Indonesian Journal of Electrical Engineering and Computer Science
Vol. 28, No. 1, October 2022, pp. 346~357
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v28.i1.pp346-357

Online hand position detection and classification system using multiple classification algorithms

Ahmed Etihad Jaleel, Hesham Adnan Alabbasi
Computers Science Department, College of Education, Mustansiriyah University, Baghdad, Iraq

Article Info

Article history:

Received Mar 19, 2022
Revised Jun 11, 2022
Accepted Jul 19, 2022

Keywords:

Hand position
Kinect sensor
K-nearest neighbors
Multilayer perceptro
Random forest
Skeleton
Support vector machines

ABSTRACT

Hand position recognition is very significant for human-computer interaction. Different kinds of devices and technologies can be used for data acquisition; each has its specification and accuracy, one of these devices is Kinect V2 sensor. A three-dimensional location of the skeleton joints is taken from the Kinect device to create three types of data, the first is joint position raw data, the second is angles between joints, the third is combined of both types. These three types of data are used to train four classifiers, which are support vector machines, random forest, k nearest neighbors, and multilayer perceptron. The experiments are done on the datasets of 30,480 frames from 127 volunteers with saved trained models are used to predict and classify the eight positions of hand in a real-time system. The results show that our proposed approach performs well with highly efficient and accuracy reaching up to 99.07% in some cases and an average time spent on checking frame by frame sequentially very short period, and some cases, it reaches 0.59*10-3 seconds. This system can used in many applications such as controlling robots or devices, comparing physical exercises, or even monitoring elderly and patients, and more.

Corresponding Author:

Ahmed Etihad Jaleel
Computers Science Department, College of Education, Mustansiriyah University
Falastin (Palestine) St. Baghdad-Resafa, Baghdad, 00964, Iraq
Email: ahmed.etihad@uomustansiriyah.edu.iq

1. INTRODUCTION

The Microsoft Kinect sensor V2 device is used in many scientific fields because of its specification like being cheap, very accurate [1], [2], easy to set up technology, and fast. To extract position skeleton data, Kinect provides to us the locations of 25 virtual anatomical joint trajectories which can be extracted from depth map with a per-pixel semantic segmentation algorithm [3], with the ability to track 6 people, the Kinect sensor provides a powerful software development kit (SDK). Its technology allowed many applications to be developed beyond the original scope of gaming, covering several categories like detection of the human body or a part of it, such as the face, hands, or legs, and distinguishing movements and gestures in the field of sign language, gait recognition as in research [4]-[9]. Also, to monitor patients and the elderly for healthcare or from falling and alert those concerned where one or several devices are used [10]-[12]. To monitor exercises with the design of an avatar to teach and display movements and compare the correctness of their implementation [10], [13]. Controlling the robot as a whole or as an arm through gestures or imitation of movements [6], [14], it has the possibility of implementation in real-time application [15], can be used as a scanner for 3D printing [16], and because artificial intelligence has a large income in controlling these areas. We apply multiple classification algorithms on three types of data extracted from the second version of the

Kinect to study, compare the effectiveness and accuracy of each classification method and apply used the best classifiers in an online test model.

Kinect V2 Sensor is a device developed by Microsoft, where it is initially launched with the Xbox game console, and then a new version of it was released for Windows, Figure 1. The powerful Kinect features like two cameras: one that is color RGB and the other that is depth (with varying resolutions). The color camera has a resolution of 1920×1080 pixels, while the depth camera has a resolution of 512×424 pixels. At any given moment, Kinect can monitor up to six skeletons, each with 25 joints as shown in Figure 2(a). The joints are labeled with numbers ranging from 0 to 24 which are color (x, y), depth (x, y), camera coordinates (x, y, z), and orientation (x, y, z), these are the 11 attributes of each joint (x, y, z, w) as shown in Figure 2(b). Figure 3. represent output data of Kinect v2 and summarize point cloud computation.

The Kinect's camera coordinates employ the infrared sensor to locate 3D locations in space where the joints are. These are the coordinates to utilize in 3D projects for joint placement. It's worth remembering that the Kinect skeleton returns "joints" rather than "bones" [17], what matters to us is the raw data represented by the three-dimensional locations of the skeletal joints, as we use it in the first type of data and we also use it to calculate the angles, which is the second type of data.

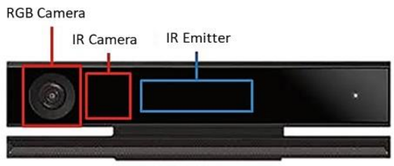
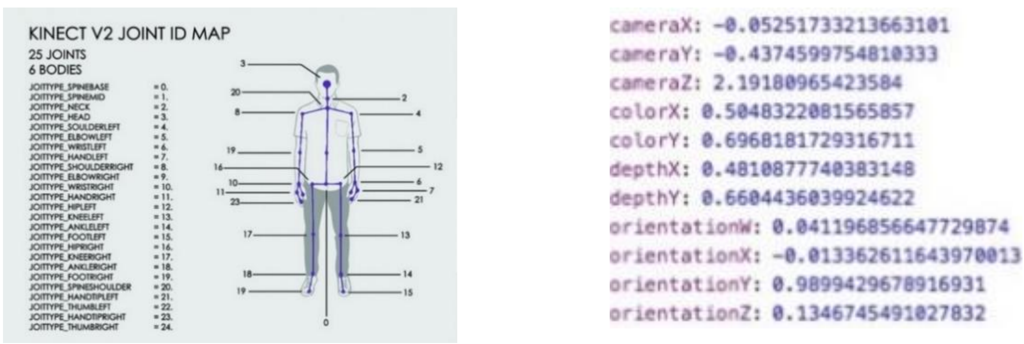


Figure 1. The face of the Kinect V2 sensor shows the placements of the cameras and emitters [18]



(a) (b)

Figure 2. Information of joints data the Kinect V2 sensor's (a) joint map of a human skeleton, and (b) an example of one Kinect joint's 11 features [19]

