 doi:10.20944/preprints202007.0625.v1

Article

Computer Vision for Elderly Care Based on Hand Gestures

Munir Oudah [1], Ali Al-Naji [1, 2, *] and Javaan Chahl [2]

[1] Electrical Engineering Technical College, Middle Technical University, Baghdad 1022, Iraq
[2] School of Engineering, University of South Australia, Mawson Lakes SA 5095, Australia
* Correspondence: ali.al-naji@unisa.edu.au; Tel.: +9647710304768

Abstract: Hand gestures may play an important role in medical applications for health care of elderly people, where providing a natural interaction for different requests can be executed by making specific gestures. In this study we explored three different scenarios using a Microsoft Kinect V2 depth sensor then evaluated the effectiveness of the outcomes. The first scenario utilized the default system embedded in the Kinect V2 sensor, which depth metadata gives 11 parameters related to the tracked body with five gestures for each hand. The second scenario used joint tracking provided by Kinect depth metadata and depth threshold together to enhance hand segmentation and efficiently recognize the number of fingers extended. The third scenario used a simple convolutional neural network with joint tracking by depth metadata to recognize five categories of gestures. In this study, deaf-mute elderly people execute five different hand gestures to indicate a specific request, such as needing water, meal, toilet, help and medicine. Then, the requests were sent to the care provider's smartphone because elderly people could not execute any activity independently. The system transferred these requests as a message through the global system for mobile communication (GSM) using a microcontroller.

Keywords: Elderly care; Hand gesture, computer vision system; Microsoft Kinect depth sensor; Arduino Nano Microcontroller; Global system for mobile communication (GSM).

1. Introduction

The aged population in the world is increasing by 9 million per year and is expected to reach more than 800 million by 2025 [1]. Health care programs are proportional directly to increase in elderly population numbers and lead to an increase in requests for home care services and also a need more healthcare organization and administration. Home services can reduce cost rather than the long term care provided inside specialized facilities. Also, it has a positive effect on elderly people when providing care service in their own homes. For enhancing the home care services, it is necessary to design a robust system that is appropriate for the case of elderly people and the type of care provided. This paper focuses on the case of elderly disabled people who are speechless due to sudden stroke or medical accident or maybe who are already deaf-mute who has difficulty to communicate with other family members at home, especially for providing daily requirements.

Human-computer interaction (HCI) based on a computer vision system is a promising technique that can provide natural interaction of hand gestures provided in front of a camera. Recently, the interface for gesture based computer interaction has been achieved using glove attached sensors, such as accelerometer, gyroscope, tactile and flex sensor. These sensors respond to hand movements and capture the proper coordinate of hand and finger. However, the glove attached sensor have some limitations, starting with the need to wear a glove, but also including wire connection that causes discomfort for elderly people, and power failure causes. Moreover, they may cause some skin problems for people with sensitive skin. Other techniques of hand gesture

 doi:10.20944/preprints202007.0625.v1

recognition utilized marked gloves, where a camera could detect the different colors on the gloves [29].

The hand gestures can be simple, such as finger detection gestures or maybe complex such as specific pose performed using one or two hands. On the other hand, the hand gesture can be dynamic (provide gestures by moving hand in a specific direction or pattern) and also static hand gestures (perform a particular arrangement of fingers and palm). Dynamic gestures may use one or both hands to execute actions, such as zooming and rotating with a continuous moving hand, including interaction with virtual reality. Whereas, the static gesture is implemented by one or two hands such as hand gestures used for sign language, home automation, medical imaging viewing and annotating.

Many proposed systems for hand gesture detection in different applications have some limitations in terms of lighting variations or background issues that affect hand segmentation and recognition rate. The Kinect depth sensor provides 3D x, y, z coordination of an object by analyzing data returned by the depth sensor based on a ray sent by an infrared projector, that effectively overcomes lighting and background limitations.

This study proposed an easy and non-contact communication method to help elderly people by sending their requests to the care provider or family member smartphone via SMS at night time or day time.

The rest of this paper is arranged as follows: Section 2 presents the related works and mentions the weaknesses of former works. Section 3 describes the materials and methods, including the participants and experimental setup, hardware design and hand gesture scenarios. Section 4 shows the experimental results and discusses the obtained results. Finally, conclusion and future research directions are provided in Section 5.

2. Related Works

In the last decade, hand gestures have become a promising interaction method, with many published studies undertaken considering different applications. The precision of the hand gesture interaction system depends on some factors, including the type of camera used and resolution, the technique utilized for hand segmentation and the recognition algorithm used. This section summarizes some techniques that have used the Microsoft Kinect depth sensor, as shown in Table 1.

A study by Ren et al. [2] proposed a new method based on the finger earth mover distance (FEMD) approach that was evaluated in terms of speed and precision, and then compared with the shape-matching algorithm using the depth map and color image acquired by the Kinect camera. Another study by Ma et al. [3] improved depth threshold segmentation by combining depth and color information using the hierarchical scan method, and then hand segmentation was used based on the local neighbor method. This approach gave results over a range of up to 2 meters. Another study by Ma et al. [4] proposed a wireless interaction system for a robot by translating hand gesture information into commands, where a slot algorithm was utilized to identify finger gestures. Lee et.al [5] presented a developed algorithm that used an RGB color frame and converted it to a binary frame using Otsu's global threshold. After that, a depth range was selected for hand segmentation, and then the two methods were aligned. Finally, the k Nearest Neighbor (kNN) algorithm was used with Euclidian distance for finger classification. In a study by Dh et al. [6], the skin–motion detection technique was used to detect the hand, and then Hu moments were applied to feature extraction, after which HMM was used for gesture recognition. In another study Li et al. [7], a depth threshold was used to segment the hand, and then a K-mean algorithm was applied to obtain pixels from both of the user's hands . Another study by Xi et al. [8] used a skeleton tracking method to capture the hand and locate fingertips, where the Kalman filter was used to record the motion of the tracked joint. The cascade extraction technique was used with a novel recursive connected component algorithm. A study by Kim et al. [9] proposed a new method based on a near depth range of fewer than 0.5 meters where skeletal data was not provided by the Kinect. This method was implemented using two image frames: depth and infrared. Next, Graham's scan algorithm was used to detect the

 doi:10.20944/preprints202007.0625.v1

convex hulls of the hand in order to merge with the result of the contour tracing algorithm to detect the fingertips. In a study by Bakar et al. [10], the segmentation used 3D depth data selected based on a threshold range. Bamwenda et al. [11] used depth information with skeletal and color data to detect the hand. The segmented hand was then matched with the dataset using a support vector machine (SVM) and artificial neural networks (ANN) for recognition. The authors concluded that ANN was more accurate than SVM. Another study by Desai et al. [12] proposed a home automation system for facility control by senior citizens who face a challenge, using a computer vision system based on a Kinect sensor. Desai et al. [13] introduced an algorithm based on an RGB color and Otsu's global threshold. After that, a depth range was selected for hand segmentation and then the two methods were aligned. Finally, the kNN algorithm was used with Euclidian distance for finger classification. In Karbasi et al. [14], the hand was segmented based on depth information using a distance method and background subtraction method. Iterative techniques were applied to remove the depth image shadow and decrease noise. Bakar et al. [15] used fingertips selected using depth threshold and the K-curvature algorithm based on depth data. Wen et. al [16] proposed a gesture recognition system to segment the hand based on skin color and used K-means clustering and convex hull to identify hand contour and finally detect fingertips. Another study by Li et al. [17] presents a developed system to combine depth information and skeletal data, facing the challenge of a complex background and illumination variation, rotation invariance, in which some constraints were set in hand segmentation. Marin et al. [18] used two techniques together to detect finger regions such as leap motion and Kinect devices to extract different feature sets. The system accuracy was increased by combining the two device features, where the leap motion provides high-level data information but lower reliability than the Kinect sensor which provides a full depth map. Extensive review on this subject can be found in [29].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

Table 1. A set of research papers that used Kinect depth sensor for hand gestures.

