

Gesture based Virtual Reality Implementation for Car Racing Game employing an Improved Neural Network

Biswarup Ganguly

Assistant Professor: Department of Electrical Engineering
Meghnad Saha Institute of Technology
Kolkata, India

Priyanka Vishwakarma

B.Tech Final Year : Department of Electrical Engineering
Meghnad Saha Institute of Technology
Kolkata, India

Rahul

B.Tech Final Year : Department of Electrical Engineering
Meghnad Saha Institute of Technology
Kolkata, India

Shreya Biswas

B.Tech Final Year : Department of Electrical Engineering
Meghnad Saha Institute of Technology
Kolkata, India

Abstract—This paper explores the formation of a smart human-computer interfacing system for computer gaming using Microsoft's Kinect sensor. Gesture based monitoring is chosen as the most prominent modality to continue the car racing game. Angle based features are extracted from skeletal images obtained from the sensor. Levenberg-Marquardt optimization technique is used for weight adaptation for a feed-forward neural network using back-propagation learning algorithm. A gesture based console for Asphalt 8: Airborne game is implemented. Experimental results reveal that the proposed method outperforms the state-of-the-art training methods of neural network.

Keywords—Gesture, computer gaming, Kinect sensor, Levenberg Marquardt neural network, back-propagation learning.

I. INTRODUCTION

Video gaming [1] is a rapidly towering form of entertainment and inclusion of young brains in this arena and increasing competition inside the industry are leading the game developers towards creating more enjoyable and realistic the game design. The video games need some heightened level of emotional experience during the game play to maximize the entertainment and enjoyment of the players. From the professional viewpoint of a gamer, he or she needs more immersion inside a game for higher involvement in the game. That is where the Human-Computer-Interaction (HCI) works dramatically [2].

HCI is a study of human centric interactive computer systems, technologies, and associative devices. The rapid influx of computer application users in the 21st century has led humanity to the scavenger hunt of a reliable and robust system to interact with a machine. To attain more intimate human-computer interaction with corporeal feedback, the recent evolution and advancing research on sensors virtual reality and rendering technology have the most important role here. All of this technological advancement has worked as a boon in the gaming industry but due to more demand for better user flexibility and comfort of gamers the arena of HCI in gaming needed something better. So, for the cases of driving based games like car racing which allows the user to drive virtually, gesture-controlled gaming consoles are most

suitable. If traditional associative devices for HCI like joysticks, wheels, and pedals etc are removed from this type of racing games, then the games become more enjoyable and comfortably flexible which is a way different experience than Microsoft Holo lens gesture which is a head-mounted simulation of virtual reality [3]. Oculus Rift is also a similar type of simulation of VR technology.

Therefore several gaming corporations boomed into the motion-controlled gaming market which involves some natural real-world actions, some combination of spoken words and gesture recognition. The first installment in motion-controlled gaming system was in 2006, Nintendo Wii [4]. Then other popular games like Microsoft Kinect [5-6] and Sony PlayStation eye [7] came. The current version of Wii MotionPlus system is in the same league as Microsoft Kinect for Sony Move add-on and X-box 360 respectively. The Wii Remote controller has accelerometer equipment which recognizes head tulte, detailed moments, a degree of acceleration etc. The PlayStation Move controller has a console plug-in named "the eye" which is tracked by the video camera which can recognize the eye's position in real time. The PlayStation also uses head position tracking and facial recognition.

The Kinect includes a depth camera and motion sensor which leads it to be controller-free. The depth camera, creates a skeleton of the image and movements are also tracked. Speech recognition software allows the system to understand spoken commands.

