperplane in Hilbert space. Those dots which touch the margin (dotted line) are known as support vectors. There are certain cases where some data are misclassified. This is where a soft margin C is implemented to allow certain error tolerance between two groups of data so that the data may be correctly classified after a few iterations. The Kernel trick is used to process data which is not linearly separable [34] [35] [36]. The common kernels for support vector machine are polynomial, Radial Basis Function (RBF) and Sigmoid function. The RBF kernel gives the best result in this research as compared to polynomial and Sigmoid function.

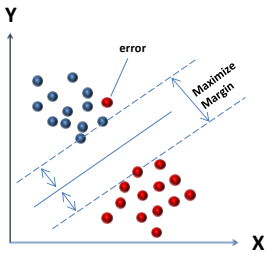


Fig. 1 Support Vector Machine classification

The objectives of the research are as follows:

1. To design a system for continuous monitoring of the driver's behavior and emotion.
2. To design and install the event data recorder to aid in automobile–related accident investigations.

I. SYSTEM OVERVIEW

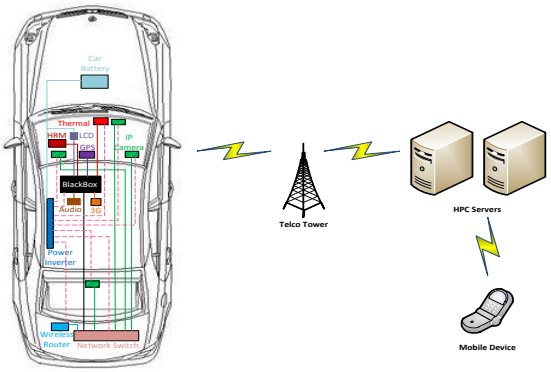


Fig. 2 System Overview[1]

The embedded system as shown in Fig. 2 consists of 6 inputs and 2 outputs. The 6 inputs are as follows:

Table I System I/O's

Device	Function	Methodology/algorithm
Thermal camera	Monitoring heat dissipated from human body	Parallel CPU + GPU based on multithreaded task parallelism architecture. (Referred to section 3 and 4C)
Camera	Monitoring facial expression	Parallel CPU + GPU using Support Vector Machine (SVM) classifier. (Referred to section 3, 4A and 4B)
Heart Rate Monitor	Monitoring heart rate	Sequential CPU architecture implementation based on Garmin ANT wireless device transfer. (Referred to section 3 and 4D)
GPS Unit	Track current vehicle location	Parallel CPU implementation using NMEA 0183 Protocol. (Referred to section 3)
LCD touch screen	Displays map, location, emotion	-
Microphone	Speech input	Built in Microsoft® OS Speech recognition. (Referred to section 3)

The 2 outputs from the Event data recorder embedded system are:

1. Wireless Network (sending SMS and data to link to the outside world for emergencies)
2. Sound (to alert the drivers)

The centralized server handles incoming data such as the speed, location of each vehicle (also known as agent). This system is designed for customizability and flexibility. It is also expandable for future development. Modules are optional for different vehicles, and each module can be customized to fit specific cars on the driver's demand.

A. Hardware Setup

The event data recorder system was designed on a 2.6GHz Intel® Centrino dual core mobile processor, running on windows 7 64 bit OS, with 4 Gigabytes of RAM and two NVidia® 8800M GTX GPU graphic card with Scale Link

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

Interface (SLI) mode disabled. The 8800M GTX consists of 768MB RAM and 6 streams multiprocessor cluster where each stream multiprocessor consist of 16 stream processors, giving a total of 96 stream processors. Each multiprocessor has 8192 32-bit registers and can have up to 768 threads. Threads are partitioned into thread blocks which consist of a maximum of 512 threads, and can be further divided into warps of 32 threads. For the 8800M GTX, the core clock frequency is 500MHz, the shader frequency is 1250MHz and the memory frequency is 800MHz. [37] The thermal camera has a white hot IR polarity; the brighter spot represents higher temperature. The data were transferred through Firewire (IEEE1394 communication protocol) connected to the event data recorder itself. The wireless heart rate monitor uses the Garmin ANT protocol for communication. [38] The data is transferred through the USB (Universal Serial Bus) from the event data recorder.