Author	Type of camera	Resolution	Techniques/ Methods for segmentation	Feature extract type	Classify algorithm	Recognition rate	No. of gestures	Application area	Invariant factor	Distance from camera
[2]	Kinect V1	640x480 320x240	depth map & color image	finger	Near-convex Decomposition & Finger-Earth Mover's Distance (FEMD).	93.9 %	10-gesture	HCI applications	No	No
[3]	Kinect V2	1920x1080 512x424	threshold segmentation & local neighbor method.	fingertip	convex hull detection algorithm	96 %	6-gesture	natural human-robot interaction.	Some small noise spots around the hands will reduce the detection performance of fingertips	0.5 to 2.0 meter
[4]	Kinect V1	640x480 320x240	depth threshold	fingertip	k-curvature algorithm	No	5-gesture	human-robot interactions	No	No
[5]	Kinect V1	640x480 320x240	3D depth sensor	fingertips	shape bases matching	91 %	6-gesture	finger painting and mouse controlling	low accuracy in rough conditions	0.5 to 0.8 meter
[6]	Kinect V1	640x480 320x240	skin and motion detection & hu moments	dynamic hand gesture	Discrete Hidden Markov Model	Table	single handed postures combination of position, orientation & 10-gesture	controlling DC servo motor action	backward movement gesture effect recognition rate	No
[7]	Kinect V1	640x480 320x240	Depth thresholds	fingertip	K-means clustering algorithm convex hulls	90%	9-gesture	real-time communication such chatting with speech	difficulty of recognize one of the nine sgesture	0.5 to 0.8 meter
[8]	Kinect V2	1920x1080 512x424	threshold and recursive connected component analysis	hand skeleton & fingertip	Euclidean distance and geodesic distance	No	hand motion	controlling actions and interactions.	occlusions may have side effects on the depth data.	No
[9]	Kinect V2	depth – 512 × 424	operation of depth and infrared images	finger counting & hand gesture	number of separate areas	No	finger count & two hand gestures	mouse-movement controlling	No	< 0.5 meters
[10]	Kinect V1	depth – 640 × 480	threshold range	hand gesture	No	No	hand gesture	hand rehabilitation	No	0.4–1.5 meters

								system		
[11]	Kinect V2	depth – 512 × 424	skeletal data stream & depth & color data streams	hand gesture	support vector machine (SVM) & artificial neural networks (ANN)	93.4% for SVM 98.2% for ANN	24 alphabets hand gesture	American Sign Language	No	0.5–0.8 meters
[12]	Kinect V1	depth – 640 × 480	range of depth image	hand gestures 1–5	kNN classifier & Euclidian distance	88%	5 gestures	electronic home appliances	No	0.25–0.65 meters
[13]	Kinect V2	RGB – 1920 × 1080 depth – 512 × 424	Otsu's global threshold	finger gesture	kNN classifier & Euclidian distance	90%	finger count	Human computer interaction (HCI)	hand not identified if it's not connected with boundary	0.25–0.65 meters
[14]	Kinect V1	depth – 640 × 480	distance method	hand gesture	No	No	hand gesture	human–computer interaction (HCI)	No	No
[15]	Kinect V1	depth – 640 × 480	depth threshold and K-curvature	finger countin g	depth threshold and K-curvature	73.7%	5 gestures	picture selection application	detection fingertips should though the hand was moving or rotating	No
[16]	Kinect V1	depth – 640 × 480	skin color segmentation and depth joint	fingertip	K-means clustering & convex hull	No	fingertip gesture	human–computer interaction (HCI)	No	No
[17]	Kinect V2	RGB – 1920 × 1080 depth – 512 × 424	double-thresh old segmentation and skeletal data	fingertip	fingertip angle characteristics & SIFT keypoints	No	10-gesture	(HCI)	some constraints are set in hand segmentation	No
[18]	Kinect V1	depth – 640 × 480	depth and color data & leap Motion	Position of the fingertip s	multi-class SVM classifier	91:3%	10-gesture	subset of the American Manual Alphabet	Leap Motion is limit while Kinect provides the full depth map.	No

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

3. Materials and Methods

3.1. Participants and Experimental Setup

This study was conducted with three different experiments, where each of them evaluated with the same group of elderly participants, including two males and one female between the ages of 65 and 75 and one adult (35 years). This study adhered to the Declaration of Helsinki ethical principles (Finland 1964) where written informed consent forms were obtained from all participants after a full explanation of the experimental procedures. The experiment was for approximately half an hour for each participant at the home environment and repeated at various times to obtain sufficient outcomes. The Microsoft Kinect v2 sensor was installed at different distances that fell approximately within 1.2-4.5 m with an angle of 0°. The videos were captured at a resolution of 512×424 and a frame rate of 30 fps. The Kinect sensor was connected to a laptop with a conversion power adaptor and a standard development kit (Microsoft Kinect for Windows SDK 2.0). Figure 1 shows the experimental setup of the proposed system.

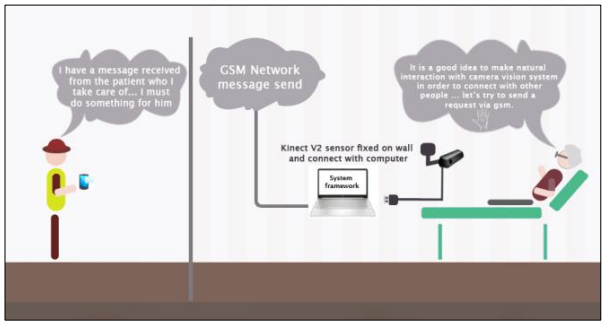

Figure 1. Experimental setup of the proposed method.

3.2. Hardware Design

The schematic diagram of the proposed method is shown in Figure 2. The system design hardware of the proposed system can be divided into four main parts: The Microsoft Kinect sensor, arduino Nano microcontroller, GSM module Sim800l and DC-DC chopper (buck).

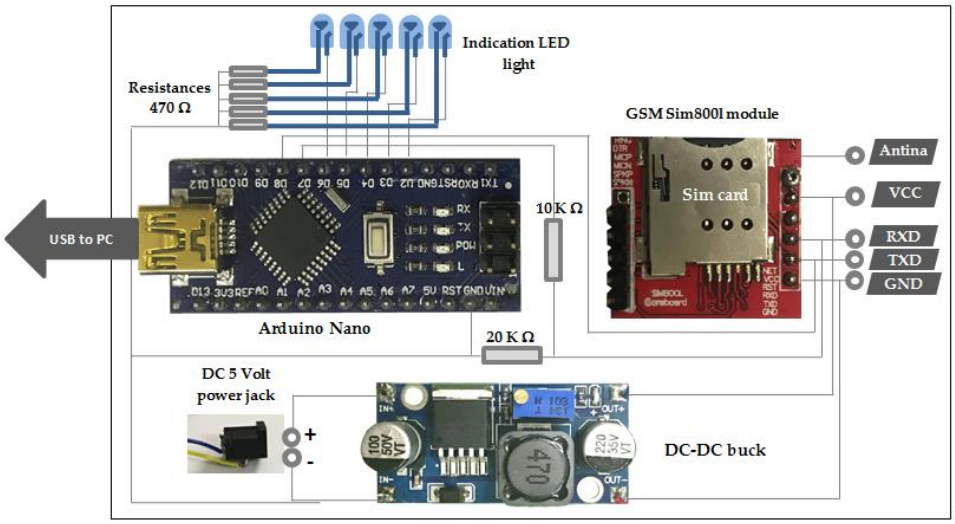

Figure 2. The schematic diagram of the proposed method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

3.2.1. Microsoft Kinect sensor

Microsoft released the Kinect V1 as a peripheral device in 2010 for game purposes. With the increasing request in the store, the company has modified it to be appropriate for operating systems like Windows using Microsoft Kinect Standard Development Kit (Microsoft Kinect for Windows SDK 2.0) and power converter [19][20][21][22]. With this, it provides an easy tool for developers and researchers to do development on a computer. After that, the new version of Kinect was released in 2014 that supports the subsequent generation of the sensor (Kinect V2) with an improvement in rendering, precision and field of view [19][20][21][22]. This is because the Kinect sensor V2 utilizes a time of flight (ToF) technology [23] instead of light coding technology [24] utilized in Kinect sensor V1. The differentiation between the two releases (Kinect V1, V2) is extensively explained in [22][24][25]. Figure 4 shows the outer view of the Microsoft Kinect sensor V2 for Xbox One.