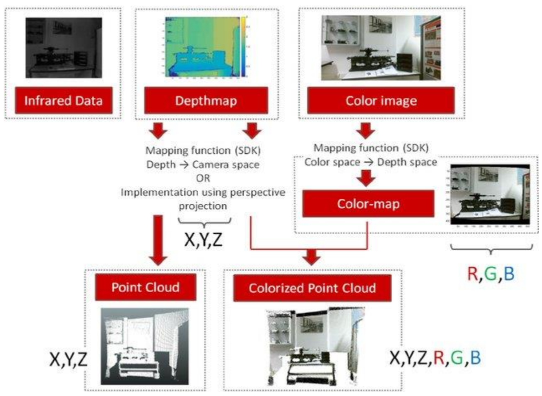


Figure 3. Schematic representation of the output data of Kinect v2 and summary of point cloud computation [20]

Different classifiers are used in this research to classify the types of hand positions. In this research, we decided to detect and classify eight positions, which are: “hands up," “right hand up," “left hand up," “hands-on head," “arms open," “stand up straight," “hands-on waist," and “hands forward". By applying the following classifiers: (support vector machines (SVMs) [21],[22], k-nearest neighbors (kNN) [23], random forests (RF) [24], multilayer perceptron (MLP) [25]). The goals of this research are:

- Finding the best accurate classifier and using it in the system to distinguish movements that can be applied in simulators and robotics control.
- Discover what kind of data derived from the skeleton provided by the Kinect device that can be used with classifiers and gives the best results in terms of speed and accuracy.
- More efficient method of storing and retrieve trained model to reduce the time of training system.
- Designed and implemented a fast system to use classifier on real-time recognition.

2. RELATED WORKS

Many researches have there attempts and approaches in this field, we present some of the recent researches related to the used classifiers in this paper. Adama, *et al* [26], offered an activity recognition learning system for use in assistive robots that uses an SVM classifier to learn everyday activity from 3D skeletal data. Byun and Lee [27], presented a survey for the use of SVMs in various applications. It was successful in applying it to several problems, including voice discrimination with knowledge of the speaker's identity, distinguishing faces with knowledge of his identity, knowing handwriting, and distinguishing numbers, and most results showed that RBF kernels were usually better than linear or polynomial kernels.

Manzi *et al.* [28], described an activity detection system that uses machine learning techniques (a multiclass SVM trained using sequential minimal optimization (SMO)) to identify actions based on skeletal data taken from a depth camera. Li *et al*. [29], developed a system for action identification based on the skeleton by mining important skeleton postures using latent SVM. The research revealed that distinguishing human actions requires only a few frames with crucial skeletal postures.

Arai and Andrie [30], created a 3D skeleton model, the Kinect sensor and Ipisoft motion capture program are used. Ipisoft is a specifically designed tool that allows users to design skeletons for their computer-generated characters. The knee angle feature will be extracted from the skeleton and used to quantify the gait disable quality. Anjum *et al* [31], created feature vectors based on the 3D location of these joints during the course of the activity, which are then utilized for SVM-based training and testing of activity identification for genuine human-robot interaction.

Piyathilaka and Kodagoda [32], offered the notion of a spatial affordance map, which uses geometric aspects of the environment to learn about human context. Rather than watching real individuals in the environment, the suggested affordance mapping approach models interaction between the environment and humans using virtual humans. The spatial affordance map learning issue is stated as a multi-label classification problem that may be learned using SVM-based learners. Experiments on an actual 3D scene dataset yielded good results, demonstrating the use of the affordance-map for mapping human context.

Elforaici *et al.* [33], created an automatic posture recognition system using an RGB-D camera (Kinect). They present two supervised algorithms for learning and detecting human poses using an RGB-D camera's multiple types of visual input. One method takes advantage of a three-dimensional configuration of body joints. The posture recognition is subsequently performed using the SVM classification of 3D skeleton-based properties.

Han *et al*. [34], to reduce the potential injury caused by falls, this study proposes a two-stage fall detection system based on human postural features. They produced additional crucial characteristics for preprocessing in this study: deflection angles and spine ratio, to describe changes in human posture based on the human skeleton, and we classified using both SVM and kNN. Ubalde *et al*. [35], represented skeletal sequences as a bag of time-stamped descriptors, and they provide a new framework for action categorization based on the kNN approach. Ramirez *et al*. [36], this paper proposes a fall detection system based on camera vision that extracts features using a KNN classifier.

Seungryul *et al.* [37], researched the challenge of activity recognition in a 24-hour monitoring scenario of patient actions in a hospital, the objective was to identify both static and dynamic actions successfully. They suggest using a kinematic-layout-aware random forest to encode scene layout and skeleton information as privileged information, collecting more geometry and kinematic-layout information, and improving action classification discriminative power. Laraba *et al.* [38], introduced a novel motion sequence representation that projects movement sequences into the RGB domain. Action classification becomes an image classification issue since the 3D coordinates of joints are transferred to values of red, green, and blue. Methods for classifying images at a basic level, such as SVM, kNN, RF, as well as CNN, were used to evaluate this representation.

Canavan *et al*. [39], suggested combining a random regression forest with a unique set of features descriptors built from bone data received from the leap motion controller to recognize automated hand gestures. Boissiere and Noumeir [40], proposed an end-to-end trainable network for human action identification utilizing skeleton and infrared data, with 2D CNN as a pose module extracting features from skeleton data and 3D CNN as an infrared module extracting visual characteristics from clips. Using a multi-layer perceptron, both feature vectors are then merged and explored together. Zhao *et al.* [41] describe a technique that uses various classifiers to identify people. By using static characteristics taken from Kinect skeletal data, and used classfiers (KNN, decision tree, Gaussian Naive Bayesian, MultiLayer perceptron, and SVM) to predect the conclution.

3. PROPOSED METHOD

Figure 4. show the diagram of proposed approach. That Use the Kinect v2 sensor and the above classifiers to represent following steps:

– Build dataset (collect datasets using the Kinect skeleton).
– Calculate angles.
– Save data in three separate CSV files containing different types of data.
– Train classifiers.
– Store trained models by use the pickle method.
– Real-time recognition using saved models.