II. SYSTEM ARCHITECTURE

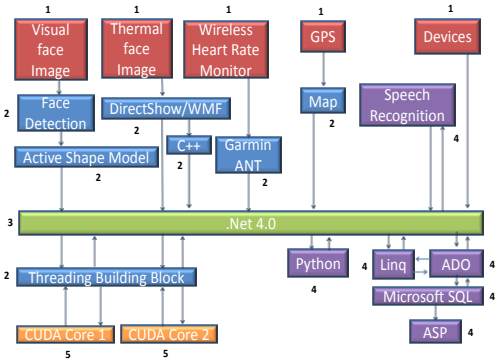


Fig. 3 System Architecture

The event data recorder system architecture can be divided into 5 layers. As shown in Fig. 3, the 1st layer is known as the hardware layer, this is where the video camera, thermal camera, GPS, and devices are. Device block is further divided into 3G broadband and LCD touch screen. This modular block can be expandable in the future to cater for modular elements to be added into the system. All the external hardware devices input (1st layer block) were connected to USB 2.0 port and IEEE 1394 (Firewire) port on the event data recorder (in the 1st layer). The maximum bandwidth shared throughout the Universal Serial Bus (USB) bus is approximately 280 Megabit per second (Mbps). However, this transfer speed will decrease if more than one hardware devices were connected to the USB bus. The IEE1394 transfer rate is around 800Mbps. The bandwidth priority is allocated to the hardware devices based on priority. For example, since visual camera needs a high speed continuous picture streaming, the data bandwidth from the USB port will be given a higher priority compare to IEEE1394 protocol. These are the limitations that affect the performance of the system and need to be taken into serious consideration throughout the research and development. The 2nd layer is the software library which handles the interface from the hardware such as video streaming from the visual camera, heart rate data from heart rate monitor, and GPS data. This layer can be further classified into two groups: Time Critical and Non-time Critical Computing. Time critical task requires real time processing and instantaneous response based on the given input. A good example is the facial expression. In order to prevent any life-threatening incident happened to the driver while driving; the system has to respond to any abnormal behaviors from the driver within split a second time frame. Due to the poor runtime performance in .Net framework [39], time critical task such as facial expression recognition was written in C++, while other non-time critical task was written in C# to promote processing efficiency. Other time critical items like the main core architecture DirectShow/WMF [40], facial detection and Facial landmark alignment are written in C++; non-time critical tasks, such as data fusion and graphic user interface (GUI) are also written in C#. The 4th layer is the layer for scripting plugins and Database communication. Language-Integrated Query (LINQ) and ActiveX Data Objects (ADO) software layer handles queries from and to the Microsoft SQL server® [41]. All the data are kept in the event data recorder and a backup copy are transmitted to a dedicated server. If broadband coverage is not available in some area, the data will be in queued and be transmitted once the vehicle is back on broadband coverage. The Python integration will act as a script plugin for modular expansion; this will enable rapid software development for future modular plugin. Currently python [42] script was integrated with the event data recorder to handle system settings and module device load balancing management. Speech recognition was integrated into the system based on the built-in function from Windows OS. The system is programmed to detect extreme voice changes such as shouting. Some pre-programmed command such as "send it", "retrieve it" and "show me the map "are used to interact with the system while the driver is driving. Once an abnormal tone is detected, the system will switch to panic mode to take control of the dangerous situation automatically. The data will be transferred to the server as needed. The 5th layer is the Compute Unified Device Architecture (CUDA) programming model, which consists of two main parts, known as Host and Device. The Host is the CPU and the Device is the GPU. The data were transferred from the Host (CPU) to the Device (GPU) in a single-program, multiple data (SPMD) for preprocessing. Thread building block [43] is used to transfer the array of data from the parallel CPU to the CUDA core 1 and CUDA core 2 for parallel processing. The final results were then transferred back from the device to the Host. The data which resides in the Host (CPU) code were compiled using standard C++ Compiler, while the Device (GPU) coding was compiled using the CUDA compiler based on ANSI C.

III. METHODOLOGY

A. *Facial Expression Image*

Human facial expressions convey emotion information which plays an important role in non-verbal communication. Humans can recognize any facial expression with minimal effort, but machine recognition relies on several factors

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

which affect the face appearance performance. Such factors include light illumination, pose, scale, occlusion and position. Moreover, it takes up vast amount of CPU power in order to process a sequence of facial data. Facial Action Coding System - FACS [44] objectively describes and measures facial expressions and movements of each individual. A mesh is created for each facial expression; each mesh is an array of matrix which consists of rows and columns known as m x n matrix:

$$\begin{bmatrix} Scale_x & Skew_y & 0 \\ Skew_x & Scale_y & 0 \\ Translate_x & Translate_y & 1 \end{bmatrix}$$

Where

$Scale_x$ = zoom in and out in terms of X position

$Scale_y$ = zoom in and out in terms of Y position

$Skew_x$ = Angle in terms of X position

$Skew_y$ = Angle in terms of Y position

$Translate_x$ = movement in terms of X position

$Translate_y$ = movement in terms of Y position

Therefore we can define the formula of a new coordinate for X and Y as:

$$X' = Scale_x * X + Skew_x * Y + Translate_x \quad (1a)$$

$$Y' = Skew_y * X + Scale_y * Y + Translate_y \quad (1b)$$

For rotation of new X and Y coordinate:

$$\begin{bmatrix} \cos\delta & \sin\delta & 0 \\ -\sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X' = \cos\delta * X - \sin\delta * Y + 0 \quad (2a)$$

$$Y' = \sin\delta * X + \cos\delta * Y + 0 \quad (2b)$$

Based on this assumption, we can train the data for the facial image set to detect any new face. The new facial landmark will be compared with the pre-built a statistical model to align the facial landmark as closely as possible. As shown in Fig. 4 is a simplify 8 point landmarks. The lighter color points are the pre-built statistical model landmark points. The darker color points are the unknown data. The concept is like an air balloon, where every point can be increased or decreased to fit the size of a facial image. Each individual point is labelled with unique numbers from 1 to 8. The main objective is to fit the lighter color to the darker color data points. The balloon like data points starting from the center of origin expand outwards until all individual points converges.

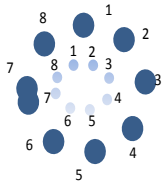


Fig. 4 Point To Point Superimpose Example

The number of each individual point cannot be swapped. For example point 1 of the lighter color point must match point 1 of the darker color. The same process continues until point 8. The same concept applied to facial image where the first facial image is marked manually with 64 landmark point. It is then trained with 200 hundred images with the same facial scale of 100 x 100 pixels. All the training images were taken from the FERET image database [45]. 200 hundred images points were gathered to form a statistical model. This is a modified version of the Active shape model (ASM) [46] which uses the Point Distribution Model [47] to represent a shape. The algorithm steps were as follows: Let S1 be the first set of landmark points of a facial image, S2 is the second set which was needed to align with S1:

$$S1 = \begin{pmatrix} x1, x2, x3 \ldots xm \\ y1, y2, y3 \ldots ym \end{pmatrix} \quad (3)$$

$$S2 = \begin{pmatrix} x1, x2, x3 \ldots xn \\ y1, y2, y3 \ldots yn \end{pmatrix} \quad (4)$$

$$\text{Mean } \bar{x} = \frac{S2_x}{m} \quad \bar{y} = \frac{S2_y}{m} \quad (5)$$

Translation as compared to the origin:

$$(x_1 - \bar{x}) \cdots (x_m - \bar{x}) \ (y_1 - y) \cdots (y_m - \bar{y}) \quad (6)$$

$$Scale = \sqrt{\frac{(x_1 - \bar{x})^2 + (y_1 - \bar{y})^2 \cdots (x_m - \bar{x})^2 + (y_m - \bar{y})^2}{m}} \quad (7)$$

