

Received September 22, 2021, accepted October 8, 2021, date of publication October 27, 2021, date of current version November 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123570

# Skeleton Joints Moment (SJM): A Hand Gesture Dimensionality Reduction for Central Nervous System Interaction

ZAINAL ABDUL KAHAR<sup>1</sup>, PUTERI SUHAIZA SULAIMAN<sup>1</sup>, FATIMAH KHALID<sup>1</sup>,  
AND AZREEN AZMAN<sup>1</sup>, (Member, IEEE)

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Kembangan 43400, Malaysia

Corresponding author: Zainal Abdul Kahar (zsuperxtreme@gmail.com)

This work was supported in part by UPM, and in part by the Ministry of Higher Education under TAM Program.

**ABSTRACT** Recent breakthroughs with numerous visual experiences using mobile devices encourage the research of human-computer interaction (HCI) involving hand gesture recognition for Holograms, Virtual Reality, and Augmented Reality. The rise of these technologies allows educators in medical segments to apply new pedagogy by interacting with virtual content in a coherent learning environment. This paper proposed the Central Nervous System (CNS) interaction using the Skeleton Joints Moment (SJM) approach for dimension reduction with k Nearest Neighbour (k-NN) for hand gesture classification. Over the past few decades, researchers have proposed various techniques in dimension reduction. One of the methods is principal component analysis (PCA). Experimental results indicated that the SJM technique has similar accuracy to PCA, where both methods showed 96% of prediction using hand skeleton joints data. In addition, PCA has a higher uncertainty of mean error 0.75 compared to SJM at only 0.01. Furthermore, PCA has the worst complexity of  $O(\min(p^3, n^3))$  where SJM  $O(n/d)$ . Evaluation results using the T-Test showed a significant difference between SJM and PCA where  $p < 0.05$ . Thus, there is evidence to reject the null hypothesis.

**INDEX TERMS** Hand gesture recognition, dimensionality reduction, machine learning, hologram.

## I. INTRODUCTION

In the era of mobile devices packed with sensors and visual experience, human-computer interaction (HCI) that involves machine learning plays a vital role in user experience. It requires supporting a variety of low-end devices. Many sectors in education practice human interaction for training purposes. The training takes place in a virtual environment. One of the most commonly used in Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (MR), and a hologram is hand interaction. The hand interaction of virtual objects is implemented using machine learning approaches to classify hand gestures. This interaction that is practised by the educational sector has changed the conventional system of pedagogy to be more interactive [1]. Due to digitisation, interaction data across the virtual environment uncovers the hand gesture patterns using machine learning algorithms [2]. It provides a new challenge in human-computer interaction [3]. The challenge in incremental learning is to retrain these

interactions to accommodate new, previously unseen data, which demands high computational time and energy requirements [4]. Other than that, the large dimension of data is difficult to classify [5]. This study intended to use hand gesture interaction in a Central Nervous System (CNS) application. CNS is one of the most challenging topics in human anatomy, as the student has difficulty visualising the complex human nerve system. A different approach to deliver CNS topics is using a hologram pyramid, where medical students can view and interact with 360-degree hologram images.

A digital hologram for a CNS application requires some interactions to navigate throughout the complete nervous system consisting of the brain and the spinal cord. It displays the brain that makes up the large portion of the nervous system divided into four parts (Brainstem, Cerebellum, Diencephalon, and Cerebrum) and connects with the vertebrae. The vertebrae, consisting of cranial nerves from the brain, protects the spinal cord.

In this study, the CNS application aims to create virtual immersion content with real-time interaction in a hologram environment. Through the pyramid hologram projection,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaogang Jin<sup>1</sup>.

educators can interact with the brain using intuitive free-hand gestures [6]. Hence, hand gestures are becoming the principal technology for realising immersive virtual interaction [7].

The CNS application uses hand gesture interaction to interact with the brain parts in a hologram setup. Hand gesture interaction is a way for a computer to start to comprehend human non-verbal communication. The general meaning of hand gestures is the capacity of a computer to identify motions and execute orders dependent on those gestures [8]. In addition, hand gestures are ubiquitous, are natural characteristics, and are a significant part of communication. Many studies have claimed that a hand gesture is a structured system during human cognition and interaction [9]. It is tactile, familiar to users, precise, and comfortable to use [10]. Hence, hand gesture study is vital in the progression of computer vision.

Hand gesture is an effective technique for hologram human-computer interaction (HCI). The main objective of this study is to make the interaction between human and hologram interface as natural as possible. Like VR, AR, and MR, hand gestures are the most suitable way to interact with 3D [11]. With the advancement of virtual peripherals and sensors, hand gestures have become a well-known approach to interact with head-mounted devices, tablets, and hologram projection. Users can control without physically touching a screen, using a keyboard or mouse. Therefore, HCI has become a vital study for pattern classification in computer vision fields [9], [12].

Hand gesture classification involves statistical learning theory and activation control schemes. The standard approach of predictive pattern recognition in machine learning like SVM, k-NN, and ANN emphasises the minimisation of empirical risk [13]. An SVM-based classification system provides excellent performance like traditional classifiers. It is strongly influenced by the quality and quantity of the labelled data used to train the classifier. A k-NN method is a popular data mining and statistics classification approach due to its simple implementation and significant classification performance [14]. However, assigning a fixed k value to all test samples is impractical for traditional k-NN methods, even though set by experts. For artificial neurons called artificial neural networks (ANNs) or synthetic neural networks (SNNs), a neural network (NNs) is an integrated collection of artificial human neurons utilising a mathematical or computational model for information processing based on a connectionist approach to computation. An artificial neural network reveals that a good choice of activation functions and control scheme will lead to a high memory capacity and increased pattern retrieval capabilities [15].

In this paper, a robust Principal Component Analysis (PCA) technique is compared with our proposed Skeleton Joints Moment (SJM) method using human hand skeleton data.

## II. RELATED WORK

applications [16]. Machine learning is applied to improve classification and experience automatically by the study of

computer algorithms. Machine learning techniques used sample data to create a mathematical model, known as training data, such that predictions or decisions can be made without explicit programming. The types of machine learning vary based on their approach, including the type of data for input and output and the type of task or problem to be solved. This literature study covers hand gesture interaction involving the central nervous, data reduction, machine learning, and uncertainty in intersected data.

