A comparison of Arabic sign language dynamic gesture recognition models

Miada A. Almasre*, Hana Al-Nuaim

Department of Computer Science, Faculty of Computing and Information Technology, King AbdulAziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Keywords: Computer science Dynamic gesture recognition models Machine learning Arabic sign language Classification

ABSTRACT

Arabic Sign Language (ArSL) is similar to other sign languages in terms of the way it is gestured and interpreted and used as a medium of communication among the hearing-impaired and the communities in which they live in. Research investigating sensor utilization and natural user interfaces to facilitate ArSL recognition and interpretation, is lacking.

Previous research has demonstrated that there is not a single classifier modeling approach that can be suitable for all hand gesture recognition tasks, therefore, this research investigated which combination of algorithms, set with different parameters used with a sensor device, produce higher ArSL recognition accuracy results in a gesture recognition system.

This research proposed a dynamic prototype model (DPM) using Kinect as a sensor to recognize certain ArSL gestured dynamic words. The DPM used eleven predictive models of three algorithms (SVM, RF, KNN) based on different parameter settings. Research findings indicated that highest recognition accuracy rates for the dynamic words gestured were achieved by the SVM models, with linear kernel and cost parameter \( c = 0.035 \).

1. Introduction

Sign language (SL) is the main method of communication among the hearing-impaired or deaf individuals and members of their community who interact with them. Definitely, there are alternative forms of communication, however, these methods may not support natural interaction, where the receiver of the communicative message does not have to learn sign language or any other method of communication.

It is here that assistive technologies become a viable solution. Technologies that support natural interaction with/among users can be utilized to assist in the hearing-impaired or deaf individuals’ successful social integration and to communicate with others in natural ways.

Assistive solutions which can interpret SL and translate it into verbal or textual cues is required to facilitate the inclusion of the hearing-impaired or deaf individuals in society. Nonetheless, and in spite of the growing research in this area, there is still a persistent need to develop efficient and easy to use assistive devices and specialized SL interpreters that perform such tasks.

Arabic Sign Language (ArSL) research acknowledges the demand for developing technologies which can be used to address the needs of the hearing-impaired or deaf people in Arabic communities as well as target the creation of inclusive environments for persons suffering from hearing loss. Various social institutions like health, education, transportation, etc. encourage and support this kind of research in Arabic countries, but the research outcomes reported until now leave much to be desired especially regarding the reliability and practicality of the developed solutions which eventually are rarely being adopted in real life situations. For example, one of the challenges for the deaf seeking the completion of their higher education is the absence of general admission to most university majors due to the small number of specialized solutions (interpreters, for example) that facilities their communication and effective inclusion.

One of the most prominent approaches in research addressing hearing-impaired or deaf individuals’ inclusion in society makes use of technologies with natural interfaces that can be utilized as SL recognition systems, which incorporate advancements in gesture recognition, depth sensors, and machine learning algorithms.

Hand gestures in SL use the palms, finger positions and shapes to generate forms that refer to different letters and phrases with different meanings in different cultural contexts [1,2]. Hence, gesture recognition refers to recognizing meaningful expressions of a motion produced by a human, including the hands, face, head, and body. Basically, gesture recognition is essential in application development ranging from ones used to interpret sign language to applications in the field of virtual reality [3]. More specifically, there are two types of hand gestures: time independent static hand gestures are those in which the hand position

---

* Corresponding author.
E-mail address: malmasre@kau.edu.sa (M.A. Almasre).

https://doi.org/10.1016/j.heliyon.2020.e03554

Received 2 April 2019; Received in revised form 3 August 2019; Accepted 3 March 2020

© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
does not change during the gesturing period, and time dependent dy-
amic hand gestures, where the hand position changes continuously with
respect to time [4].

In SL recognition systems that rely on natural interaction, depth
sensors are a requirement. Depth sensors provide essential data about
every object near the device, which will help in extracting many of the
user body's features, such as the neckline, thumb and index fingers. In
real life, hand and finger recognition needs extra extraction of features
and uses complex procedures to achieve accurate gesture recognition
using machine learning approaches [5].

In such recognition systems, machine learning algorithms are used to
translate SL based on human a gesture recognition process, which passes
through a human detection phase and a gesture recognition or classifi-
cation phase. Moreover, in the science of statistics and machine learning,
basic learning algorithms that work on part of the data do not provide
superior predictive performance than the ensemble methods that use
several learning algorithms. The main reason for combining models of
learning algorithms is their collaborations, where all models break down
to capturing the high dimensional data in a more profound comprehen-

dion [6,7].

The remaining content of this article is organized as follows: Section 2
presents the research objective, followed by Section 3, which introduces
a literature review of some of the related works. Then, a thorough
description of the gesture recognition pipeline is provided in Section 4.
Section 5 details the proposed model (DPM). Section 6 highlights the
results and discussion of important findings. Finally, Section 7 reiterates
the importance of the study, the results and suggestions for future work.

2. Research objective

Previous research has shown that there is no single classifier
modeling approach that is suitable for all gesture recognition tasks,
which leads to a need for developing a system using a multi classifica-
tion model [8]. Therefore, this paper's objective is to achieve higher ArSL
cognition accuracy results in live settings by comparing various combi-

gations of algorithms, set with different parameters, used with a sensor
device. The model of the system proposed could be the backbone of a
gesture recognition system. The researchers believe that the use of
ensemble algorithms can result in better recognition decisions; espe-
cially, with sensor devices (Kinect, for example) which have an average
speed in capturing human movement. Therefore, to improve the recog-
nition process, we recommend the implementation of a stacked
algorithm.

3. Review of literature

Hand posture plays an important role in gesture recognition systems.
It is defined as the pose or configuration of the hand in a single image,
whether static or dynamic, that is used to implement commands such as
selection, navigation, and image manipulation with specific movements
that the application requires [6,8].

