Multimodal deep learning model for human handover classification

Islam A. Monir,[1] Mohamed W. Fakhr,[2] Nashwa El-Bendary[3]
[1]College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Smart Village, Egypt
[2]College of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Cairo, Egypt
[3]College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Aswan, Egypt

Article Info

Article history:

Received Oct 12, 2021
Revised Jan 27, 2022
Accepted Feb 23, 2022

Keywords:

Action recognition
Human handover
Multimodality
Robot-human interaction
Robotic grasping

ABSTRACT

Giving and receiving objects between humans and robots is a critical task which collaborative robots must be able to do. In order for robots to achieve that, they must be able to classify different types of human handover motions. Previous works did not mainly focus on classifying the motion type from both giver and receiver perspectives. However, they solely focused on object grasping, handover detection, and handover classification from one side only (giver/receiver). This paper discusses the design and implementation of different deep learning architectures with long short term memory (LSTM) network; and different feature selection techniques for human handover classification from both giver and receiver perspectives. Classification performance while using unimodal and multimodal deep learning models is investigated. The data used for evaluation is a publicly available dataset with four different modalities: motion tracking sensors readings, Kinect readings for 15 joints positions, 6-axis inertial sensor readings, and video recordings. The multimodality added a huge boost in the classification performance; achieving 96% accuracy with the feature selection based deep learning architecture.

Corresponding Author:

Islam A. Monir
College of Computing and Information Technology
Arab Academy for Science and Technology and Maritime Transport, B2401, Smart Village, Giza, Egypt
Email: islammonir@aast.edu

1. INTRODUCTION

Live interaction of robots with human beings is one of the emerging technologies with several research points to be addressed. One of the main challenges with robot's interaction is its ability to take/give an object from/to a human successfully the robot must be able to classify the type of human's handover action; so that it can successfully determine the apt response given any situation. Human handover classification is a research area with limited number of researchers working on it.

Thus, automatic classification of different types of handover actions is one of the important challenging tasks for robots' live interactions. Most of the previous works related to handover focus mainly on handover detection, hand grasping, the type of motion to take a steady object, and the classification of handover motion type from merely one perspective. The challenging task is to be able to classify different types of human motion; to make a successful object handover from both perspectives.

Deep learning is one of the most recently used techniques in classification with many developed models for different tasks. These models are separated into two different types, which are unimodal and multimodal architectures. Unimodal architectures are the most widely used models, these types of models usually train on one type of data, unlike multimodal architectures that may take many different types of data as input. Multimodal architectures exploit different types of inputs with different features to make a more accurate and specific representation of the input situation.

The concept of multimodality depends on multimodal data. Multimodal data comes in different formats, such as different types of sensors readings, videos, and images. These data formats can be used separately in different unimodal classification models, or be integrated together to be used in one multimodal classification model. As an example, the work done by R. González-Ibáñez *et al.* [1] differentiates between unimodal and multimodal architectures for detection of relevance in interactive IR. Moreover, another work presents an overview on the state-of-the-art deep learning models used for classification of sensor-based data that are unimodal and multimodal [2]. Not to mention, it has been used in the field of video classification in other works [3], [4].

As multimodality proved its efficiency in many previous research areas as described in the related work section; the multimodal deep learning human handover classification model (MDHHC) proposed in this paper is designed with different architectures, mainly based on long short term memory (LSTM) network with different feature selection techniques; to investigate their effect on the model performance. Additionally, Unimodal architectures are designed for different data inputs with different model architectures to scrutinize their effect on the model performance and the effect of using concept of multimodality on the classification task.

Object grasping by hand is a research area that has been studied lately in many previous works. Satish *et al.* [5] tried to train a robot policy that analyses millions of grasp candidates in 4-DOF using a fully convolutional network architecture. Prior works in this area presented many ways in grasp detection and/or classification. A research has been done on real time robotic grasp detection [6]. This work tries to find local optimal grasps in candidate grasp rectangles. It used enhanced segmentation techniques to separate the object from the background, along with morphological image processing techniques to generate candidate grasp rectangles sets for a Random Forest to be trained on; instead of searching on global grasp rectangles sets. Grasp detection technique was done using Random Forest model which achieved an accuracy of 94.26%.

Grasping of objects may come in different shapes and techniques and grasp may fail if the object was not carried correctly. Some works that were done lately lets robots learn the correct grasping techniques based on human experience in grasping of objects. A work done in 2019 depends on predicting what action the human will perform to grasp the object [7]. A classifier is well trained on taking visual information of the object and predict the human grasping technique to be done, and use the prediction information for deciding the suitable action for robots' soft hands to grasp the object. In the same line another work also depends on human experience but in a slightly different way, it utilizes human intervention to control the robot's hand to grasp the object [8]. The data acquired from the human operator to do the grasp action is being used to train a Decision Tree that is used later to generate the hand movement action required to grasp the object.

Furthermore, Arapi *et al.* [9] proposed a system to decide a hand grasp of objects. The objective of this work is to predict soft hand's grasp failure before occurrence using IMU sensors data as input. Two deep learning architectures have been implemented and used in this work, none of which achieved 100% accuracy. Another work [10] enhances robots working in factories which enables robots to carry many heavy objects. This work depends on using deep convolutional neural networks (CNN) for hand grasping prediction of both single and multiple hand poses all at once using RGB images as input. Very few past works concentrated on handovers in specific, all work in this area focused on the detection of a handover action being performed and how the action is being performed.

Detection of a handover action task was carried in many different forms. A work established in 2017 [11] made use of kinematic features with SVM classifier to detect a handover action. Kinematic features like joint angles and distances between joints of the giver were measured and selected using bagged Random Forests. The extracted important kinematic features are input to SVM classifier to classify the giver's intent to hand an object to the robot. The SVM classifier was able to classify 97.5% of the data correctly but from the receiver's perspective only. Moreover, another work makes use of classifying object grasping and holding made by human in making successful handover [12]. It makes use of a human grasping and holding poses for grasping classification. With the classification data, the algorithm quickly plans a trajectory accordingly for the robot to meet the human half way and receive the object.