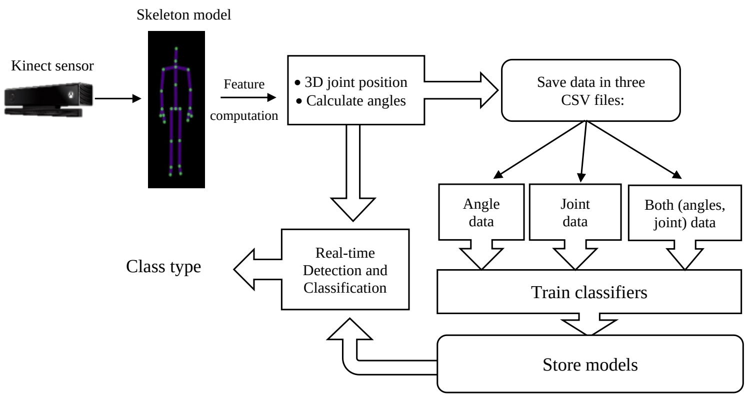


Figure 4. Diagram of the proposed approach

3.1. Build dataset

The database we collected for eight fixed positions came from 127 volunteers (men and women), whose ages ranged from 20 to 41, with different heights (1.45–1.91 m) and different body sizes. Each person from the volunteers imitates or performs the eight positions or poses: “hands up”, “right hand up”, “left hand up”, “hands-on head”, “arms open”, “stand up straight”, “hands-on waist" and “hands forward” as shown in Figure 5, interspersed with a simple movement that falls under the same position. For each person, we record 240 frames (each frame contains 15 joint camera coordinates in X, Y, and Z, and 6 angles). The record total frames are 30,480 frames. 72% are used for training data and 28% are used for testing data.

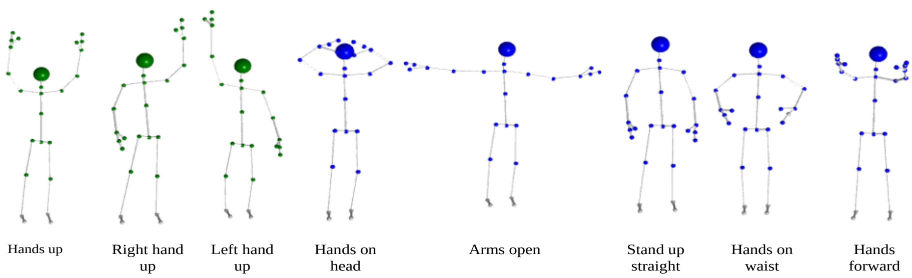


Figure 5. Eight hand positions

3.2. Calculate angles

If we have space coordinate positions of their joint points, we can calculate an angle by using three 3D points to make space vectors between them. Like vector (ER-SR) (SR-SS), where ER represents the joint point of the elbow right, SR represents the joint point of the shoulder right, and SS represents the joint point of the shoulder spine. As shown in Figure 6.

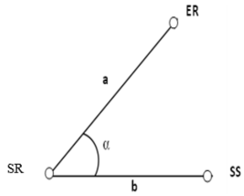


Figure 6. Diagram of joint angle

By assuming the coordinates of the elbow-right joint point are (x_1, y_1, z_1), the coordinates of the shoulder-right joint point are (x_2, y_2, z_2), and the joint point coordinates of the spine-shoulder are (x_3, y_3, z_3), then the vector a=(x_2–x_1, y_2–y_1, z_2–z_1), vector b=(x_3–x_2, y_3–y_2, z_3–z_2), Assume (a, b) included angle is α, then:

$$\cos\alpha = \frac{a.b}{|a||b|} \tag{1}$$

$$a.b = (x_2 - x_1)(x_3 - x_2) + (y_2 - y_1)(y_3 - y_2) + (z_2 - z_1)(z_3 - z_2) \tag{2}$$

$$|a| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{3}$$

$$|b| = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2} \tag{4}$$

to get the angle between the vectors created by the three essential bone joint sites joined in pairs, substitute the following equations into (1)-(4). This strategy was used by Liu *et al.* [12].

3.3. Save data

This research is based on distinguishing the upper half of the body, specifically the location of the hands, we focused on the 15 upper joints and the angles that determine the movement of the hands. For this, the lower half does not affect the determination of the movements adopted in the search, to reduce processing operations we saved data in three separate files. First file used to save joints coordinate (X, Y, Z) of upper joints (head, nick, spin shoulder, spin mid, spin base, shoulder (left, right), elbow (left, right), wrist (left, right), hand (left, right), hip (left, right)), second file to save calculate six angles shown in Figure 7 which is shoulder angle calculated using points (spine shoulder–shoulder-elbow), elbow angle calculated using points (shoulder-elbow-wrist), wrist angle calculated using points (elbow–wrist -hand) for right and left side, The third file is used to save data by combining the first and second files, meaning we use both joints and angles to train the algorithm.

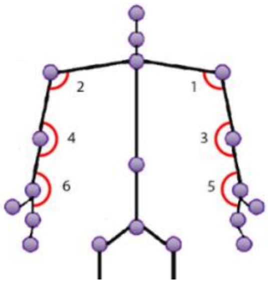


Figure 7. Positions of the six calculated angles

3.4. Train the classifiers

Three types of data to are used to train the classifiers: the first comes from the Kinect device represented by the skeleton joint coordinate position; the second is the calculation of six angles shown in Figure 7 which are calculated by using the aforementioned method and the third type of data used for training is by using the joints and angles together. These datasets are used to train a set of classifiers (SVM, random forest, k-nearest neighbors, multilayer perceptron), as mentioned above 72% from the dataset are used for training the classifiers.

3.5. Store models

It is known that training any algorithm takes a longer time than the rest of the steps. To shorten the time and not have to repeat the training of the classifier at each run of the real-time system, we used a method to save the module after it has been trained and load them when needed. Using Python's built-in persistence model, namely pickle, and use the models in real-time classifiers as shown in Figure 4.

3.6. Real-time detection and classification

After training the classifier and saving it as a pickle, the stage of using the classifier to distinguish patterns begins with running a special program written in visual basic by C++ language to choose the type of classifier and the type of data Figure 8 that used in real-time detection system. After that, loading the saved model based on the choice and starting the Kinect device to track the person and send his data to a Python script that extracts the data from each frame individually and stores it in the form of a list.