Mittal, Kumar, Roy, Balasubramanian, & Chaudhuri (2019) proposed a
modified long short-term memory (LSTM) model for continuous SLR in
order to identify connected signs. The model split the continuous ges-
tures into smaller units to be modeled with the neural networks. The
researchers chose 942 sentences and 35 sign words from the Indian Sign
Language (ISL) to test their model. The results showed an average ac-
curacy of 72.3% for the sentences, and 89.5% for the individual words
[9].

Vo, Huynh, Doan, & Meunier (2017) presented an approach to extract
features and classify the continuous dynamic gestures of the Vietnamese
Sign Language (VSL). Microsoft Kinect depth sensor was used to capture
the data. They employed Support Vector Machine (SVM) algorithm in
their work, along with Hidden Markov Model (HMM) technique to
calculate the recognition accuracy. The approach was tested on 3000
samples formed of 30 dynamic gestures, and the results showed an
average accuracy of 95% [10].

Varshini and Vidhyapathi (2016) proposed a dynamic finger-gesture
recognition system using the color camera of Microsoft’s Kinect, with the
clustering algorithm K-means that clusters the pixels into a specific range.
They also applied a trigger based dynamic action recognition to recog-
nize the single hand gesture [11].

D’Orazio et al., (2016) discussed the recent trends in gesture recog-
nition, and how depth data has improved classical approaches. They also
analyzed the perspective of the recent state-of-the-art; RGB-Depth (RGB-
D) color-depth images as a new gesture recognition approach [12]. They
explored the Support Vector Machine (SVM) classification algorithm and
highlighted the use of angles, directions, and orientation which were the
most recently used features in the field of processing of RGB-D data for
gesture recognition [12].

Halim & Abbas (2015) proposed a system that processes images and
recognizes patterns to detect and translate sign language into speech by
using a custom-built software tool. A Dynamic Time Warping (DTW)
algorithm was employed to recognize certain gestures, an off-the-shelf
software tool was used to generate the spoken language, and video
streams were recorded via Microsoft Kinect tool. The proposed system
succeeded in detecting gestures with 91% accuracy. They conducted an

![Figure 1. Gesture recognition pipeline.](image-url)
Experiment in which they establish a communication between people with hearing impairments and others without any disabilities; and 87% of the sample found it helpful [8].

In addition, Rehman, Halim, and Ahmad (2014), presented an idea of designing an ArSL interface for desktop application applied for Urdu language which belongs to the family of ArSLs. The American Standard Code for Information Interchange (ASCII) codes used for mapping the keystrokes to the Urdu character's images using Phonetic keyboard styles. Their proposed modules provide entities for applications like Urdu translation system and other interface based software [13].

Among the important requirements for hand detection, and ultimately for recognition systems, are the sensors which track human movement. Nevertheless, one has to consider that the depth sensors' current movement tracking have a constrained zone of tracking and recognition, depending on natural lighting, lens occlusion (a special lens that removes visible light below 820 nm from being read), and hardware factors [14].

Pisharady and Saerbeck (2015) defined depth data as a position determined by the three coordinates (X,Y,Z) localizing a point corresponding to an object in 3D space. This is the main object feature retrieved by the depth sensor. Furthermore, depth sensors retrieve data about each object close in distance to it, which improves gesture recognition beyond the limit of standard approaches based only on colorful images [4].

Most of the research papers discussed previously follow similar steps to recognize SL gestures, so the following section will explain the most popular process pipeline and all technical terminologies that the researchers used for this paper.

4. Gesture recognition process

Recognition systems usually extract numerous values from the data, called features, and use complex procedures to achieve accurate gesture recognition [1]. Following the extraction of the features step, classification becomes an important requirement. Classification can be achieved by using either one algorithm or a combination of multiple algorithms, referred to as an ensemble, such as boosting, stacking and bagging [6].

Hence, recognizing and interpreting sign-language gestures in a sequential process goes through three steps: Figure 1-A presents a generic pipeline for a gesture recognition process, where data is input (via sensors), process of selecting certain data from the input through (feature extraction), and finally the output with classification Figure 1-A [4]. These are the steps which the researcher adapts and follows in the proposed model (Figure 1-B). They will be explained in detail in Section 5 (The Dynamic Prototype Model).

4.1. Sensors

The hardware component which is mainly used in this model is Kinect. It is considered a depth sensor which is usually used by other researchers to capture crucial data about each human close in distance to the sensors, by extracting many of the user's hand and body features. Advances in 3D depth cameras that were utilized by Kinect have produced many opportunities for multimedia computing [15]. Kinect relies on depth technology, which allows users to deal with any system via a web camera [2,16].

Kinect has a wide sensing range that can track the complete user's skeleton and detect two fingertips from each hand and 20 joints from the whole body [17]. The SDK Kinect class library converts the position point from the regular camera view to the depth camera view (which are considered the points in the real world). Figure 2 shows the Kinect field of view range, where Kinect converts each position point from Camera-Space (X, Y) to Depth-Space (X, Y,Z) with a device range view of 75° [18].

4.2. Feature extraction

Selecting data or features is crucial for gesture recognition, since body gestures are very rich in shape and motion variations [19]. There are many features that are retrieved from sensor devices, but it is useless to use them all in a recognition system. So, these features should be either reduced or only important values could be extracted from them to reduce the data processing time [20]. Data interpolation in feature extraction is a method of constructing new data points within the range of a discrete set of known data points such as piecewise constant interpolation, linear interpolation, polynomial interpolation, and spline interpolation [21].

4.3. Classification

Classification is the task of learning a target function (classification model) that maps each attribute set X to one of the predefined class labels (Y). A classifier is a systematic approach for building classification models from an input data set.

