Real-time Prediction in Hands-free Computing for Disabled

Pradeep V
Associate Professor, Department of ISE
Alva's Institute of Engineering & Technology, Moodbidri, DK, Karnataka, India
E-mail: writetopv@gmail.com

Abstract

Over the past three decades, numerous authors have proposed alternatives to the mouse for people in the movement with disabilities who have not yet had a fair opportunity to utilise the standard input methods of a personal computer. In camera-based systems, the overhead of using head-mounted devices is reduced by using the web camera as the mouse. Tracking user facial expressions as they are being captured by the camera and accurately translating them into mouse cursor movement and click events are research challenges and opportunities. The current systems can only move the pointer in a slanting manner due to the user's accidental head movements losing the tracked feature. The movement has not yet allowed people with impairments the same chances as others to interact with computers. They have trouble using the input devices on computers due to their mobility problems. Controlling mouse pointer navigation is still difficult, despite the fact that on-screen virtual keyboards may be used to simulate a physical keyboard and speech recognition can be easily utilized to map mouse click events. The development of hands-free mouse replacement technologies has undergone much advancement. There are a number of limitations to mouse replacement systems that use a webcam as a mouse. In order to enable people with movement disabilities to use a standard PC, this research suggests enhancing the ability of camera-based hands-free computing systems to control the mouse cursor by predicting the user's selection of the target item in the GUI-based system using neural network techniques. Using samples where the mouse cursors predicted position values are closer to the user's actual selection region on the computer screen, the system is put to the test, and the anticipated outcome is achieved in every sample.

Keywords: predicting mouse cursor targets; assistive technology; neural networks; hands-free computing; camera mouse; disabled users.

1. Introduction

1.1 Background

Around one billion individuals worldwide suffer from some sort of physical disability, with over 20% suffering from severe disabilities such as quadriplegia [1]. More than 25 million individuals in India are disabled, with more than 5 million of them having a movement disability, which includes people who are paralyzed without both legs and/or both arms [2]. Assistive technologies are a type of hardware or software, or both, that is designed to make it easier for people with disabilities or impairments to use computers [3]. Hands-free computing, which eliminates the need for human interface devices like the keyboard and mouse, is an option for disabled people. They can use their head, tongue, lips, eyes, or speech recognition software to work on a computer without using their hands [4]. On-screen virtual keyboards can be used to mimic a physical keyboard, so a pointing device mouse is sufficient for using the computer with GUI applications. Controlling the mouse pointer navigation remains a challenge, even though mouse click events may be easily mapped using speech recognition [5][6]. Various hands-free mouse replacement systems have been developed over the last three decades, with various advances such as Head-Mounted Pointing, Smartnav, Headmouse Extreme, Camera Tracking, iTracker, Camera Mouse, Enable ViaCam etc.

1.2 Challenges

Their movement issues make it difficult for People with disabilities to use a computer's input devices. The majority of mouse replacement options necessitate additional gear that allows the user to operate the computer by wearing it on their face or head and operating it through their eyes [5] [7] [8] [9] [10] [11]. There are a number of limitations to mouse replacement systems that use a webcam as a mouse.

1.3 Motivation

People with disabilities in the movement have yet to be given the same opportunities as others to interact with computers. They do, however, deserve all the help they can get to become fully integrated into the Information Society. The development of assistive technologies that allow people to use normal personal computers is highly desirable, and various attempts have been made to meet the needs.

1.4 Objectives

This research proposes to improve the capacity of camera-based hands-free computing systems to control the mouse cursor by predicting the user's selection of target item in the GUI-based system using neural network techniques, allowing people with movement disabilities to utilize a normal PC.

1.5 Contributions

A user's head motions acquired using a webcam are utilized to operate the mouse cursor, reducing the overhead of needing a high-cost hardware system and head-mounted devices. The trendiest subfields of machine learning and artificial intelligence, computer vision, offers significant promise to execute these applications [12]. Face detection algorithms, such as those developed by Viola and Jones [13], and facial analysis algorithms, such as those developed by Kazemi and Sullivan [14], which recognize facial landmarks, are critical in the development of such applications. When enough examples have been provided, Artificial Neural Networks (ANNs) can be utilised for applications that require item prediction and can find new patterns in data by classifying the given data [15].

2. Previous work

Chareonsuk et al. [16] described a method for tracking the human face and imitating mouse pointer movement using the tracking information. Kumar et al. [17] created an assistive HCI system for people with motor limitations that uses Haar classifier techniques to capture the eyes and face to operate the computer's input devices. Chathuranga et al. [6] employed the user's nose movement to control mouse cursor movement, as well as vocal commands to control mouse click events. For mouse control, Bian et al. [18] have captured the position of the nose, as well as the open and closed status of the mouth. Left cheek movement is used to control cursor movement in the left direction; right cheek movement is used to control cursor movement in the right direction; eyebrow rise is used to control cursor movement in the upward direction; eyebrow down is used to control cursor movement in the downward direction, and mouth open is used to implement the mouse click event by Vasanthan et al. [19]. Kraichan et al. [20] used a haar-like characteristic to detect the face, eyes, and the location between the two eyes, which they then mapped to the mouse cursor, coordinates on the computer screen. Parmar et al. [21] have created a technique that estimates the position between the eyes for computer tracking. To replicate mouse cursor movement, Eric et al. [22] extract the upper lip, left eyebrow, and right eyebrow. For mimicking mouse events, Fathi et al. [23] employed skin color features to determine the user's face. Using the Viola-Jones algorithm based on Haar features, Islam et al.[24] track the position of the nose for cursor control. Asad et al. [25] devised a method for controlling mouse motions by moving one's head. For manipulating the user's computer cursor with the face tracking application, Antunes et al.[26] implemented the Viola-Jones object detection algorithm. Meena et al. [27] used their faces and eyes to mimic mouse behavior. For mouse control, Baştu et al.[28] tracked eye movements. For mouse operations, Anantha et al.[29] used facial recognition and eye motions. Computer mouse cursor movement with vision focus was implemented by Ghadekar et al.[30].