To scale the S2 as the same ratio as S1:

Compose the Singular Value Decomposition (SVD) of the matrix:

$$S1' = \begin{pmatrix} x_1 - \bar{x}, \cdots x_m - \bar{x} \\ y_1 - \bar{y}, \cdots y_m - \bar{y} \end{pmatrix} \quad (8)$$

$$S2' = \begin{pmatrix} x_1 - \bar{x}, \cdots x_m - \bar{x} \\ y_1 - \bar{y}, \cdots y_m - \bar{y} \end{pmatrix} \quad (9)$$

To obtain the Frobeinus Norm, Substitute equation (8) and (9):

S1'' = Sum of **S1'*S1'** (10)

S2'' = Sum of **S2' * S2'** (11)

Frobenius Norm F1 = $\sqrt{S1"}$ (12)

Frobenius Norm F2 = $\sqrt{S2"}$ (13)

$$\text{Matrix multiplication A} = (S1'X \ldots S1'Y) \begin{pmatrix} S2'X \\ \vdots \\ S2'Y \end{pmatrix} \quad (14)$$

Compute SVD from A to return three 2x2 matrices of U[2][2], S[2][2] and V[2][2]

Take the diagonal sum of matrix S' = S[1][1] + S[2][2] (15)

If the point is scaled the Distance: $1 - S'*S'$ (16)

If the point is not scaled the Distance:

$$\frac{1 + S2"}{S1"} - \frac{2 * A * F2}{F1} \quad (17)$$

Table II shows the steps to compute the algorithm using multi-GPU and CPU Cores:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

Table II. Algorithm for Landmark Alignment Using Multi GPU and CPU

1. From the visual camera image 512 x 512 pixels, track the facial image. From the facial image check for strong edge. The new facial 64 landmark points (name as *F1*) are as follows:

   I. Outer facial shape = 15 points
   II. Left and Right Eyebrow = 6 points each
   III. Left and Right Eye = 4 points each
   IV. Nose = 12 points
   V. Mouth = 17 points

2. Create a task and name it as Task1 using Task parallel Library (one of the .Net framework 4.0 features) and split the threads into two threads using threading building block so that it can be processed using CUDA 1 and CUDA 2.

3. Set Block , Size and ThreadBlock for CUDA GPU:

   I. CUDA_Block = 64

   II. CUDA_Size = 128

   III. CUDA_ThreadBlock = Block * Size

4. Place the pre-trained statistical facial 64 landmark points (name as *T1*) at the center of the origin of *F1* (below the eyes and slightly above the nose area):

   Shift the matrix array of F1[64][64] and T1[64][64] to GPU CUDA 1 and CUDA 2 Global memory to compute the mean and transfer the final value back to CPU and name it as Mean_F1 and Mean_T1.

   a. Similar steps for equation (8) to (11) to obtain the Frobenius norm F1 and F2.

   b. From 4(b) scale the matrix T1 to the same ratio as F1

5. Once the center of origin is obtained, the next step is to expand the pre-trained points in a *balloon likeform* by doing translation and scaling to fit the pre-trained T1 points to the original F1 points:

   a. Shift the matrix S1' and S2' to CUDA 1 and CUDA 2 global memory to calculate the matrix multiplication A and transfer the matrix back to the CPU.

   b. Shift the matrix A to CUDA to get the SVD. With return value of 2x2 matrix of U[2][2] , S[2][2] and V[2][2].

   c. Check for translation to see how close we are to the original F1 points.

   d. Scale the points using equation (7).

   e. Check the distance of each point between Starting from F1 point 1 to point 64 and T1 point 1 to point 64.

   f. If the distance is > 1, use equation (15) , else if distance is > 2 use equation (16)

Move or superimpose the point of T1 to F1. If it does not converge go to equation 5(a) again.

B. MultiClass Support Vector Machine (SVM)

In order to solve a Multiclass SVM, the most common methods are One Against One and One Against All. One Against All method is chosen in this research. The concept of One against All is shown in Fig. 5 and Table III. There are three classes, Happy, Sad and Neutral. Here are the possibilities:

Table III Multiclass SVM

Facial Expression	Happy Classification	Sad Classification	Neural Classification
Happy	+1	-1	-1
Sad	-1	+1	-1
Neutral	-1	-1	+1

- Data input sample falls in the neutral category (Blue), classifier y = +1, Happy and Sad will be y = -1.
- Data input sample falls in the happy category (Green), classifier y = +1, Neutral and Sad will be y = -1.
- Data input sample falls in the Sad category (Red), classifier y = +1, Neutral and Happy will be y = -1.
- Data input sample falls in between Happy and Sad (Between red and green line), the decision will be given to which the training sample is closer to either happy or sad category. If the data input is closer to the red line than green, the weight carries for the red will be more than the green. Therefore the data input will be categorized as sad.

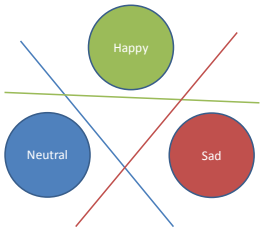


Fig. 5 Multiclass SVM

The same process continues for the rest of the facial expression until all the classification such as Happy, Neutral, Sad, Angry and surprise expression were classified.

In order to solve for an SVM classification:

$$\frac{1}{2}W^T W + C\sum_{i=1}^{l}\xi$$

Where C > 0;

Subject to constraints: $\xi_i \geq 0, y_i\left(W^T x_i + b\right) \geq 1 - \xi_i$

ξ is known as slack variable

C is known as soft margin

W is known as weight vector

This is an optimization problem, with two constraints; the main objectives are to find α (support vectors). Once the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

support vectors were known, W and b can be solved. By using Lagrange multiplier, the equation can be rewritten as:

$$L(W,b,\xi,\alpha,\beta)=\frac{1}{2}W^TW+C\sum_{i=1}^{l}\xi_i-$$
$$\sum_{i=1}^{l}\alpha_i\ [y_i\left(W^Tx_i+b\right)-1+\xi_i]-\sum_{i=1}^{l}\beta_i\xi_i$$

Where $\alpha \geq 0$, $\beta_i \geq 0$

The equation can be rewritten as:

$$L_p=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\ \alpha_i y_i KY\alpha^T$$

The facial expression classification algorithm using SVM is as follows:

Table IV Algorithm to classify facial expression through SVM

1. X1 = *happy* and X2 = *sad* equal to the number of sample training where:

   X1 = 17 points x 100 = 1700 array, X2 = 17 points x 100 = 1700 arrays.