The main goal of the classification process is to predict a category of a new observation using earlier observations through a classification model [22]. In classification, usually one of the two following methods is used:

- Hold-out, where the mostly large dataset is divided to three subsets: a training set, validation set and testing set, yet most researchers utilize only training and testing sets [12]. The main function of the training set is to build a predictive model, then the testing set asesses the performance of this predictive model.
- Cross-Validation (CV), which is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one is used to train a model and the other is used to validate it [23]. In typical cross-validation, the training and validation sets must cross-over in successive rounds, so that each data point gets a chance of being validated against [23]. The basic form of cross-validation is k-fold cross-validation, the data is first partitioned into k equally or nearly equally sized segments or folds [23].

Each method employs a learning algorithm to identify a model that best fits the relationship between the attribute set and the class label of the input data. A key objective of the learning algorithm is to build models with good generalization capabilities, i.e. models that accurately predict the class labels of previously unknown records [24].

Classification machine learning algorithms are used to fit a predictive model to data. The top ten data mining algorithms identified and well researched by the IEEE International Conference on Data Mining (ICDM) are C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, KNN, Naïve Bayes, and CART [25].

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain a better predictive performance than what could be obtained from any of the constituent learning algorithms [7]. In addition, the overall meaning of ensembling (combination) is to combine the predictions of multiple different models together [26]. There are
many reasons to ensemble models, but it usually comes down to capturing a deeper understanding of high dimensional data [26].

The three most popular methods for combining predictions from different models are bagging, boosting, and stacking. Stacking trains a learning algorithm by combining the predictions of several other learning algorithms. It is carried out in two steps: 1) training using the original data; 2) training using a suitable resampling method, such as: sum, maximum, minimum, and weighted majority voting of the predictions generated by the other algorithms, as extra inputs [27].

One important part of the recognition systems is the evaluation of the classification algorithm. It is a vital step to determine which classification algorithm is suitable to represent the data, and how well this chosen algorithm will work in the future (Sayad, 2010). Statistical models that are based on the values of Accuracy, Area Under Curve (AUC), and logarithmic loss (logLoss) can be used to evaluate the model [16].

5. The Dynamic Prototype Model (DPM)

The prototype software which was developed for this research connected a sensor (Kinect) with an easy to use application interface, developed by the researcher using Visual Studio 2013 with C#, to capture the data of the human body (dataset) and transfer it to a personal computer. The users were interacting with the interface while they were standing in front of the sensor device in order to gesture the words (Figure 3).

The general implementation of the DPM model structure that was used in this research consisted of three main steps, which constituted the gesture recognition pipeline described previously. The steps were:

1. Choosing the input device: A Kinect sensor was used to capture the ArSL words that were gestured by the participants. Each gestured word was considered a class.
2. Feature extraction: The bone directions and joint angles were the main features used to describe gestures. These values were calculated from the data which was retrieved by the sensors. These values formed a dataset in which each gesture was classified into a certain class.
3. Classification: Each gestured sign was assigned to a class using three of the most commonly used classification algorithms (SVM, RF, KNN). These algorithms were utilized to examine and classify gestured signs with less processing time while achieving a high level of accuracy.

Dynamic gestures in sign language stand for a word or a phrase; however, for the purposes of this research, dynamic gestures are only considered in the context of single words. Although the prototype could capture and save phrases as dynamic gestures without changing the technical requirements or recognition process, it demanded more advanced linguistic processing; which was outside the scope of this research.

Since some participants were able to gesture faster than others, some of the gestures spanned more frames than others. To address this issue; pre-processing methods were followed for representing gesture values.
The researchers used Caret, which has an interface to call multi-classification packages that were used in this research (SVM, RF, KNN) as well as an evaluation metric such as logLoss, ROC, and AUC.

A number of steps have been followed to prepare the data which are: data interpolation, feature representation and deleting redundancy and null values. Then, DPM tested the accuracy of the stacking ensemble by adding features using the predictive models of the three algorithms SVM, KNN, and RF on the dataset of the dynamic words.

5.1. The experiment setup

The setup of the experiment entailed arranging the environment of the sensor with application that captured the data, determining the participants’ tasks, and monitoring the dataset collection.

The experiment’s environment was a typical room, a 4 × 6 space with normal Florence light. The Kinect was mounted on a table of a 100 cm height from the floor (Figure 4). Although participants could stand away from Kinect in range of one to four meters while using it, the researcher asked them to stand two meters away when capturing dynamic gestures. This was needed to accurately capture dynamic gestures for all participants.

As for the ArSL dataset used for this study, it is important to note the unavailability of online ArSL datasets, which focus on image depth values and which can be reliable in terms of quality. Therefore, the researchers created their own dataset. It took one week to capture the required data based on the participants’ free time.

The dataset was collected from 10 non-deaf participants, who were able to sign a few words in ArSL. The recruitment of participants was through an internal broadcast for volunteers with clear details about the experiment and what will be required of each volunteer.

The ten participants were asked to gesture 6 selected dynamic words (classes) (Figure 5), where each participant stood in front of Kinect and gestured the same word one or more times. Table 1 illustrates the 222 observations that were collected for each word and the proportion of each sign in the dataset.

The researchers’ role was to monitor the participants’ interaction with the prototype, and click the save button, whenever a gesture was performed for two reasons: 1) the participants were using both of their hands during the gesturing process, 2) the researchers had to ensure that the participants did not perform extra gestures while the device was capturing the frames that might distort the gesture input (like the movement to click save). The result was a dataset of 222 observations.

The dynamic gesture dataset for the same gestured word had multi-frames, even if the same participant had made the same gesture. Each observation sequence was considered as a word with many features if it was captured from a specific participant. For example, Figure 6 illustrates the sequence of signing the phrase “شاعر” (celebrity), which is a sequence of 10 frames that represents one user’s observation.

The dataset was divided eventually into two or more sets. The two main sets existing in this division were a training and a testing set. The training set was used to train the recognition system on the meaning of data values, while the testing set was used to assess the performance of the recognition system, as to whether it would be capable of accurately recognizing new incoming gestures.