### A. INCREMENTAL LEARNING

Nowadays, incremental learning is gaining more attention in the context of growing datasets. It's in contrast with the traditional approach that requires complete data for classification. Although many studies have proposed different techniques for incremental learning, the suitability of each method for a specific task and how they perform often remains unclear. It is vital in incremental learning classification that when a new batch of data arrives, it does not need to access the previous data [17]. Next, the data extends the existing machine learning model.

Retraining these learning models to accommodate new data demands high computational power and impedes the hand gesture model in large applications [4]. It will not recalculate the trained model during the retraining. Hence, each successive training data cause the model to forget the base model and previous training sample.

Incremental machine learning algorithms are known as algorithms that can facilitate incremental learning. Most machine learning algorithms inherently support incremental learning. Incremental learning algorithms include incremental SVM, incremental kNN 1NN, and aNN (RBF network, Learn++, Fuzzy ARTMAP, and TopoART). SVM requires calculation complexity and a large memory requirement when trained in batch mode on large data sets. Wu *et al.* [18] in his paper used convex hulls to reduce the calculation complexity. The convex hulls algorithm utilises the trained model with retained information to reduce computational cost. The convex hull vectors of the trained model and the new dataset constitute the current training dataset for the convex hull calculation. Guo *et al.* [19] in his paper described incremental learning as training a deep model with dynamic connections which can be either 'activated' or 'deactivated' on different datasets of the training stages that lead to better training performance. Jie *et al.* [20] proposed 11NN on the basis of the 1-NN classifier for kNN incremental learning.

The incrementally learned model helps to classify the unstructured dataset from tasks. Equation 1 describe in brief the process of classification tasks in incremental machine learning. The continuously new data breaks into sequential tasks as in figure 1. The model, denoted by  $M$  accumulated and classified from a sequence of tasks  $T$ .

$$M_i = f(T_i, M_{i-1}) \quad (1)$$

Figure 1 describes the process of incremental learning. Machine learning uses the task to continuously train

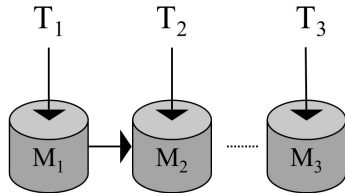


FIGURE 1. Process of incremental learning for a new streaming dataset.

the model. The incremental learning process represents a dynamic technique. It can be applied to the existing model when new data is gradually available over time. This technique overcomes the catastrophic forgetting that happens during training [20]. In incremental learning, the model parameters become input to form knowledge distillation and similar class consolidation. The model can better extend the dataset while learning the new ones for the recognition ability of the old classes.

**B. PRINCIPAL COMPONENT ANALYSIS**

Hand gesture classification for the CNS application involves high dimensional data. There is a large number of published studies that describe high-dimensional data classification as computationally expensive. Most research in data reduction has indicated the usage of PCA [21], [22]. Hence, this paper examines the PCA approach to solving the problem.

In a previous study of data reduction, Principal Component Analysis (PCA) was used to retain data information. PCA is a technique used in machine learning applications for dimensionality reduction. The PCA algorithm is a robust technique in high-dimensional data reduction. Li et al. [23] in his paper uses PCA with k-NN to reduce MIT-BIH Arrhythmia features. PCA condenses object features from a large dataset without losing the information by applying a transformation from linearly correlated variables into uncorrelated variables. The PCA can compress the redundancy information from correlated variables. When two variables are correlated, it's not crucial to retain both variables. PCA transfers the variance of the second variable to the first variable in the k dimension. The direction of the k dimension is determined using eigenvalues and eigenvectors. Therefore, the first transformed principal component is not losing much information from the original variables, whereas PCA can remove the second variable that contains noise with negligible feature information. Reducing the number will significantly minimise the size of the variable with minimal loss of feature information. Pattanadech and Nimsanong [24] in his paper claimed that PCA-kNN gave the highest accuracy and minimal loss of 95%, Li et al. [23] using the Arrhythmia data loss of 98.9%. Experiments conducted by Zhu et al. [25], based on the Cambridge ORL face database, showed that PCA-ANN has the accuracy of 83% and PCA-kNN 72.5% respectively.

Given a collection of points in higher dimensional space, the PCA uses the best fitting algorithm to define the minimum average squared distance from the input to the line.

It performs a preprocessing feature selection to improve computational time and accuracy [2].

Figure 2 described how the two variables are reduced into one variable without losing the information of variable  $x_1$  and  $x_2$  by transferring the information of  $x_1$  and  $x_2$  into eigenvector  $e_{11}$  and eigenvector  $e_{12}$  into  $PC_1$ .  $PC_2$  is removed if the value is highly correlated to  $PC_1$ .

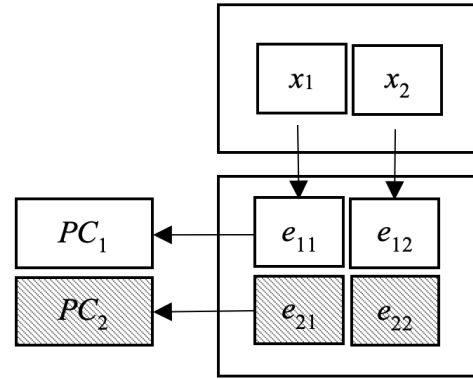


FIGURE 2. One dimensional projection of two dimensional points in the original space.

Assume that a dataset  $x_1, x_2, \dots, x_n$  has  $m$  dimension of inputs and reduced to a smaller space namely as  $k$  where  $m$  is bigger than  $k$ .

- 1) Calculate covariance matrix from the gesture input from feature extraction.

$$\sum = \frac{1}{m} \sum_i^m (x_i)(x_i)^T \tag{2}$$

$$\sum \in R^{n \times n} \tag{3}$$

- 2) Retrieve eigenvector and eigenvalue from the previous covariance matrix.

$$u^T \sum = \lambda \mu \tag{4}$$

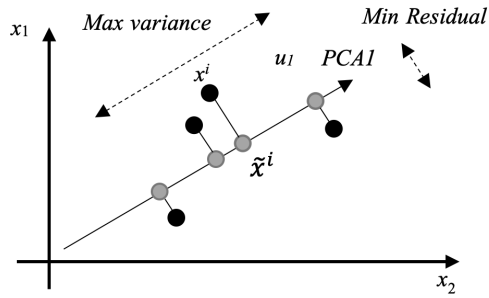
$$U = \begin{bmatrix} | & | & \dots & | \\ u_1 & u_2 & \dots & u_3 \\ | & | & \dots & | \end{bmatrix}, \quad u_i \in R^n \tag{5}$$

- 3) The  $k$ -dimensional subspace projects the hand gesture data where the highest eigenvector of covariance is selected. Hence, it will be a new form of data for hand gesture classification.

$$x_i^{new} = \begin{bmatrix} u_1^T x^i \\ u_2^T x^i \\ \vdots \\ u_k^T x^i \end{bmatrix} \in R^k \tag{6}$$

The PCA reduces the data to a new  $k$ -dimensional space that consists of  $n$  dimensionality. Figure 3 illustrates how the two-dimensional data converted into one dimension.