According to the type of data to be classified, if it is of the first type the data of the skeleton joints shall be placed in the list. And if it is of the second type the required angles shall be placed after calculating them, and if it is the third type each of the previous two types is placed and sent. Then the classifier makes the prediction and displays it on the screen as shown in Figure 9.

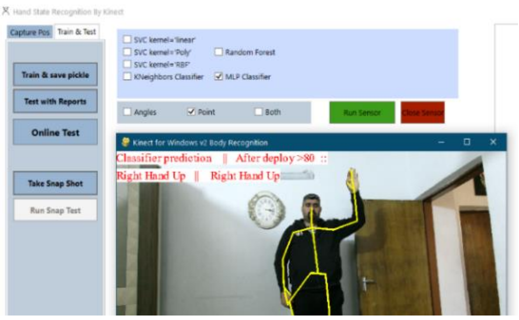


Figure 8. Online hand position detection and classification system; the main window

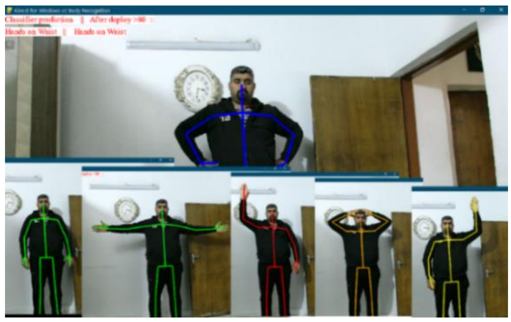


Figure 9. An example of real-time recognition

4. EXPERIMENTAL RESULTS

Applying the classifiers using our written code with Python version 3.9 and the scikit-learn version 1.0.1 libraries [42]. These tests were done on a computer with following specifications: Software (Microsoft window 10 Pro 64-bit version 21H2). Hardware (processor: Intel Core i7-4510U 2000 GHz, memory: 16 GB, harddisk: 1 TB SSD). From the implementations of the classifiers, the following experimental results are examined to determine which one is the best classifier based on the accuracy and the kind of the used data. As we can see in Tables 1-2, the classifiers achieve the best performance on point data, except for random forests, which have the best accuracy on the third type of data. The most important thing is that the accuracy of classifiers, in some cases, exceeded 93 percent and reached 99 percent in MLP and SVM with the poly kernel.

Table 1. The classifier test result of SVM types on three types of data

Data Type	Position Name	SVM with Linear kernel			SVM with Polynomial kernel			SVM with RBF kernel		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Angles Dataset	hands up	0.31	0.28	0.3	0.41	0.42	0.41	0.52	0.4	0.45
	right hand up	0.62	0.61	0.61	0.55	0.53	0.54	0.64	0.57	0.61
	left hand up	0.48	0.39	0.43	0.45	0.4	0.42	0.52	0.39	0.44
	hands on head	0.86	0.93	0.89	0.9	0.89	0.9	0.89	0.93	0.91
	arms open	1	0.93	0.96	0.97	0.97	0.97	0.99	0.94	0.96
	stand up straight	0.44	0.79	0.57	0.53	0.74	0.62	0.5	0.76	0.6
	hands on waist	0.8	0.85	0.83	0.91	0.87	0.89	0.91	0.86	0.88
	hands forward	0.54	0.24	0.34	0.57	0.45	0.5	0.57	0.63	0.6
	Accuracy		**62.72%**			**65.76%**			**68.54%**	
Points Dataset	hands up	0.98	0.94	0.96	0.98	0.99	0.99	0.96	0.98	0.97
	right hand up	0.93	1	0.96	0.99	1	1	1	1	1
	left hand up	0.99	0.99	0.99	1	1	1	1	0.99	1
	hands on head	1	0.95	0.98	0.99	0.99	0.99	0.98	0.97	0.97
	arms open	0.98	0.99	0.98	1	0.99	1	0.97	0.99	0.98
	stand up straight	0.94	1	0.97	0.97	0.99	0.98	0.97	0.99	0.98
	hands on waist	1	0.97	0.98	0.99	0.97	0.98	0.99	0.96	0.98
	hands forward	1	0.96	0.98	1	0.99	0.99	1	0.97	0.98
	Accuracy		**97.48%**			**99.07%**			**98.26%**	
Both Dataset	hands up	0.96	0.99	0.97	0.43	0.34	0.37	0.37	0.27	0.31
	right hand up	0.99	0.99	0.99	0.67	0.62	0.64	0.64	0.57	0.6
	left hand up	0.99	0.98	0.99	0.57	0.46	0.51	0.56	0.43	0.48
	hands on head	0.99	0.96	0.98	0.84	0.95	0.89	0.86	0.95	0.9
	arms open	0.95	0.99	0.97	1	0.93	0.96	1	0.93	0.96
	stand up straight	0.85	1	0.92	0.46	0.83	0.59	0.42	0.85	0.57
	hands on waist	1	0.86	0.92	0.79	0.89	0.84	0.8	0.89	0.84
	hands forward	1	0.92	0.96	0.6	0.29	0.39	0.57	0.26	0.36
	Accuracy		**96.16%**			**66.41%**			**64.34%**	