The three used algorithms were SVM, RF, and KNN. SVM is considered one of the most sophisticated algorithms; RF is one of the most widely used algorithms, while KNN is one of the easiest algorithms to implement (needs less computational time).
All three classification algorithms, which had been chosen for this research, had parameters that could be set with different values, which would affect the results. Therefore, the researcher explored each classification algorithm with different parameter settings to explore the recognition rate.

This research implemented SVM using two of the most commonly used kernels: SVM with linear kernel, and SVM with a radial kernel. In addition, SVM with a linear kernel has only one parameter to tune, which is the cost parameter. This parameter could be set to default (caret package has been used to choose the cost parameter) or tuned by values to see if a better performance could be achieved by tuning the cost parameter. On the other hand, SVM with a radial kernel has two parameters: cost and gamma. First, the models were trained by setting the caret package in R to choose different combinations of the two parameters (cost and gamma) to see which combination would yield a better performance.

Therefore, all parameters of the three classification algorithms were set twice: the first set of parameters were done with default values using caret package and the second set of parameters were done with tuning of a range of values set by the researchers. The default parameter would provide a good starting point from which tuned values could be used in order to assess which parameter setting would yield a better performance.

This approach created predictive models of the three algorithms. The top three predictive models, which had the highest training performance, were chosen for the ensemble approach. Therefore, three new predictive models appeared. For example, if the top three models were SVMLD, RFD, and SVMRT, the new three stacked models would be SVMLDS, RFDS, and SVMRTS – adding the suffix (-S).

The following is the list of abbreviations for the eleven predictive models of the three algorithms (SVM, RF, KNN) based on different parameter settings:

1. KNND: Nearest Neighbor with Default values
2. KNNT: Nearest Neighbor with Tuning values
3. SVMRD: Support Vector Machine with Radial kernel and Default values
4. SVMLD: Support Vector Machine with Linear kernel and Default values
5. SVMLT: Support Vector Machine with Linear kernel and Tuning values
6. SVMRT: Support Vector Machine with Radial kernel and Tuning values
7. RFT: Random Forest with Tuning values
8. RFD: Random Forest with Default values
9. SVMLDS: Support Vector Machine with Linear kernel, Default values and Stacking.
10. RFDS: Random Forest with Default values and Stacking.
11. SVMRTS: Support Vector Machine with Radial kernel, Tuning values and Stacking.

5.2. Smoothing gestures with interpolation

Interpolation has been used to make all signs span an equal number of frames (8 frames). Therefore, data interpolation in this research means equalizing the number of captured frames for the same gesture word. For example, if user1 gestured the dynamic word “بكتيريا” (Bacteria) within 5 frames and user 2 gestured the same word within 9 frames, data interpolation makes the number of frames = 8 for both.

The function ‘spline’ (which is used for non-linear interpolation) from R has been used to perform cubic interpolation of the data points. This function has been chosen over a simpler linear interpolator, like the ‘approx’ function, to imitate the natural hand movement of a human signer. Spline interpolation is smoother and more suitable for representing the natural hand movement. Therefore, the median value of frames across all gestures has been chosen to be the unifying number, which happened to be 8 frames.

5.3. Feature representation

Feature representation in dynamic gestures means the process of transforming dataset values into features that better represent the underlying problem of the predictive models, resulting in an improved category recognition accuracy of data in the testing set. So, the
algorithm’s success in predicting data in the testing set is represented by the features that the algorithm can understand. Since each 8 rows represent one observation, and machine learning classification algorithms are designed to deal with each row as a single observation, some feature representations are needed in order to represent each block of the 8 rows as a single row.

The researchers have chosen to represent each feature, which spanned 8 rows, with two statistical parameters: the mean, to capture the centrality, and the standard deviation, to capture the dispersion. Therefore, each original feature has been transformed into two features, and each 8 rows have been transformed into a single row.

5.4. Classification implementation for DPM

The dynamic words’ dataset was split into three parts: training set 1, training set 2, and a testing set. Then, the predictive models of the three algorithms (SVM, KNN, RF) were trained and evaluated individually, using training set 1 and the testing set. The results of the three algorithms were used to apply the ensemble stacking with added features. In particular, a 5-fold CV was used to train the predictive models, using training set 2, and to evaluate the performance also using the same three evaluation metrics (Accuracy, AUC, and logLoss). The top three categories were used to provide new features for the stacking ensemble.
Figure 9. Evaluation results of the 5-fold CV for SVMRD

Figure 10. Evaluation results of the 5-fold CV for SVMRT

Figure 11. Evaluation results of the 5-fold CV for KNND
Figure 12. Evaluation results of the 5-fold CV for KNN

Figure 13. Evaluation results of the 5-fold CV for RF

Figure 14. Evaluation results of the 5-fold CV for RFT
The classification implementation for the first predictive models used in DPM were SVM’s predictive models. Figure 7 and Figure 8 show the results of SVMLD and SVMLT respectively.

A slight improvement was noticed with the fine tuning of the cost parameter, as logLoss dropped from 0.7297 (in the default case) to 0.7031 (in the tuned case).

Figure 9 and Figure 10 illustrate the SVMRD and SVMRT evaluation results respectively.

When using a tuned value, where cost = 8 and gamma = 0.001, logLoss was reduced to 0.6704, compared to 0.8174 in SVMRD.

The classification implementation for the second predictive model used in DPM were KNN’s predictive models. First, KNN was set with a default parameter fit (Figure 11), then the k parameter was tuned (Figure 12).

The classification implementation for the third predictive model used in DPM were RF’s predictive models, with only one tuning parameter (mtry). Figure 13 and Figure 14 show the results of RFD and RFT respectively.

RFD results show that Accuracy has the highest value when choosing mtry = 2, while AUC and logLoss had mtry = 111. A slight improvement in RFT results was noticed due to the fine tuning.