PCA involves a challenging issue of computing eigenvectors and eigenvalues from an incrementally arriving high-dimensional data stream. Incremental PCA (IPCA)



**FIGURE 3.** Variance and residual of one dimension of principal component analysis.

expects the computation without the corresponding covariance matrix and knowing data in advance. Weng proposed a candid covariance-free IPCA (CCIPCA). It is fast in convergence rate, and low in computational complexity for IPCA [26]. Stutafford claimed that using CCIPCA reduces the amounts of data and has a high accuracy [27].

The results of this study indicate that IPCA ran into a convergence problem with high dimensional image vectors. On the contrary, the CCIPCA technique involves a statistical estimation problem with a lower dimension ratio. Therefore, this paper continues to study the moment technique.

### C. HU MOMENTS

As mentioned in the PCA literature review, the IPCA approach is facing a convergence problem. Hence, this paper examines the moment’s technique to see the feasibility of data reduction. Previous studies have explored the relationships between data reduction and retaining data information. For data reduction, many researchers use PCA to retain data information. This paper proposes an image moments approach to retain dataset information for incremental data. Image moment is a weighted average of function in such moments that are not affected by translation, rotation, and scaling of an image using identity transformation according to the characteristic of each moment.

A features distribution approach can prove moments of various orders. It must have a nonzero point of a finite number in a multi-dimensional space. Equation 7 shows the formula where  $f(x, y)$  is the feature distribution function.

$$k_{gh} = \sum_{x=1}^M \sum_{y=1}^N x^g y^h f(x, y) \quad (7)$$

Center moments  $\eta_{gh}$  that are not affected by translation is taken as the origin coordinate of an image.  $\eta_{gh}$  is obtained by the following formula where  $\bar{x} = k_{10}/k_{00}$  and  $\bar{y} = k_{01}/k_{00}$ .

$$n_{gh} = \sum_{x=1}^M \sum_{y=1}^N (x - \bar{x})^g (y - \bar{y})^h f(x, y) \quad (8)$$

The normalized center moment of  $\eta_{gh}$  is obtained using normalization conversion in following expression

where  $\gamma = (g + h)/2 + 1$ .

$$u_{gh} = \eta_{gh} / \eta_{00}^\gamma \quad (9)$$

Regular and centre moments express the shape characteristic of an image. Nevertheless, they have no invariability. Therefore Hu proposed seven invariant moments. They are not affected by translation, rotation, scaling and mirroring using the expressions as follows.

$$K_1 = u_{20} + u_{02} \quad (10)$$

$$K_2 = (u_{20} - u_{02})^2 + 4u_{11}^2 \quad (11)$$

$$K_3 = (u_{30} - 3u_{12})^2 + (3u_{21} - u_{03})^2 \quad (12)$$

$$K_4 = (u_{30} + u_{12})^2 + (u_{21} + u_{03})^2 \quad (13)$$

$$K_5 = (u_{30} - 3u_{12})(u_{30} + u_{12}) \times [(u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2] + (3u_{21} - u_{03})(u_{21} + u_{03}) \times [3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2] \quad (14)$$

$$K_6 = (u_{20} - u_{02})[(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2] + 4u_{11}(u_{30} + u_{12})(u_{21} + u_{03}) \quad (15)$$

$$K_7 = (3u_{21} - u_{03})(u_{30} + u_{12}) \times [(u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2] + (3u_{12} - u_{30})(u_{21} + u_{03}) \times [3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2] \quad (16)$$

$K_7$  in equation 16 is a skewed invariant formula that enables mirror images to be distinguished

This study uses hand skeleton data from a hand tracking device and requires an approach to keep the feature with the invariance of translation, scale, and rotation. Therefore, this paper examines the moment’s technique for hand gesture data reduction. It prevents convergence problems and a low dimension ratio.

### D. K NEAREST NEIGHBOURS

The k-NN classification algorithm was proposed by P.E Hart and T.M. Cover. It is often used to classify future data owing to its simplicity, ease of implementation and because it is highly effective. k-NN is the simplest lazy learning or instance-based algorithm, which leads to high computational time during classification [28]. However, using PCA-kNN shows a significant improvement in terms of recognition [29], [30]. It reduces the size of dimension without losing the prominent features for kNN [31]. kNN has no explicit training or decision boundaries and can be used for classification and regression. The classification of k-NN is through a voting process of training samples with the size of k value. Kamencay *et al.* [32] in his paper described that kNN is a classifying method based on close training samples in the feature vector. However, k-NN has three main shortcomings. All attributes are considered equal by using standard Euclidean distance. Unbalanced data caused inaccurate results from the neighbourhood size taken as an input parameter. The voting process for probability estimation has a high calculation cost depending on the number of training data.



k-NN classifies using training data by measuring using standard Euclidean distance. Euclidean distance is defined as follows, where  $a_i$  is the attribute vector of  $x$  [2]:

$$d(x_i, x_j) = \sqrt{\sum (a_r(x_i) - a_r(x_j))^2} \quad (17)$$

In Cartesian coordinates,  $x_i$  and  $x_j$  are two samples in Euclidean n-space, then the distance denoted as  $d$  from  $x_i$  to  $x_j$  is measured by the Pythagorean formula. The position of  $x_i$  in Euclidean space is known as the Euclidean vector. The k-NN classification is based on the most common class of  $k$  nearest neighbours to estimate the test instance. The equation is defined as follow:

$$c(x) = \arg \max \sum_{i=1}^k \delta(c, c(y_i)) \quad (18)$$

$c$  is denoted as the finite set of class, where  $y_i$  is the  $k$  nearest neighbours of test instance [33].