The field of activity recognition and classification is a powerful research point that many researches worked on lately. Activity data may come into different forms like sensors, images, and videos. Wearable sensors are the most widely used data sources for human activity recognition. Meanwhile, a literature review of the sensor-based datasets that are used in the activity recognition task was done by De-La-Hoz-Franco *et al.* in 2018 [13]. Wang and Liu [14] presented a hierarchical LSTM architecture that accepts inputs from wearable sensors like accelerometer and gyroscope. Three public UCI datasets have been used in the experimental setup and proved the outperformance of this model by 99.15% accuracy. Serving the same task, Tang *et al.* [15]

proposed a lightweight CNN that is capable of making human activity recognition from wearable sensors readings. The datasets used for evaluation are UCI-HAR, PAMP2, WISDM, and OPPORTUNITY datasets, and the accuracy came to be 96.90%, 92.97%, 98.82%, and 88.09% respectively. Continuing in the same line a work [16] performs systematic-study on on-body sensor positioning and data acquisition. It depended on eight body worn inertial measurement sensors on different positions on body, sensors readings are being input to LSTM network for training. Afterwards, late fusion process is performed by having the output of different classes probabilities from each sensor classification as an input for an ensemble model to perform final classification; applying the concept of multimodality. The work done in 2021 [17] also tries to make supervised classification of different hand motions using EMG signals. The MYO-ARM BAND dataset was used for evaluation in this work and achieved a high accuracy of 83.9%.

Feature extraction also plays an important role in all deep learning classification techniques. Some researches start to focus on how to improve feature extraction to get more accurate classification of the action being performed. Nafea *et al.* [18] in their research makes use of CNN and Bi-LSTM for feature extraction. Using UCI datasets as input the architecture extracts spatial and temporal features from the input sensor data for classification of the action being performed. This type of feature extraction actually gave good results (97-98%).

Most of the researches made and demonstrated above either focus on activity recognition in general or on hand grasping, which means focusing on taking an object from anywhere. The rest of the researches who dealt with human handovers either worked on detecting the handover process or took only the perspective of the receiver and how the receiver shall deal with the handover. The Human Handover detection and classification task is a combination of all of the previously-mentioned tasks, in which the object is being grasped but by another person. The handover technique being done is a special type of activity that is recognized; and finally this activity involves both giver and receiver in the same action.

In the upcoming sections, the proposed model is first introduced, which is able to classify the whole handover process, with its different architectures (unimodal and multimodal). The setup of the training environment and the dataset being used for evaluation are also discussed. Subsequently, a comparative study is being made on the results of all proposed architectures, outlining the architectures with the best outcomes.

2. RESEARCH METHOD

2.1. The proposed approaches

This section illustrates the proposed multimodal deep learning human handover classification (MDHHC) model, which tries to introduce multimodality in human handover classification and investigate its effect on the performance of the model. The model gets to work on a complete handover not just grasping and can work from either giver or receiver point of view without any difference. Furthermore, the different architectures tested for the MDHHC model are introduced. Multiple experiments with multiple architectures were performed to investigate their effect on the classification performance. First, a unimodal architecture was designed to take multiple sensors of only one type with two different architectures (with and without feature selection). Second, a multimodal architecture was designed to take several data formats as input with many different architectures being tested. The following sections go into details of every experiment model architecture.

2.1.1. Unimodal architectures

The unimodal architecture is designed to have only one type of sensors, such as motion tracking sensors, Kinect 15 joints positions readings, or inertial sensor readings as input for handover action classification. The architecture of this model takes as input multiple sensors readings of the same type, thereafter extracts features for each input sensor readings. The features are then merged together in different ways as discussed below.

Figures 1 and 2 show both parts of the unimodal architecture. As shown in Figure 1, each sensor readings (x, y and z) are passed through time distributed fully connected layer for feature extraction. A time distributed layer is a wrapper layer which deals with sequential data. It takes each time-step data as a different input and deals with it as a normal input; and produces an output for that time-step. The concept of time distributed layer is introduced and used in [19]. The time distributed FC layer produces a 2-D feature vector containing feature vectors at each time-step.

As shown in Figure 2, each 2-D vector is passed through LSTM network for temporal feature extraction producing a 1-D feature vector for each input sensor. Afterwards, all feature vectors are concatenated into one vector for first level of fusion. Two different paths for the concatenated feature vector were investigated here.

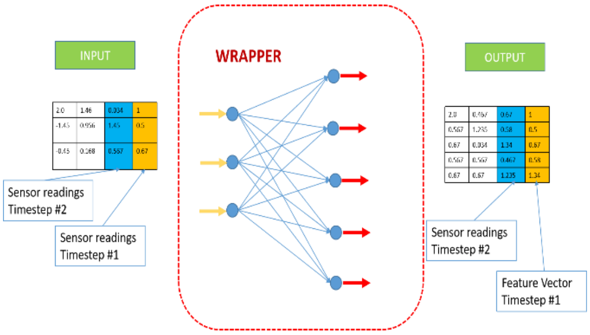

INPUT
WRAPPER
OUTPUT
Sensor readings Timestep #2
Sensor readings Timestep #1
Sensor readings Timestep #2
Feature Vector Timestep #1

Figure 1. Time distributed layer architecture

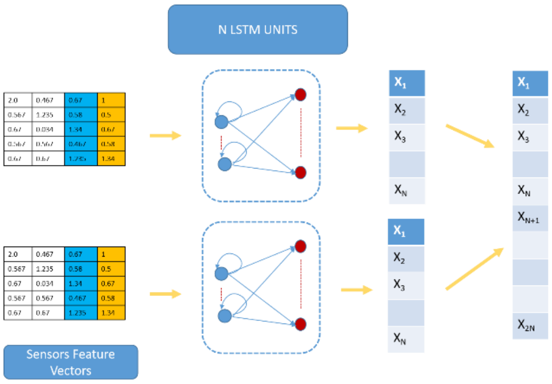


Figure 2. Architecture for LSTM network

Experiment 1 (Unimodal-no feature selection architecture (Unimodal_No_FS))

The first architecture takes the feature vector as it is and passes it to the final classification layer. The final layer is a fully connected layer with a number of outputs the same as the number of classes to differentiate between them. The output is passed through a Sigmoid activation function to calculate the probabilities of the final classes.