2. Assigned Y1 = +1 for happy expression classification, Y2 = -1 for all the rest of the classes Y1 = 17 x 100 * 1, Y2 = 17 x 100 * -1.

3. Set 80% of X1 , X2 = 17 x 80 = 1360 points for training, 20% of X1 , X2 = 17 x 20 = 340 for predicting. 80% of Y1 for training, 20% of Y2 for predicting.

4. Combine 80% X1 matrix with 80% X2 matrix training, 20% X1 matrix with 20% X2 matrix for predicting. Combine 80% of Y1 for training, 20% of Y2 for predicting

5. Let RBF kernel = $e^{-\left(\frac{\|X-Y\|^2}{2\sigma^2}\right)}$, C = 0.1, $\sigma = 0.5$

6. Compute the gram matrix for the RBF kernel from X1 and X2, where X1 = X of RBF, X2 = Y of RBF

$$K\begin{bmatrix} X1_1X2_1 & \cdots & X1_1X2_n \\ \vdots & \ddots & \vdots \\ X1_1X2_n & \cdots & X1_nX2_n \end{bmatrix}$$

7. Using Mosek QP solver with 6 variable P, q, R, s, T, u represents the following equation:

$$\frac{1}{2}X^TPx+q^Tx$$

Where:

$R_x \leq s$

$T = u$

From y_train = [y1…..yn], y_train = [y'1…..y'n]

$$P=\begin{bmatrix} y1y'1 & \cdots & y1y'n \\ \vdots & \ddots & \vdots \\ yny'1 & \cdots & yny'n \end{bmatrix} * K\begin{bmatrix} X1_1X2_1 & \cdots & X1_1X2_n \\ \vdots & \ddots & \vdots \\ X1_1X2_n & \cdots & X1_nX2_n \end{bmatrix}$$

$$q=\begin{bmatrix} -1 & \cdots & -1 \\ \vdots & \ddots & \vdots \\ -1 & \cdots & -1 \end{bmatrix}$$

8. After the QP calculation α_i can be obtained. Using KKT rules discard the α_i which is not within 0 and C, that is $0 < \alpha_i < C$

9. Calculate $W = \alpha_i X^T Y$

10. Calculate b

$$b=\frac{1}{Total\ number\ of\ support\ vector\ samples}$$
$$\sum_{m\in s}^{l}\left(y_s-\sum\alpha_i X^TYx_s\right)$$

11. The same process is repeated for neutral, surprise and angry expression.

During prediction, an unknown data points are obtained from the camera. The prediction data is then compared to the trained data points to predict the facial expression from either 1 of the 5 facial expression, angry, sad, neutral, surprise or happy.

C. Thermal Image

The histogram can be used to represent the color distribution of an image as well as the frequent occurrence of each color pixel element. The thermal camera has 16-bit dynamic range, with an operating temperature range from -20 degree Celsius to 120 degree Celsius. The average normal human temperature is at 37 degree Celsius. Thus, by converting to 16 bit gray scale (range from 2 to the power 16 = 0 - 65535 gray colors) to represent the Thermal camera range from -20 to 120 (140 steps) degree Celsius, each degree Celsius increment will be:

65535/140<= 468.107(every 1 degree Celsius increment)

468.107 Per degree Celsius * 25 degree Celsius<= 11702.678

468.107 Per degree Celsius * 25 degree Celsius<=17788.071

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

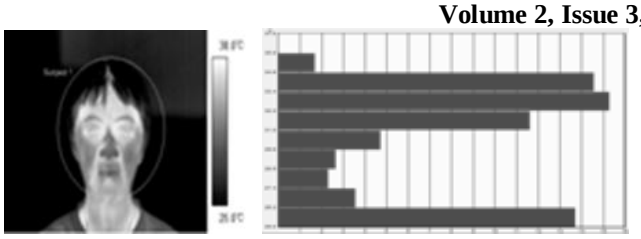


Fig. 6 Normal Human Temperature Histogram (Temperature In Descending Order)

The thermal camera was installed approximately 1 foot from the driver; the temperature dissipated from the body is directly proportional to the distance of the placement of the thermal camera. The further the camera was placed, the less accurate the temperature was. As shown in Fig. 6 above, the histogram sorted in a descending order from top (highest temperature) to bottom (lowest temperature). The temperature ranged from 25 degree Celsius to 38 degree Celsius.

Table V Algorithm for Histogram computation using Multi GPU and CPU

1. Create a task and name it as Task2 using Task parallel Library (one of the .Net framework 4.0 features) and splits the thread into two threads using threading building block so that it can be processed using CUDA 1 and CUDA 2.
2. Compute Thermal Image of array size 640 x 480
3. Transfer the Histogram data from Host to Device and parallel process the sum.
4. Global memory:
   a. Let *I* = blockIdx.x* blockDim.x + threadIdx.x
   b. Let *J* = blockDIm.x * gridDim.x
      i. While *I* smaller than Step 1 size
      ii. Sum the histogram with atomicAdd function
      iii. *I* = *I* + *J*
5. Transfer the histogram data from the Device to Host
6. Take the top two highest histogram temperatures and compare with normal human temperature. If the sum is more than 40% of normal range, categorize as abnormal temperature.

D. Heart Rate

The Heart rate measurement is another key feature for tracking or diagnosing certain medical symptoms. As shown in Fig.7, the number of heartbeats per unit in a unit time is known as heart rate. One heartbeat peak to another is known as the R-R interval. Heart rate is measured in terms of Beats per Minute (BPM). The average for heart beat per minute is 70 BPM for the first peak, 76 BPM for the second and 83 BPM for the third peak. The R-R interval for the first unit interval is 0.659 second, followed by 0.793 second and 0.726 second. (This project used only information from the first R-R peak.) One can observe that the peak heart rate for different peaks is not consistent.