Figure 4 illustrates the voting process of nearest neighbours using k-NN with a  $k$ -value. We assume that the  $k$ -value is 3. The test instance denoted as  $y_i$  with a distance of  $k$ -value consists of  $x_1, x_2$  and  $x_3$  as the training instance.  $c_1$  and  $c_2$  describe as the sample class of dataset where  $c_i \in S$ . As a result,  $c_1$  consists of more training instances compared to  $c_2$ , which indicates that  $y_i$  belongs to class  $c_1$ .

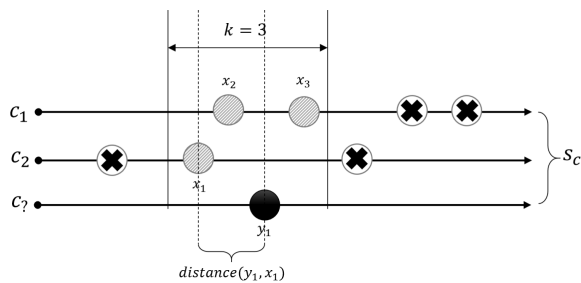


FIGURE 4. One dimensional voting process for sample  $S$  denoted  $c_i$  where  $y_i$  is a test instance of  $k$ -value equals to 3.

### III. METHODOLOGY

Previous studies have based their selection on the PCA. It modifies an  $n$  to  $k$  dimensional space from the nonzero characteristic vector of a linear transformation known as eigenvector [25]. The corresponding eigenvalue is the scale of the eigenvector denoted by  $\lambda$ . Unlike PCA, SJM applies a central moment technique with invariant anchor parenting to the data without calculating the covariance matrix for eigendecomposition. Invariant anchor parenting applied in skeleton data keeps the property unchanged for a specified transformation in hand gestures. The SJM is a dimension reduction method proposed to decrease the  $n$  dimension to three-dimensional space represented in  $x, y,$  and  $z$ -axis for skeleton data. Therefore, the SJM supports incremental learning without the need to recalculate the dataset.

### A. SJM DIMENSION REDUCTION

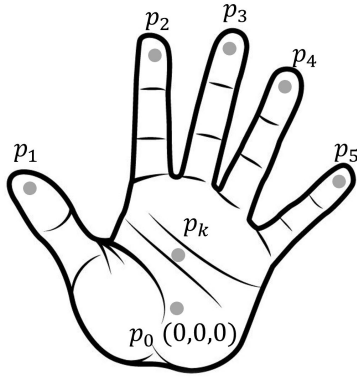
Initially, this approach starts by acquiring hand skeleton features denoted as  $\{X \in \mathbb{R}^3 | X \subset J\}$  from a hand tracking device known as Leap Sensor where  $J$  is the joints captured. The device's controller read the data into memory and performed resolution adjustments. This data is then streamed to Leap Motion tracking software. After compensating for background objects (such as heads) and ambient environmental lighting, the images are analyzed to reconstruct a 3D hand skeleton for both hands. Figure 5 illustrates feature selection  $x_i \in X$  from the skeletal information output from the device, and there are 22 features available as skeletal joints. Lai et al. [34] stated that vision-based fingertips detection for hand gesture recognition has been widely used and often defines hand pose. Maisto et al. [35] described the use of fingertips coordinates could represent a system that allows an advanced and suitable form of interaction. Therefore, in this paper, only the palm and fingertips: thumb, index, middle, ring, little are selected as significant features for this study where  $\{x_i \in X | 0 \leq i \leq 5\}$ . The fingertips were chosen as attributable to significant movement while performing the hand gestures pose. In addition, this paper aimed to reduce the dimension without losing the hand gesture information. Thus, in this paper, the selected feature  $x_i$  will be examined in terms of prediction accuracy using training and test data from the  $x_i$ . The  $x_i$  is recorded as a raw dataset for SJM, while  $\neg x_i$  is not recorded as a dataset and discarded. The  $x_i$  are extracted from local to world coordinates.

Assume that hand skeleton joints are extracted from a hand tracking device. The raw number of instances is 79,744, which is too big to process in machine learning with the worst complexity of  $O(n)$  for the k-NN approach, and  $O(n^3)$  for SVM, and  $O(nt * (ij + jk + kl))$  for ANN. Consequently, the computational complexity requires high processing and is unsuitable for low-end devices. Therefore, the SJM reduces the  $n$  dimensions by selecting the skeleton joints features and downsizing them as in Table 1. Table 1 demonstrates the reduction process from the raw image that consists of 79,744 data to feature selection with 19,936 data. As illustrated in figure 5, there are 6 points extracted from the joints where there are 18 dimensions denoted as  $p_0, p_2 \dots p_5$ .  $p_1$  to  $p_5$  are fingertips, and  $p_0$  is the centre position of the palm. The SJM transforms these joints into a world coordinate. The palm  $p_0$  is selected as an anchor to  $p_1 \dots p_5$ .

TABLE 1. Feature reduction of hand skeleton.

Raw Skeleton	Feature selection
79744	19936

After feature extraction, the SJM reduces the skeleton data dimension using centroid moment by anchoring the fingertips to the palm to retain the invariant data information and retrieve the centroid of  $p_0 \dots p_5$  known as  $p_k$ . Therefore,  $p_0 \dots p_5$  is reduced to  $p_k$ , consisting of three dimensions  $x, y,$  and  $z$ -axis for each sample of every situation in a hand



**FIGURE 5.** Skeleton joints as features of hand divided into 6 parts, the finger tips and the palm.

gesture. The  $p_k$  is used to classify the hand gestures using the nearest neighbours approach. Equation 19, 20 and 28 described as the higher dimension of moments [5] used in the SJM. The following equations describe the process of the SJM.

- 1) Raw moment for continuous three dimensional function  $f(x, y, z)$  of order  $(p + q + r)$  is defined as

$$m_{pqr} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q z^r f(x, y, z) dx dy dz \quad (19)$$

- 2) for  $p, q, r = 0, 1, 2, \dots$  Adapting this to scalar (Boolean) of skeleton with vector transformation  $S(x, y, z)$ , raw image moments  $M_{ijk}$  are calculated by

$$m_{ijk} = \sum_x \sum_y \sum_z x_i y_j z_k S(x, y, z) \quad (20)$$

- 3) Scale the skeleton points by multiplying the points by scaling factor of  $S_x, S_y$  and  $S_z$

$$S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (21)$$

- 4) Transform the skeleton points into x, y, and z-axis rotation denoted as  $R_x(\theta), R_y(\theta)$ , and  $R_z(\theta)$ .