5.4.3. New features for DPM

The statistical comparison between the DPM’s predictive models was based on their CV performance for each evaluation metric (Table 6).

The top three categories were chosen based on the mean logLoss on CV. These top three were used to provide new features for the stacking model.

Table 2. Testing set results for all individual predictive models.

| model    | logLoss | AUC   | Accuracy |
|----------|---------|-------|----------|
| SVMRT    | 0.5610002 | 0.9848406 | 0.7924528 |
| SVMRD    | 0.6339350 | 0.9773189  | 0.7924528 |
| RFD      | 0.7559546 | 0.9494649  | 0.7358491 |
| RFT      | 0.7662624 | 0.9480142  | 0.7358491 |
| SVMLD    | 0.7790614 | 0.9554407  | 0.7924528 |
| KNND     | 0.7981315 | 0.9379291  | 0.7169811 |
| KNNT     | 0.7981315 | 0.9379291  | 0.7169811 |
| SVMLT    | 0.8385668 | 0.9599782  | 0.7735849 |

Table 3. Individual class evaluation using LogLoss metric for dynamic words.

| class    | SVMLD | SVMLT | SVMRD | SVMRT | KNND | KNNT | RFD | RFT |
|----------|-------|-------|-------|-------|------|------|-----|-----|
| Bacteria | 0.2736 | 0.2458 | 0.2646 | 0.2296 | 0.327 | 0.327 | 0.3519 | 0.3468 |
| Celebrity | 0.2162 | 0.2429 | 0.224 | 0.208 | 0.2797 | 0.2797 | 0.2638 | 0.2704 |
| Common | 0.2264 | 0.2475 | 0.1438 | 0.1352 | 0.1882 | 0.1882 | 0.2433 | 0.242 |
| Disaster | 0.1651 | 0.179 | 0.1439 | 0.1556 | 0.2229 | 0.2229 | 0.1382 | 0.1405 |
| Protein | 0.3213 | 0.3668 | 0.1738 | 0.1494 | 0.1981 | 0.1981 | 0.1539 | 0.1645 |
| Stick | 0.0782 | 0.1017 | 0.1178 | 0.0908 | 0.1871 | 0.1871 | 0.0958 | 0.0939 |
ensemble. It has been observed that RFT, RFD, and SVMRT were the top three models (Table 6).

Moreover, plotting the distribution of each of the three-evaluation metrics showed the degree of variance that each category exerted (Figure 16). The box, such as SVM predictive models, that was more centred around its median value (the black point in the middle), the less variance it exerted; hence it would be more reliable when exposed to the data in the testing set, and vice versa.

Figure 16 and Figure 18 show box plots for the predictive models for the AUC and Accuracy metrics. The box of RFT predictive model is shorter and centred around its median value; hence it would be more reliable when exposed to the data in the testing set.

Table 4. Individual class evaluation using AUC metric for dynamic words.

| Class      | SVMLD  | SVMLT  | SVMRD  | SVMRT  | KNND  | KNNT  | RFD  | RFT  |
|------------|--------|--------|--------|--------|-------|-------|------|------|
| Bacteria   | 0.9481 | 0.9481 | 0.9524 | 0.9675 | 0.8615 | 0.8615 | 0.8615 | 0.8810 |
| Celebrity  | 0.9651 | 0.9535 | 0.9651 | 0.9698 | 0.9116 | 0.9116 | 0.9279 | 0.9233 |
| Common     | 0.9596 | 0.9627 | 0.9876 | 0.9907 | 0.9658 | 0.9658 | 0.9255 | 0.9130 |
| Disaster   | 0.9848 | 0.9874 | 0.9949 | 0.9949 | 0.9470 | 0.9470 | 1.0000 | 1.0000 |
| Protein    | 0.8750 | 0.8722 | 0.9667 | 0.9861 | 0.9556 | 0.9556 | 0.9819 | 0.9736 |
| Stick      | 1.0000 | 1.0000 | 0.9972 | 1.0000 | 0.9861 | 0.9861 | 1.0000 | 0.9972 |

Table 5. Individual class evaluation using accuracy metric for dynamic words.

| Class      | SVMLD  | SVMLT  | SVMRD  | SVMRT  | KNND  | KNNT  | RFD  | RFT  |
|------------|--------|--------|--------|--------|-------|-------|------|------|
| Bacteria   | 0.7608 | 0.8063 | 0.7944 | 0.7608 | 0.7944 | 0.7489 | 0.7489 | 0.7608 |
| Celebrity  | 0.8651 | 0.8151 | 0.8651 | 0.8535 | 0.6919 | 0.6919 | 0.8035 | 0.8151 |
| Common     | 0.8254 | 0.8245 | 0.8463 | 0.8463 | 0.7640 | 0.7640 | 0.7205 | 0.7205 |
| Disaster   | 0.9104 | 0.9104 | 0.9659 | 0.9545 | 0.8676 | 0.8763 | 0.9773 | 0.9669 |
| Protein    | 0.9264 | 0.8639 | 0.8639 | 0.8639 | 0.8528 | 0.8528 | 0.8750 | 0.8639 |
| Stick      | 0.9889 | 0.9889 | 0.9264 | 1.0000 | 0.8639 | 0.8639 | 0.9375 | 0.9375 |

Table 6. Statistics summary for all individual models: LogLoss metric.

|            | Min.   | 1st Qu. | Median | Mean  | 3rd Qu. | Max.   |
|------------|--------|---------|--------|-------|---------|--------|
| SVMLD      | 0.4856224 | 0.5663899 | 0.7479859 | 0.7296798 | 0.8916693 | 0.9567313 |
| SVMLT      | 0.4591032 | 0.5206693 | 0.7595137 | 0.7031278 | 0.8771076 | 0.8992454 |
| SVMRD      | 0.5998265 | 0.6081160 | 0.7877364 | 0.8173809 | 0.9877253 | 1.1035006 |
| SVMRT      | 0.4354733 | 0.4550325 | 0.7073525 | 0.6703756 | 0.8214575 | 0.9325619 |
| KNND       | 0.7370785 | 0.7802288 | 0.9786960 | 0.9037623 | 0.9894172 | 1.0333910 |
| KNNT       | 0.7370785 | 0.7802288 | 0.9786960 | 0.9037623 | 0.9894172 | 1.0333910 |
| RFD        | 0.4998879 | 0.5464180 | 0.7224055 | 0.6556990 | 0.7276816 | 0.7821022 |
| RFT        | 0.5095183 | 0.5635101 | 0.6927703 | 0.6460835 | 0.7248194 | 0.7397994 |

Figure 16. Distribution of the logLoss Evaluation Metric.