Figure 3. Microsoft Kinect sensor v2 for Xbox.

Kinect sensor V2 includes three visual sensors, a RGB sensor, an IR sensor, an IR projector that provides outputs, an RGB image, and a depth image. These features permit body tracking, 3D human rebuilding, human skeletal tracking and human joint tracking. Because Kinect V2 is supplied with PC adapter and has particular advantages at a low cost, and is intended for the gaming uses, it is common equipment for many biomedical implementations in both clinical and non-clinical applications [7].

3.2.2. Arduino Nano Microcontroller

The microcontroller Arduino Nano, based on the ATmega328P, acts as an interface between GSM module and a computer, which links with the module through two digital pins with a computer through Mini-B USB serial cable. This microcontroller has 14 digital I/O pins and 8 analog pins. The clock frequency is 16MHz [26]. It also can receive data from Matlab and do the processing to control message sending. In addition, it has some advantages, such as small size, low cost, easy to program with an open-source platform software integrated development environment (IDE).

3.2.3. GSM Module Sim800l

The GSM (module Sim-800l) is a small modem approximately 0.025 m² that can operate in a voltage range from 3.4V to 4.4V [27, 28], which is used for communication purposes, such as sending and receiving SMS messages, GPRS data and making voice calls. Therefore, it is suitable to use for sending a patient request to the care provider cellphone, containing five messages controlled by data processing in a microcontroller via the Matlab program environment. Therefore, the GSM module transmitter and receiver pins (TX and RX) are connected with two digital pins of the microcontroller. Also, the ground pins (Vcc) of the module are connected with 5 Volt chargers through the DC to DC step down buck converter LM2596, to avoid the drop voltage of the microcontroller and to feed GSM with a proper voltage at 3.7 V.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

3.2.4. DC-DC Chopper (buck)

It is a dc to dc step-down converter. The simplest way to reduce the voltage of a DC supply is to use a linear regulator (such as a 7805) yet linear regulators waste energy as they operate by dissipating excess power as heat. Buck converters, on the other hand, can be remarkably efficient (95% or higher for integrated circuits). It utilizes a MOSFET switch (IRFP250N), a diode, an inductor and a capacitor. Few resistors are also used in the circuit for the protection of the main components. When the MOSFET switch is 'ON' current rises through inductor, capacitor and load. The inductor is used to store energy. When the switch is 'OFF', the energy in the inductor circulates current through the inductor, capacitor freewheeling diode and load. The output voltage will be less than or equal to the input voltage. In this study an LM2596 dc-dc buck converter step-down power module with a high-precision potentiometer for adjusting output voltage was used that is capable of driving a load up to 3A with high efficiency.

3.3. Software

In this study, the following software and tools have been used:

1. Matlab R2018a (Image processing toolbox, Computer vision toolbox, Deep learning toolbox).
2. Microsoft standard development kit (SDK) for Kinect V2.
3. Kinect for windows Runtime V2.
4. Arduino (IDE).

4. Experimental Results & Discussion

4.1. Experimental Results

4.1.1.The first Scenario: Hand Detection Using Depth Threshold and Depth Metadata

Depth information and skeleton data were obtained through the Kinect V2. A threshold-based segmentation algorithm to z-axis was adopted to extract the hand mask. The resulting image was then smoothed by using a median filter [17]. The filtered image was combined with the cropped hand based on a joint tracking to improve the result of hand segmentation. The diagram that describes the process for the first scenario is shown in Figure 4.

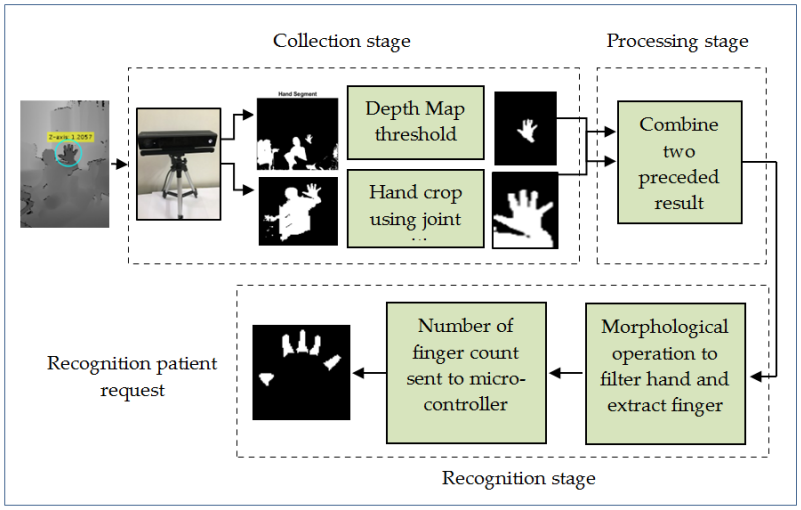

Figure 4. The proposed method based on depth threshold and depth metadata.

 doi:10.20944/preprints202007.0625.v1

- The steps illustrated in Figure 4 are summarized as follows:

- After acquiring the depth frame from the Kinect depth sensor, it can be easy to locate the center of hand palm from depth metadata using the joint position property. This point is mapped onto the depth map, and their depth values are saved for the next step.
- As every skeleton point in 3D space is associated with a position and an orientation, we can obtain the position of the central palm in real-time.
- The depth metadata returned by the depth sensor gave body tracking data so that the body index frame property enabled segmentation of the full human body into six bodies.
- After segmenting the body, a rectangular region was selected (for example, with size 200×200) around the central point of the hand/palm in the depth images. Initial segmentation was conducted based on the hand crop using the tracking point of the central palm. Because the right hand conforms more to the habit of human-computer interaction, we chose the right hand as the identification target.
- The depth threshold was provided for the depth map and the hand segment using z-axis threshold.
- The hand cropped result was combined with the depth threshold result to improve the outcome.
- The binary image was smoothed using a median filter and we set 5 as the linear aperture size.
- Using some morphological operations, such as erosion and dilation and image subtraction to extract the palm by drawing a circle covering the whole area of the palm using a tracked joint of the central palm. The fingers were then segmented, where the numbers of fingers counts appear as a white area and were then connected with a specific request.
- Finally, five fingers carried out five requests according to finger count that was sent by the microcontroller as a numeric value via the serial port to control the GSM module. The experimental results for the first scenario are shown in Figure 5 at five different gestures based on finger counting.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