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (22)$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (23)$$

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (24)$$

- 5) The world position of skeleton points

$$t_{\delta x \delta y \delta z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \delta x & \delta y & \delta z & 1 \end{pmatrix} \quad (25)$$

- 6) Skeleton joints moment where translation (t), rotation(r), and scale(s) are vectors in three dimensional space.  $S(t, r, s)$  is the skeleton joints world transformation. It is defined as

$$\mu_{ijk} = \sum_t \sum_r \sum_s (s/s_{p'})^i (r.r_{p'})^j (t - t_{p'})^k S(t, r, s) \quad (26)$$

- 7) Moment in local space

$$\{x, y, z\} = \mu_{100} * \mu_{010} * \mu_{001} \quad (27)$$

- 8) Centroid of the SJM in local space that is invariant to translation, rotation & scaling

$$\{\bar{x}, \bar{y}, \bar{z}\} = \left\{ \frac{x}{M_{000}}, \frac{y}{M_{000}}, \frac{z}{M_{000}} \right\} \quad (28)$$

It has previously been identified that central moment and hu moment derive invariant concerning transformation classes. Both moments are derived from equation 19 known as continuous moment for two dimensional space of  $p$  and  $q$  [36]. Where  $p, q \in \{0, 1\}$  and  $f$  is the function of the feature. This study proposes an SJM method to adapt the central moment and hu moment that derive invariant to the transformation classes involving a hand skeleton. Parent moment consists of three dimensions as the principal component. The initial process is to apply equation 20 to calculate the mean of x, y, and z. Next, the scale, rotation, and translation from skeleton points obtained from equation 21, 22, 23, 24, and 25. Then, equation 4 is the moment to modify the point into the local space of the hand skeleton root defined as the palm by multiplying with the inversed parent. Finally, equation 28 is to get the centroid of the local space of the hand skeleton. Algorithm 1 explains the process to reduce the six features' skeleton data into three dimensions.

### B. SJM K-NN CLASSIFICATION

The final step is to process the SJM dataset with a k-NN algorithm. The SJM dataset consists of the centroid of the six features and target class. Initially, the k-NN algorithm split the dataset into training and test data with (ratio 60%:40%). The next step is to calculate the distance of the neighbour of training and test data. The distances are sorted out from shortest to longest. Then, the number of neighbours with the distance of k-value is retrieved. Finally, The number of prediction errors is defined by comparing the prediction output with the actual test output denoted as  $Y_{test}$  where  $k - value$  is  $\{x \in \mathbb{Z} | x < k_n\}$ .

**Algorithm 1:** SJM Dimension Reduction

```

Result: SJM Invariant dataset
data := HandTrackingDevice( $X_1, X_2 \dots X_n$ );
data := SelectFeature(data,  $X_1, X_2 \dots X_{18}$ );
target := SelectPose({hand – open, hand –
grab, hand – point, hand – close});
if isOutlier(data, target,  $SJM_{dataset}$ ) then
|   return null ;
end
 $T_{anchor} :=$  Transform(data[16], data[17], data[18]);
 $T_{world} :=$  TransformExclude(data,  $T_{anchor}$ );
foreach  $t \in T_{world}$  do
|    $M_{000} := M_{000} + 1$ ;
|    $\mu_{100} := \mu_{100} + t(s)/T_{anchor}(s)'$ ;
|    $\mu_{010} := \mu_{010} + t(r) * T_{anchor}(r)'$ ;
|    $\mu_{001} := \mu_{001} + t(t) - T_{anchor}(t)'$ ;
end
 $T_{local} := \mu_{100} * \mu_{010} * \mu_{001}$ ;
 $x := T_{local}(x)/M_{000}$ ;
 $y := T_{local}(y)/M_{000}$ ;
 $z := T_{local}(z)/M_{000}$ ;
 $SJM_{dataset}(x, y, z, target)$ ;
    
```

**IV. EXPERIMENT**

This experiment involves comparing the SJM and PCA approaches using k-NN, where the evaluation results from the accuracy of hand gesture recognition. Thirty participants were involved in this experiment for feature extraction (Average age = 21.65).

**A. EXPERIMENT SETUP**

The CNS application is installed in a Microsoft Surface 6 Pro tablet and wired to an infrared hand tracking device to conduct this experiment. The development of this education application is by using Unity Engine 2019 to create immersive visual content of the central nervous system with HCI. The engine rendered the central nervous system model in a 3D hologram, as in figures 6 and 7. The 3D models are interactive using hand gestures to display brief information in video or textual form. There are a few parts of the brain selected for the HCI:

Figure 8 depicts the layout of the brain hologram projection relating to the pepper ghost technique [37]. The Windows Surface Pro is facing upward at the projection base, and the folded pyramid acrylic is lying above it. In the middle-bottom of the pyramid is a small hand tracking device to detect hand gestures within a meter distance for the interaction.

Figure 9 shows the grab hand gesture to move or rotate the brain. To release the brain, perform an open hand pose as in figure 11. Figure 10 is a pose to trigger video or information of the brain part and figure 12 to hide the brain part and reveal the details inside the brain. The component setup for these situations are as in figure 13.

**Algorithm 2:** SJM k-NN Classification

```

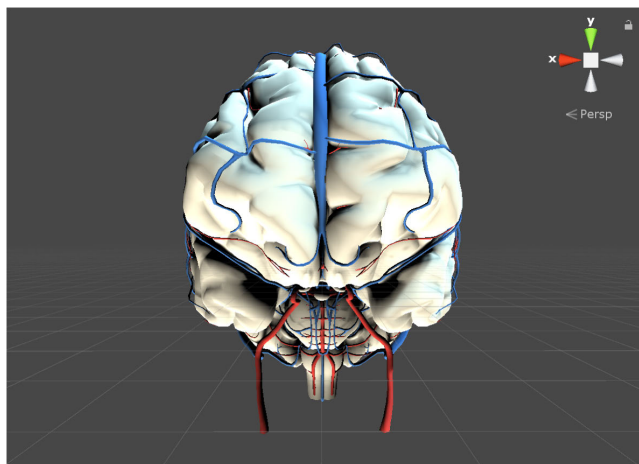
Result: Number of error in specification
Dataset := DataFrame(list(zip( $SJM_{dataset}$ )), columns =
["x", "y", "z", "target"]);
 $X_{train}, X_{test}, Y_{train}, Y_{test} :=$ 
trainTestSplit(Dataset, test_size = 0.60)
for  $i = 0$  to range( $0, k_n$ ) do
|    $pred_i =$  knnPredict( $X_{train}, X_{test}, i$ );
|   error.append( $pred_i \neq Y_{test}$ );
end
Function euclideanDistance (train, test) :
|   for  $i = 0$  to train.length do
|   |   distance := distance + (train[i] – test[i])2;
|   end
|   return distance;
End Function
Function getNeighbors (train, test, kValue) :
|   for  $i = 0$  to train.length do
|   |   distance := euclideanDistance(train[i], test);
|   |   distanceStack.append(train[i], distance);
|   end
|   distanceStack.sort();
|   for  $i = 0$  to kValue do
|   |   neighbors.append(distanceStack[i][0]);
|   end
|   return neighbors;
Function kNNPredict (train, test, kValue) :
|   neighbors = getNeighbors(train, test, kValue);
|   prediction := maxCount(neighbors);
|   return prediction;
End Function
    
```