Saha *et al.* [8] have developed a gesture based car driving game using Microsoft's Kinect sensor to track the human gestures while driving. Steering angle and diameter, the extracted features, are fed to a Mamdani Fuzzy Type-1 system. Acceleration and brake signal of the car is produced by the z-coordinate's difference of the right and left feet respectively. Authors are trying to expand the degrees of freedom of the driving simulator to apply it on an aircraft system using a virtual control stick. Wada and Kimura [9] have presented a paper where stability of car-driving with a joystick system is analyzed. A dynamic model for vehicles has been derived for analyzing various car velocities. Zhu and Yuan [10] have presented a real time gesture recognition

technology which can be effectively used to track both single hand and double hand gestures. One of the most popular RGB-D sensor named Kinect, released by Microsoft is used for taking RGB images and depth images at a specific resolution and sampling rate. The recognition procedure follows the steps; preprocessing, hand detection, tracking, classification of palm and fist, and finally gesture recognition. The classification is carried out by template matching, SVM and some other light methods. The practical uses of this work lie in real time tasks such as power point system for controlling slides and car racing games achieving an accuracy rate of 91% and 98% respectively. This system is robust and efficient for real time applications.

Shivamurthy *et al.* [11] have presented a paper where interactive car gaming has been interfaced via personal computer. Kinect is used to take the skeleton of human gestures while playing the racing game. Four basic skeletal gestures namely right, left, forward, normal have been introduced to play the computer game. It has been found that this novel gesture based car racing game is more exciting than the conventional keyboard oriented game. Zhao *et al.* [12] have proposed a control system for improving the handling stableness of all the motor wheels of an electric vehicle employing Mamdani's minimum fuzzy rule. The aim of this research was to control the braking and driving forces of the wheels driven by the motors. Pfleging *et al.* [13] have approached a versatile technique upon the steering wheel of a driving car implementing both gestures and speech. Controlling the car by proper feedback has been provided using some manual gestures like right, left *etc.* A speech allowing multi touch steering wheel for controlling a computer based driving simulator has been installed for this experiment. A gesture and voice recording app is also designed for presenting different scenes. Authors are trying to enlarge their work by adding more modalities like body postures, gazes for analyzing driver's attention.

The proposed work is an improvement of previous work [14], where the back-propagation neural network with steepest descent technique is used for weight adaptation. The convergence speed of steepest descent training algorithm is very slow due to its small step size. It has fast movements in large gradient (steep regions) and slow motion in small gradient (valley regions) of the error surface. The performance of steepest descent training algorithm can be improved by



Fig. 1. Kinect sensor depicting camera and sensors.

Gauss-Newton (GN) algorithm by choosing a proper step size. Taking second order derivatives of error function into the account, the GN training method evaluates the curvature of error surface and finds its convergence rate faster than steepest descent. But the former fails when the quadratic approximation of error surface is not possible. The Levenberg-Marquardt

(LM) optimization based training combines the above two training methods with respect to faster convergence and stability and here lies the contribution of the proposed work.

The presentation of the paper is as follows. Preliminaries, *i.e.* Kinect sensor and Levenberg-Marquardt neural network (LM-NN) are discussed in section II. The detailed scheme of the work is in section III. Section IV explains the experimental results. Finally, conclusion is drawn in section V.

II. PRELIMINARIES

This section explains Microsoft's Kinect sensor used for experimental setup and Levenberg-Marquardt based neural network for classification of gestures.

A. Kinect Sensor

Kinect [15], the official name of the Xbox 360 Console, is developed by Prime Sense company along with Microsoft in June 2010. The device has become popular in gaming [3], surveillance [16], man-machine interaction [17] and in many other fields of robotics. The main reasons to use Kinect is, firstly it blends the tools of speech and gesture in a common device, a multimodal interfacing and secondly it permits users to communicate with the gaming console only using gestures, keeping hands free for other communication. Kinect is a black horizontal bar sensor and it looks like a webcam. The required sensors for this work are infra-red (IR) and RGB camera. Using IR emitter-receiver sensor pair, the device is capable to get the depth of the subject from the sensor. Fig. 1 shows the sensors required for the proposed work.