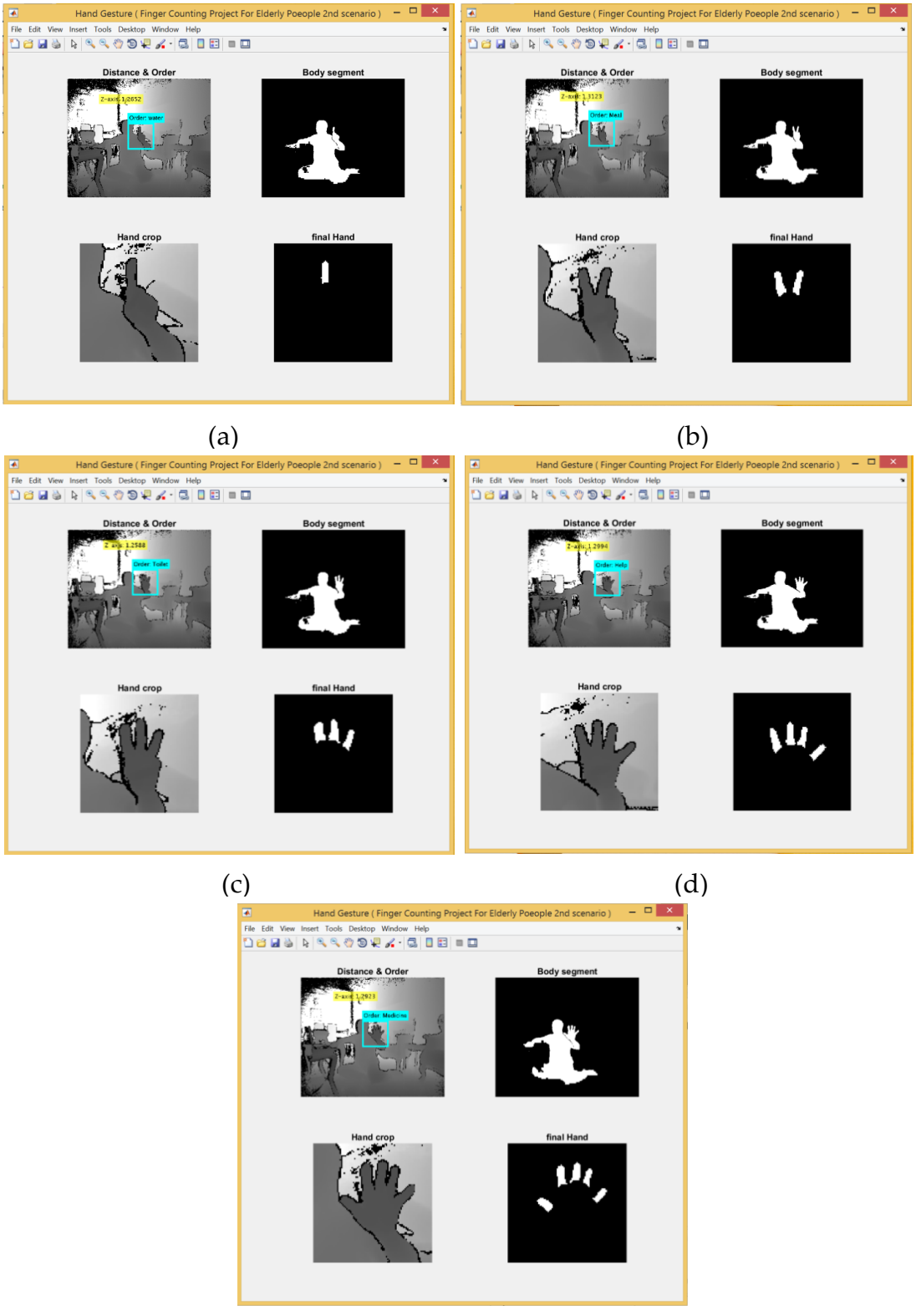

Figure 5. Finger count interpreted as patient requests (a-b-c-d-e).

Table 2 shows the experimental results for all participants with every single gesture. The results were recorded for all participants and we took the mean of these recorded results. The recognition rate for the overall gestures was 83.07% at detection distance between 1.2-1.5 meters.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

Table 2. The results analysis for the total number of tested gestures for each participant (First scenario).

Hand gesture type	Total number of sample per tested gesture	Number of Recognize gesture	Number of Un-recognize gesture	Percentage of Correct Recognition for Total number of sample gesture %	Percentage of Fault Recognition for Total number of sample gesture %
0	65	65	0	100.00	0
1	65	55	10	84.62	15.38
2	65	52	13	80.00	20.00
3	65	50	15	76.92	23.08
4	65	49	16	75.38	24.62
5	65	53	12	81.54	18.6
Total	390	324	66		

It is clear from Table 2 that the confusion matrix was adopted to provide predicted and actual results for all gestures that could help to observe the deviation and behaviour of the proposed method. Figure 6 shows the results of the confusion matrix and summaries the predicted results and actual results in the form of row and column.

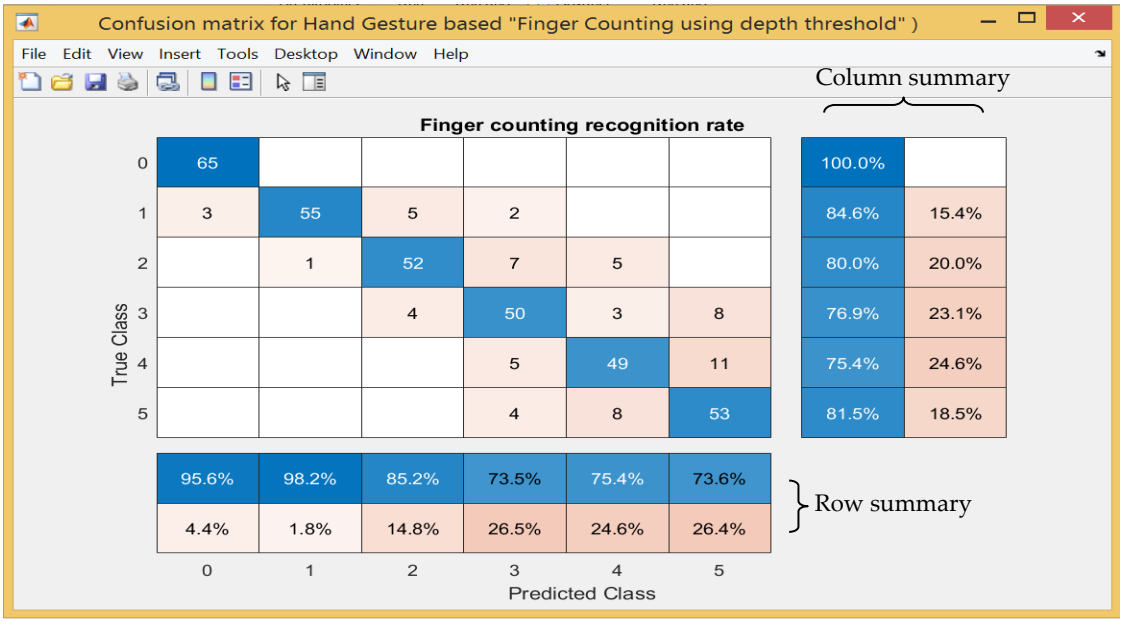

Figure 6. Confusion matrixes for the first scenario.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

4.1.2.The Second Scenario: Hand Detection and Tracking using Kinect V2 Embedded System

In this scenario, both RGB and depth sensors were used to acquire color image and body data. The output of the color sensor has a set of device-specific properties. These properties are read-only for Kinect V2, such as exposure time, frame interval, gain and gamma. The output of the depth sensor has one specific property associated with body tracking where the depth sensor collects body metadata by turning on the body tracking property, while the metadata provides the parameters of the body data as listed in Table 3.

Using get data property in the depth sensor, it can easily access to body tracking data as metadata on the depth stream. The function returns frames of size 512x424 in mono 13 formats and uint16 data type. We look at the metadata to see the parameters in the body data which bring eleven different properties, these metadata fields are related to tracking the bodies as listed in Table 3.