**TABLE 2.** Central nervous system parts for HCI.

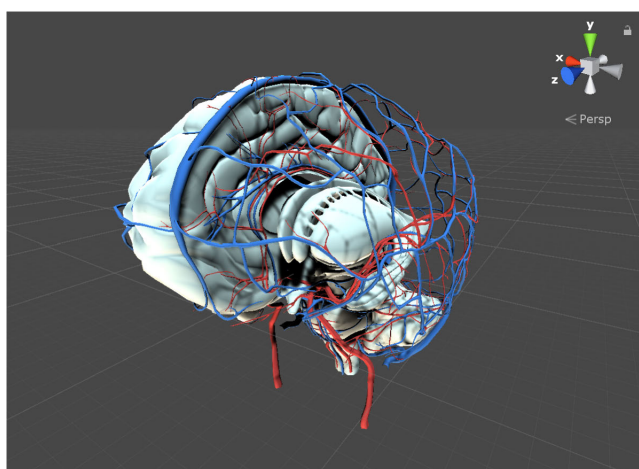
Brain Parts	Description
1. Cerebellum	Its role is in motor control, active in the coordination, precision, and timing of movements.
2. Cerebrum	Responsible for receiving and giving meaning to information from the sense organs, as well as controlling the body
3. Amygdaloid	Part of the limbic systems and olfactory. Role in the sense of smell, motivation, and emotional behavior.
4. Hypothalamus	Releasing hormones. regulating body temperature
5. Thalamus	Relay motor and sensory signals to the cerebral cortex.
6. Medulla oblongata	Passing messages between spinal cord and brain. It's is essential for regulating cardiovascular and respiratory systems.

**B. MEASURES**

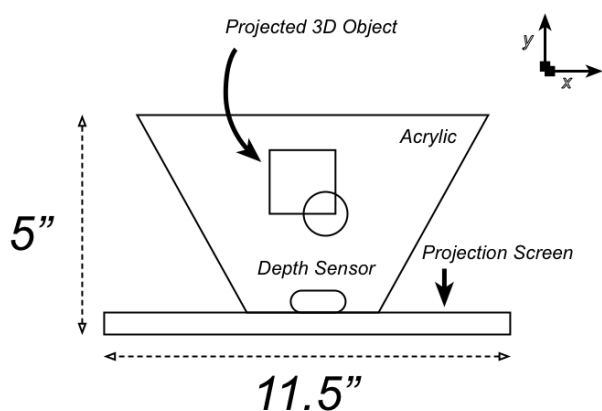
The measurements in this experiment consist of quantitative and qualitative measurements to deal with the high dimensional data for machine learning addressed in the problem statement, where it becomes a constraint for low-end device



**FIGURE 6.** Outer part of the brain that consists of cerebrum and cerebellum hemispheres.



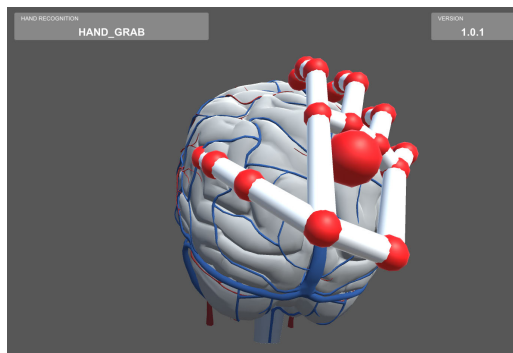
**FIGURE 7.** Internal part of the brain in detail that consists trigger points for interaction.



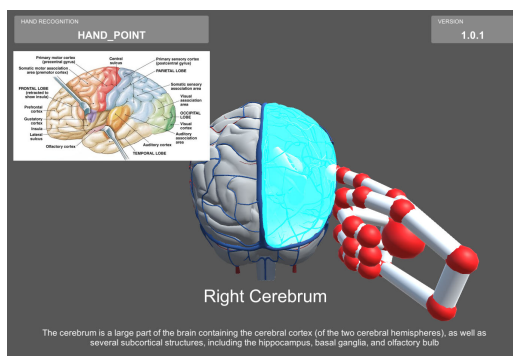
**FIGURE 8.** Hologram using paper ghost technique to project the 3D model, the top side is exposed for interaction.

processing. The measurements used for SJM evaluation are as follows:

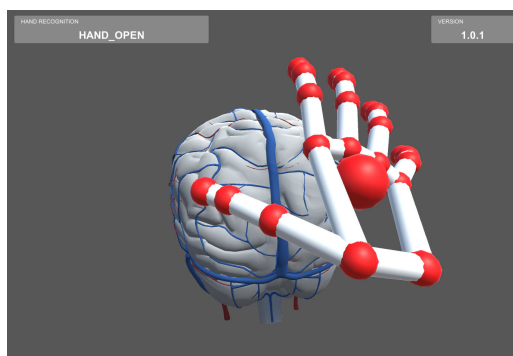
- Dimensional Reduction: the data reduction in percentile after applying the SJM to the raw data using the moment process.



**FIGURE 9.** Grabbing the brain recognition for pan and rotation.



**FIGURE 10.** Point to part of the brain to trigger information.



**FIGURE 11.** Open hand gesture to release from transformation.

- Error rate: Examine the mean error of hand gesture classification with k value for SJM k-NN. The mean error signifies the optimal size of the k-value and the distinction between SJM and PCA integration with k-NN.
- Accuracy estimation: Analyse the accuracy of hand gesture classification using the F1 score for SJM k-NN and PCA k-NN. The F1 score will apply to entire segments in the convex hull operation.
- Significance different: Analyse the significance difference of hand gesture dataset using a T-test for SJM Convex Hull and PCA Convex Hull. The T-test will signify the distinction between both methods.