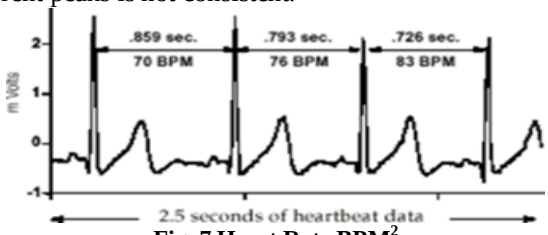


Fig. 7 Heart Rate BPM[2]

There are various heart rate monitoring devices available in the market. The device used in this particular research is the Garmin Forerunner 405 with an additional wireless heart rate monitor. The heart rate monitor uses the Garmin ANT+ wireless sensor network protocol [38] to transfer the data from the device to the PC. The ANT+ device software layer is built directly on top of the Microsoft® .Net Framework 4.0. Real time heart rate monitoring will be taken. If a sequence of abnormal heart rate occurred, the driver will be alerted with an audible alarm.

IV. IMPLEMENTATION

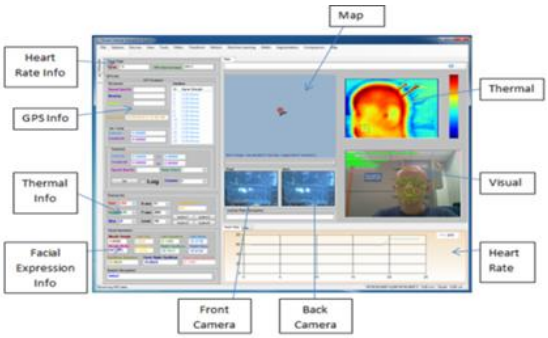


Fig. 8 System Implementation

The embedded system from Fig. 8 has the following features:
From the Graphic User Interface,

- The visual image on the center right, Facial expression info on the lower left showing the coordinate axis (width and height) of the mouth, nose, eyes and eyebrow.
- Thermal Image on the upper right, Thermal Info on the lower center left showing the maximum temperature of the first two rows of the histogram.
- Heart Rate graph on the bottom right, Heart Rate Info on the upper right showing the Heart rate beat per minute and R-R interval.

From GPS location,

1. Query for Location-Based Services (LBS): Traffic report, nearest restaurants /business/places of interest,

[2]*Image taken from http://www.heartmath.org*

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

etc. Considering that we have nationwide cellular GPRS/3G coverage, albeit weak signal in certain areas, the LBS info can be updated on a periodic basis or until a network has been detected. The availability of Wi-Fi chip in the system will broaden the network coverage as well as reduce cellular usage cost.

2. Query the traffic police's server for speed limit in the area and check vehicle speed against speed limit. Use 3G chip coupled with a cellular subscription to text message offenders' violation speed and vehicle registration directly to the traffic police hotline
3. Query for local emergency phone numbers around the area and in case of emergency, uses 3G chip to make an emergency call through the cellular network.
4. Obtain driving direction from current location to the desired location.

While the last item is easily accessed by tapping services such as Google Maps, item 1, 2 and 3 however will require an implementation of a nationwide LBS database (which include local emergency number) and traffic police speed database.

V. TEST RESULT AND DISCUSSION

A. *Visual Image Results*

Table VI is the result of the visual imaging quality test. 4 different resolutions setting for the visual camera were tested. The resolutions are: 128 x 128 = 16,384 pixels; 256 x 256 = 6,5536 pixels; 384 x 384 = 14,7456 pixels; and 512 x 512 = 262,144 pixels respectively. The table clearly indicates that the speedup time of Parallel CPU + GPU is roughly 1.7 to 13.7 times improved over the speedup time of Parallel CPU. As a comparison, the Parallel CPU's performances are dropping tremendously as the image resolution increasing. For the same task of searching for a facial image within the scene, the speedup time of the 512 x 512 resolution is at least 10 times slower than the speedup time of the 256 x 256 solution.

Table VI Results for facial prediction - multiple resolutions Parallel CPU + GPU

Resolution	Parallel CPU (millisecond)	Parallel CPU + GPU (millisecond)	Parallel CPU + GPU Speedup over CPU
128x128	76.923	43.478	1.769
256x256	357.143	91.743	3.893
384x384	769.231	98.039	7.846

Fig. 9 is the bar graph of visual facial camera resolution versus speed (shorter time bar graph is better). As the resolution of the visual image is getting larger, the time gap between Parallel CPU and Parallel CPU + GPU is getting wider. As the Parallel CPU process takes 1428.571 (millisecond) to detect a facial expression on a 512 x 512 resolutions, parallel CPU + GPU only takes 104.166 (millisecond) to execute the same task. The Algorithm is set to detect only a single face in the visual camera scene. If more than one facial image exists within the camera scene of the driver seat, the system categorizes this as an abnormal driving behavior. This can happen if the driver is holding a child on the lap. However, there is also a possibility that the image of the passenger on the back seat or passenger seat overlaps together with the driver in one visual camera scene. In this case, the visual facial algorithm will select the facial image which is closest to the camera.

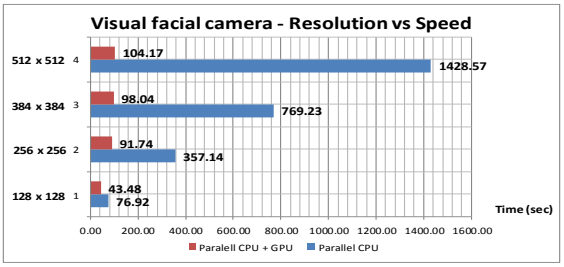


Fig. 9 Visual Facial Camera – Resolution versus Speed using parallel CPU+GPU

B. *Thermal and Heart Rate Results*

For thermal and heart rate experimental setup, the test was divided into two phases:

1. The test subject is in normal condition and is being monitored for a long period of time to ensure the accuracy of the results matches the real condition.
2. The test subject is in abnormal condition and under the influence of alcoholic test for temperature and heart rate monitoring.