Figure 17. Distribution of the AUC evaluation metric.

Figure 18. Distribution of the accuracy evaluation metric.
The performance of DPM was evaluated, with and without stacking, by comparing the individual and stacked predictive models (overall and individual classes).

Figure 19 and Table 7 illustrate that each stacked predictive category has surpassed its individual counterpart. Moreover, the performance of each predictive category on an individual class basis was applied (Table 8, Table 9, and Table 10).

### 5.4.4. Training and evaluating a stacking ensemble of dynamic words

The performance of DPM was evaluated, with and without stacking, by comparing the individual and stacked predictive models (overall and individual classes).

Figure 19 and Table 7 illustrate that each stacked predictive category has surpassed its individual counterpart.

Moreover, the performance of each predictive category on an individual class basis was applied (Table 8, Table 9, and Table 10).

### 6. Data interpretation of DPM’s results

- The stacking ensemble models had higher recognition accuracy rates than its individual counterpart. For example, SVMRTS had higher recognition accuracy rates than SVMRT. In addition, RFTS had higher recognition accuracy rates than RFT. However, it is not absolute that all stacking ensemble models outperformed an individual category. For example, SVMRT had higher recognition accuracy rates than RFTS.

- In SVMLT, a slight improvement was noticed with the fine tuning of the cost parameter, as logLoss dropped from 0.7297 (using the default parameter values) to 0.7031 (in the tuned case).

- The top three models, according to their mean logLoss statistics, were: RFT, RFD, and SVMRT. However, SVM’s models outperformed other models on the testing set.

- The highest variance was noticed among SVM models. On the other hand, the RF models had a moderate degree of variance, while the variance degree of KNN models was midway between SVM and RF.

- Overall, the RF and SVM models had higher recognition accuracy rates than KNN’s models because, on average, they had less logLoss value.

- Two gestured words have been predicted by the models with a very high degree of recognition accuracy, "ﻖﺼﺘﻟﺍ" (Stick) achieved 100% and "ﺀﻼﺘﺑﺍ" (Disaster) achieved 97.73%, while the two words, "بكتريا" (Bacteria) achieved 74.89% and "تشكيله" (Celebrity) achieved 69.19% were significantly lower. That could be due to the

| Table 7. Comparing individual and stacked models (overall). |
|--------------------------|--------------------------|--------------------------|
| model                   | logLoss                  | AUC                      | Accuracy                  |
| SVMRTS                  | 0.5566221                | 0.9823148                | 0.8301887                 |
| SVMRT                   | 0.5610002                | 0.9848406                | 0.7924528                 |
| SVMRD                   | 0.6339350                | 0.973189                 | 0.7924528                 |
| RFTS                    | 0.7318362                | 0.9502115                | 0.7358491                 |
| RFD                     | 0.7559546                | 0.9499948                | 0.7358491                 |
| RFT                     | 0.7662624                | 0.9480142                | 0.7358491                 |
| SVMLD                   | 0.7790614                | 0.9554407                | 0.7924528                 |
| KNND                    | 0.7981315                | 0.9379291                | 0.6792453                 |
| KNNT                    | 0.7981315                | 0.9379291                | 0.6981132                 |
| SVMLT                   | 0.8385668                | 0.9539782                | 0.7735849                 |

| Table 8. Class comparison using LogLoss for dynamic words. |
|--------------------------|--------------------------|--------------------------|
|                          | SVMLD | SVMLT | SVMRD | SVMRT | KNND | KNNT | RFD | RFT | SVMRTS | RFTS | RFDS |
| Bacteria                 | 0.2736 | 0.2458 | 0.2646 | 0.2296 | 0.327 | 0.327 | 0.3519 | 0.3468 | 0.2282 | 0.328 | 0.3306 |
| Celebrity                | 0.2162 | 0.2429 | 0.224 | 0.208 | 0.2797 | 0.2797 | 0.2638 | 0.2704 | 0.2073 | 0.2695 | 0.2715 |
| Common                   | 0.2264 | 0.2475 | 0.1438 | 0.1352 | 0.1882 | 0.1882 | 0.2433 | 0.242 | 0.1351 | 0.2411 | 0.2544 |
| Disaster                 | 0.1651 | 0.179 | 0.1639 | 0.1556 | 0.2229 | 0.2229 | 0.1382 | 0.1405 | 0.1463 | 0.1344 | 0.1313 |
| Protein                  | 0.3213 | 0.3668 | 0.1738 | 0.1494 | 0.1981 | 0.1981 | 0.1539 | 0.1645 | 0.1519 | 0.1609 | 0.1631 |
| Stick                    | 0.0782 | 0.1017 | 0.1178 | 0.0908 | 0.1871 | 0.1871 | 0.0958 | 0.0939 | 0.0909 | 0.0934 | 0.0916 |
participants signing these words in a similar manner (not much discrepancy).

- The box plots in Figure 16, Figure 17, and Figure 18 illustrate that:
  - The SVM models show the highest degree of variance among the eight models.
  - The RF models show compact, although not quite centred distributions, which means they have a moderate degree of variance.
  - The KNN models show a degree of variance that is ‘narrower’ than the SVM models, but wider than RF, because their boxes are shorter than the SVM’s, but longer than the RF’s.