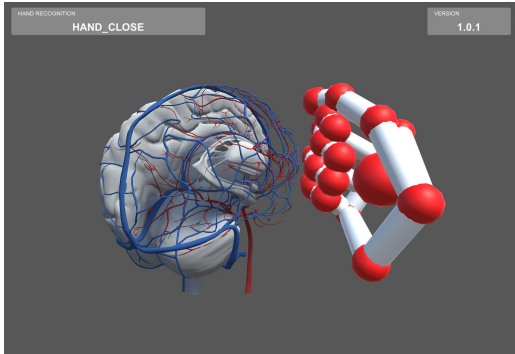


FIGURE 12. Close hand to hide brain part.

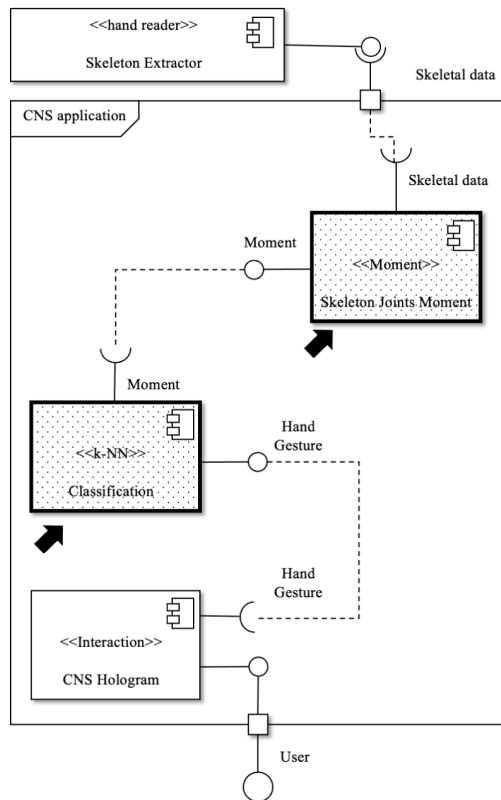


FIGURE 13. SJM component setup.

C. PROCEDURES

Briefing and demonstrating SJM k-NN data capture using a hand tracking device for skeleton data took place before the feature extraction started. During the experiment, every participant interacted with the brain 3D model using hand gestures. They performed four different poses: open hand, close hand, pointing, and grabbing.

V. RESULT AND DISCUSSION

This topic will study the findings of the results from the SJM experiment applied to the raw skeletal data held in this chapter. The findings involve dimension reduction, prediction

TABLE 3. Comparison SJM and PCA complexity.

Method	Training & Incremental Complexity
SVM	$O(n^3)$
ANN	$O(nt * (ij + jk + kl))$
kNN	$O(1)$
KDTree kNN	$O(1)$
PCA kNN	$O(\min(p^3, n^3))$
SJM k-NN	$O(n/d)$
Prediction Complexity	
SVM	$O(1)$
ANN	$O(l * k * t)$
kNN	$O(n)$
KDTree kNN	$O(\log(n))$
PCA kNN	$O(p)$
SJM k-NN	$O(n/d)$

TABLE 4. Computational speed for PCA and SJM.

Method	Data Size	Time
PCA	4,984	1.7781 sec
SJM	4,984	0.3200 sec
Difference (PCA-SJM)		1.4581 sec

mean error and accuracy, and the distinction between SJM and PCA.

A. DETERMINING COMPUTATIONAL COMPLEXITY

Table 3 shows the complexity of several techniques in machine learning. The complexity to train for k-NN is  $O(1)$  where training is not needed to form a model. However, for SJM k-NN, the complexity of training is  $O(n/d)$  where  $d$  is the dimension.  $d$  is derived from the moment where  $d$  is the number of dimensions. According to Table 3, SJM k-NN Has no training compared to SVM, ANN and PCA k-NN for training and Incremental learning. The complexity of prediction shows that SJM k-NN is optimal compared to ANN and equal to PCA k-NN. SVM complexity  $O(1)$  for prediction is better than SJM k-NN  $O(n/d)$ . However, the incremental learning for SVM has the worst case of  $O(n^3)$ . As a result, SJM k-NN has the most optimal complexity for incremental learning compared to SVM, KNN and ANN.

As shown in Table 4, the computational cost for PCA is much higher compared to SJM. PCA records 1.7781 seconds to calculate covariance and Eigen decomposition for 4,984 data. However, SJM calculation is reduced to 0.3200 seconds, which is 82% faster than PCA.

B. DIMENSION REDUCTION

Table 5 below shows the SJM dimension reduction for the raw hand skeleton data. Initially, the raw data retrieved from the hand tracking device was 79,794 points. Feature selection reduced the points to 19,936 with 75% reduction. Using SJM, the data reduced to 4,984 where the number of data points is significantly reduced, up to 94% for SJM. The raw data consists of  $\{x \in \mathbb{R}|x_1 \dots x_{63}\}$  and target  $k = \{ho, hg, hp, hc\}$ , and feature selection consist of  $\{x \in \mathbb{R}|x_1 \dots x_{15}\}$  and  $k$ . While the SJM consists of  $\{x \in \mathbb{R}|x_1 \dots x_3\}$  and  $k$ .

Figure 14 and 15 illustrates the variables taken from 1.2k samples from 30 participants. Each participant was required

TABLE 5. SJM reduces the size of dataset.

	Raw data	Feature Selection	SJM Method
Hand-open	19968	4992	1248
Hand-grab	19968	4992	1248
Hand-point	19904	4976	1244
Hand-close	19904	4976	1244
Dimension	63	15	3
Reduction	0 %	75%	94%

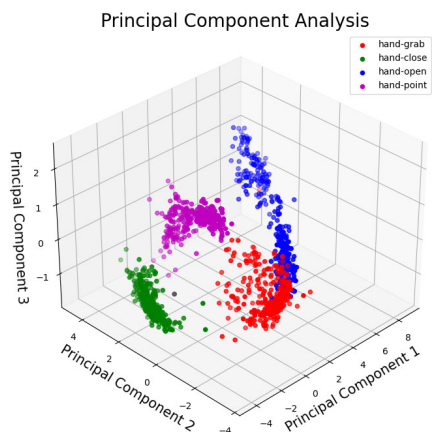


FIGURE 14. The classes are clustered into four segments to check for uncertainty using PCA.