The thermal camera is installed approximately 1 foot away from the test subject. 2 main factors are taken into account when deciding on the placement of the camera. The first consideration is the safety of the driver; the thermal camera is not blocking the driver's view during driving. The second consideration is the heat loss as the distance from the thermal camera is directly proportional to the heat loss detection from the body.

Fig. 10 shows the thermal result of a drunk driver. The change from the 1st to the 2nd row of the histogram shows that the body temperature has raised drastically.

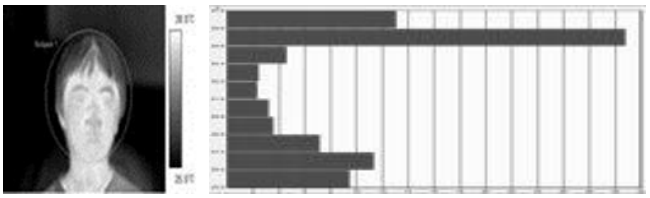


Fig. 10 Abnormal Human Temperature Histogram (Temperature In Descending Order)

Fig. 11 shows the changes of the body temperature before and after alcoholic consumption within a period of time. The normal human temperature is approximately 37 degree Celsius. For the test object, the temperature rose from 35.5 degree up to 36.6 degree within 7 minutes after consuming alcohol. Although there is a slight dropped by 1 degree, but the body temperature increased sharply again until 37 degree Celsius within a very short period of time. Based on the sequence changes of temperature, the system will be able to predict whether the driver is driving under normal condition or under abnormal condition (driving while intoxicant). If an abnormal condition happened, the driver will be alerted by an alarm. If the driver insists on continue driving, the abnormal

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

data will be logged into the system and will be sent out to the dedicated server through broadband whenever is needed.

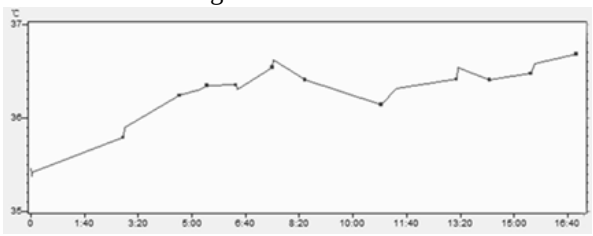


Fig. 11 Results Before and After Alcoholic Consumption

Fig. 12 shows a sequence of abnormal heart rate within 20 minutes time period after alcohol consumption. The heart beat is getting into an irregular route pattern; begin from the normal 60 plus heart Beat Per Minute (BPM) and raised up to 90 plus heart beat per minute. Although heart BPM may differ from individual to individual, but the significant changes in the pattern of a normal heart beat to abnormal heart rate can be easily spotted and recorded by the system.

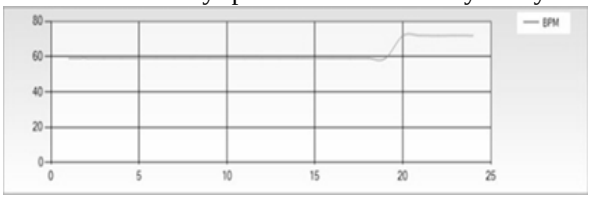


Fig. 12 Normal and Abnormal Heart Rate

C. *Database Info*

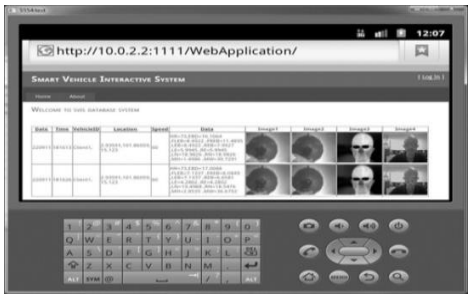


Fig. 13 Data Retrieved From SQL Server Using a Mobile Device

When all related information has been uploaded to the dedicated server, authorized user can browse the information using any Smartphone devices. Fig. 13 is the sample view of the data being displayed on an Android emulator. What is shown is the date in the first column, followed by time, Vehicle ID, Location, Speed, Emotion data, Front and Back image capture from the camera, as well as thermal image and visual image.

D. *Parallel CPU and GPU*

There are several factors that lead to abnormal driving behavior and irregular heart rate such as medical condition, driving under the influence of alcohol and aggressive emotions. For the system to process such complex information in real time, a large processing capacity is required. The normal CPU cannot handle such high processing demand. But parallel processing offers a solution to this problem. Among all parallel processing methods, parallel CPU and GPU give the most promising result. The result from Table VII proves that by using parallel CPU and GPU, the total time for completing the entire process has reduced tremendously. Execution time required by the Parallel CPU for 1 completion of emotion recognition is 2.91 second, but coupling with parallel GPU accelerate the processed by at least 4 times .Total time taken to complete one process is not more than 1.02 seconds. The Facial detection mechanism work flawlessly under dim light condition and work on the subject wearing glasses. The upper left green bars are the gauge bar that indexing different emotion. By examining the graph, one able to tell if the driver is drunk, as the heart rate and body heat which dissipated from the driver is different compared to when the driver is behaving normally. The speedup time graph for parallel CPU + GPU against parallel CPU can be represented in Fig. 14 (the shorter the time bar graph is, the better the speedup is). It can be divided into 7 categories: Temperature, Heart Rate, Angry, Sad, Surprise, Happy and Neutral. From the bar graph, the advantage of using parallel CPU + GPU over parallel CPU can clearly be seen. By Parallel CPU+GPU, the speedup time is at least 3.6 times faster than using parallel CPU only.