Experiment 2 (Unimodal-feature selection architecture with decision trees (Multimodal_DT_FS))

This architecture applied a type of feature selection using decision trees to choose only the important features out of the whole vector for classification. The feature selection technique here is seldom used. It makes use of decision trees for selecting the important features. Decision trees are one of the machine learning classification techniques. It consists of nodes and edges that form a tree. The node represents the feature on which a decision is to be made, whereas the edge represents the decision criteria based on the feature.

The tree construction algorithm at each node selects the feature that best classifies the problem according to some measures of impurity. One of these measures is the Gini Index. The feature that gets the least Gini Index at that split is the best feature to be used. The process of exploiting the Gini Index for deciding the best feature that best splits the node into two sub-nodes is briefly described in [20], [21] and applied by Theodoridis and Gkikas using genetic algorithm [22]. A flow chart for the algorithm is shown in Figure 3.

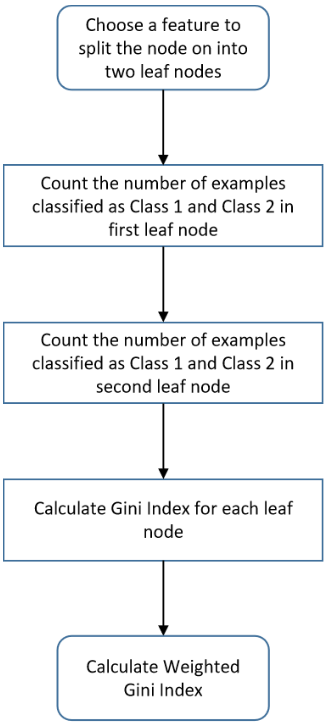

Choose a feature to split the node on into two leaf nodes
Count the number of examples classified as Class 1 and Class 2 in first leaf node
Count the number of examples classified as Class 1 and Class 2 in second leaf node
Calculate Gini Index for each leaf node
Calculate Weighted Gini Index

Figure 3. Decision tree process structure

In the model presented in this paper, the Gini Index calculation is used in decision tree building. Given the fact that decision trees sort the features according to their importance, it is used here as a feature selection method for the concatenated feature vector; to select the features that are deemed salient. The selected features are then passed to the same final layer described in the first architecture as shown in Figure 4.

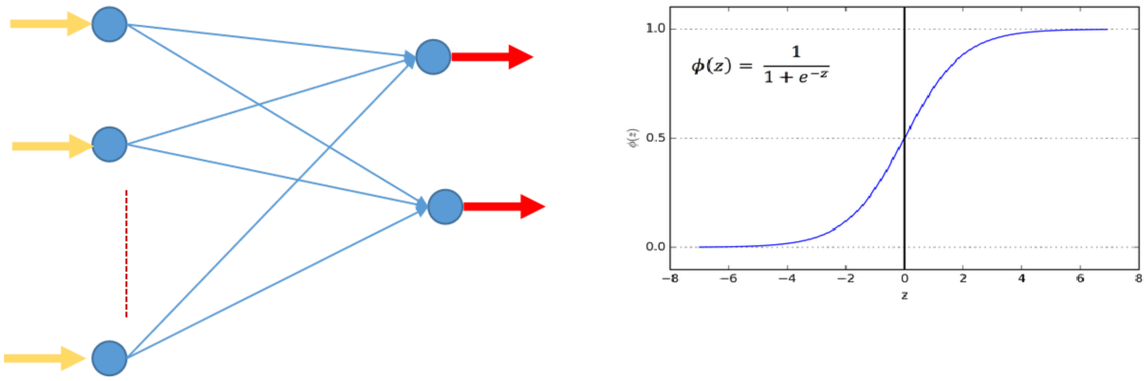


Figure 4. Classification layer with Sigmoid activation function

2.1.2. Multimodal architectures

This subsection presents the whole multimodal architecture, and introduces the concept of multimodality in many different model architectures. Different architectures are designed to have three channels for three different data modalities to be input together at once (motion tracking sensors readings, Kinect 15 joints positions, Inertial 6-axis sensor readings). Architectures were investigated with and without different types of feature selection techniques.

Experiment 3 (Multimodal-no feature selection architecture (Multimodal_No_FS))

The architecture of the model here is simple; for each input channel different readings are being input and passed through the aforementioned time distributed FC layer. Following that, the output vector of each sensor reading is passed through the same LSTM network. Concatenating the outputs of all LSTM networks of all 3 input channels without any type of selection is the concept being applied in this experiment. The resulted feature vector is then passed through a fully connected hidden layer. The output feature vector is then passed through the final classification layer mentioned above, as shown in Figure 5(a).

Experiment 4 (Multimodal-feature selection architecture with decision trees (Multimodal_DT_FS))

As mentioned above, Decision Trees can be used in feature selection. For 3 input channels, each channel will have its feature selection done on feature vector of its own type of input. The feature vector selected in each channel is passed through a semi-final classification layer with two outputs representing the classes' probabilities predicted by the channel. The three channels outputs are then passed to the final classification layer, as shown in Figure 5(b).

Experiment 5 (Multimodal-feature selection architecture with random forests (Multimodal_RF_FS))

This experiment has the same architecture as Multimodal_DT_FS with replacing the feature selection technique. Random forest is a machine learning algorithm that consists of more than one decision tree, they are deeply discussed in [23]. It is being trained through bagging or bootstrap aggregation ensemble algorithms. As a result, random forests can be used in the same way for feature selection akin to decision trees.