When analyzing box plots in general, many factors affect the recognition accuracy of each category, some are attributed to the design structure (how complex the model implementation structure is) and others are attributed to the algorithm being used with the model.

To simplify, Table 11 presents comparisons of algorithms models’ results with the mean values achieved by each algorithm when applied in different models for predicting the testing dataset. The three-evaluation metrics used in this research were (Minimum logLoss, Maximum AUC, Maximum Accuracy). In general, Table 11, indicates that evaluating algorithm performance relies on the agreement between at least two metrics of the three that represent the highest value, as indicated by the shaded cells in each category. Table 11 shows that among the 11 results for dynamic words, the highest recognition rates were achieved by the SVM models. The researchers had previously reviewed related research, which used Kinect and a number of classification algorithms; however, the focus was not on ArSL, and the methodology and experiment processes used, were run in different ways [8–10]. Considering the results of our current research, as demonstrated in Table 11, one notes that the algorithm we implemented (SVMRTS) has achieved an accuracy value of 83% higher than the accuracy rates achieved by the researchers in [9]; nonetheless, lower than that which is demonstrated in [8,10].

### 7. Conclusion

Technology plays an important role in enhancing communication with the deaf, yet the available solutions lack accuracy and speed, which make them difficult to implement in real life applications to improve their quality of life. Several combinations of a depth sensor and algorithms set with different parameters were investigated for recognition accuracy of ArSL.

Although the recognition accuracy depends on the size of the dataset in general, the SVM algorithm set to radial kernel and tuned parameters achieved the highest recognition accuracy rates when using Kinect. In

---

### Table 9. Class comparison using AUC for dynamic words.

|        | Bacteria | Celebrity | Common | Disaster | Protein | Stick |
|--------|----------|-----------|--------|----------|---------|-------|
| SVMLD  | 0.9481   | 0.9651    | 0.9596 | 0.9484   | 0.8750  | 1.000 |
| SVMLT  | 0.9481   | 0.9535    | 0.9627 | 0.9874   | 0.8722  | 1.000 |
| SVMRD  | 0.9524   | 0.9651    | 0.9876 | 0.9949   | 0.9667  | 1.000 |
| SVMRT  | 0.9675   | 0.9698    | 0.9907 | 0.9949   | 0.9667  | 1.000 |
| KNND   | 0.8615   | 0.9116    | 0.9658 | 0.9470   | 0.9556  | 0.9861|
| KNNT   | 0.8615   | 0.9279    | 0.9685 | 0.9470   | 0.9556  | 0.9861|
| RFD    | 0.8810   | 0.9233    | 0.9255 | 0.9470   | 0.9556  | 0.9861|
| RFT    | 0.9610   | 0.9209    | 0.9130 | 0.9470   | 0.9556  | 0.9861|
| SVMRTS | 0.8712   | 0.9286    | 0.9876 | 1.0000   | 0.9861  | 0.9806|
| RFTS   |          |           |        |          |         | 1.000 |
| RFDS   |          |           |        |          |         | 1.000 |

### Table 10. Class comparison using accuracy for dynamic words.

|        | Bacteria | Celebrity | Common | Disaster | Protein | Stick |
|--------|----------|-----------|--------|----------|---------|-------|
| SVMLD  | 0.7608   | 0.8651    | 0.8254 | 0.9104   | 0.9264  | 0.9889|
| SVMLT  | 0.8063   | 0.8151    | 0.8463 | 0.9104   | 0.8639  | 0.9889|
| SVMRD  | 0.7944   | 0.8525    | 0.8463 | 0.9545   | 0.8639  | 0.9364|
| SVMRT  | 0.7608   | 0.6919    | 0.7640 | 0.8434   | 0.8639  | 0.9264|
| KNND   | 0.7706   | 0.7419    | 0.7640 | 0.8434   | 0.8639  | 0.9364|
| KNNT   | 0.7489   | 0.8035    | 0.7205 | 0.9773   | 0.8639  | 0.9364|
| RFD    | 0.7608   | 0.8151    | 0.7205 | 0.9773   | 0.8639  | 0.9364|
| RFT    | 0.8517   | 0.8651    | 0.8463 | 0.9659   | 0.8639  | 0.9364|
| RFTS   | 0.7489   | 0.8651    | 0.8463 | 0.9659   | 0.8639  | 0.9364|
| SVMRTS |          |           |        |          |         | 1.000 |
| SVMRTS |          |           |        |          |         | 1.000 |

### Table 11. Summary based on algorithm.

|       | Min of logLoss | Max of AUC | Max of Accuracy |
|-------|----------------|------------|-----------------|
| KNND  | 0.798132       | 0.9379291  | 67.92453%       |
| KNNT  | 0.798132       | 0.9379291  | 69.81132%       |
| RFD   | 0.755955       | 0.9494649  | 73.58491%       |
| RFDS  | 0.750522       | 0.9499948  | 73.58491%       |
| RFT   | 0.766262       | 0.9480142  | 73.58491%       |
| RFTS  | 0.731836       | 0.9502116  | 73.58491%       |
| SVMLD | 0.779061       | 0.9554407  | 79.24528%       |
| SVMLT | 0.838567       | 0.9539782  | 77.35849%       |
| SVMRD | 0.633935       | 0.9773189  | 79.24528%       |
| SVMRT | 0.561000       | 0.9848406  | 79.24528%       |
| SVMRTS| 0.556622       | 0.9823148  | 83.01887%       |
addition, using the SVM algorithm is sufficient to classify gestures with selected features by using angles and bone orientation. If the dataset is small, then using a 5-Fold CV to train the classification model yields higher recognition rates. In conclusion, evaluation of the algorithms’ performance using more than one metric is essential to compare the recognition accuracy rates.