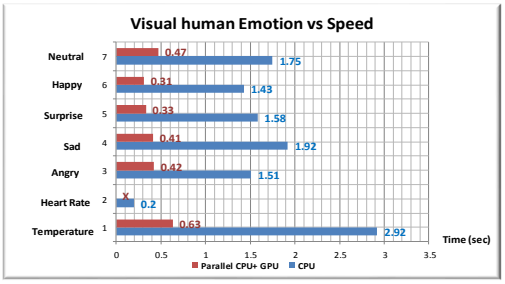


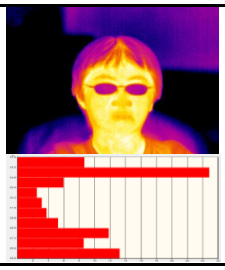
Fig. 14 Visual Human Emotions Using Parallel CPU + GPU

Among all, thermal imaging detection required the maximum time (approximately 0.63 seconds) using parallel CPU + CPU. The same process takes 2.92 second to be completed on parallel CPU. Noticed that there is no parallel CPU + GPU processing in heart rate monitoring, as the algorithm is purely based on parallel CPU processing only, since the heart rate Beat Per Minute (BPM) is only in single dimensional data. Hence, there is no need for the use of the GPU. The misuse of GPU in this data might even obtain slower results than using parallel CPU processing. The priority of the emotion processing is given to thermal imaging, heart rate then followed by the visual image. For example a drive might express sudden aggressive emotion, but the temperature of the driver and heart beat is normal; there could be a chance that this is a false alarm. However, if the aggressive emotion has been detected for a period of time, the changes of temperature and heart rate will be compared with the initial data. The emotion data will be logged and send to the server whenever is needed. At the same time the audible speech warning will be activated.

Table VII Results Comparison With and Without Parallel GPU Acceleration

Emotion/Heart Rate (BPM)	Image	Parallel CPU recognition time (sec)	Parallel CPU + GPU recognition time (sec)

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

Drunk (with glasses) **Heart Rate : 81 - 89**		2.91	0.63
Neutral (with glasses) **Heart Rate 70 - 83**		1.75	0.47
Angry (low light condition + glasses) **Heart Rate 70 – 83**		1.51	0.42
Sad (with glasses) **Heart Rate 70 – 83**		1.92	0.41
Sleepy(with glasses) **Heart Rate 70 – 83**		1.98	0.44
Happy (with glasses) **Heart Rate 70 - 83**		1.43	0.31

VI. CONCLUSION

In conclusion, using a parallel CPU GPU system enables the real time processing of visual, thermal and heart rate data in order to determine the state of a driver and make a decision on whether the driving is abnormal and dangerous. The key objective of the project is to develop a system which executes the real time emotion recognition on minimal power consumption while maintaining the optimal speed. The CUDA programming model is used to optimize the overall system's performance. This paper has demonstrated the efficiency of parallel computing by coupling both CPU and GPU technology to achieve real-time emotion recognition in a non-controlled environment such as the outdoor environment or moving vehicle. By using parallel CPU alone, each completion of a single emotion recognition takes 1.43 to 2.91 second. While with the aid of parallel CPU and GPU, the whole process took not more than 0.63 second, it is at least 4 times faster in comparison. The CPU usage is constantly maintained below 75 percent of all time while all system modules are running simultaneously. The results can be even improved further if the latest CUDA GPU card is installed and Quadcore CPU is used. The portability and compatibility of the system have been taken consider during development. Software framework is planned and designed from the beginning in modular basis; these allow the whole architectural framework to be migrated to different platforms such as ARM [48] based embedded hardware when necessary. Since the Graphic User Interface is written in C#, is it easy to be replaced with other cross platform GUI such as GTK [49], WxWidget [50] DirectShow and WMF can be replaced by FFMPEG [51]. The Microsoft Task Parallel Library can be replaced by OpenMP [52]. Further enhancements include designing the proposed system around a CPU GPU combination which is enabled to handle more threads and blocks during data processing. As such major improvements can only be made if the hardware is improved thus making even more massive parallelism economically feasible.

REFERENCES

[1] UN, "Road Safety report," 2012. [Online]. Available: http://www.un.org/ar/roadsafety/pdf/roadsafetyreport.pdf.

[2] L. J.-C. Gomi T, "Behavior-based AI techniques for vehicle control," in Vehicle Navigation and Information Systems Conference, 1993. Proceedings of the IEEE-IEE, Ottawa, Ont., 1993.

[3] B. J. K. P. Unsal C, "Intelligent control of vehicles: preliminary results on the application of learning automata techniques to automated highway system," in Tools with Artificial Intelligence, 1995. Proceedings. Seventh International Conference, Herndon, VA, 1995.

[4] L. V. Xiang Y, "A constructive Bayesian approach for vehicle monitoring," in Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference, 2000.

[5] E.-S. A. E.-S. N. Noureldin A, "Adaptive neuro-fuzzy module for inertial navigation system/global positioning system integration utilizing position and velocity updates with real-time cross-validation," Radar, Sonar & Navigation, IET, vol. 1, no. 5, pp. 388 - 396 , 2007.

[6] C. M. S. A. J. M. Yongchang Ma, "Real-Time Highway Traffic Condition Assessment Framework Using Vehicle–Infrastructure Integration (VII) With Artificial Intelligence (AI)," Intelligent Transportation Systems, IEEE Transactions, vol. 10, no. 4, pp. 615 - 627 , 2009.

[7] X. H. W. W. M. W. Xiaofeng He, "Intelligent SINS/RDSS integrated algorithms for land vehicle navigation," Aerospace and Electronic Systems Magazine, IEEE, vol. 4, no. 3, pp. 4-11, 2009.

[8] J. S. A. G. David Culler, Parallel Computer Architecture, MORGAN KAUFMANN, 1998.

[9] W.-m. H. David Kirk, Programming Massively Parallel Processors, MORGAN KAUFMANN, 2010.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

[10] P. Pacheco, an Introduction to Parallel Programming, MORGAN KAUFMANN, 2011.

[11] E. H. L. D. M. Betke, "Highway scene analysis in hard real-time.," IEEE in Intelligent Transportation Systems., 1997.