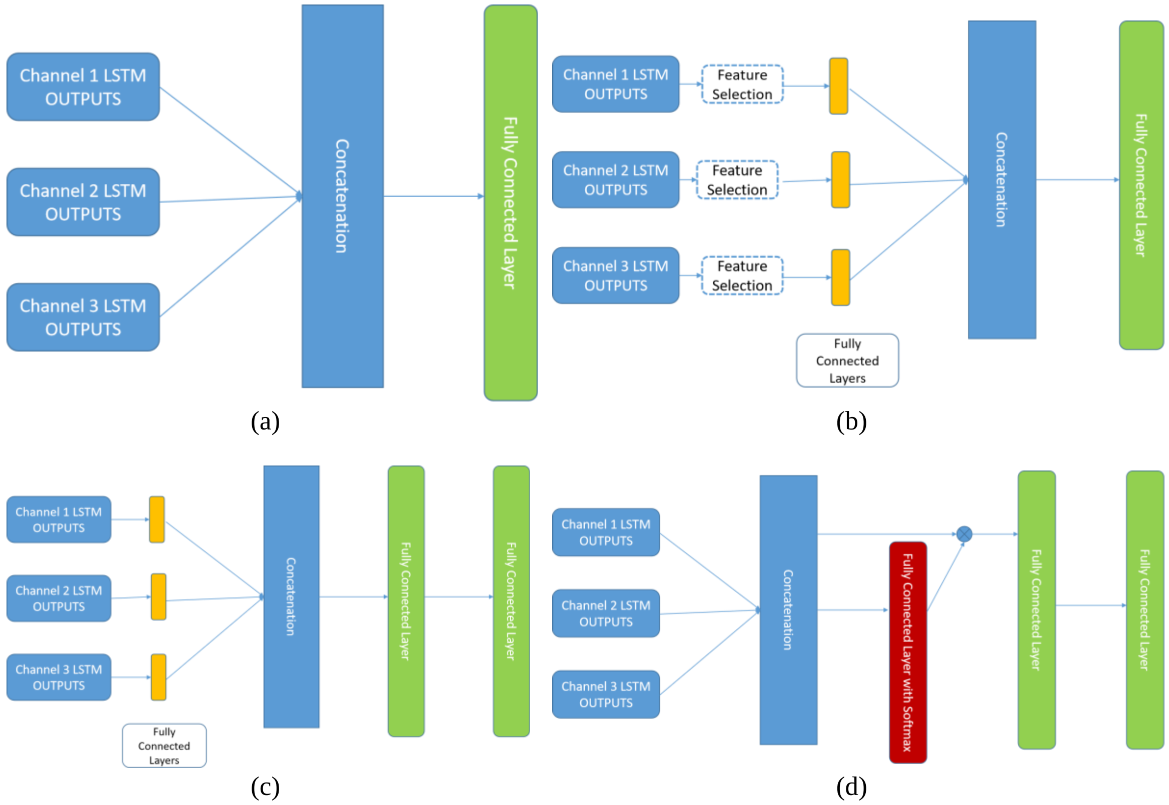
Experiment 6 (Multimodal with late fusion)

Late Fusion can come into different forms, one of which is described in a work done by Pandeya and Lee [24]. The architecture used here is similar to Multimodal_No_FS architecture, different readings are being input and passed through the previously-mentioned Time Distributed FC layer for each input channel. Subsequently, the output vector of each sensor reading is passed through the same LSTM network. The outputs of the all LSTMs of each channel are later joined together and passed through a fully connected layer with two outputs representing the classes' probabilities predicted by the channel. The three channels outputs are then passed to a semi-final classification layer to apply Late Fusion of the feature vectors with FC layer. The output feature vector is then passed through the final classification layer mentioned above, as shown in Figure 5(c).

Experiment 7 (Multimodal with attention layer)

The architecture investigated here used the Attention Layer concept. Attention Layer is a layer that is responsible for paying attention to only important features, it was also mentioned and used in other researches

[25]. The architecture here takes the concatenation of all feature vectors being output by all LSTM networks and passes them through an Attention Layer. The layers here consist of a fully-connected layer with Softmax output, with an equal number of inputs. Thus, the output scores represent the importance of each feature in the input feature vector. Multiplying the output scores with the feature vector, a new feature vector is produced that pays attention to the important features. The new feature vector is then passed through a fully-connected layer that then outputs a vector that will pass through the final classification layer mentioned above, as shown in Figure 5(d).



(a) (b)

(c) (d)

Figure 5. Design of different proposed multimodal architectures (a) Multimodal_No_FS, (b) Multimodal_DT_FS/Multimodal_RF_FS, (c) multimodal with late fusion, and (d) multimodal with attention layer

2.2. Experimental setup

2.2.1. Dataset

The dataset used for the evaluation purpose is a publicly available one [26]. Eighteen volunteers participated in the construction of this dataset making two types of experiments (single-blinded and double-blinded). The double-blinded experiment was to investigate whether or not the giver/receiver need to move to get to the handover; and to classify the object being handed. Ergo, this type of experiments was not of concern. The single-blinded type of experiment had each volunteer facing an experimenter. The experimenter makes a handover process with the volunteer, then checks how the volunteer will react and make the handover process, sometimes with the volunteer being the receiver and other times being the giver. Some of these experiments mandated the volunteer to make a move to reach the correct place for handover. Same as before these samples were excluded.

The data in this dataset came into four different modalities. The first is the motion tracking sensors. 20 motion tracking sensors were placed on the volunteer and the experimenter in different places with 10 sensors on each, the last 5 sensors came to have missing readings in some experiments so they were not used in the experiment. The second modality is the Kinect readings for positions of 15 different joints of the volunteer, such as left/right hip, left/right foot, left/right elbow, left/right shoulder, left/right knee, left/right hand, neck, torso, and head. The third modality is a 6-axis inertial sensor in an LG smart watch worn by the volunteer. The fourth modality which is yet to be exploited for this research is video recording of the handover action itself.

2.2.2. Dataset preparation

A problem in this dataset is that all handovers happen to have very similar motion, only one pair of handover types have some recognizable difference (normal handover, wrongpose handover). Consequently, only those two classes were used in experiments. Each record in the data accounts as a whole action. Hence, the use of sliding window technique was abated. Sliding window would have caused a problem as the sequence being input to the model is only part of a handover action not the whole action. The next step was to ensure that all data sequences have the same length. By taking all records into consideration, the record with the least sequence is recognized so that all other sequences are truncated from the beginning; in order to have the same length of the smallest sequence. Sequence truncation has been made to all records of data as shown in Figure 6.