The common GUI interactive features such as menus and scroll bars, which require horizontal or vertical as the cursor moves in a slanting direction, were difficult to use for Chareonsuk et al. [16] and Sancheti et al. [31]. It is difficult for Vasanthan et al. [19], Nasor et al. [32], Magee et al. [33], Sugano et al. [34], Valenti et al. [35], and Riad et al. [36] to synchronize the mouse pointer movement with the rate of head movement. When users turn their heads fast or out of the screen, Betke et al. [37] found that the facial characteristic they had chosen to follow was frequently lost. The limitations of environment control were experienced by Chareonsuk et al. [16], Vasanthan et al. [19], and Salunkhe et al. [38]. The detection result is extremely sensitive to lighting conditions, and their method performs best in a bright environment. Two key issues with tracking facial features are occlusions and variable stances [39].

Kim et al. [40] used KLT and SegNet landmark detectors to obtain 97.98 percent for slow films and 91.58 percent for fast videos in tracking face landmarks. Silwal et al. [41] used a Convolutional Neural Network with a Softmax classifier to improve the accuracy of facial recognition by 94.37 percent. When the moving speed is no more than 20 pixels per action, Wang et al. [42] found that the accuracy of controlling the mouse cursor is greater than 90%. Participants were irritated by slight cursor changes when the cursor entered the target and the participant tried to continue gazing at the target, according to Murata et al. [43]. Esiyok et al.[44] have stated the importance of allowing those with limited head movements to use a single gesture. For the head and eye blink classifications, Ghadekar et al. [45] attained accuracies of 98.12 percent and 93.48 percent, respectively.

3. Methods

In a GUI-based computer system, the proposed system focuses on predicting the user's target item. Based on previous learning experience, while the user is attempting to move the mouse, neural networks can be utilized to predict which object will be selected on the screen. The neural network should be trained to produce the intended outputs for the known set of inputs [46].

The screen area is divided into a number of potential S_P square pixels for the pixel $P_H \times P_V$ computer screen resolution with landscape orientation, each of which is believed to be a selection region for the mouse cursor.

The total number of regions which are the user selection areas is $m_{max} \times n_{max}$, where m_{max} the total no. of rows is and n_{max} is the total no. of columns and their possible values are shown in Eq. (3.1) and Eq. (3.2).

$$n_{max} = \begin{cases} 1, & if\ P_H < S_P \\ \left\lceil \frac{P_H + 1}{S_P} \right\rceil, & Otherwise \end{cases} \tag{3.1}$$

$$m_{max} = \begin{cases} 1, & if\ P_H < S_P \\ \left\lceil \frac{P_V + 1}{S_P} \right\rceil, & Otherwise \end{cases} \tag{3.2}$$

The User Selection area is a matrix of regions that can be accessed by row and column numbers. The position of a particular region in the computer screen is identified by the values of $\{X_{Min}(n), Y_{Min}(m)\}$ as the top left corner and $\{X_{Max}(n), Y_{Max}(m)\}$ as the bottom right corner where *m* and *n* are the rows and column numbers of the region. The relation between the row/column numbers and the location of the region is shown in Eq. (3.3) to Eq. (3.6).

$$X_{Min}(n) = (n-1)S_P \tag{3.3}$$

$$Y_{Min}(m) = (m-1)S_P \tag{3.4}$$

$$X_{Max}(n) = X_{Min}(n) + S_P - 1 = (n-1)S_P + S_P - 1 = nS_P - 1 \tag{3.5}$$

$$Y_{Max}(m) = Y_{Min}(m) + S_P - 1 = (m-1)S_P + S_P - 1 = mS_P - 1 \tag{3.6}$$

where,

$$0 \leq (X_{Min}(n), X_{Max}(n)) \leq P_H$$

$$0 \leq (Y_{Min}(m), Y_{Max}(m)) \leq P_V$$

The basic design of the proposed neural network system is presented in Fig. 3.1. In the input layer, there are four neurons, and in the output layer, there are two neurons. Depending on the complexity necessary, any number of hidden layers with any number of neurons in each hidden layer can be utilized in the interim.

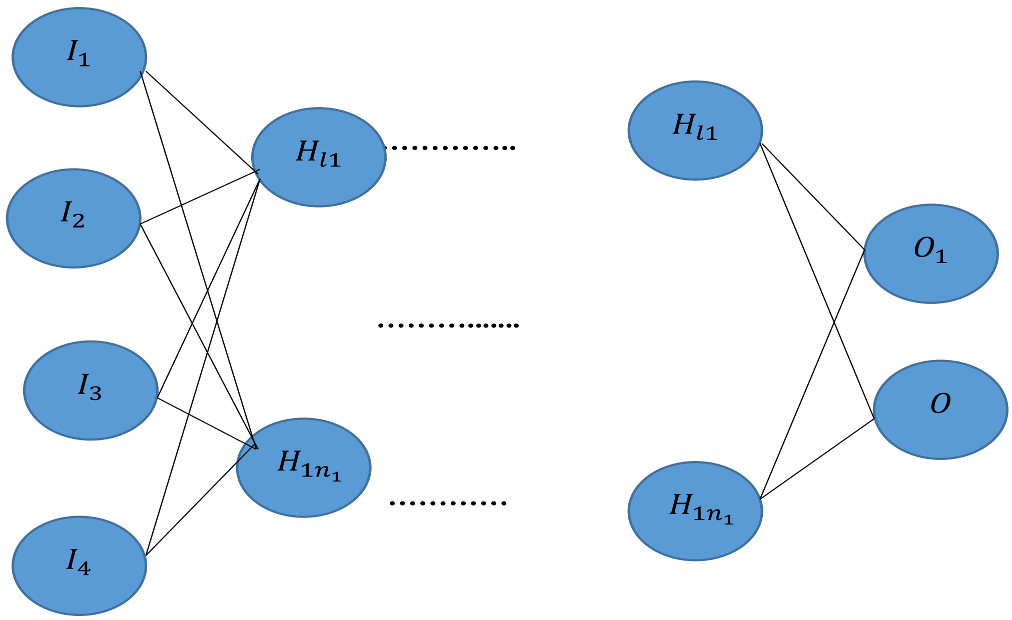

Fig. 3.1 The basic design of the proposed neural network.

I_1, I_2, I_3, and I_4 are input neurons of the input layer that takes the following values.