Table 2. The classifier test result of k-NN, RF, and MLP on three types of data

Data Type	Position Name	K Nearest Neighbors			Random Forests			Multilayer Perceptron		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Angles Dataset	hands up	0.39	0.37	0.38	0.45	0.35	0.4	0.28	0.32	0.3
	right hand up	0.55	0.58	0.57	0.59	0.63	0.61	0.63	0.53	0.58
	left hand up	0.5	0.45	0.47	0.52	0.47	0.5	0.45	0.39	0.42
	hands on head	0.89	0.91	0.9	0.92	0.93	0.92	0.75	0.9	0.82
	arms open	0.98	0.93	0.95	1	0.94	0.97	0.9	0.91	0.91
	stand up straight	0.49	0.72	0.58	0.61	0.65	0.63	0.51	0.72	0.6
	hands on waist	0.86	0.85	0.85	0.9	0.85	0.87	0.81	0.66	0.73
	hands forward	0.63	0.42	0.5	0.56	0.72	0.63	0.54	0.4	0.46
	Accuracy		**65.33%**			**69.27%**			**60.35%**	
Points Dataset	hands up	0.81	0.72	0.76	0.85	0.88	0.86	0.98	1	0.99
	right hand up	0.99	0.96	0.97	0.99	1	0.99	1	1	1
	left hand up	1	0.96	0.98	0.98	1	0.99	1	0.99	1
	hands on head	0.69	0.81	0.75	0.95	0.78	0.86	0.99	0.99	0.99
	arms open	1	0.94	0.97	0.75	0.99	0.85	1	0.99	1
	stand up straight	0.74	0.52	0.61	0.9	0.85	0.87	0.95	1	0.97
	hands on waist	0.58	0.83	0.68	0.83	0.89	0.86	0.99	0.94	0.97
	hands forward	0.96	0.87	0.91	0.96	0.75	0.84	0.99	0.99	0.99
	Accuracy		**82.67%**			**89.19%**			**98.79%**	
Both Dataset	hands up	0.55	0.5	0.52	0.92	0.87	0.9	0.93	0.82	0.87
	right hand up	0.67	0.7	0.68	0.99	1	0.99	0.99	0.98	0.98
	left hand up	0.63	0.57	0.6	0.98	1	0.99	0.94	0.98	0.96
	hands on head	0.89	0.91	0.9	0.89	0.94	0.91	0.83	0.93	0.88
	arms open	0.96	0.93	0.94	0.89	0.99	0.94	1	0.93	0.96
	stand up straight	0.6	0.8	0.69	0.93	0.99	0.96	0.96	0.99	0.97
	hands on waist	0.86	0.87	0.87	0.98	0.95	0.97	0.9	0.96	0.93
	hands forward	0.72	0.56	0.63	0.99	0.82	0.9	0.96	0.91	0.93
	Accuracy		**73.07%**			**94.29%**			**93.63%**	

We also noticed that the classifiers do not work correctly when using angles data in some actions, specifically in the movements of hands forward, some errors occur, as shown in the confusion matrix in Tables 3-5. We also noticed when using angles data, the accuracy is lower than the rest. The reason may be the fact that the ranges of these angles are not large enough. Furthermore, the values are similar in most of the movements or contain more noise.

Table 3. Confusion matrix of SVM with Linear kernel and SVM with Poly kernel classifiers on three types of data

Data type	True position label	SVM with Linear kernel								SVM with Polynomial kernel							
Angles Dataset	hands up	298	144	107	91	0	281	102	27	438	133	92	55	1	232	31	68
	right hand up	59	636	103	30	0	150	13	59	117	555	139	0	0	138	0	101
	left hand up	177	98	409	6	0	290	7	63	159	168	419	5	0	225	7	67
	hands on head	0	0	6	977	0	7	21	39	43	5	20	935	1	0	18	28
	arms open	27	2	1	0	977	0	43	0	14	1	3	0	1015	0	15	2
	stand up straight	89	18	85	0	0	825	0	33	98	32	92	2	0	776	2	48
	hands on waist	113	22	2	8	0	14	890	1	62	16	0	6	0	10	917	39
	hands forward	199	105	137	30	1	291	30	257	135	108	167	32	29	90	20	469
Points Datas-et	hands up	983	31	12	0	24	0	0	0	1036	0	3	11	0	0	0	0
	right hand up	0	1050	0	0	0	0	0	0	0	1050	0	0	0	0	0	0
	left hand up	0	0	1040	0	0	10	0	0	0	0	1050	0	0	0	0	0
	hands on head	8	40	0	1002	0	0	0	0	7	0	0	1043	0	0	0	0
	arms open	8	0	0	0	1042	0	0	0	9	0	0	0	1041	0	0	0
	stand up straight	0	0	0	0	0	1047	3	0	0	0	0	0	0	1044	6	0
	hands on waist	0	0	0	0	0	33	1017	0	0	0	0	0	0	31	1019	0
	hands forward	0	13	0	0	0	29	0	1008	0	9	0	0	0	2	0	1039
Both Datas-et	hands up	1036	0	14	0	0	0	0	0	352	120	102	103	0	262	87	24
	right hand up	0	1036	0	0	0	14	0	0	67	646	77	30	0	166	19	45
	left hand up	0	0	1034	2	0	14	0	0	42	75	484	11	0	319	42	77
	hands on head	38	0	0	1012	0	0	0	0	1	0	0	1001	0	4	17	27
	arms open	6	2	1	0	1041	0	0	0	50	0	0	0	976	0	24	0
	stand up straight	0	0	0	0	0	1048	2	0	51	16	73	0	0	875	3	32
	hands on waist	0	0	0	0	0	144	904	2	92	1	1	9	0	7	938	2
	hands forward	0	11	0	8	50	14	0	967	173	111	109	35	1	263	51	307
Predicted positi-on label		han-ds up	right hand up	left hand up	han-ds on head	arms open	stand up straig-ht	hands on wai-st	hands forw-ard	han-ds up	right hand up	left hand up	hands on head	arms open	stand up straight	hands on waist	hands forward

Table 4. Confusion matrix of SVM with RBF kernel and kNN classifiers on three types of data