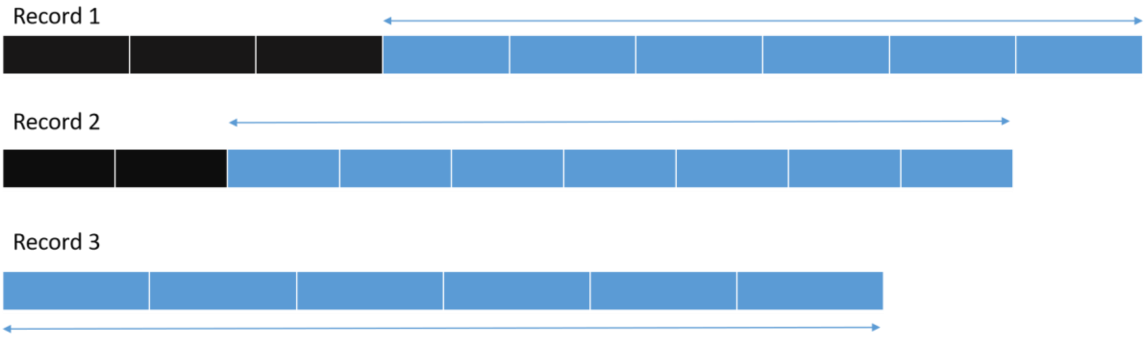


Figure 6. An example of sequence truncation assuming that record 3 has minimum length

2.2.3. Training

The training had two different processes whether for unimodal or multimodal architectures. Without feature selection using decision trees or random forests, the training happened to be a normal end-to-end training using 80% of the dataset samples, whilst the other 20% was for testing.

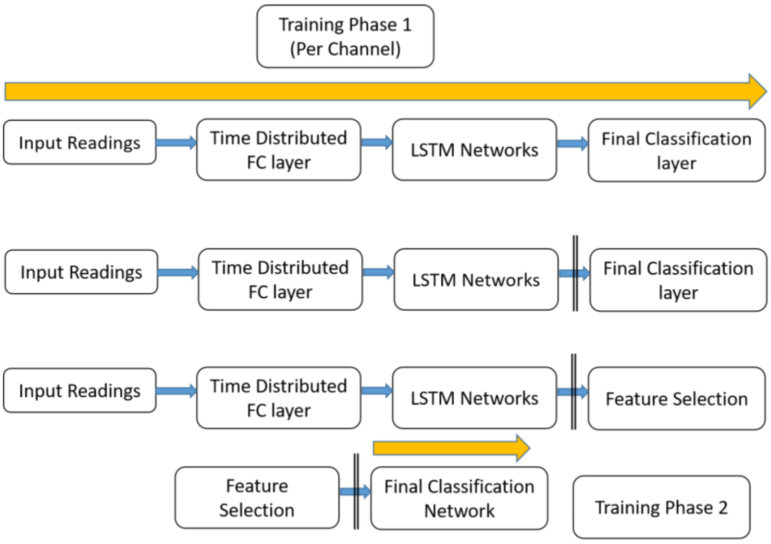


Figure 7. Training steps when using feature selection techniques

Decision trees and random forests are machine learning techniques that are trained via training data induction, unlike deep learning algorithms which are trained via looping on training data. As a result of both having different training techniques, this architecture training will have two steps. First, train each unimodal channel end-to-end training without using feature selection. After training finishes, training and testing data samples are input to the model with cutoff before the feature selection part. Take the extracted features of the training data to train the machine learning part and extract important features for training and testing data. Second, use the new feature vectors to train the final classification part of the model again, as shown in Figure 7. The training parameters that were used in all of the training experiments were chosen by trial and error. It was clear that these parameters were the ones that gave the best performance for all the mentioned architectures. The parameters configuration is summarized in Table 1.

Table 1. Summary of fixed training parameters

Parameter	Value
Number of training samples	80% (120 samples)
Number of testing samples	20% (24 samples)
Number of epochs	100
Batch size	10
Learning rate	0.01

2.2.4. Hyper parameter optimization

For the proposed model to get the highest performance, the concept of hyper parameter tuning was used in a number of sections in the model. Hyper parameter is a normal parameter in the model. Nonetheless, instead of having a fixed value, a range of values is defined. The model starts to run many times while changing the values for this parameter in the defined range, searching for the parameter value that will give the model the best performance. A detailed description of the hyper parameter optimization is in [27].

The MDHHC model had four different defined hyper parameters that are the same in each of the architectures Multimodal_No_FS, Multimodal with late fusion, and multimodal with attention layer. The fully connected layer that was used in the time distributed wrapping layer must have a specific number of perceptrons to make the calculations, this was defined as a hyper parameter ranging from 5 to 20 perceptrons with step of +1 every trial. In the same way the number of units in each LSTM network was a hyper parameter ranging from 25 to 500 with step of +25 every trial. Moreover, the final hidden layer before the classification layer had its number of perceptrons defined as a hyper parameter in the same way ranging from 5 to 20 with step +1. These ranges and the number of steps were changed and tested multiples times and were found to be optimal. In addition to that, the activation function of the final classification layer was optimized by searching between Sigmoid and Softmax activation functions which are mentioned in [28]; as they both can work for classification between two classes. Ten test runs ran on the model as a whole having different combinations of the defined hyper parameters resulting in the combination that best suits the model.

Furthermore, the MDHHC model had three different hyper parameters that are the same in each of the architectures Multimodal_DT_FS and Multimodal_RF_FS. The fully connected layer defined after every feature selection block had its number of perceptrons defined as hyper parameter ranging from 1 to 10 with step +1. The second hyper parameter is the same one described above in the final classification layer between Sigmoid and Softmax activation functions. The third and final hyper parameter is a manually tuned hyper parameter to choose the best number of features to be selected by the feature selection mechanism. The number of features ranged from 1 to 20 with step +5. As a result of manual tuning every value chosen, the model runs 10 trials to find the best values for the other two hyper parameters that give good performance with this number of features, resulting in a 20*10 trial runs.