Table 3. The metadata fields related to tracking the bodies

	Parameters of the body data obtained by the depth sensor	**struct array**
1	IsBodyTracked	[1x6 logical]
2	BodyTrackingID	[1x6 double]
3	BodyIndexFrame	[424x512 double]
4	ColorJointIndices	[25x2x6 double]
5	DepthJointIndices	[25x2x6 double]
6	HandLeftState ✓	[1x6 double] ✓
7	HandRightState ✓	[1x6 double] ✓
8	HandLeftConfidence	[1x6 double]
9	HandRightConfidence	[1x6 double]
10	JointTrackingStates	[25x6 double]
11	JointPositions	[25x3x6 double]

The property of the left hand state and right hand state provides a 1 x 6 double array that identifies possible states for both the left and right hands of the tracked bodies. Where the values obtained by the depth sensor include (0= unknown, 1= not tracked, 2= open, 3= closed, and 4= lasso) corresponding to a specific gesture performed by the participant.

In this scenario, the metadata parameters were encoded for three different gestures performed by the right hands and two gestures performed by the left hands to perform five different requests and sent via GSM. The requests represented by the right hand gestures were open hand, closed hand and lasso gestures, which indicate 'Water', 'Meal', 'Toilet', respectively. Whereas, the requests represented by the left hand were only two gestures (open hand and closed hand) that indicate 'Help' and 'Medicine', respectively. This experiment used both hands to implement five different gestures, where every gesture indicates a specific request as a reverse of the first experiment that used only one hand to perform these five requests. Figure 7 shows five gestures provided by the left and right hands, whereas, Figure 8 shows the detection range between 0.5 ~ 4.5 meters for applying this scenario.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

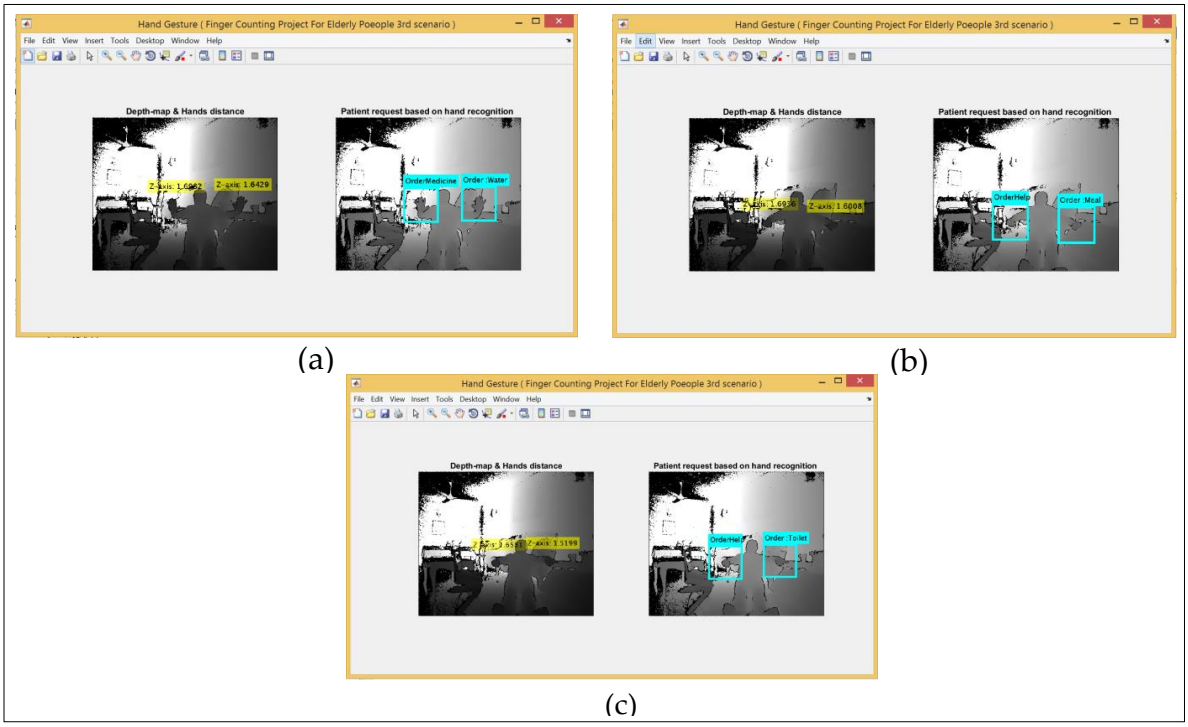

Figure 7. The results of the proposed method for the second scenario for both hands (a, b, c).

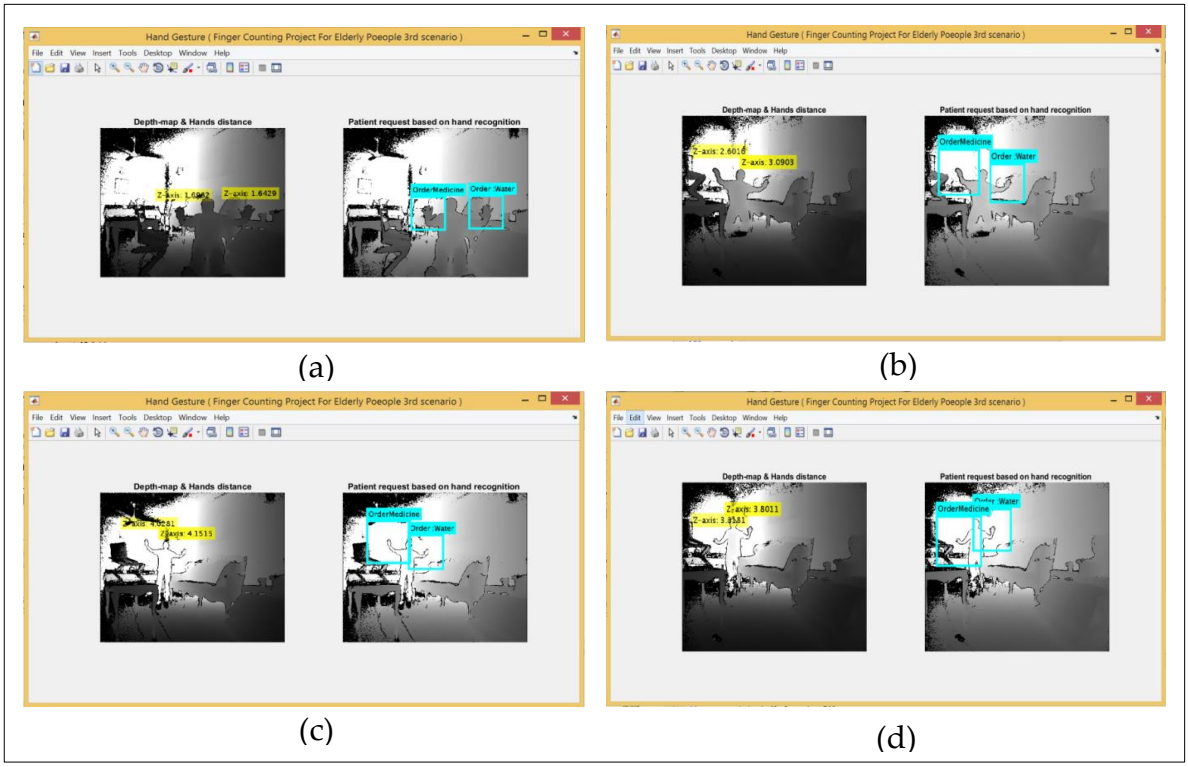

Figure 8. The test of the detection range for the second scenario (a, b, c, d).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

Table 4 shows the experimental results for all participants regarding every single gesture performed by both hands together. The recognition rate for the overall gestures in this scenario was 95.2 % at detection distance between 0.5 ~ 4.5 meters.