B. Levenberg-Marquardt Neural Network

The Levenberg-Marquardt (LM) algorithm [18-19], originated by Kenneth Levenberg and Donald Marquardt, finds its application in minimizing a non-linear function due to its stability and fast convergence. Artificial neural network with the traditional training algorithms, *viz.* steepest descent, Newton's method, Gauss-Newton's algorithm has been considered as the most promising breakthroughs in neural network training. But LM based training outperforms all the previous training methods in terms of speed and stability.

A feed-forward neural network [20] consists of an input layer, multiple hidden layers, and one output layer. The number of neurons in the hidden layers can be varied whereas the neurons in the input layer and the output layer specify the input features and output classes respectively. Weights are initialized randomly and a weighted sum of input is forwarded from input to output layer. All the layers have connections from its immediate previous layer only. Back-propagation (BP) learning algorithm is implemented for weight adaptation so that the difference between the target value and the output value is minimized. The LM optimization technique is used for updating weights in BP learning mode as it outperforms steepest descent algorithm and Gauss-Newton algorithm for small and medium-sized training problems.

The error at output layer is defined as

$$e_{p,q} = t_{p,q} - o_{p,q} \quad (1)$$

where, t is target vector and o is the output vector for P number of patterns and Q number of outputs. The sum square error (SSE) is provided to train the neural network as

$$E(f, w) = \frac{1}{2} (\sum_{p=1}^P \sum_{q=1}^Q e_{p,q}^2) \quad (2)$$

where, f is the feature vector and w is the weight vector.

Now the first order derivative of SSE is defined as

$$g = \frac{\partial E(f, w)}{\partial w} = \left[\frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_2} \quad \dots \quad \frac{\partial E}{\partial w_N} \right]^T \quad (3)$$

and second order derivatives of error function form a Hessian matrix (H) as shown below,

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \quad (4)$$

$$J = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \dots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \dots & \frac{\partial e_{1,2}}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{1,Q}}{\partial w_1} & \frac{\partial e_{1,Q}}{\partial w_2} & \dots & \frac{\partial e_{1,Q}}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{P,1}}{\partial w_1} & \frac{\partial e_{P,1}}{\partial w_2} & \dots & \frac{\partial e_{P,1}}{\partial w_N} \\ \frac{\partial e_{P,2}}{\partial w_1} & \frac{\partial e_{P,2}}{\partial w_2} & \dots & \frac{\partial e_{P,2}}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{P,Q}}{\partial w_1} & \frac{\partial e_{P,Q}}{\partial w_2} & \dots & \frac{\partial e_{P,Q}}{\partial w_N} \end{bmatrix} \quad (8)$$

The relation between H and J can be written as

$$H = J^T J \quad (9)$$

Hence the weight adaption rule of Gauss- Newton algorithm becomes,

$$w_{n+1} = w_n - (J_n^T J_n)^{-1} J_n e_n \quad (10)$$

In order to verify whether Hessian matrix is invertible, Levenberg and Marquardt have introduced another modification to Hessian matrix as

$$H = J^T J + \mu I \quad (11)$$

where, μ is the combination coefficient and I is the identity matrix. It is noticeable that the elements on the principal diagonal of the modified Hessian matrix H are positive and hence invertible. Combining both (10) and (11), Levenberg-Marquardt updated the rule as,

$$w_{n+1} = w_n - (J_n^T J_n + \mu I)^{-1} J_n e_n \quad (12)$$

outperforming both gradient descent and Gauss-Newton training of back-propagation neural network.

III. PROPOSED SOLUTION OF THE WORK

Kinect sensor has been used to identify human gestures. With the help of the camera- sensor pair and software development kit (SDK), human skeletal images are formed by 20 body joints in three dimensional spaces. For implementing virtual reality in car racing game, six joints (Hip center (HC), Shoulder Center (SC), Hand Left (HaL), Hand Right (HaR), Foot Left (FL) and Foot Right (FR)), shown in black bold points, have been used for the proposed solution shown in Fig. 3.