- *OUT* (I_1), the horizontal position of the mouse cursor in the latest selection event.
- *OUT* (I_2), the vertical position of the mouse cursor in the latest selection event.
- *OUT* (I_3), the horizontal direction of the mouse cursor in the n^{th} frame
- *OUT* (I_4), the vertical direction of the mouse cursor in the n^{th} frame

The values of *OUT* (I_3) and *OUT* (I_4) are the differences in the horizontal and vertical positions of the mouse cursor in the n^{th} frame from the locations *OUT* (I_1) and *OUT* (I_2) respectively. The neurons take the input value, '1' if the difference is positive, '-1' if the difference is negative, and '0' if no change. This is depicted in Fig. 3.2 and shown in Eq. (3.7) and Eq. (3.8).

$$OUT(I_3) = \begin{cases} 1, & if\ x' > x \\ -1, & if\ x' < x \\ 0, & Otherwise \end{cases} \tag{3.7}$$

$$OUT(I_4) = \begin{cases} 1, & if\ y' > y \\ -1, & if\ y' < y \\ 0, & Otherwise \end{cases} \tag{3.8}$$

where (x, y) is the position of the mouse cursor in the latest selection event and (x', y') is the position of the mouse cursor in the n^{th} frame in pixels.

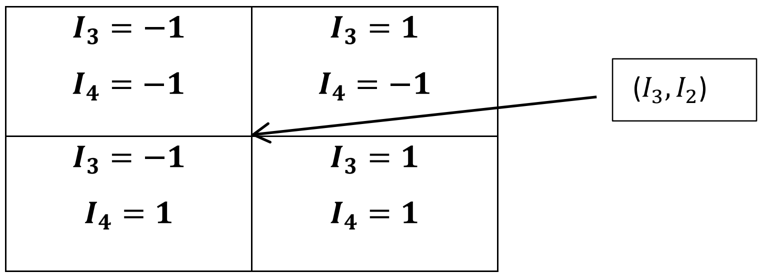

Fig. 3.2 Input neurons of the proposed Neural Network.

For the user selection point (x,y) pixel for (I_1, I_2), the region of the user selection area is identified as m^{th} row and n^{th} column, where $m = \left\lceil \frac{y+1}{S_P} \right\rceil$ and $n = \left\lceil \frac{x+1}{S_P} \right\rceil$. H_{jk} denotes Hidden neuron of jth layer and kth neuron where $1 \leq j \leq l$, $1 \leq k \leq n_j$, l is the total number of hidden layers and n_j is the total number of neurons in the jth hidden layer.

The activation function, which is the sum of the product of the previous layer values and the weights of the connections that are connected to the corresponding neurons of the hidden layer, is used to calculate the net inputs to the hidden layer neurons. The net input to a 1st hidden layer neuron is represented in Eq. (3.9).

$$IN(H_{1k}) = \sum_{z=1}^{4} \mathrm{W}(I_z, H_{1k})\mathrm{OUT}(\mathrm{I}_z) \tag{3.9}$$

The net input to 2^{nd} to l^{th} hidden layer neurons is represented in Eq. (3.10).

$$IN(H_{jk}) = \sum_{z=1}^{n_{j-1}} \mathrm{W}(H_{(j-1)z}, H_{jk})\mathrm{OUT}(H_{(j-1)z}) \tag{3.10}$$

The sigmoid function, which takes the matching net input values of the hidden neurons as the parameter, is used to calculate the output values of the neurons of the hidden layer. The hidden layer's neurons' output values are as denoted in Eq. (3.11).

$$OUT(H_{jk}) = \frac{1}{1+e^{-IN(H_{jk})}}, \quad 1 \leq j \leq l \tag{3.11}$$

The activation function, which is the sum of the product of the output values of the *l*th hidden layer and the weights of the links that connect the corresponding neurons of the *l*th hidden layer and output layer, is used to calculate the net inputs to the two neurons of the output layer, as shown Eq. (3.12) and Eq. (3.13).

$$IN(O_1) = \sum_{z=1}^{n_l} \mathrm{W}(H_{lz}, O_1)\mathrm{OUT}(H_{lz}) \tag{3.12}$$

$$IN(O_2) = \sum_{z=1}^{n_l} \mathrm{W}(H_{lz}, O_2)\mathrm{OUT}(H_{lz}) \tag{3.13}$$

The sigmoid function is used to generate the output values of the output layer's neurons, which takes the matching net input values of the output neurons as the parameter, as shown in Eq. (3.14) and Eq. (3.15).

$$OUT(O_1) = \frac{1}{1+e^{-IN(O_1)}} \tag{3.14}$$

$$OUT(O_2) = \frac{1}{1+e^{-IN(O_2)}} \tag{3.15}$$

The projected positions of the user's next mouse selection region by the proposed neural network are$OUT(O_1)$for the horizontal position and $OUT(O_2)$ for the vertical position of the mouse pointer in the computer monitor screen.

The purpose of the backpropagation approach, in which the neural network learns how to map random inputs to outputs correctly, is to optimize the weights. The backpropagation approach changes the weights of the links linking the neurons of the hidden layer and the output layer by first processing the output layer. After that, it returns to the hidden layer to update the weights of the links linking the hidden neurons and the input neurons [47].

As demonstrated in Eq. (3.16) and Eq. (3.17), the squared error function is utilized to calculate the errors in the output neurons' values.

$$Error(O_1) = \frac{1}{2}(Target(O_1) \quad - OUT(O_1))^2 \tag{3.16}$$

$$Error(O_2) = \frac{1}{2}(Target(O_2) \quad - OUT(O_2))^2 \tag{3.17}$$

where $(OUT(O_1),\ OUT(O_2))$ are the predicted horizontal and vertical positions and $(Target(O_1), Target(O_2))$ are the actual horizontal and vertical positions of the user’s targeted mouse cursor selection area in the monitor screen of the GUI based computer system.

The total error is the sum of the errors of the two neurons in the output layer, as shown in Eq. (3.18) or Eq. (3.19).

$$Error(Tot) = Error(O_1) + Error(O_2) \tag{3.18}$$

$$Error(Tot) = \frac{1}{2}(Target(O_1) \quad - OUT(O_1))^2 + \frac{1}{2}(Target(O_2) \quad - OUT(O_2))^2 \qquad (3.19)$$

The backpropagation method's main goal is to reduce error by making modifications to all of the weights. Each link connecting the neurons of the layers in the neural network must be adjusted such that the mouse cursor's predicted position values go closer to the target value, which is the real user's selection area on the computer screen.