3. RESULTS AND DISCUSSION

This section presents the detailed results of evaluation of the different proposed architectures in this paper. Table 2 shows the detailed data splitting between training and testing. The data were split in a standard way with 80% for training and 20% for testing. In the evaluation aspect, multiple evaluation metrics were used in evaluating and comparing the performance of different architectures.

Table 2. Dataset training/testing splitting details

Data	Total number of samples	Number of class 1 samples (normal handover)	Number of class 2 samples (wrongpose)
Training data	120	60	60
Testing data	24	12	12

$$\frac{TP + FP}{TP + FP + TN + FN} = \text{Accuracy} \quad (1)$$

$$\frac{TP}{TP+FP}\text{=Precision} \tag{2}$$

$$\frac{TP}{TP+FN}\text{=Recall} \tag{3}$$

$$\frac{2*(Precision * Recall)}{Precision + Recall}\text{=F1-score} \tag{4}$$

Where TP (true positive) count represents the number of class 1 samples that have been correctly classified. TN (true negative) count represents the number of class 2 samples that have been correctly classified. On the other hand, FP (false positive) count represents the number of class 2 samples that have been miss-classified as class 1 and vice-versa with FN (false negative) count. A specific part in the work done by El-Razzaz explains these metrics briefly and utilizes it in other evaluations [29].

In addition to these metrics, there is another one that is precise and widely used in many evaluations: precision recall area under curve (PR-AUC), which plots a curve between precision and recall values and evaluate the area under the curve. A study of the performance of each of the aforementioned architectures in section 2.1 with the setup mentioned in section 2.2 is done. It starts by evaluating the performance of the unimodal architecture with its two different architectures on the three different data modalities used. Table 3 summarizes different data inputs used to evaluate the unimodal architecture.

The evaluation is performed on Unimodal_No_FS architecture described above on each of the inputs mentioned in Table 3. The accuracy, precision, and recall came almost equal for every input. For the motion tracking sensors, the performance seemed to increase by increasing the number of sensors until it reached 10 sensors with 80% performance in all measures, after that, a drop occurred in performance with 15 sensors. This may be because the 10 sensors relate to giver or receiver only and the extra 5 sensors related to the other person. The result that caught the attention is the performance of the LG-smart watch inertial sensor, which scored 91% in accuracy and precision with 83% recall. This is marked to be the highest in the three performance measures.

Table 3. Summary of inputs for testing the unimodal architecture

Data modality	Number of readings
Motion tracking sensors	2-3D sensors 4-3D sensors 10-3D sensors 15-3D sensors
Kinect volunteer joints positions	15 3-axis readings of 15 different joints of volunteer's body positions
LG smart watch inertial sensor	1 6-axis sensor divided into two 3D readings

When unimodal_DT_FS architecture was evaluated on the same inputs, it made an improvement in the model performance with some inputs and a drop with others. In general, no significant change was observable. Table 4 summarizes the difference in performance between both architectures. Not to mention, LG watch sensor readings gave the highest performance 91% in accuracy and precision with 88% in recall (+5% recall), which indicates that the sensor readings in this watch is able to give precise classification of the handover action being done. As a result, this modality type will give a boost in performance to other modalities when used together.

Table 4. Comparison between performances of unimodal_No_FS and Unimodal_DT_FS architectures on different data inputs

Model	Unimodal-No-FS			Unimodal-DT-FS		
Data inputs/measures	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)
2-Sensors	46	46	46	37.5	37.5	37.5
4-Sensors	58	58	58	75	69	67
10-Sensors	80	80	80	54	51	62
15-Sensors	66	66	66	62.5	63	58
Kinect	70	70	70	70	74	70
LG-Smart Watch	**91**	**91**	**83**	**91**	**91**	**88**

From the above experiments, a deduction can be made that the impact of having the smart watch inertial sensor readings as input to the classification model is the highest among all inputs. The second-best input is the Kinect readings. The purpose of this work is to investigate the impact of putting all of these inputs together in one classification network. In the coming part, two of the proposed MDHHC architectures multimodal_DT_FS and multimodal_RF_FS, that contain feature selection are studied, one with Decision Trees and the other with random forests. The first point that was to be studied is finding the best number of features to be selected from the total concatenated feature vector.

In this experiment, both models were run with changing the number of features to be selected (5, 10, 15, and 20). With decision Trees the selection of 20 best features gave the highest possible performance of about 96% in all evaluation metrics; proving that it was able to classify all test samples correctly. Whereas with random forests, the selection of 15 and 20 best features gave the best performance of about 96% in all evaluation metrics. The difference in the resulted number of features selected in both algorithms may come to that the Random Forests consists of many decision trees resulting in choosing the more important features. Below is a summary of both models' performance on different numbers of selected features. As shown in Table 5, random forests gave better results when selecting 15 and 20 best features, while decision trees gave better results when selecting exactly 20 best features. As an example, Table 6 shows the output confusion matrix of the model when choosing best 15 features using random forests.

Table 5. Summary of Multimodal_DT_FS and Multimodal_RF_FS architectures performance with different number of features selecting

	Multimodal_DT_FS					Multimodal_RF_FS				
# of features	Acc (%)	Prec (%)	Rec (%)	F1-score(%)	PR-AUC(%)	Acc (%)	Prec (%)	Rec (%)	F1-score(%)	PR-AUC(%)
5	91.67	92.86	91.67	91.67	92.86	83.33	84.29	83.33	83.33	87.2
10	91.67	92.86	91.67	91.67	92.86	87.5	87.76	87.5	87.5	90.22
15	91.67	92.86	91.67	91.67	92.86	**95.83**	**96.15**	**95.83**	**95.83**	**96.15**
20	**95.83**	**96.15**	**95.83**	**95.83**	**96.15**	**95.83**	**96.15**	**95.83**	**95.83**	**96.15**

Table 6. MDHHC with random forests selecting 15 features confusion matrix

		Actual values	
		Class 1	Class 2
Predicted values	Class 1	12	1
	Class 2	0	11

Investigating the effect of other MDHHC architectures multimodal_No_FS, multimodal with late fusion and multimodal with attention layer, there is a little drop in the performance. The worst performance was for the architecture having the late fusion mechanism with 87.5% accuracy, recall, and F1-score with 87.76% precision and 90.22% PR-AUC. This model misclassified 3 samples out of 24 divided into 2 false positives and 1 false negative.