Approaching towards the solution, the first decision by the player is whether to steer and afterwards in which direction to steer. Henceforth the speed of the car is decided by the player's leg gesture. Skeletal data of 30 frames per second are collected from the sensor for the proposed algorithm.

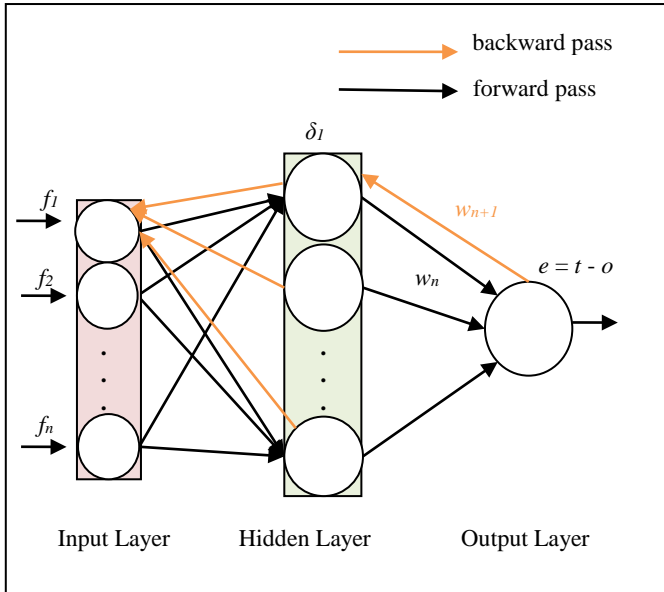


Fig. 2. Structure of neural network with LM based back-propagation learning
Combining (3) and (4),

$$-g = H \Delta w \quad (5)$$

Or,

$$\Delta w = -H^{-1} g \quad (6)$$

Hence, the weight updating rule becomes

$$w_{n+1} = w_n - H_n^{-1} g_n \quad (7)$$

If Newton's weight updating rule is applied, the calculation for second order derivatives of error function becomes quite complex. For circumventing this problem, Jacobian matrix (J) has been implemented as

A. Determination of steering direction

Two angular features are extracted from the driving gesture, obtained from the skeleton image of the player, are given as

$$\alpha_L = \text{angle}(SC, HC, HaL) \quad (13)$$

$$\alpha_R = \text{angle}(SC, HC, HaR) \quad (14)$$

where, α_L is the angle between SC , HC and HaL , and α_R is the angle between SC , HC and HaR . For straight driving, $\alpha_L \approx \alpha_R$. For steering left α_R decreases and α_L increases; and is just reverse for steering right condition. When the gamer steers the car gradually to the left, $\alpha_R < \alpha_L$. Therefore if $(\alpha_L - \alpha_R)$ exceeds a specified threshold, the virtual car is steered towards left. The opposite phenomenon occurs for steering the car to right.

B. Calculation of steering angle

The angle (θ) between the straight line joining HaR and HaL with the transverse plane is taken as the feature shown in Fig. 5(b). The feature is modified after multiplying with the updated weight obtained via LM-NN discussed in section II (B).

$$\theta' = \theta \cdot w_{n+1} \quad (15)$$

C. Evaluation of speed of the car

The speed of the car is monitored by applying acceleration or brake by leg. Usually acceleration is applied by right leg and brake by left leg. So, angle of directional cosines (β_R or β_L) of the line segment joining FL and FR are taken as the feature shown in Fig. 5(e). The entire scheme is shown in a flowchart given in Fig. 4.

IV. EXPERIMENTAL RESULTS

The proposed method is tested on 50 subjects playing Asphalt 8: Airborne game. The ratio of male to female is 3:2 within the age group of 22 to 35 years. 20 subjects (12 male and 8 female) have prior knowledge to play this racing game using Kinect Xbox 360.