The partial derivative of the total error with respect to the weight of the given link is the amount of change necessary in each of the link's weights. The changes required in the weights connecting the output layer's neuron O_1 and the neurons of the *l*th hidden layer are shown in Eq. (3.20) to Eq. (3.22).

$$\frac{\partial Error(Tot)}{\partial W(H_{lz},O_1)} = \frac{\partial Error(Tot)}{\partial OUT(O_1)} * \frac{\partial OUT(O_1)}{\partial IN(O_1)} * \frac{\partial IN(O_1)}{\partial W(H_{lz},O_1)}, \quad 1 \le z \le n_l \qquad (3.20)$$

where,

$$\frac{\partial Error(Tot)}{\partial OUT(O_1)} = OUT(O_1) - Target(O_1),$$

$$\frac{\partial OUT(O_1)}{\partial IN(O_1)} = OUT(O_1)(1 - OUT(O_1))$$

and

$$\frac{\partial IN(O_1)}{\partial W(H_{lz}, O_1)} = \mathrm{OUT}(H_{lz})$$

Therefore,

$$\frac{\partial Error(Tot)}{\partial W(H_{lz}, O_1)} = \left(OUT(O_1) - Target(O_1)\right) * OUT(O_1)\left(1 - OUT(O_1)\right) * \mathrm{OUT}(H_{lz}), \;\; 1 \le z \le n_l \qquad (3.21)$$

For the sake of minimizing the inaccuracy, the weights have been modified as shown in Eq. (3.22).

$$W(H_{lz}, O_1) = W(H_{lz}, O_1) - \eta * \frac{\partial Error(Tot)}{\partial W(H_{lz},O_1)} \qquad (3.22)$$

where η is the learning rate that determines how quickly the neural network must train in order to minimize the error.

The changes required in the weights connecting the output layer's neuron $O_2$and the neurons of the *l*th hidden layer are shown in Eq. (3.23) to Eq. (3.25).

$$\frac{\partial Error(Tot)}{\partial W(H_{lz},O_2)} = \frac{\partial Error(Tot)}{\partial OUT(O_2)} * \frac{\partial OUT(O_2)}{\partial IN(O_2)} * \frac{\partial IN(O_2)}{\partial W(H_{lz},O_2)}, 1 \le z \le n_l \qquad (3.23)$$

where,

$$\frac{\partial Error(Tot)}{\partial OUT(O_2)} = OUT(O_2) - Target(O_2),$$

$$\frac{\partial OUT(O_2)}{\partial IN(O_2)} = OUT(O_2)\big(1 - OUT(O_2)\big)$$

and

$$\frac{\partial IN(O_2)}{\partial W(H_{lz}, O_2)} = \mathrm{OUT}(H_{lz})$$

Therefore,

$$\frac{\partial Error(Tot)}{\partial W(H_{lz}, O_2)} = \big(OUT(O_2) - Target(O_2)\big) * OUT(O_2)\big(1 - OUT(O_2)\big) * \mathrm{OUT}(H_{lz}), 1 \leq \mathrm{z} \leq n_l \quad (3.24)$$

The weights are updated as shown in Eq. (3.25) for minimizing the error.

$$W(H_{lz}, O_2) = W(H_{lz}, O_2) - \eta * \frac{\partial Error(Tot)}{\partial W(H_{lz}, O_2)} \quad (3.25)$$

The changes required in the weights connected between the hidden layers are shown in Eq. (3.26).

$$\frac{\partial Error(Tot)}{\partial W(H_{(j-1)z}, H_{jk})} = \frac{\partial Error(Tot)}{\partial OUT(H_{jk})} * \frac{\partial OUT(H_{jk})}{\partial IN(H_{jk})} * \frac{\partial IN(H_{jk})}{\partial W(H_{(j-1)z}, H_{jk})} \quad (3.26)$$

where $1 \leq \mathrm{z} \leq n_{(j-1)}$, $2 \leq \mathrm{j} \leq l$ and $1 \leq k \leq n_j$.

The weights are updated as shown in Eq. (3.27) for minimizing the error.

$$W(H_{(j-1)z}, H_{jk}) = W(H_{(j-1)z}, H_{jk}) - \eta * \frac{\partial Error(Tot)}{\partial W(H_{(j-1)z}, H_{jk})} \quad (3.27)$$

The changes required in the weights connected between neurons of the input layer and the neurons of the first hidden layer are shown in Eq. (3.28).

$$\frac{\partial Error(Tot)}{\partial W(I_z, H_{1k})} = \frac{\partial Error(Tot)}{\partial OUT(H_{1k})} * \frac{\partial OUT(H_{1k})}{\partial IN(H_{1k})} * \frac{\partial IN(H_{1k})}{\partial W(I_z, H_{1k})} \quad (3.28)$$

where $1 \leq \mathrm{z} \leq 4$ and $1 \leq k \leq n_1$.

The weights are updated as shown in Eq. (3.29) for minimizing the error.

$$W(I_z, H_{1k}) = W(I_z, H_{1k}) - \eta * \frac{\partial Error(Tot)}{\partial W(I_z, H_{1k})} \quad (3.29)$$

Each time the neural network is updated, all of the weights of the links connecting the neurons of the layers are adjusted. The projected mouse cursor position values grow closer to the target value at each iteration, which is the actual user's selection area on the computer screen, which minimizes the inaccuracy [47].

4. Results

The proposed system employs backpropagation neural network techniques to introduce the prediction of products that users want to select. The most extensively used neural network for forecasting/prediction is the backpropagation algorithm [48-62]. The screen resolution is set to 1366 × 768 pixels with landscape orientation. The mouse pointer selection region on the computer screen is believed to be 40 square pixels.