Data type	True position label	SVM with RBF kernel								KNN							
Angles Data set	hands up	421	123	51	75	0	247	36	97	389	161	62	66	1	219	77	75
	right hand up	78	603	85	0	0	149	15	120	95	610	150	0	0	150	5	40
	left hand up	63	120	408	8	14	344	3	90	110	134	476	2	17	251	9	51
	hands on head	0	0	8	978	0	0	9	55	23	0	3	956	0	0	25	43
	arms open	11	0	16	0	982	0	9	32	23	32	15	0	975	0	3	2
	stand up straight	78	26	88	1	0	798	0	59	103	60	109	0	0	753	6	19
	hands on waist	63	12	6	5	0	2	906	56	90	9	0	21	3	12	890	25
	hands forward	94	56	124	30	0	62	22	662	162	100	141	35	0	148	25	439
Points Data set	hands up	1032	0	0	18	0	0	0	0	755	0	0	278	0	0	0	17
	right hand up	0	1050	0	0	0	0	0	0	12	1008	0	0	0	0	30	0
	left hand up	0	0	1044	6	0	0	0	0	0	0	1011	9	0	15	15	0
	hands on head	36	0	0	1014	0	0	0	0	144	0	0	853	0	30	0	23
	arms open	9	0	0	0	1041	0	0	0	25	0	0	8	985	2	30	0
	stand up straight	0	0	0	0	0	1044	6	0	0	0	0	0	0	551	499	0
	hands on waist	0	0	0	0	0	37	1011	2	0	0	0	30	0	147	873	0
	hands forward	0	3	0	0	29	0	0	1018	0	11	0	59	0	0	71	909
Both Data set	hands up	280	132	82	84	0	330	113	29	525	172	63	71	5	97	66	51
	right hand up	63	594	84	27	0	216	15	51	45	735	82	0	0	135	14	39
	left hand up	68	63	448	13	0	373	16	69	104	38	596	3	17	237	8	47
	hands on head	0	0	0	997	0	4	21	28	28	1	4	955	0	0	24	38
	arms open	38	4	0	0	976	0	22	10	19	38	8	0	975	0	8	2
	stand up straight	52	18	60	0	0	897	4	19	58	41	76	0	0	845	8	22
	hands on waist	89	2	1	8	0	7	937	6	38	8	1	16	20	17	917	33
	hands forward	166	114	132	29	1	292	40	276	135	71	112	31	0	86	25	590
Predicted position label		hands up	right hand up	left hand up	hands on head	arms open	stand up straight	hands on waist	hands forward	hands up	right hand up	left hand up	hands on head	arms open	stand up straight	Hands on waist	hands forward

It is worth noting that the use of any classifier model saved in real-time testing will work without problems or delays in the presentation, Table 6 The table shows the average time taken to test each frame and show the results. We note that the best classifier is MLP in terms of speed, then SVM with Linear kernel follows, and the slowest classifier is random forest, but all falls within the real-time of the test.

Table 5. Confusion matrix of RF and MLP classifiers on three types of data

Data type	**True position label**	**RF**								**MLP**							
Angles Dataset	hands up	370	169	93	45	0	163	60	150	180	214	100	108	36	222	90	100
	right hand up	82	657	165	0	0	57	0	89	57	687	115	0	14	103	0	74
	left hand up	89	129	496	9	0	157	0	170	44	137	438	10	33	285	1	102
	hands on head	12	0	4	977	0	2	13	42	15	6	12	895	0	1	77	44
	arms open	10	0	31	0	987	0	14	8	1	3	41	0	974	4	19	8
	stand up straight	109	78	99	2	0	682	1	79	18	49	81	0	0	853	1	48
	hands on waist	53	17	1	7	0	20	894	58	79	31	2	19	46	0	822	51
	hands forward	97	59	65	25	2	34	12	756	84	124	153	32	2	88	47	520
Poin-ts Dataset	hands up	920	0	15	39	60	0	0	16	1047	0	0	3	0	0	0	0
	right hand up	0	1050	0	0	0	0	0	0	0	1050	0	0	0	0	0	0
	left hand up	0	0	1050	0	0	0	0	0	0	0	1045	5	0	0	0	0
	hands on head	156	0	0	820	61	0	0	13	10	0	0	1040	0	0	0	0
	arms open	8	0	0	0	1042	0	0	0	10	0	0	0	1040	0	0	0
	stand up straight	0	0	0	0	0	888	162	0	0	0	0	0	0	1047	3	0
	hands on waist	0	0	0	0	9	104	937	0	0	0	0	0	0	52	990	8
	hands forward	0	11	5	0	217	0	32	785	0	2	0	0	0	0	0	1048
Both Dataset	hands up	918	0	15	117	0	0	0	0	973	1	18	57	1	0	0	0
	right hand up	0	1050	0	0	0	0	0	0	0	1050	0	0	0	0	0	0
	left hand up	0	0	1050	0	0	0	0	0	0	0	1011	0	0	39	0	0
	hands on head	68	0	0	982	0	0	0	0	62	14	3	970	0	0	0	1
	arms open	9	0	0	0	1036	0	0	5	11	0	0	0	1039	0	0	0
	stand up straight	0	0	0	0	0	1044	6	0	0	0	0	0	0	1038	12	0
	hands on waist	0	0	0	0	0	52	998	0	0	0	0	3	0	151	890	6
	hands forward	0	13	4	0	127	28	13	865	0	29	0	27	0	0	1	993
Predicted position label		hands up	Ri-ght hand up	left hand up	hands on head	arms op-en	stand up straight	hands on waist	hands forward	hands up	right hand up	left hand up	hands on head	arms open	stand up straight	hands on waist	hands for ward

Table 6. Average time to test one frame in online test

Classifier	Data type	Average time (second)	Classifier	Data type	Average time (second)
SVM with Linear kernel	Angles	0.00232	kNN	Angles	0.00255
	Points	0.00088		Points	0.01190
	Both	0.00082		Both	0.01518
SVM with Polynomial kernel	Angles	0.00164	RF	Angles	0.02546
	Points	0.00124		Points	0.02549
	Both	0.00316		Both	0.03023
SVM with RBF kernel	Angles	0.00424	MLP	Angles	0.00078
	Points	0.00325		Points	0.00072
	Both	0.00677		Both	0.00059

5. CONCLUSION

In this research, we tested three types of data extracted from the skeleton of the Kinect device on four classifiers with the presentation of the results. The classifier that achieved the best performance on points data is random forests, which had the best accuracy on the third type of data. It is observed that high results achieved up to 99% in SVM with polynomial kernel and 98.79% in MLP by using points data. Post-training classifiers can be used to save in the model, and the saved model can be used for real-time detection and classification. In the test procedure, results demonstrated that human position can be recognized by only one frame of data, by examining the incoming data sequentially for each frame. Numbers of problems or difficulties occurred, including the inability to train some classifiers, such as SVM with the polynomial kernel, which failed to classify data above the 4th degree, and the time it takes to train SVM is longer than other classifiers. In future work, we will study the use of other algorithms with the possibility of linking them with devices to execute orders, or even using raspberry pi instead of PC.