While playing the racing game, a few frames of the gamer are captured by Kinect, shown in Fig. 5 and also the screenshots for those suitable gestures are depicted in Fig. 6.

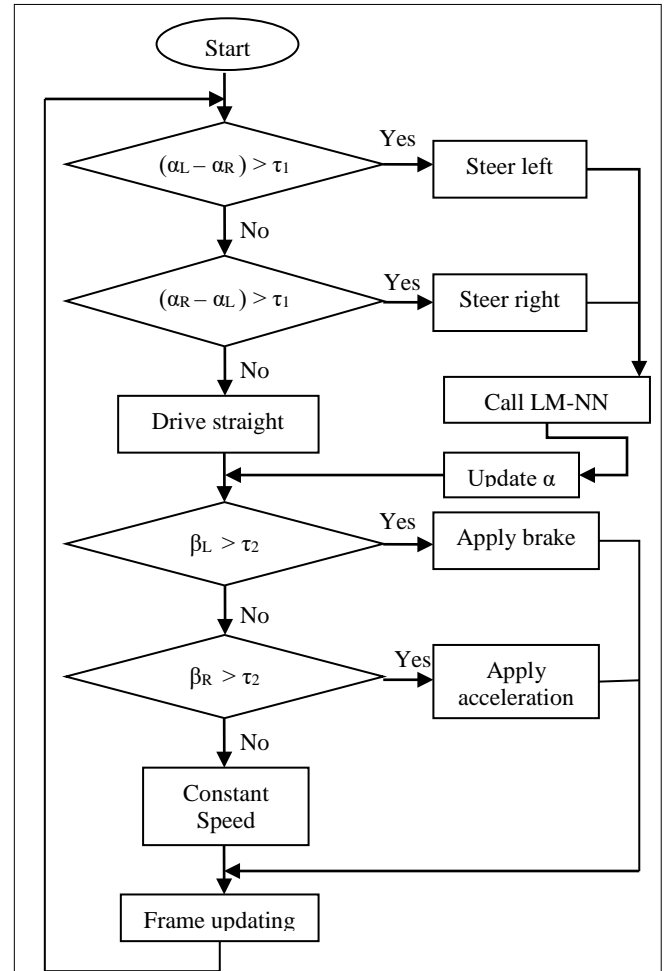


Fig. 4. Flowchart for implementing virtual reality in car racing game.

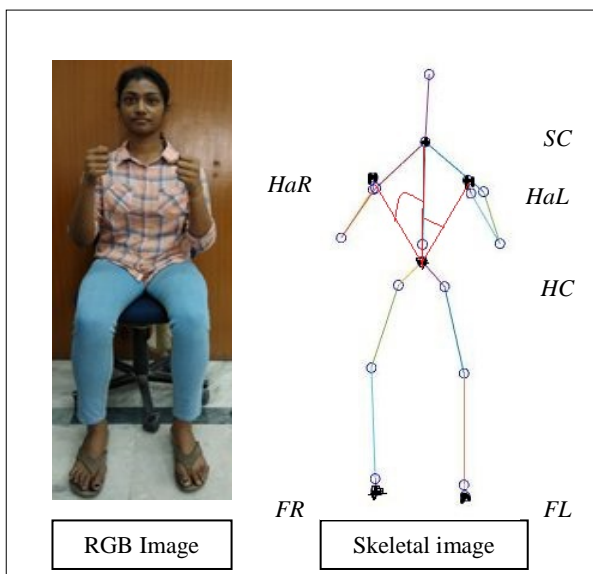


Fig. 3. RGB with skeletal image showing required joints.