Table 4.1: Horizontal (X) regions of the computer screen

X Region	X_{Min}	X_{Max}	X Region	X_{Min}	X_{Max}
1	0	39	19	720	759
2	40	79	20	760	799
3	80	119	21	800	839
4	120	159	22	840	879
5	160	199	23	880	919
6	200	239	24	920	959
7	240	279	25	960	999
8	280	319	26	1000	1039
9	320	359	27	1040	1079
10	360	399	28	1080	1119
11	400	439	29	1120	1159
12	440	479	30	1160	1199
13	480	519	31	1200	1239
14	520	559	32	1240	1279
15	560	599	33	1280	1319
16	600	639	34	1320	1359
17	640	679	35	1360	1366
18	680	719	36	1380	1378

Table 4.2: Vertical (Y) regions of the computer screen

Y Region	Y_{Min}	Y_{Max}	Y Region	Y_{Min}	Y_{Max}
1	0	39	11	400	439
2	40	79	12	440	479
3	80	119	13	480	519
4	120	159	14	520	559
5	160	199	15	560	599
6	200	239	16	600	639
7	240	279	17	640	679
8	280	319	18	680	719
9	320	359	19	720	759

10	360	399	20	760	768

The 20 vertical (Y) zones are shown in Table 4.2, while the 35 horizontal (X) regions are shown in Table 4.1. As a result, the forecast will use user chosen areas from a total of 700 locations (the product of 35 horizontal regions and 20 vertical regions). As indicated in Table 4.3, the system is tested using five samples of succeeding pixel square sections. It is assumed that the next target selection area and the user's most recent selection area are two consecutive selection areas. The most recent or initial selection area is j_i, $1 \leq j \leq 5$, and the succeeding or target selection area is j_o, $1 \leq j \leq 5$.

Table 4.3: Pixel areas of the sample square regions

Sample (*j*)	**Selection area**									
	j_i					j_o				
	Region (x,y)	X_{Min}	X_{Max}	Y_{Min}	Y_{Max}	***Region (x,y)***	X_{Min}	X_{Max}	Y_{Min}	Y_{Max}
1	(1,19)	0	39	720	759	(6,7)	200	239	240	279
2	(16,2)	600	639	40	79	(33,17)	1280	1319	640	679
3	(17,8)	640	679	280	319	(13,18)	480	519	680	719
4	(21,14)	800	839	520	559	(8,11)	280	319	400	439
5	(31,8)	1200	1239	280	319	(11,15)	400	439	560	599

Table 4.4: Input neuron values of the sample regions of Table 4.3.

Sample	**1**	**2**	**3**	**4**	**5**
I_1	0	600	640	800	1200
I_2	720	40	280	520	280
I_3	1	1	-1	-1	-1
I_4	-1	1	1	-1	1

Table 4.4 shows the values of the input neurons I_1 to I_4 for the five sample square sections of Table 4.3. All of the weights in the neural network are given random values at first.

Table 4.5: Actual vs predicted regions of mouse cursor

Sample	**1**	**2**	**3**	**4**	**5**
Actual Region (j_o)	R(6,7)	R(33,17)	R(13,18)	R(8,11)	R(11,15)
Initial Predicted Region (j_p)	R(17,15)	OUT	R(19,13)	R(18,8)	R(19,11)
After 5 passes (j_p)	R(8,8)	R(31,18)	R(14,18)	R(9,8)	R(12,11)
After 10 passes (j_p)	R(6,7)	R(32,17)	R(13,18)	R(8,8)	R(11,11)
After 20 passes (j_p)	R(6,7)	R(33,17)	R(13,18)	R(8,8)	R(11,11)

After 40 passes (j_p)	R(6,7)	R(33,17)	R(13,18)	R(8,11)	R(11,11)
After 80 passes(j_p)	R(6,7)	R(33,17)	R(13,18)	R(8,11)	R(11,15)

Table 4.5 and Fig. 4.1 show the initially predicted regions and anticipated the mouse pointer positions on the computer screen after 5, 10, 20, 40, and 80 passes for each of the five examples.

5. Discussion

The disparity between the real or target selection area j_o, $1 \leq j \leq 5$, and the anticipated region of the mouse cursor location j_p, $1 \leq j \leq 5$ is regarded as a clearly an evident error. The predicted location values of the mouse cursor get closer to the user's actual selection region on the computer screen by repeating the backpropagation procedure in each pass and updating all the weights. After 10 passes, the predicted position values of the mouse cursor are the same as the user's real selection region on the computer screen, resulting in the expected output for samples 1 and 3. The neural network is thus trained for samples 1 and 3. Sample 2 gets its output after 20 passes, sample 4 after 40 passes, and sample 5 after 80 passes, in the same way.

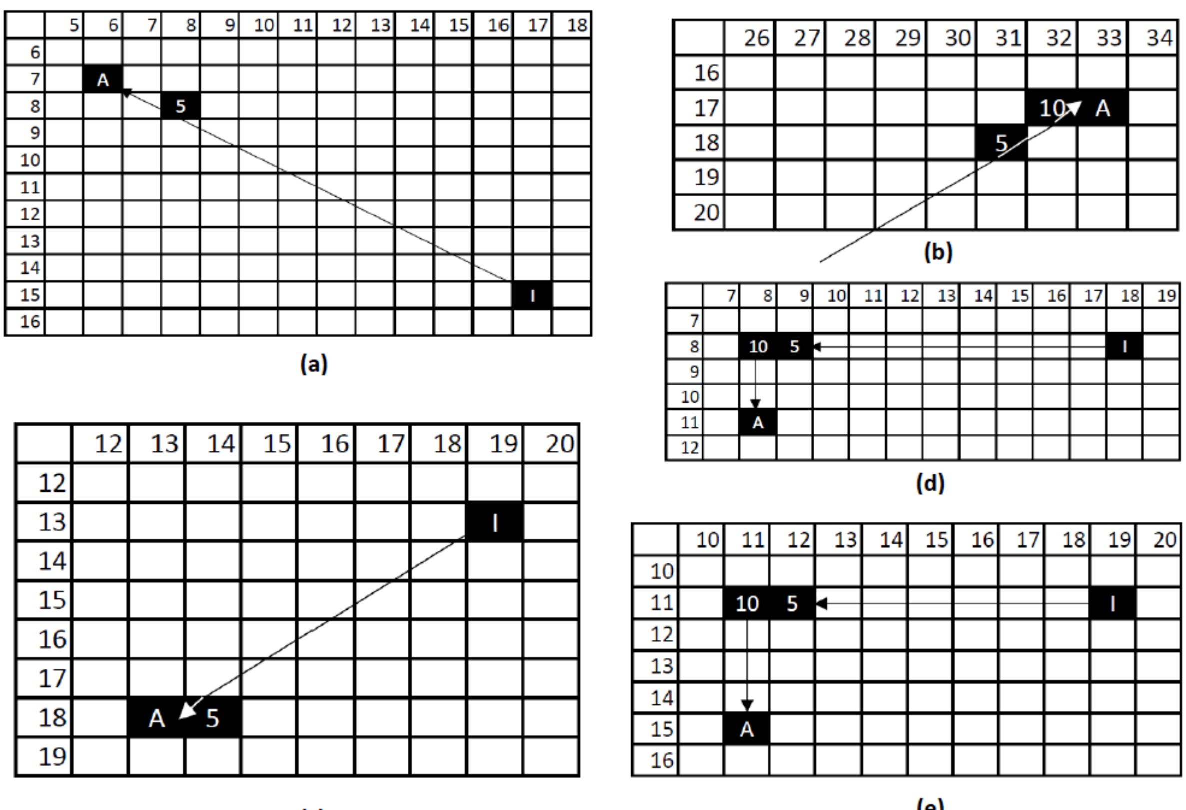
(a)
(b)
(c)
(d)
(e)