ACKNOWLEDGEMENTS

The authors are thankful to Mustansiriyah University hosted by the Department of Computer Science in the College of Education. Baghdad, Iraq, for supporting them with the investigations.

REFERENCES

[1] M. Tölgyessy, M. Dekan, and Ľ. Chovanec, "Skeleton tracking accuracy and precision evaluation of Kinect V1, Kinect V2, and the azure kinect," *Applied Sciences*, vol. 11, no. 12, p. 5756, Jun. 2021, doi: 10.3390/app11125756.

[2] B. Galna, G. Barry, D. Jackson, D. Mhiripiri, P. Olivier, and L. Rochester, "Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease," *Gait & Posture*, vol. 39, no. 4, pp. 1062–1068, Apr. 2014, doi: 10.1016/j.gaitpost.2014.01.008.

[3] J. Shotton *et al.*, "Real-Time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013, doi: 10.1145/2398356.2398381.

[4] H.-D. Yang, "Sign language recognition with the kinect sensor based on conditional random fields," *Sensors*, vol. 15, no. 1, pp. 135–147, Dec. 2014, doi: 10.3390/s150100135.

[5] B. M. V. Guerra, S. Ramat, G. Beltrami, and M. Schmid, "Automatic pose recognition for monitoring dangerous situations in ambient-assisted living," *Frontiers in Bioengineering and Biotechnology*, vol. 8, no. May, pp. 1–12, May 2020, doi: 10.3389/fbioe.2020.00415.

[6] M. Ababneh, H. Shaban, D. AlShalabe, D. Khader, H. Mahameed, and M. AlQudimat, "Gesture controlled mobile robotic arm for elderly and wheelchair people assistance using kinect sensor," in *2018 15th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2018, pp. 636–641, doi: 10.1109/SSD.2018.8570547.

[7] L. Wang, "Analysis and evaluation of Kinect-based action recognition algorithms," *arXiv preprint*, pp. 1–22, Dec. 2021, doi: 10.48550/arxiv.2112.08626.

[8] N. Chen, Y. Chang, H. Liu, L. Huang, and H. Zhang, "Human pose recognition based on skeleton fusion from multiple kinects," in *2018 37th Chinese Control Conference (CCC)*, 2018, vol. 2018-July, pp. 5228–5232, doi: 10.23919/ChiCC.2018.8483016.

[9] N. Li and X. Zhao, "A Benchmark for gait recognition under occlusion collected by multi-kinect SDAS," *arXiv preprint*, vol. 1, no. 1, pp. 1–20, Jul. 2021, doi: 10.48550/arxiv.2107.08990.

[10] H. Alabbasi, A. Gradinaru, F. Moldoveanu, and A. Moldoveanu, "Human motion tracking & evaluation using Kinect V2 sensor," *2015 E-Health and Bioengineering Conference, EHB 2015*, pp. 2–5, 2016, doi: 10.1109/EHB.2015.7391465.

[11] K. Ryselis, T. Petkus, T. Blažauskas, R. Maskeliūnas, and R. Damaševičius, "Multiple Kinect based system to monitor and analyze key performance indicators of physical training," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, p. 51, Dec. 2020, doi: 10.1186/s13673-020-00256-4.

[12] L. Xidong, "KinectV2 Sensor Real-Time Dynamic Gesture Recognition Algorithm," *Instrumentation and Equipments*, vol. 08, no. 01, pp. 8–20, 2020, doi: 10.12677/iae.2020.81002.

[13] E. W. Trejo and P. Yuan, "Recognition of yoga poses through an interactive system with kinect device," in *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*, 2018, pp. 1–5, doi: 10.1109/ICRAS.2018.8443267.

[14] Z. A. Mundher and Z. Jiaofei, "A real-time fall detection system in elderly care using mobile robot and kinect sensor," *International Journal of Materials, Mechanics and Manufacturing*, vol. 2, no. 2, May 2014, doi: 10.7763/IJMMM.2014.V2.115.

[15] A. D. Q. Burle, T. B. D. G. Lafayette, J. R. Fonseca, V. Teichrieb, and A. E. F. Da Gama, "Real-time approach for gait analysis using the Kinect v2 sensor for clinical assessment purpose," in *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, 2020, pp. 144–153, doi: 10.1109/SVR51698.2020.00034.

[16] N. E. Zain and W. N. Rahman, "Three-dimensional (3D) scanning using Microsoft® Kinect® Xbox 360® scanner for fabrication of 3D printed radiotherapy head phantom," *Journal of Physics: Conference Series*, vol. 1497, 2020, doi: 10.1088/1742-6596/1497/1/012005.

[17] H. P. H. Shum and E. S. L. Ho, "Real-time physical modelling of character movements with microsoft kinect," *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, pp. 17–24, 2012, doi: 10.1145/2407336.2407340.

[18] R. Brancati, C. Cosenza, V. Niola, and S. Savino, "Experimental measurement of underactuated robotic finger configurations via RGB-D sensor," in *Mechanisms and Machine Science*, vol. 67, Springer International Publishing, 2019, pp. 531–537.

[19] K. Loumponias, N. Vretos, G. Tsaklidis, and P. Daras, "An Improved Tobit Kalman Filter with Adaptive Censoring Limits," vol. 1, no. 1, pp. 41–53, Nov. 2019, doi: 10.48550/arxiv.1911.06190.

[20] E. Lachat, H. Macher, T. Landes, and P. Grussenmeyer, "Assessment and calibration of a RGB-D camera (Kinect v2 Sensor) towards a potential use for close-range 3D modeling," *Remote Sensing*, vol. 7, no. 10, Oct. 2015, doi: 10.3390/rs71013070.

[21] C. C. Chang and C. J. Lin, "LIBSVM: A Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–39, 2011, doi: 10.1145/1961189.1961199.

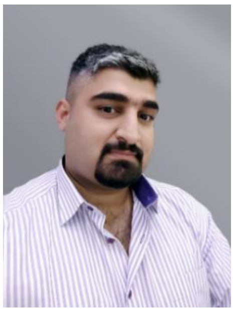
[22] A. Mathur and G. M. Foody, "Multiclass and binary SVM classification: implications for training and classification users," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 241–245, Apr. 2008, doi: 10.1109/LGRS.2008.915597.