Table 4. The results analysis for the total number of tested gestures for each participant (second scenario)

Hand gesture type	Total number of sample per tested gesture	Number of Recognize gesture	Number of Un-recognize gesture	Percentage of Correct Recognition for Total number of sample gesture %	Percentage of Fault Recognition for Total number of sample gesture %
1	50	47	3	94.00	6.00
2	50	48	2	96.00	4.00
3	50	47	3	94.00	6.00
4	50	48	2	96.00	4.00
5	50	48	2	96.00	4.00
Total	250	238	12		

It is clear from Table 4 that the confusion matrix was adopted to provide the predicted and actual results for all gestures that could help to observe the deviation and behavior of the proposed method. Figure 9 shows the result of the confusion matrix and summarises the predicted and actual results in the form of row and column.

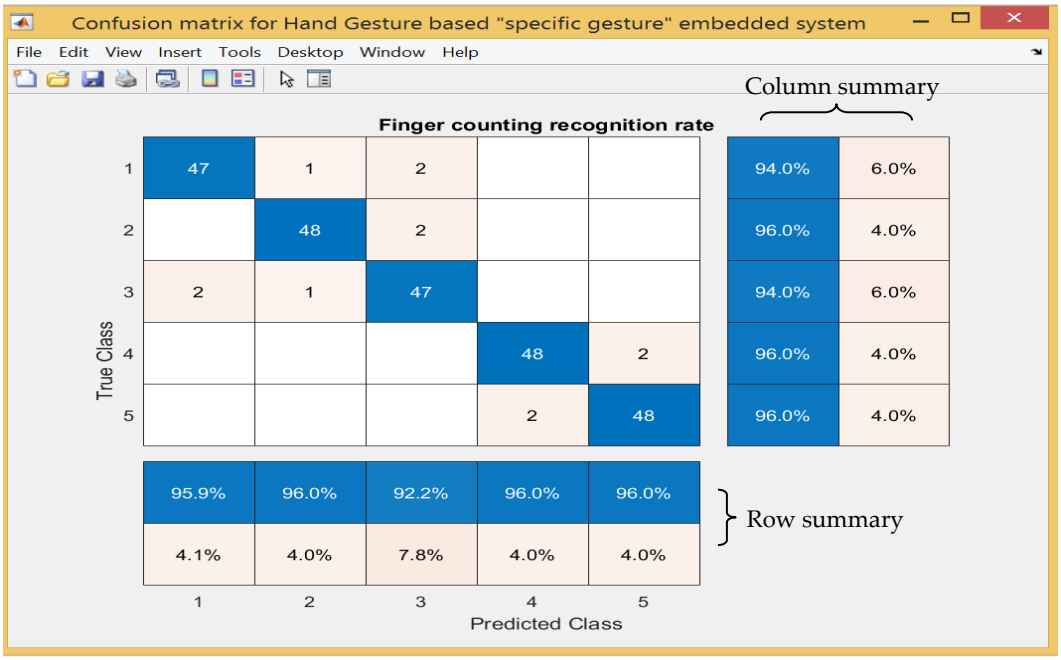

Figure 9 Confusion matrixes for the second scenario.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

4.1.3. The Third Scenario: Hand Gestures Based on SCNN and Depth Metadata

In this scenario, the experiment was conducted using a deep learning classifier based on a simple convolutional neural network (SCNN). CNN is a suitable tool for building an image recognition system.

The hand image samples were captured by an automatic program created by the author where the image data was resized and stored in one folder to separate into different categories related to five gestures manually. These categories were named image data-store. The image data-store in this folder category was labeled based on folders' names with storage of the image as an object. The images data-store can store a large amount of image data and efficiently read a batch of images while training the CNN.

The data store includes 125 images for every category of hand gestures from 1-5 and a total of 625 images for all categories. The number of classes was specified at the last fully connected layer in the output of the network. Also, the input image size was specified at the input layer. Each image must be stored as 28-by-28-by-1 pixels. Figure 10 shows five hand gestures used in this experiment, where the dataset categories were created by the authors using the Kinect depth sensor.

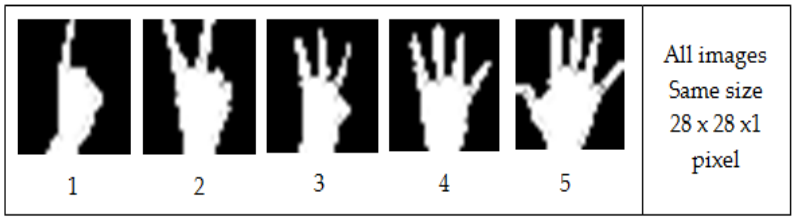

Figure 10. Five gestures created and stored for training and testing.

The image dataset was separated into training and validation data-sets, where the training-set includes 70 images and the remaining images for validation-set. Each label splits the data store into two new data stores, training hand gestures data and validation hand gestures data.

To specify the training based on a CNN structure build, this step needs to determine the training parameters, where the network trained using stochastic gradient descent with momentum (SGDM) with a learning rate initially of 0.01 and max-epoch number 4. The epoch is the full training cycle for the input training dataset.

The network trained using GPU by default. Otherwise, it uses only the CPU. Figure 11 shows the deep-learning-training-progress and plots the mini-batch-loss (cross-entropy loss), the validation loss and accuracy (percentage of images classified by the network correctly).

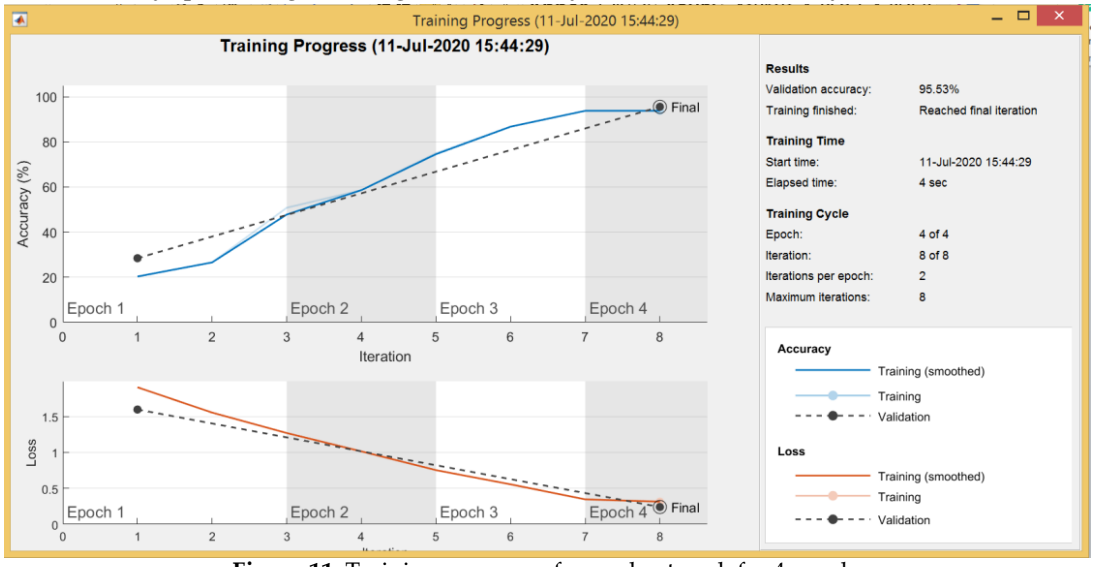

Figure 11. Training progress of neural network for 4 epochs.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