Fig 4.1 Actual vs predicted regions of the mouse cursor (a-e) for samples (1-5) of Table 4.5

5.1 Limitations

Real-time prediction in hands-free computing for disabled individuals faces challenges such as accuracy issues in dynamic environments, reliance on quality training data, adapting to user variability, potential latency, power consumption concerns, and privacy considerations.

5. Conclusion and Future work

The proposed system has used neural network techniques to introduce the prediction of things that the user wishes to select to improve the performance of selecting the user's target item in the GUI-based computer system. In the input layer, there are four neurons, and in the output layer, there are two neurons, with hidden layers in between. The technique is put to the test by taking five samples of such locations in a row. The weights connecting the neurons in the network are changed in consecutive runs and predicted values move closer to the target mouse cursor location values by minimizing the error. When compared to the current system, the suggested system's accomplishments in achieving the cursor's horizontal and vertical mobility are noteworthy. When the user maintains a stable head position, the system also properly ignores any slight head movement captured by the web camera, preventing any feature loss. The model's performance is measured in this case by the precision of mouse clicks.

Exploring more sophisticated neural network architectures will further improve prediction accuracy and reduce training time. Considering user-specific adaptations by incorporating personalized training data or allowing users to fine-tune the system according to their preferences could contribute to a more user-centric approach.

References

[1] World Health Organization. (2011). World Report on Disability. (Report No. WHO/NMH/VIP/11.01). https://www.who.int/disabilities/world_report/2011/report.pdf

[2] Social Statistics Division - Ministry of Statistics and Programme Implementation – Government of India. (2017). Disabled Persons in India: A statistical profile 2016. http://mospi.nic.in/sites/default/files/publication_reports/Disabled_persons _in_India_2016.pdf

[3] Wikipedia. (n.d.). Computer accessibility. https://en.wikipedia.org/wiki/Hands-free_computing

[4] Wikipedia. (n.d.). Hands-free computing. https://en.wikipedia.org/wiki/Computer_accessibility

[5] Honye, S. and Thinyane, H. (2012) WiiMS: Simulating mouse and keyboard for motor-impaired users, Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, Pretoria, South Africa, Pages 188-195, 1-3 Oct 2012, ACM, New York.

[6] Chathuranga, K., Samarawickrama, C., Chandima, L., Chathuranga, D. and Abeykoon, S. (2010) Hands free interface for Human Computer Interaction, Proceedings of Fifth International Conference on Information and Automation for Sustainability, Colombo, Sri Lanka, Pages 359-364, 17-19 Dec 2010, IEEE.

[7] Kim, S., Park, M., Anumas, S. and Yoo, J. (2010) Head mouse system based on gyro- and opto-sensors, Proceedings of 3rd International Conference on Biomedical Engineering and Informatics, Yantai, China, 16-18 Oct 2010, Pages 1503-1506, IEEE.

[8] Takahashi, J., Suezawa, S., Hasegawa, Y. and Sankai, Y. (2011) Tongue Motion-based Operation of Support System for Paralyzed Patients, Proceedings of IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, Pages 1-6, 29 June-1 July 2011, IEEE.

[9] Kuzume, K. (2012) Evaluation of tooth-touch sound and expiration based mouse device for disabled persons, Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops, Lugano, Switzerland, Pages 387-390, 19-23 March 2012, IEEE.

[10] Arai, K. and Mardiyanto, R. (2011) Eye based HCI with moving keyboard for reducing fatigue effects, Proceedings of Eighth International Conference on Information Technology: New Generations,Las Vegas, NV, USA, Pages 417–422, 11-13 April 2011, IEEE.

[11] Lupu, G., Bozomitu, G., Păsărică, A. and Rotariu, C. (2017) Eye Tracking User Interface for Internet Access Used in Assistive Technology, Proceedings ofE-Health and Bioengineering Conference (EHB), Sinaia, Romania, Pages 659-662, 22-24 June 2017, IEEE.

[12] Szeliski, R. (2011) Computer vision: Algorithms and applications, New York: Springer, ISBN 978-1-84882-935-0.

[13] Viola, P. and Jones, M. (2004) Robust Real-Time Face Detection, International Journal of Computer Vision, Volume 57, Issue 2, Pages 137-154.

[14] Kazemi, V. and Sullivan, J. (2014) One Millisecond Face Alignment with an Ensemble of Regression Trees, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision Foundation, Pages 1867-1874.

[15] Cilimkovic, M. (2010) Neural Networks and Back Propagation Algorithm, Institute of Technology Blanchardstown, Ireland [online] Available at: <http://dl.icdst.org/pdfs/files/c9ea365e15acbb738862b8114769e96b.pdf >.

[16] Chareonsuk, W., Kanhaun, S., Khawkam, K. and Wongsawang, D. (2016) Face and Eyes mouse for ALS Patients, Proceedings ofFifth ICT International Student Project Conference, Nakhon Pathom, Thailand, Pages 77-80, 27-28 May, IEEE.

[17]. Kumar, S., Rai, A., Agarwal, A. and Bachani, N. (2010) Vision based human interaction system for disabled, Proceedings of 2nd IEEE International Conference on Image Processing Theory, Tools and Applications, Paris, France, Pages 441-446, 7-10 Jul 2020, IEEE.

[18] Bian, P., Hou, J., Chau, P. and Thalmann, M. (2016) Facial position and expression-based human-computer interface for persons with tetraplegia, IEEE Journal of Biomedical and Health Informatics, Volume 20, Issue 3, Pages 915-924.