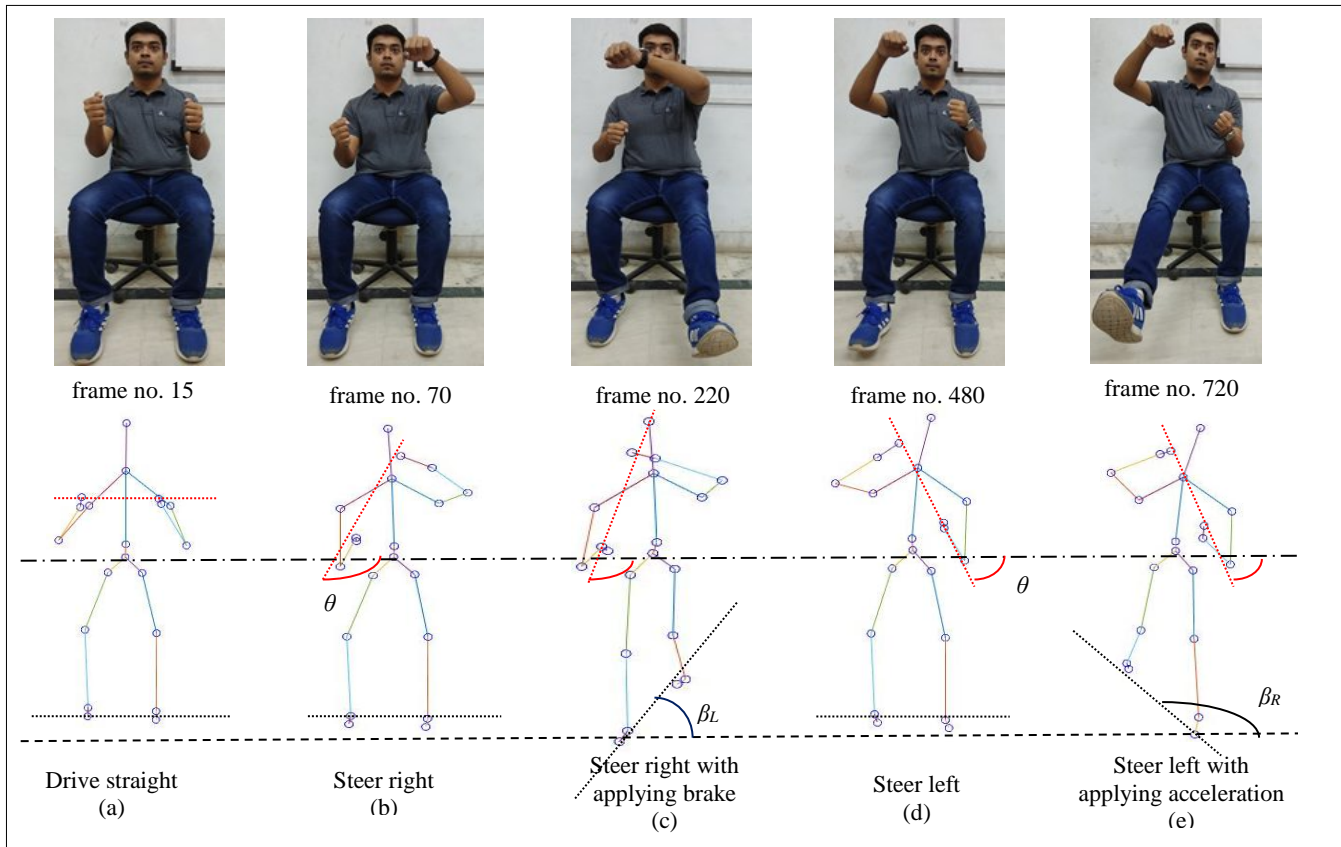


Fig. 5. RGB and subsequent skeletal images for different frames

Table I shows the corresponding features of the player according to his gestures.

The frame-by-frame method for car driving in virtual reality has been shown in Fig. 5. Starting from straight driving (Fig. 5(a)), while playing the car racing game, the gamer steers right (Fig. 5(b)), whenever required brake is applied (Fig. 5(c)). Also steering left with acceleration (Fig. 5(d-e)) is implemented whenever needed to play the game. The useful features (θ and β) are marked in Fig. 5 while the driver progresses the game.