[12] T. E. T. P. C. Papageorgiou, "A trainable pedestrian detection system." IEEE In Intelligent Vehicles, pp. 241-246, 1998.

[13] P. B. R. C. A. Rajagopalan, "Higher order statistical learning for vehicle detection in images," in In Proceedings of 7th International Conference on Computer Vision, 1999.

[14] T. P. Constantine P. Papageorgiou, "A Trainable Object Detection System: Car Detection in Static Image," Biological and Computational Learning, vol. 180, no. C.B.C.L Paper, October 1999.

[15] K.-M. L. L.-S. S. Dang-HuiLiua, "Illumination invariant face recognition," Pattern Recognition, vol. 38, no. 10, p. 1705–1716, 2004.

[16] K.-M. L. XudongXie, "Face recognition under varying illumination based on a 2D face shape model," Pattern Recognition, vol. 38, no. 2, p. 221–230, 2005.

[17] P. O. ,. W. L. AlisterCordiner, Illumination Invariant Face Detection Detecting Faces In Challenging Lighting Conditions, VDM Verlag Dr. Müller, March 25, 2010.

[18] D. A. S. Andrea Selinger, "Face Recognition in the Dark," in 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 8, 2004.

[19] P. D. J. T. P. S. D. P. I. F. M. Buddharaju, "Automatic Thermal Monitoring System (ATHEMOS) for Deception Detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005., 20-25 June 2005.

[20] P. Buddharaju, I. Pavlidis and P. Tsiamyrtzis, "Pose-Invariant Physiological Face Recognition in the Thermal Infrared Spectrum," in Computer Vision and Pattern Recognition Workshop, 17-22 June 2006

[21] .S. Z. C. R.-F. L. S.-C. Z. L. Li, "Illumination Invariant Face Recognition Using Near-Infrared Images," in Pattern Analysis and Machine Intelligence, , April 2007.

[22] S. Erkanli, "Fusion of visual and thermal images using genetic algorithms," Old Dominion University, 2011.

[23] "Ford Motorcraft Vehicle Data Recorder (VDR)," Ford, 2011. [Online]. Available: http://www.dgtech.com/aboutus/success_stories/ford_motorcraft_vdr.html.

[24] "Automotive Data Loggers," MicroDAQ, 2011. [Online]. Available: http://www.microdaq.com/data-logger/automotive.php.

[25] G. S. J. Saxena V, "A real time face tracking system using rank deficient face detection and motion estimation," in Cybernetic Intelligent Systems, 2008. CIS 2008. 7th IEEE , London, 2008.

[26] Y.-q. X. P. X. X.-w. J. Yue-cheng Wu, "he Design of an Automotive Anti-Drunk Driving System to Guarantee the Uniqueness of Driver," in Information Engineering and Computer Science, 2009. ICIECS, Wuhan, 2009.

[27] J. A. S. S. Omidyeganeh M, "Intelligent driver drowsiness detection through fusion of yawning and eye closure," in Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2011 IEEE International , Ottawa, ON, 2011.

[28] G. Dimitrakopoulos, "Intelligent transportation systems based on internet-connected vehicles: Fundamental research areas and challenges," in ITS Telecommunications (ITST), 2011 11th International Conference, St. Petersburg, 2011.

[29] D. Matolak, "Performance of LTE in Vehicle-to-Vehicle Channels," in Vehicular Technology Conference (VTC Fall), 2011 IEEE, San Francisco, CA, 2011.

[30] C. C. VLADIMIR VAPNIK, "Support-Vector Networks," Kluwer Academic Publishers, vol. 20, pp. 273-297, 1995.

[31] Vapnik, "The Nature of Statistical Learning Theory," Springer, 1995.

[32] N. C. John Shawe-Taylor, An introduction to Support Vector Machines., Cambridge University Press, 2000.

[33] J. S. Bernhard Schölkopf, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2001.

[34] C. J. C. S. A. J. SchölkopfBernhard, Advances in Kernel Methods, MIT Press, 1999.

[35] S. A. J. Schölkopf Bernhard, Learning with Kernels, MIT Press, 2002.

[36] N. Shawe-Taylor John, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

[37] "NVidia CUDA Programming Guide," 2011. [Online]. Available: http://developer.download.nvidia.com/compute/cuda/1_0/NVIDIA_CUDA_Programming_Guide_1.0.pdf.

[38] ANT+, 2010. [Online]. Available: http://thisisant.com/.

[39] Dmihailescu, "Benchmark start-up and system performance for .Net, Mono, Java, C++ and their respective UI," 2 Sep 2010. [Online]. Available:

[40] http://www.codeproject.com/KB/dotnet/RuntimePerformance.aspx.

[41] "Microsoft DirectShow," 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/dd375454(v=vs.85).aspx.

[42] "Microsoft SQL Server," Microsoft, 2011. [Online]. Available: http://www.microsoft.com/sqlserver/en/us/default.aspx.

[43] "Python," Python Software Foundation, 2011. [Online]. Available: http://python.org/.

[44] "Threading Building Blocks," Intel, 2011. [Online]. Available: http://threadingbuildingblocks.org/.

[45] "Paul Ekman," 1972. [Online]. Available: http://face.paulekman.com/default.aspx?override=1.

[46] FERET, "FERET," 2000. [Online]. Available: http://www.itl.nist.gov/iad/humanid/feret/feret_master.html.

[47] "Active Shape Model," 1995. [Online]. Available: http://webdocs.cs.ualberta.ca/~nray1/CMPUT615/Snake/cootes_cviu95.pdf.

[48] T. Cootes, "Statistical Shape Models," 1997. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/COOTES/pdms.html.

[49] "ARM," 2011. [Online]. Available: http://www.arm.com/.

[50] "GTK," 2011. [Online]. Available: http://www.gtk.org/.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

[51] "WxWidgets," 2011. [Online]. Available: http://www.wxwidgets.org/.

[52] FFMEPG, "FFMEPG," 2011. [Online]. Available: http://www.ffmpeg.org/.

[53] "OpenMP," 2011. [Online]. Available: http://openmp.org/wp/.