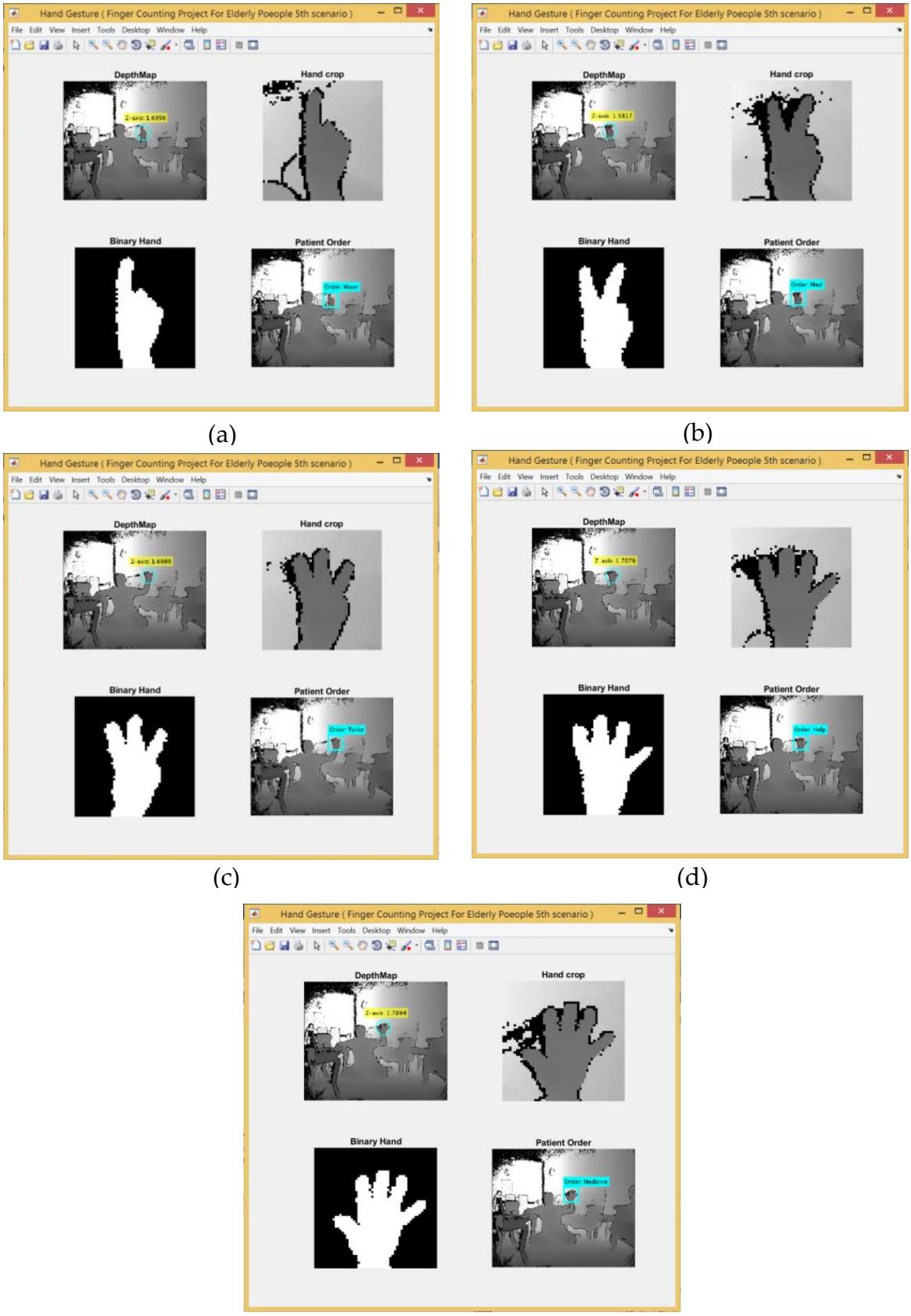

Figure 12. Fingers count interpreted at different participant requests using a deep learning method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

Table 5 shows the experimental results for all participants regarding every single gesture performed by both hands together. The recognition rate for the overall gestures in this scenario was 95.53 % at detection distance between 1.5 ~ 1.7 meters.

Table 5. The results analysis for the total number of tested gestures for each participant (third scenario).

Hand gesture type	Total number of sample per tested gesture	Number of Recognize gesture	Number of Un-recognize gesture	Percentage of Correct Recognition for Total number of sample gesture %	Percentage of Fault Recognition for Total number of sample gesture %
1	65	64	1	98.46	1.54
2	73	73	0	100.00	0.00
3	49	49	0	100.00	0.00
4	65	53	12	81.54	18.46
5	61	60	1	98.36	1.6
Total	313	299	14		

It is clear from Table 5 that the confusion matrix was adopted to give the predicted and actual results for all gestures that could help to observe the deviation and behaviour of the proposed method. Figure 13 shows the result of the confusion matrix and summarises the predicted and actual results in the form of row and column.

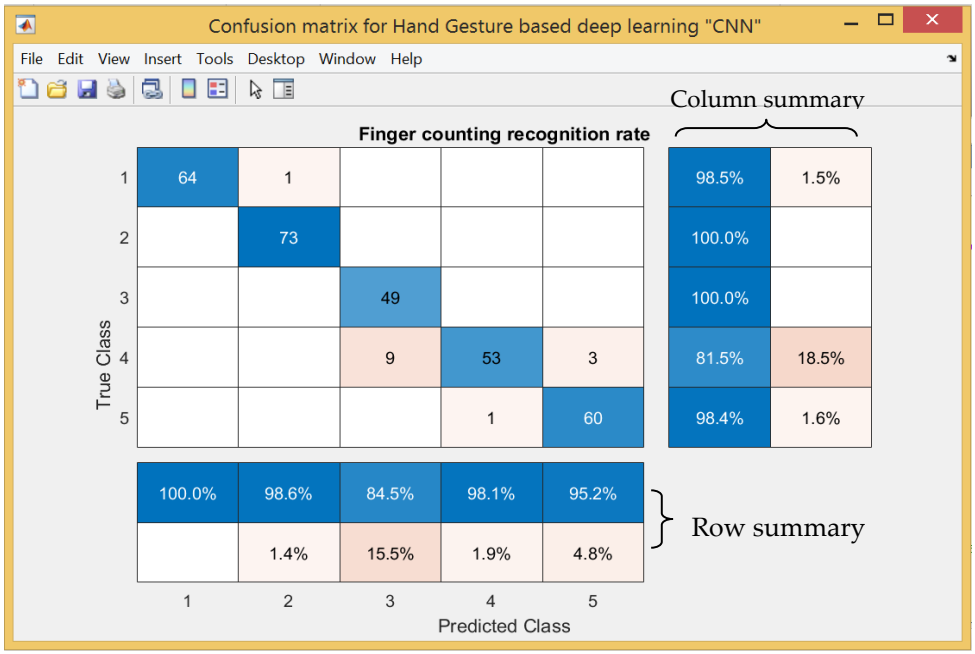

Figure 13 Confusion matrixes for the third scenario.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2020 doi:10.20944/preprints202007.0625.v1

4.2. Discussion

In this study, the three different hand gestures recognition scenarios were conducted using the Microsoft Kinect V2 sensor. These scenarios can be categorised into three main approaches: Finger counting, embedded system and deep learning. In this section, the main key points for these three categories are compared and summarized in Table 6.

Table 6. The key points of every approach for three scenarios and their performance.