From the pictorial analysis of Fig. 5, it is noticeable that the steering angle increases from straight driving to steering right as well as straight driving to steering left. But it is observed that angle obtained is acute for steering left and angle produced is obtuse for steering right. In case of applying acceleration or brake, angle of directional cosines (β) have been taken. From repetitive experiments it is found that the

angle is acute for brake condition and obtuse for acceleration condition. The experimental values for τ_1 and τ_2 are 79 and 140 degree respectively.

The proposed method employing LM-NN is compared with other types of neural network including gradient descent rule based neural network (GD-NN), radial basis kernel based neural network (RB-NN), probabilistic neural network (PNN) and recurrent neural network (RNN).

The receiver operating characteristic curve (ROC) for various types of neural network is shown in Fig. 6, where performance between true positive rate (sensitivity) and false positive rate (specificity) has been focused.

TABLE I. FEATURES EXTRACTED FROM DIFFERENT FRAMES

Frame no.	α_L	α_R	θ	β_L or β_R
15	78.26	76.15	2.04	0.89
70	68.22	23.46	122.34	2.04
220	77.19	19.23	107.12	46.63
480	26.20	77.89	67.77	0.45
720	13.23	58.35	54.32	134.23

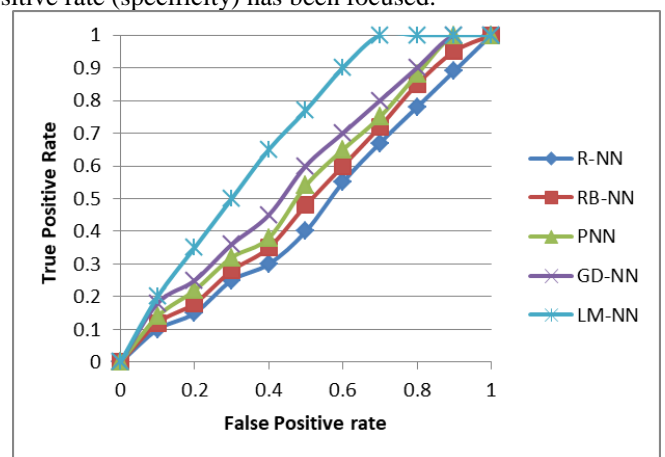


Fig. 6. ROC curves for various methods of neural network.



Fig. 7. Screenshots of Asphalt 8: Airborne game with three instances.

V. CONCLUSION

The main contribution of this work lies in its efficiency and applicability towards optimized training in neural network. Although this method is an improvement of [14], a high level precision is obtained. Employing this gesture based car driving, a number of degrees of freedom can be enjoyed by the gamer. LM-NN outperforms other training methods of neural network in terms of sensitivity and specificity. Further in this work, a reduced number of features compared to [14] have been used for virtual implementation of car gaming system. The proposed work has redefined the significance of Kinect sensor for modern day gaming interfaces.

ACKNOWLEDGMENT

The research work is supported by Artificial Intelligence Laboratory, Department of Electronics & Tele-communication Engineering, Jadavpur University, India.