[19] Vasanthan, M., Murugappan, M., Nagarajan, R., Ilias, B. and Letchumikanth, J. (2012) Facial expression based computer cursor control system for assisting physically disabled person, Proceedings of International Conference on Communication, Networks and Satellite, Bali, Indonesia, Pages 172-176, 12-14 July 2012, IEEE.

[20] Kraichan, C. and Pumrin, S. (2014) Face and Eye Tracking for Controlling Computer Functions, Proceedings of 11th International Conference on Electrical Engineering / Electronics, Computer,

Telecommunications and Information Technology, Nakhon Ratchasima, Thailand, Pages 1-6, 14-17 May 2014, IEEE.

[21] Parmar, K., Mehta, B. and Sawant, R. (2012) Facial-feature based Human-Computer Interface for disabled people, Proceedings of International Conference on Communication, Information & Computing Technology, Mumbai, India, Pages 1-5, 19-20 Oct 2012, IEEE.

[22]. Eric, M. and Betke, M. (2010) Blink and wink detection for mouse pointer control, Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments, Samos, Greece, 23-25 Jun 2010, ACM, New York.

[23] Fathi, A. and Mohammadi, A. (2015) Camera-based eye blinks pattern detection for intelligent mouse, Signal, Image and Video Processing, Volume 9, Issue 8, Pages 1907–1916.

[24] M. N. Islam et al. (2021) Developing a Novel Hands-Free Interaction Technique Based on Nose and Teeth Movements for Using Mobile Devices, IEEE Access, vol. 9, pp. 58127-58141, doi: 10.1109/ACCESS.2021.3072195.

[25] C. M. Asad et al. (2018) Removing Disabilities: Controlling Personal Computer Through Head Movements and Voice Command, IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), 2018, pp. 1-4, doi: 10.1109/ICAICT.2018.8747123.

[26] R. A. Antunes, L. B. Palma, F. V. Coito, H. Duarteramos and P. Gil (2016) Intelligent human-computer interface for improving pointing device usability and performance, 12th IEEE International Conference on Control and Automation (ICCA), pp. 714-719, doi: 10.1109/ICCA.2016.7505363.

[27] K. Meena, M. Kumar and M. Jangra (2020) Controlling Mouse Motions Using Eye Tracking Using Computer Vision, 4th IEEE International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 1001-1005, doi: 10.1109/ICICCS48265.2020.9121137.

[28] R. S. Baştuğ, B. Yeşilkaya, M. Unay and A. Akan (2018) Virtual Mouse Control by Webcam for the Disabled, IEEE Medical Technologies National Congress (TIPTEKNO), 2018, pp. 1-4, doi: 10.1109/TIPTEKNO.2018.8597147.

[29] Anantha Prabha, P., Srinivash, K., Vigneshwar, S., Viswa, E. (2022). Mouse Assistance for Motor-Disabled People Using Computer Vision. In: Mahapatra, R.P., Peddoju, S.K., Roy, S., Parwekar, P., Goel, L. (eds) Proceedings of International Conference on Recent Trends in Computing. Lecture Notes in Networks and Systems, vol 341. Springer, Singapore. https://doi.org/10.1007/978-981-16-7118-0_35

[30] Ghadekar, P., Korpal, P., Chendake, P., Bansal, R., Pawar, A., Bhor, S. (2021). Real-Time Hands-Free Mouse Control for Disabled. In: Swain, D., Pattnaik, P.K., Athawale, T. (eds) Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1311. Springer, Singapore. https://doi.org/10.1007/978-981-33-4859-2_16

[31] Sancheti, K., Suhaas, A. and Suresh, P. (2018) Hands-free Cursor Control using Intuitive Head Movements and Cheek Muscle Twitches, Proceedings of TENCON 2018: IEEE Region 10 Conference, Jeju, Korea (South), 28-31 October 2018, Pages 0356-0361.

[32] Nasor, M., Rahman, M., Zubair, M., Ansari, H. and Mohamed F. (2018) Eye-controlled mouse cursor for physically disabled individual, Proceedings of 2018 Advances in Science and Engineering Technology International Conferences (ASET), 6 Feb.-5 April 2018, Dubai, Pages 1-4, IEEE.

[33] Magee, J., Epstein, S., Missimer, S., Kwan, C. and Betke, M. (2011) Adaptive Mouse-Replacement Interface Control Functions for Users with Disabilities, Universal Access in Human-Computer Interaction, Users Diversity, UAHCI 2011, Lecture Notes in Computer Science, Volume 6766, Springer, Berlin, Heidelberg.

[34] Sugano, Y., Matsushita, Y., Sato, Y. and Koike, H. (2015) Appearance-based gaze estimation with online calibration from mouse operations, IEEE Transactions on Human-Machine Systems, Volume 45, Issue 6, Pages 750-760.

[35] Valenti, R., Lablack, A., Sebe, N., Djeraba, C. and Gevers, T. (2010) Visual gaze estimation by joint head and eye information, Proceedings of 20th International Conference on Pattern Recognition, Istanbul, Turkey, Pages 3870-3873, 23-26 Aug 2010, IEEE.

[36] R. Riad Saboundji and R. Adrian Rill (2020) Predicting Human Errors from Gaze and Cursor Movements, IEEE International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207189.

[37] Betke, M., Gips, J. and Fleming, P. (2002) The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities, IEEE Transactions on Neural Systems and Rehabilitation Engineering, Volume 10, Issue 1, Pages 1-10.

[38] P. Salunkhe and A. R. Patil (2016) A device controlled using eye movement, IEEE International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, pp. 732-735, doi: 10.1109/ICEEOT.2016.7754779.

[39] Kai, W., Peng, W., Yang, J, Meng, D. and Yu, Q. (2020) Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition, IEEE Transactions on Image Processing, Vol. 29, 2020, 4057 to 4069

[40] Kim, Hj., Kim, Hw. and Hwang (2018) E.,Real-time shape tracking of facial landmarks, Multimedia Tools and Applications, Springer Nature, doi 10.1007/s11042-018-6814-7.

[41] Silwal, R., Alsadoon, A., Prasad, P.W.C. et al. (2020) A novel deep learning system for facial feature extraction by fusing CNN and MB-LBP and using enhanced loss function, Multimedia Tools and Applications, Springer Nature, 31027–31047 (2020), doi: 10.1007/s11042-020-09559-1.