Method	Type of gesture	Principle	classification	Image pixel	Recognition rate	Distance from the camera
Scenario 1	Finger count (0-5) Single hand	Depth threshold and skeleton joint tracking using metadata information	Appearance of white area	512 x 424	83.07 %	1.2 ~ 1.5 meter
Scenario 2	Specific gesture both hand	Metadata parameter	Hand left state, hand right state parameter by kinect depth	512 x 424	95.2 %	0.5 ~ 4.5 meter
Scenario 3	Finger count image pattern (1-5) Single hand	SCNN Depth metadata	CNN	Dataset 28 x 28 x1	95.53 %	1.5 ~ 1.7 meter

From Table 6, it can easily be observed which is the best approach in regard to recognition rate and distance from the camera and easy to perform hand gestures. But, taking consideration of some challenges facing every category can be summarise as follows:

- The first scenario offers acceptable results, but it has limitations in regard to classification, where the number of fingers based on a white area affected by any speckle changes the results.
- The second scenario provides a high recognition rate because it offers better flexibility in regard to distance during capturing the gestures in real-time if compared with other categories. But, the only type of gestures that can be read are three active gestures for every hand (from the default of the embedded system provided by the Kinect) and five hand gestures must perform by both hands with three for every hand, respectively.
- The third scenario provides a good recognition rate but suffered due to the distance limitation related to the range sensor used when the dataset was created.

5. Conclusion

In conclusion, this study explored the feasibility of extracting hand gestures in real-time from the Microsoft Kinect v2 sensor under three scenarios: finger counting, the embedded system provided by the Kinect itself, and a deep learning technique. The proposed method used the same practical circuit for each scenario, including depth threshold, dataset matching, and specific gesture for an embedded system, which reports that the correct SMS message sent to the care provider correlated directly with the results and accuracy of the recognition system. The experimental evaluation of the proposed method has been conducted in real-time for all participants under three different scenarios. The experimental results were recorded and analyzed using a confusion matrix which gave acceptable outcomes making this study a promising method for future home care applications.

Author Contributions

Conceptualization, Ali Al-Naji and Munir Oudah; Methodology, Munir Oudah and Al-Naji.; Investigation, Munir Oudah and Ali Al-Naji; Data curation, Munir Oudah.; Project administration, Ali Al-Naji and Javaan Chahl.; Resources, Munir Oudah.; Software, Munir Oudah and Ali Al-Naji; Supervision, Ali Al-Naji and Javaan Chahl.; Validation, Munir Oudah.; Funding acquisition, Ali Al-Naji and Javaan Chahl; Writing—original draft preparation, Munir Oudah; Writing—review and editing, Ali Al-Naji and Javaan Chahl. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors of this manuscript have no conflict of interest relevant to this work.

References

[1] T. Truelsen, R. Bonita, and K. Jamrozik, "Surveillance of stroke: a global perspective.," *Int. J. Epidemiol.*, vol. 30, no. suppl_1, p. S11, 2001.

[2] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in human-computer-interaction," in *2011 8th International Conference on Information, Communications & Signal Processing*, 2011, pp. 1–5.

[3] X. Ma and J. Peng, "Kinect sensor-based long-distance hand gesture recognition and fingertip detection with depth information," *J. Sensors*, vol. 2018, 2018.

[4] B. Ma, W. Xu, and S. Wang, "A robot control system based on gesture recognition using Kinect," *TELKOMNIKA Indones. J. Electr. Eng.*, vol. 11, no. 5, pp. 2605–2611, 2013.

[5] U. Lee and J. Tanaka, "Finger identification and hand gesture recognition techniques for natural user interface," in *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, 2013, pp. 274–279.

[6] D. H. Pal and S. M. Kakade, "Dynamic hand gesture recognition using kinect sensor," in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, 2016, pp. 448–453.

[7] Y. Li, "Hand gesture recognition using Kinect," in *2012 IEEE International Conference on*

Computer Science and Automation Engineering, 2012, pp. 196–199.

[8] C. Xi, J. Chen, C. Zhao, Q. Pei, and L. Liu, "Real-time Hand Tracking Using Kinect," in *Proceedings of the 2nd International Conference on Digital Signal Processing*, 2018, pp. 37–42.

[9] M.-S. Kim and C. H. Lee, "Hand Gesture Recognition for Kinect v2 Sensor in the Near Distance Where Depth Data Are Not Provided," *Int. J. Softw. Eng. Its Appl.*, vol. 10, no. 12, pp. 407–418, 2016.

[10] M. Z. A. Bakar, R. Samad, D. Pebrianti, and N. L. Y. Aan, "Real-time rotation invariant hand tracking using 3D data," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, 2014, pp. 490–495.

[11] J. Bamwenda and M. S. Özerdem, "Recognition of static hand gesture with using ANN and SVM," 2019.

[12] S. Desai and A. Desai, "Human Computer Interaction through hand gestures for home automation using Microsoft Kinect," in *Proceedings of International Conference on Communication and Networks*, 2017, pp. 19–29.

[13] S. Desai, "Segmentation and Recognition of Fingers Using Microsoft Kinect," in *Proceedings of International Conference on Communication and Networks*, 2017, pp. 45–53.

[14] M. Karbasi, Z. Bhatti, P. Nooralishahi, A. Shah, and S. M. R. Mazloomnezhad, "Real-time hands detection in depth image by using distance with Kinect camera," *Int. J. Internet Things*, vol. 4, pp. 1–6, 2015.

[15] M. Z. A. Bakar, R. Samad, D. Pebrianti, M. Mustafa, and N. R. H. Abdullah, "Finger application using K-Curvature method and Kinect sensor in real-time," in *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, 2015, pp. 218–222.

[16] Y. Wen, C. Hu, G. Yu, and C. Wang, "A robust method of detecting hand gestures using depth sensors," in *2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings*, 2012, pp. 72–77.

[17] J. Li, J. Wang, and Z. Ju, "A novel hand gesture recognition based on high-level features," *Int. J. Humanoid Robot.*, vol. 15, no. 02, p. 1750022, 2018.

[18] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," in *2014 IEEE International conference on image processing (ICIP)*, 2014, pp. 1565–1569.

[19] M. Samir, E. Golkar, and A. A. A. Rahni, "Comparison between the Kinect™ V1 and Kinect™ V2 for respiratory motion tracking," in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2015, pp. 150–155.

[20] C. Kim, S. Yun, S.-W. Jung, and C. S. Won, "Color and depth image correspondence for Kinect v2," in *Advanced Multimedia and Ubiquitous Engineering*, Springer, 2015, pp. 111–116.

[21] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. El Saddik, "Evaluating and improving the depth accuracy of Kinect for Windows v2," *IEEE Sens. J.*, vol. 15, no. 8, pp. 4275–4285, 2015.

[22] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus Time-of-Flight Kinect," *Comput. Vis. image Underst.*, vol. 139, pp. 1–20, 2015.

[23] C. D. Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-flight cameras and microsoft kinect (TM)*. Springer Publishing Company, Incorporated, 2012.

[24] A. Al-Naji, K. Gibson, S.-H. Lee, and J. Chahl, "Real time apnoea monitoring of children using the Microsoft Kinect sensor: a pilot study," *Sensors*, vol. 17, no. 2, p. 286, 2017.

[25] A. Al-Naji and J. Chahl, "Detection of cardiopulmonary activity and related abnormal events using microsoft kinect sensor," *Sensors*, vol. 18, no. 3, p. 920, 2018.

[26] R. H. Kumar, A. U. Roopa, and D. P. Sathiya, "Arduino ATMEGA-328 microcontroller," *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng.*, vol. 3, no. 4, pp. 27–29, 2015.

[27] S. Mluyati and S. Sadi, "Internet Of Things (IoT) Pada Prototipe Pendeteksi Kebocoran Gas Berbasis MQ-2 Dan SIM800L," *J. Tek.*, vol. 7, no. 2, 2019.

[28] M. Oudah and A. Alnaji and J. Chahl, " Hand Gestures for Elderly Care Using a Microsoft Kinect," *Nano biomedicine and Engineering,* http://nanobe.org/Data/View/652, 2020.

[29] M. Oudah and A. Alnaji and J. Chahl, " Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *Journal of Imaging*, vol, 6, no, 8, pp, 73, 2020.