REFERENCES

- [1] Kang, Hyun, Chang Woo Lee, and Keechul Jung. "Recognition-based gesture spotting in video games." *Pattern Recognition Letters* 25, no. 15 (2004): 1701-1714.
- [2] Pavlovic, Vladimir I., Rajeev Sharma, and Thomas S. Huang. "Visual interpretation of hand gestures for human-computer interaction: A review." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 7 (1997): 677-695.
- [3] Chung, David D., and Walter R. Klappert. "Systems and methods for transmitting media associated with a measure of quality based on level of game play in an interactive video gaming environment." U.S. Patent 8,657,680, issued February 25, 2014.
- [4] Deutsch, Judith E., Arielle Brettler, Caroline Smith, Jamie Welsh, Roshan John, Phyllis Guarrera-Bowlby, and Michal Kafri. "Nintendo wii sports and wii fit game analysis, validation, and application to stroke rehabilitation." *Topics in stroke rehabilitation* 18, no. 6 (2011): 701-719.
- [5] Zhang, Zhengyou. "Microsoft kinect sensor and its effect." *IEEE multimedia* 19, no. 2 (2012): 4-10.
- [6] Han, Jungong, Ling Shao, Dong Xu, and Jamie Shotton. "Enhanced computer vision with microsoft kinect sensor: A review." *IEEE transactions on cybernetics* 43, no. 5 (2013): 1318-1334.
- [7] Neil, A., S. Ens, R. Pelletier, T. Jarus, and D. Rand. "Sony PlayStation EyeToy elicits higher levels of movement than the Nintendo Wii: Implications for stroke rehabilitation." *European journal of physical and rehabilitation medicine* 49, no. 1 (2013): 13-21.
- [8] Saha, Chiranjib, Debdipta Goswami, Sripama Saha, Amit Konar, Anna Lekova, and Atulya K. Nagar. "A novel gesture driven fuzzy interface system for car racing game." In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pp. 1-8. IEEE, 2015.
- [9] Wada, Masayoshi, and Yohei Kimura. "Stability analysis of car driving with a joystick interface." In *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pp. 493-496. IEEE, 2013.
- [10] Zhu, Yanmin, and Bo Yuan. "Real-time hand gesture recognition with Kinect for playing racing video games." In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pp. 3240-3246. IEEE, 2014.
- [11] Shivamurthy, R. C., and M. B. Manjunatha. "Gesture recognition based interactive car game." In *Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on*, pp. 1-6. IEEE, 2015.
- [12] Zhao, Yongli, Yuhong Zhang, and Yane Zhao. "Stability control system for four-in-wheel-motor drive electric vehicle." In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, vol. 4, pp. 171-175. IEEE, 2009.
- [13] Pflöging, Bastian, Stefan Schneegass, and Albrecht Schmidt. "Multimodal interaction in the car: combining speech and gestures on the steering wheel." In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 155-162. ACM, 2012.
- [14] Saha, Sripama, Rimita Lahiri, Amit Konar, Anca L. Ralescu, and Atulya K. Nagar. "Implementation of gesture driven virtual reality for car racing game using back propagation neural network." In *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*, pp. 1-8. IEEE, 2017.
- [15] Ren, Zhou, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. "Robust part-based hand gesture recognition using kinect sensor." *IEEE transactions on multimedia* 15, no. 5 (2013): 1110-1120.
- [16] Ganguly, Biswarup *et al.* (December 2018) "Kinect sensor based gesture recognition for surveillance application." [Online]. Available: <https://arxiv.org/abs/1812.09595>
- [17] Saha, Sripama, Biswarup Ganguly, and Amit Konar. "Gesture based improved human-computer interaction using Microsoft's Kinect sensor." In *Microelectronics, Computing and Communications (MicroCom), 2016 International Conference on*, pp. 1-6. IEEE, 2016.
- [18] Lera, Gabriel, and Miguel Pinzolas. "Neighborhood based Levenberg-Marquardt algorithm for neural network training." *IEEE transactions on neural networks* 13, no. 5 (2002): 1200-1203.
- [19] Wilamowski, Bogdan M., and Hao Yu. "Improved computation for Levenberg-Marquardt training." *IEEE transactions on neural networks* 21, no. 6 (2010): 930-937.
- [20] Svozil, Daniel, Vladimir Kvasnicka, and Jiri Pospichal. "Introduction to multi-layer feed-forward neural networks." *Chemometrics and intelligent laboratory systems* 39, no. 1 (1997): 43-62.