Attention and No FS mechanisms described above achieved the same performance evaluation with 91.67% for accuracy, precision, recall and F1-score; and 93.75% PR-AUC. The performance is slightly better than the late fusion architecture and slightly lower than the feature selection architectures with 2 samples being misclassified. Table 7 shows a comparison between all MDHHC architectures performance.

Table 7. Summary of performance measures of all MDHHC architectures

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	PR-AUC (%)
Multimodal-NO-FS	91.67	91.67	91.67	91.67	93.75
Multimodal-DT-FS	**95.83**	**96.15**	**95.83**	**95.83**	**96.15**
Multimodal-RF-FS	**95.83**	**96.15**	**95.83**	**95.83**	**96.15**
Multimodal with late fusion	87.5	87.76	87.5	87.5	90.22
Multimodal with attention layer	91.67	91.67	91.67	91.67	93.75

Comparing the results of all experiments, it was found that the architectures that applied inner feature selection mechanisms achieved higher performance in both unimodal and multimodal architectures. In unimodal architecture, the best input discussed above is the smart watch sensor, which gave better recall with feature selection being applied. Also, in multimodal architectures, feature selection models gave the best performance. Other than that, the above results of all architectures deeply show the effect of using the concept of multimodality. In unimodal architectures the best performance achieved was having 91% accuracy and precision, and 88% recall, while out of 5 proposed multimodal MDHHC architectures 4 architectures outperformed these results with the least having 91.67% accuracy, precision and recall. Comparing that with the previously-mentioned works, the proposed models yielded high performance measures dealing with the very specific task of human handover. It included, grasping and action recognition from both giver and receiver perspectives.

4. CONCLUSION AND FUTURE WORK

In this study, multiple architectures for MDHHC model to make successful handover classification were proposed. Architectures were divided into unimodal and multimodal architectures, and the concept of multimodality proved efficiency in the task of handover classification. Along the same line, embedded feature selection techniques proved to have high influence on the model performance. All of the different proposed architectures did not take into consideration the handover action from only one perspective like previous works, but rather, from all perspectives. The next step is to work more on this model to deal with few samples of data and missing data readings; an endeavor to ameliorate the model to classify more deeply other classes which seem to have almost the same input readings.

REFERENCES

[1] R. González-Ibáñez, A. Esparza-Villamán, J. C. Vargas-Godoy, and C. Shah, “A comparison of unimodal and multimodal models for implicit detection of relevance in interactive IR,” *Journal of the Association for Information Science and Technology*, vol. 70, no. 11, pp. 1223–1235, Apr. 2019, doi: 10.1002/asi.24202.

[2] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, “Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2022, doi: 10.1145/3447744.

[3] H. Tian, Y. Tao, S. Pouyanfar, S.-C. Chen, and M.-L. Shyu, “Multimodal deep representation learning for video classification,” *World Wide Web*, vol. 22, no. 3, pp. 1325–1341, May. 2019, doi: 10.1007/s11280-018-0548-3.

[4] T. Zhao, “Deep multimodal learning: An effective method for video classification,” in *2019 IEEE International Conference on Web Services (ICWS)*, pp. 398-402, 2019, doi:10.1109/ICWS.2019.00071.

[5] V. Satish, J. Mahler and K. Goldberg, "On-Policy Dataset Synthesis for Learning Robot Grasping Policies Using Fully Convolutional Deep Networks," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357-1364, April 2019, doi: 10.1109/LRA.2019.2895878.

[6] J. Zhang, M. Li, Y. Feng, and C. Yang, “Robotic grasp detection based on image processing and random forest,” *Multimedia Tools and Applications*, vol. 79, no. 3–4, pp. 2427–2446, 2020, doi:10.1007/s11042-019-08302-9.

[7] C. D. Santina *et al*., "Learning From Humans How to Grasp: A Data-Driven Architecture for Autonomous Grasping With Anthropomorphic Soft Hands," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1533-1540, April 2019, doi: 10.1109/LRA.2019.2896485.

[8] C. Gabellieri *et al*., "Grasp It Like a Pro: Grasp of Unknown Objects With Robotic Hands Based on Skilled Human Expertise," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2808-2815, April 2020, doi: 10.1109/LRA.2020.2974391.

[9] V. Arapi *et al.*, “To grasp or not to grasp: an end-to end deep-learning approach for predicting grasping failures in soft hands,” *IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 653-660, doi:10.1109/RoboSoft48309.2020.9116041.

[10] L. Bergamini, M. Sposato, M. Pellicciari, M. Peruzzini, S. Calderara, and J. Schmidt, “Deep learning-based method for vision-guided robotic grasping of unknown objects,” *Advanced Engineering Informatics*, vol. 44, no. 101052, pp. 101052, Apr. 2020, doi: 10.1016/j.aei.2020.101052.

[11] M. K. Pan, V. Skjervøy, W. P. Chan, M. Inaba, and E. A. Croft, “Automated detection of handovers using kinematic features,” *The International Journal of Robotics Research*, vol. 36, no. 5–7, pp. 721–738, 2017, doi: 10.1177/0278364917692865.

[12] W. Yang, C. Paxton, M. Cakmak, and D. Fox, “Human grasp classification for reactive human-to-robot handovers,” arXiv [cs.RO], 2020, doi:10.1109/IROS45743.2020.9341004.

[13] E. D-L-H-Franco, P. A-Colpas, J. M. Quero, and M. Espinilla, “Sensor-based datasets for human activity recognition – A systematic review of literature,” *IEEE Access*, vol. 6, pp. 59192–59210, 2018, doi:10.1109/ACCESS.2018.2873502.