[23] A. Napolitano, *Classification techniques for noisy and imbalanced data*, no. December. Boca Raton: Florida Atlantic University, 2009.

[24] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, no. 2, pp. 197–227, Jun. 2016, doi: 10.1007/s11749-016-0481-7.

[25] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer perceptron: architecture optimization and training," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 1, 2016, doi: 10.9781/ijimai.2016.415.
[26] D. A. Adama, A. Lotfi, C. Langensiepen, K. Lee, and P. Trindade, "Human activity learning for assistive robotics using a classifier ensemble," *Soft Computing*, vol. 22, no. 21, pp. 7027–7039, Nov. 2018, doi: 10.1007/s00500-018-3364-x.
[27] H. Byun and S. W. Lee, "Applications of support vector machines for pattern recognition: A survey," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2388, pp. 213–236, 2002, doi: 10.1007/3-540-45665-1_17.
[28] A. Manzi, P. Dario, and F. Cavallo, "A human activity recognition system based on dynamic clustering of skeleton data," *Sensors*, vol. 17, no. 5, p. 1100, May 2017, doi: 10.3390/s17051100.
[29] X. Li, Y. Zhang, and D. Liao, "Mining key skeleton poses with latent SVM for action recognition," *Applied Computational Intelligence and Soft Computing*, vol. 2017, pp. 1–11, 2017, doi: 10.1155/2017/5861435.
[30] K. Arai and R. Andrie, "3D Skeleton model derived from kinect depth sensor camera and its application to walking style quality evaluations," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, 2013, doi: 10.14569/IJARAI.2013.020705.
[31] M. L. Anjum, S. Rosa, and B. Bona, "Tracking a subset of skeleton joints: an effective approach towards complex human activity recognition," *Journal of Robotics*, vol. 2017, pp. 1–8, 2017, doi: 10.1155/2017/7610417.
[32] L. Piyathilaka and S. Kodagoda, "Understanding Human Context in 3D Scenes by Learning Spatial Affordances with Virtual Skeleton Models," *arXiv preprint*, pp. 1–25, Jun. 2019, doi: 10.48550/arxiv.1906.05498.
[33] M. El Amine Elforaici, I. Chaaraoui, W. Bouachir, Y. Ouakrim, and N. Mezghani, "Posture recognition using an rgb-d camera: exploring 3d body modeling and deep learning approaches," in *2018 IEEE Life Sciences Conference (LSC)*, 2018, doi: 10.1109/LSC.2018.8572079.
[34] K. Han, Q. Yang, and Z. Huang, "A two-stage fall recognition algorithm based on human posture features," *Sensors*, vol. 20, no. 23, p. 6966, Dec. 2020, doi: 10.3390/s20236966.
[35] S. Ubalde, F. Gomez-Fernandez, N. A. Goussies, and M. Mejail, "Skeleton-based action recognition using citation-kNN on bags of time-stamped pose descriptors," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, vol. 2016-Augus, no. Mil, pp. 3051–3055, doi: 10.1109/ICIP.2016.7532920.
[36] H. Ramirez, S. A. Velastin, E. Fabregas, I. Meza, D. Makris, and G. Farias, "Fall detection using human skeleton features," in *11th International Conference of Pattern Recognition Systems (ICPRS 2021)*, 2021, vol. 2021, doi: 10.1049/icp.2021.1465.
[37] S. Baek, Z. Shi, M. Kawade, and T.-K. Kim, "Kinematic-layout-aware random forests for depth-based action recognition," *arXiv preprint*, pp. 1–10, Jul. 2016, doi: 10.48550/arXiv.1607.06972.
[38] S. Laraba, M. Brahimi, J. Tilmanne, and T. Dutoit, "3D skeleton-based action recognition by representing motion capture sequences as 2D-RGB images," *Computer Animation and Virtual Worlds*, vol. 28, May 2017, doi: 10.1002/cav.1782.
[39] S. Canavan, W. Keyes, R. Mccormick, J. Kunnumpurath, T. Hoelzel, and L. Yin, "Hand gesture recognition using a skeleton-based feature representation with a random regression forest," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, vol. 6785, no. 4, pp. 2364–2368, doi: 10.1109/ICIP.2017.8296705.
[40] A. M. De Boissiere and R. Noumeir, "Infrared and 3D skeleton feature fusion for RGB-D action recognition," *IEEE Access*, vol. 8, pp. 168297–168308, 2020, doi: 10.1109/ACCESS.2020.3023599.
[41] W. Zhao, S. Yang, T. Qiu, and X. Luo, "Person identification based on static features extracted from Kinect Skeleton data," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021, doi: 10.1109/SMC52423.2021.9659013.
[42] D. K. Barupal and O. Fiehn, "Generating the blood exposome database using a comprehensive text mining and database fusion approach," *Environmental Health Perspectives*, vol. 127, no. 9, p. 097008, Sep. 2019, doi: 10.1289/EHP4713.

BIOGRAPHIES OF AUTHORS



Ahmed Etihad Jaleel was born in Baghdad, Iraq in 1989. He received the B. degree in computer science from the University of Mustansiriyah, in 2011. His current research interests pattern recognition, artificial intelligence. He can be contacted at email: ahmed.etihad@uomustansiriyah.edu.iq.

Hesham Adnan Alabbasi was born in Baghdad, Iraq in 1969. He received his B.Sc. degree in Computer Science in 1992 from Mustansiriyah University and his M.Sc. degree in Computer Science in 1999 from the University of Technology, Iraq, Baghdad. He received his Ph.D. degree in 2016 in Computer Science/Software from Politehnica University in Bucharest, Romania. In 2004, he joined the faculty of Engineering/Computers Engineering Dept. at the Mustansiriyah University in Baghdad. Now, he is Assist. Prof. In the college of education\computer science department. His recent research activities are Face Recognition, Emotions Detection, and Body Motions Tracking with Kinect Sensor, Wireless Sensor Network Systems, Speech Signal Processing, and Chaotic Modulation. Now he is a lecturer at the Mustansiriyah University, college of education, He can be contacted at email: hesham.alabbasi@uomustansiriyah.edu.iq.