[42] Z. Wang et al. (2019) AirMouse: Turning a Pair of Glasses Into a Mouse in the Air, IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7473-7483, Oct. 2019, doi: 10.1109/JIOT.2019.2900887.

[43] A. Murata and W. Karwowski (2021) Automatic Lock of Cursor Movement: Implications for an Efficient Eye-Gaze Input Method for Drag and Menu Selection, IEEE Transactions on Human-Machine Systems, vol. 49, no. 3, pp. 259-267, June 2019, doi: 10.1109/THMS.2018.2884737.

[44] Cagdas, E., Ayhan, A., Aliye, T., Sahin, A. (2021) Novel hands-free interaction techniques based on the software switch approach for computer access with head movements, Universal Access in the Information Society, 20:617–631, Publisher: Springer, doi: 10.1007/s10209-020-00748-1

[45] Rahib, H. A., Murat, A. (2021) Head mouse control system for people with disabilities, Wiley: Expert Systems. 2019;e12398, Accepted: 22 February 2019, DOI: 10.1111/exsy.12398

[46] Wikipedia. (n.d.). Soft computing. https://en.wikipedia.org/wiki/Soft_computing

[47] Kriesel, D. (2007) A Brief Introduction to Neural Networks[online] Available at: <http://www.dkriesel.com/en/science/neural_networks>.

[48] Eldon, R., López, E., Kim, J. and Park, H. (2013) Back Propagation Neural Network Model for Predicting the Performance of Immobilized Cell Biofilters Handling Gas-Phase Hydrogen Sulphide and Ammonia, Biotechnology in Environmental Monitoring and Pollution Abatement, Volume 2013, Article ID 463401, doi: 10.1155/2013/463401.

[49] Rathish, C. R., and A. Rajaram. "Efficient path reassessment based on node probability in wireless sensor network." International Journal of Control Theory and Applications 34.2016 (2016): 817-832.

[50] S Rahamat Basha, Chhavi Sharma, Farrukh Sayeed, AN Arularasan, PV Pramila, Santaji Krishna Shinde, Bhasker Pant, A Rajaram, Alazar Yeshitla, "Implementation of Reliability Antecedent Forwarding Technique Using Straddling Path Recovery in Manet," Wireless Communications & Mobile Computing (Online), vol. 2022, 2022.

[51] Rathish, C. R., and A. Rajaram. "Hierarchical Load Balanced Routing Protocol for Wireless Sensor Networks." International Journal of Applied Engineering Research 10.7 (2015): 16521-16534.

[52] D. N. V. S. L. S. Indira, Rajendra Kumar Ganiya, P. Ashok Babu, A. Jasmine Xavier, L. Kavisankar, S. Hemalatha, V. Senthilkumar, T. Kavitha, A. Rajaram, Karthik Annam, Alazar Yeshitla, "Improved Artificial Neural Network with State Order Dataset Estimation for Brain Cancer Cell Diagnosis", BioMed Research International, vol. 2022, 10 pages, 2022.

[53] P. Ganesh, G. B. S. R. Naidu, Korla Swaroopa, R. Rahul, Ahmad Almadhor, C. Senthilkumar, Durgaprasad Gangodkar, A. Rajaram, Alazar Yeshitla, "Implementation of Hidden Node Detection Scheme for Self-Organization of Data Packet", Wireless Communications and Mobile Computing, vol. 2022, 9 pages, 2022. https://doi.org/10.1155/2022/1332373.

[54] M. Dinesh, C Arvind, S.S Sreeja Mole, C.S. Subash Kumar, P. Chandra Sekar, K. Somasundaram, K. Srihari, S. Chandragandhi, Venkatesa Prabhu Sundramurthy, "An Energy Efficient Architecture for

Furnace Monitor and Control in Foundry Based on Industry 4.0 Using IoT", Scientific Programming, vol. 2022, Article ID 1128717, 8 pages, 2022. https://doi.org/10.1155/2022/1128717.

[55] S Kannan, A Rajaram, "Enhanced Stable Path Routing Approach for Improving Packet Delivery in MANET," Journal of Computational and Theoretical Nanoscience, vol. 4, no. 9, pp. 4545-4552, 2017.

[56] RP Prem Anand, A Rajaram. "Effective timer count scheduling with spectator routing using stifle restriction algorithm in manet," IOP Conference Series: Materials Science and Engineering, vol. 994, no. 1, pp. 012031, 2022.

[57] Rathish, C. R., and A. Rajaram. "Efficient path reassessment based on node probability in wireless sensor network." International Journal of Control Theory and Applications 34.2016 (2016): 817-832.

[58] Kumar, K. Vinoth, and A. Rajaram. "Energy efficient and node mobility based data replication algorithm for MANET." (2019).

[59] CR Rathish, A Rajaram, "Sweeping inclusive connectivity based routing in wireless sensor networks," ARPN Journal of Engineering and Applied Sciences, vol. 3, no. 5. pp. 1752-1760, 2018.

[60] K. Mahalakshmi, K. Kousalya, Himanshu Shekhar, Aby K. Thomas, L. Bhagyalakshmi, Sanjay Kumar Suman, S. Chandragandhi, Prashant Bachanna, K. Srihari, Venkatesa Prabhu Sundramurthy, "Public Auditing Scheme for Integrity Verification in Distributed Cloud Storage System", Scientific Programming, vol. 2021, Article ID 8533995, 5 pages, 2021. https://doi.org/10.1155/2021/8533995.

[61] J. Divakaran, Somashekhar Malipatil, Tareeq Zaid, M. Pushpalatha, Vilaskumar Patil, C. Arvind, T. Joby Titus, K. Srihari, M. Ragul Vignesh, Baswaraj Gadgay, Venkatesa Prabhu Sundramurthy, "Technical Study on 5G Using Soft Computing Methods", Scientific Programming, vol. 2022, Article ID 1570604, 7 pages, 2022. https://doi.org/10.1155/2022/1570604.

[62] S. Shitharth, Pratiksha Meshram, Pravin R. Kshirsagar, Hariprasath Manoharan, Vineet Tirth, Venkatesa Prabhu Sundramurthy, "Impact of Big Data Analysis on Nanosensors for Applied Sciences Using Neural Networks", Journal of Nanomaterials, vol. 2021, Article ID 4927607, 9 pages, 2021. https://doi.org/10.1155/2021/4927607.