[14] L. Wang and R. Liu, “Human activity recognition based on wearable sensor using hierarchical deep LSTM networks,” *Circuits, Systems, and Signal Processing,* vol. 39, no. 2, pp. 837–856, Feb. 2020, doi:10.1007/S00034-019-01116-Y.

[15] Y. Tang, Q. Teng, L. Zhang, F. Min and J. He, "Layer-Wise Training Convolutional Neural Networks With Smaller Filters for Human Activity Recognition Using Wearable Sensors," in *IEEE Sensors Journal*, vol. 21, no. 1, pp. 581-592, 1 Jan.1, 2021, doi: 10.1109/JSEN.2020.3015521.

[16] S. Chung, J. Lim, K. J. Noh, G. Kim, and H. Jeong, “Sensor data acquisition and multimodal sensor fusion for Human Activity Recognition using deep learning,” *Sensors,* vol. 19, no. 7, pp. 1716, Apr. 2019, doi: 10.3390/s19071716.

[17] H. A. Javaid *et al*., “Classification of hand movements using MYO armband on an embedded platform,” *Electronics,* vol. 10, no. 11, pp. 1322, May. 2021, doi: 10.3390/electronics10111322.

[18] O. Nafea, W. Abdul, G. Muhammad, and M. Alsulaiman, “Sensor-based human activity recognition with spatio-temporal deep learning,” *Sensors (Basel),* vol. 21, no. 6, pp. 2141, Mar. 2021, doi:10.3390/s21062141.

[19] A. Sen and K. Deb, “Categorization of actions in soccer videos using a combination of transfer learning and Gated Recurrent Unit,” *ICT Express*, 2021, doi: 10.1016/j.icte.2021.03.004.

[20] L. E. Raileanu, and K. Stoffel, “Theoritical comparison between the gini index and information gain criteria,” *Annals of Mathematics and Artificial Intelligence,* vol. 41, no. 1, pp. 77-93, 2004, doi:10.1023/B:AMAI.0000018580.96245.c6.

[21] A. Mangal and E. A. Holm, “A comparative study of feature selection methods for stress hotspot classification in materials,” *Integrating Materials and Manufacturing Innovation,* vol. 7, no. 3, pp. 87-95, 2018, doi: 10.1007/s40192-018-0109-8.

[22] P. K. Theodoridis and D. C. Gkikas, “Optimal feature selection for decision trees induction using a genetic algorithm wrapper - A model approach,” in *Strategic Innovative Marketing and Tourism, Cham: Springer International Publishing*, 2020, pp. 583–591, doi: 10.1007/978-3-030-36126-6_65.

[23] A. Sarica, A. Cerasa, and A. Quattrone, “Random Forest algorithm for the classification of neuroimaging data in Alzheimer’s disease: A systematic review,” *Frontiers in aging neuroscience,* vol. 9, pp. 329, Oct. 2017, doi: 10.3389/fnagi.2017.00329.

[24] Y. R. Pandeya and J. Lee, "Deep learning-based late fusion of multimodal information for emotion classification of music video," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 2887–2905, 2021, doi: 10.1007/s11042-020-08836-3.

[25] Z. Kun, W. W. Yong, H. Teng, and W. C. Huang, "Comparison of Time series forecasting based on statistical ARIMA model and LSTM with attention mechanism," *Journal of Physics: Conference Series*, vol. 1631, no. 1, pp. 012141, Sep. 2020, doi:10.1088/1742-6596/1631/1/012141.

[26] A. Carfì, F. Foglino, B. Bruno, and F. Mastrogiovanni, "A multi-sensor dataset of human-human handover," *Data Brief*, vol. 22, pp. 109–117, Feb. 2019, doi: 10.1016/j.dib.2018.11.110.

[27] M. Feurer and F. Hutter, "Hyperparameter Optimization," *in Automated Machine Learning, Cham: Springer International Publishing*, 2019, pp. 3–33, doi: 10.1007/978-3-030-05318-51.

[28] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," arXiv [cs.LG], 2018.

[29] M. E-Razzaz, M. W. Fakhr, and F. A. Maghraby, "Arabic gloss WSD using BERT," *Arabic gloss WSD using BERT*, vol. 11, no. 6, pp. 2567, Mar. 2021, doi: 10.3390/app11062567.

BIOGRAPHIES OF AUTHORS

Islam A. Monir a Computer Engineer graduate from the Arab Academy for Science and Technology and Maritime Transport College of Engineering and Technology in 2018. Since then he has been a Teaching Assistant at College of Computing and Information Technology. He is currently an MSc student doing researches in the field of machine learning and deep learning. He can be contacted at email: islammonir@aast.edu

Mohamed W. Fakhr finished his Ph.D. at the University of Waterloo, Canada, 1993, in the field of neural networks and machine learning; he then joined the speech research lab at NORTEL, Montreal, Canada, for 5 years where he was a researcher investigating and implementing different speech processing, speech recognition, language modeling, and statistical error analysis techniques and has 2 patents with NORTEL. Since 1999 he has been a professor with the Arab academy for science and technology (Cairo, Egypt) with 3 years sabbatical at the University of Bahrain. He has been doing research in the areas of time series forecasting, deep neural networks, natural language processing and privacy-preserving computing. He can be contacted at email: waleedf@aast.edu

Nashwa El-Bendary received the Ph.D. degree in information technology from the Faculty of Computers and Artificial Intelligence, Cairo University, Egypt, in 2008. She is currently a Full Professor with the College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport (AASTMT), Egypt. Her publication history spans more than 80 publications in reputed international journal articles, conference papers, and book chapters, and several international research projects and numerous plenary talks at flagship venues. Her research interests include machine learning and deep learning, NLP, and image processing. She has received several recognition, including the UNESCO-ALECSO Award for creativity and technical innovation for young researchers, in 2014, and the L'Oréal-UNESCO for women in science fellowship, in 2015. She currently serves as an Editorial Board Member for the Applied Soft Computing journal (Elsevier) and also she is a Senior Member in IEEE. She can be contacted at email: nashwa.elbendary@aast.edu