A main contribution of this paper was the developed software prototype to capture and manage the dataset collected from the participants’ gestures, which could also be used by other researchers in the field, to achieve higher ArSL recognition rates.

The dynamic gesture pose estimation is still an active field of research, and the model designed to extract the features which were used in this research prototype can be enhanced. Therefore, future work could focus on developing algorithms that are able to produce higher accuracy rates, such as using deep learning algorithms after adding more observations to the dataset.

Declarations

Author contribution statement

M. A. Almasre: Performed the experiments; Contributed reagents, materials, analysis tools or data; Wrote the paper.
H. Al-Nuaim: Conceived and designed the experiments; Analyzed and interpreted the data; Wrote the paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] H. Liang, J. Yuan, Hand parsing and gesture recognition with a commodity depth camera, Comput. Vis. Mach. Learn. RGB-Sens. (2014) 259–265.
[2] L. Chen, F. Wang, H. Deng, K. Ji, “A survey on hand gesture recognition,” 2013, Int. Conf. Comput. Sci. Appl. CSA (Dec. 2013) 313–316.
[3] S. Mitra, T. Acharya, Gesture recognition: a survey., IEEE Trans. Syst. Man Cybern. C Appl. Rev. 37 (3) (May 2007) 311–324.
[4] P.K. Pisharady, M. Saerbeck, Recent methods and databases in vision-based hand gesture recognition: a review, Comput. Vis. Image Understand. 141 (2015) 152-165.
[5] H. Liang, J. Yuan, D. Thalmann, Z. Zhang, Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization, Vis. Comput. 29 (6–8) (Jun. 2013) 837–848.
[6] M. Wozniak, M. Grana, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (Mar. 2014) 3–17.
[7] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intel. 35 (8) (Aug. 2013) 1798–1828.
[8] Z. Halim, G. Abbas, A kinect-based sign language hand gesture recognition system for hearing and speech-impaired: a pilot study of Pakistani sign language, Assist. Technol. Off. J. RESNA 27 (1) (2015) 34–43.
[9] A. Mittal, P. Kumar, P.P. Roy, R. Balasubramanian, R.B. Chaudhuri, A modified LSTM model for continuous sign language recognition using leap motion, IEEE Sensor. J. 19 (16) (Aug. 2019) 7056–7063.
[10] D.-H. Vo, H.-H. Huynh, P.-M. Doan, J. Meunier, Dynamic gesture classification for Vietnamese sign language recognition, Int. J. Adv. Comput. Sci. Appl. 8 (3) (2017).
[11] M.R.L. Varshini, C.M. Vidhya Pathi, Dynamic finger gesture recognition using KINECT, in: 2016 International Conference on Advanced Communication Control and Computing Technologies, ICACCT, 2016, pp. 212–216.
[12] T. D’Orazio, R. Marani, V. Resio, G. Cicirelli, Recent trends in gesture recognition: how depth data has improved classical approaches, Image Vis. Comput. 52 (Aug. 2016) 56–72.
[13] B. Rehman, Z. Halim, M. Ahmad, ASCII based GUI system for Arabic scribbled languages: a case of Urdu, Int. Arab J. Inf. Technol. 11 (2014) 329–337.
[14] M.T.K. Tsun, B.T. Lau, H.S. Jo, S.L. Lau, A human orientation tracking system using Template Matching and active Infrared marker, in: 2015 International Conference on Smart Sensors and Application, ICSSA, 2015, pp. 116–121.
[15] Z. Zhang, Microsoft Kinect sensor and its effect, IEEE Multimedia. 19 (2) (Feb. 2012) 4–10.
[16] M. Almasre, H. Al-Nuaim, Comparison of four SVM classifiers used with depth sensors to recognize Arabic sign language words, Computers 6 (2) (Jun. 2017) 20.
[17] M. Almasre, H. Al-Nuaim, “Using the Hausdorff algorithm to enhance kinect’s recognition of Arabic sign language gestures, Int. J. Exp. Algorithms IJEA 7 (1) (Apr. 2017) 18.
[18] A. Jana, Kinect for Windows SDK Programming Guide: Build Motion-Sensing Applications with Microsoft’s Kinect for Windows SDK Quickly and Easily, Packt Publ, Birmingham, 2012.
[19] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, “A survey on human motion analysis from depth data,” time-flight depth imaging, Sensors. Algorithms Appl. (2013) 149–187.
[20] Y.-C. Ou, et al., The learning curve for reducing complications of robotic-assisted laparoscopic radical prostatectomy by a single surgeon, BJU Int. 108 (3) (Aug. 2011) 420–425.
[21] J. Li, A.D. Heap, Spatial interpolation methods applied in the environmental sciences: a review, Environ. Model. Software 53 (Mar. 2014) 173–189.
[22] K. Miura, Basics of image processing and analysis, Cent. Mol. Cell. Imaging EMBL Heidelberg. (2006).
[23] P. Refaelizadeh, L. Tang, H. Liu, Cross-validation, in: L. Liu, M.T. OZSU (Eds.), Encyclopedia of Database Systems, Springer US, 2009, pp. 532–538.
[24] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, 3 edition, Morgan Kaufmann, Haryana, India; Burlington, MA, 2011.
[25] N. Settouti, M.E.A. Bechar, M.A. Chikh, Statistical comparisons of the top 10 algorithms in data mining for classi cation task, Int. J. Interact. Multimed. Artif. Intell. 4 (1) (2016) 46.
[26] L. Rokach, Ensemble-based classiﬁers, Artif. Intell. Rev. 33 (1–2) (Feb. 2010) 1–39.
[27] L. Li, Q. Hu, X. Wu, D. Yu, Exploration of classiﬁcation conﬁdence in ensemble learning, Pattern Recogn. 47 (9) (Sep. 2014) 3120–3131.