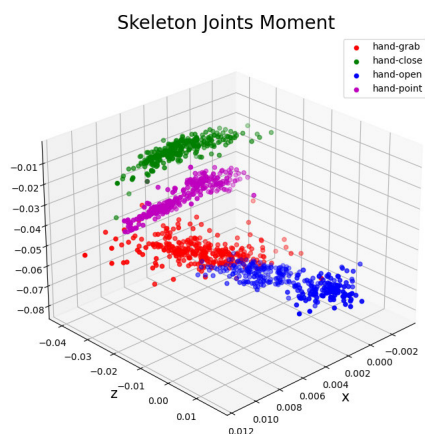


FIGURE 15. Four classes of hand recognition is scatter plotted in three dimensional space using SJM output.

to pose with four different gestures: closing hand, grab the brain, open hand, and point at a particular part of the 3D model. Using PCA, the hand gesture dataset is reduced into three principal components  $PC_1, PC_2, PC_3$ . Figure 14 demonstrates that *hand – open* and *hand – grab* classes overlap and increase the uncertainty of classification between these two classes. However, the result of SJM in figure 15 shows that the four classes are separable.

C. ACCURACY ESTIMATION

Table 6 and 7 shows that SJM and PCA have a similar accuracy of 96%. Therefore, despite SJM not measuring the covariance, SJM still has a similar accuracy in hand gesture classification that is practical for incremental learning.

TABLE 6. F1 score for PCA.

Class	Precision	Recall	F1-Score
hand-close	0.98	0.99	0.99
hand-grab	0.94	0.93	0.93
hand-open	0.94	0.95	0.94
hand-point	1.00	0.98	0.99
Accuracy		0.96	0.96
Macro avg	0.96	0.96	0.96
Weighted avg	0.96	0.96	0.96

TABLE 7. F1 score for SJM.

Class	Precision	Recall	F1-Score
hand-close	1.00	0.99	1.00
hand-grab	0.92	0.92	0.92
hand-open	0.93	0.92	0.93
hand-point	0.99	1.00	0.99
Accuracy		0.96	0.96
Macro avg	0.96	0.96	0.96
Weighted avg	0.96	0.96	0.96

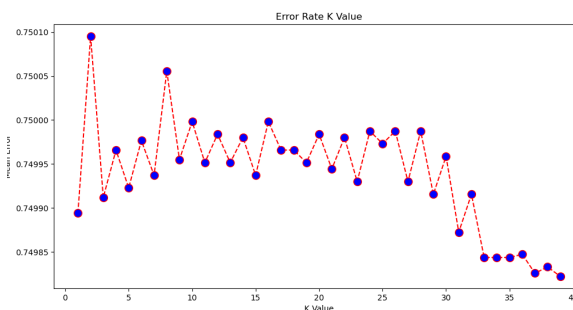


FIGURE 16. The mean error using PCA method where  $k = 40$ .

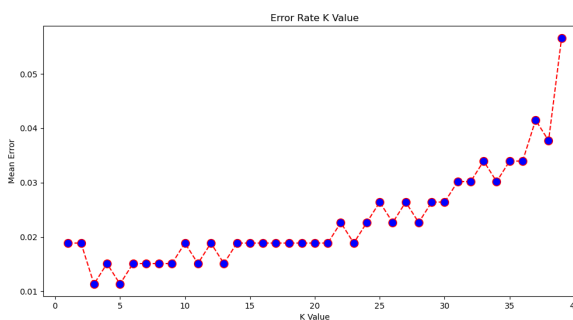


FIGURE 17. The mean error using SJM method where  $k = 40$ .

D. SJM ERROR RATE

As demonstrated in figure 14 and 15, results show that using SJM for relative data is better than PCA in data clustering and for retaining information for skeletal joints. To prove this claim, figure 16 and 17 show the mean error of PCA is  $\{\bar{x} \in \mathbb{R} | 0.74985 > \bar{x} > 0.74970\}$  and SJM is  $\{\bar{x} \in \mathbb{R} | 0.05 > \bar{x} > 0.01\}$ . According to the PCA in figure 16, the higher the size of  $k - value$  the lower the mean error rate. However, the SJM in figure 17 shows that the lower the size of  $k - value$  the lower the mean error rate gets. Evaluation of the mean error

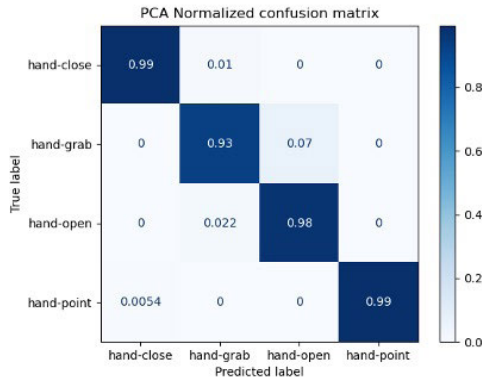


FIGURE 18. Confusion matrix in percentage for PCA.

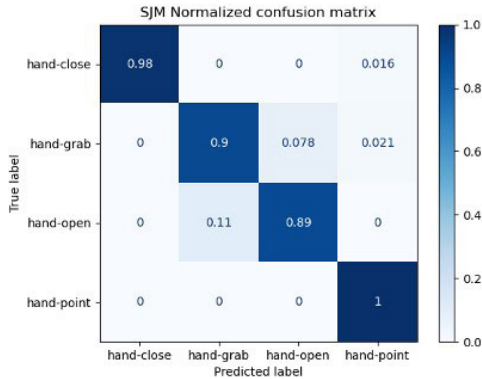


FIGURE 19. Confusion matrix in percentage for SJM.

explains that the SJM has better recognition compared with the PCA.

Figure 18 indicates the confusion matrix of the hand gesture classes for the PCA method. A confusion matrix, also known as the error matrix, is used to visually describe the performance of classification and the intersected region between the hand gesture classes. The largest confusion percentage using PCA for a data reduction method is 7% for hand-grab and hand-open. 1% confusion for hand-close and hand-grab, 2.2% confusion for hand-open and hand-grab, and 0.5% confusion for hand-point and hand-close. The confusion percentage of hand-close, hand-grab and hand-open show that there are similar poses of hand gesture that represent the classes and cause uncertainty during the recognition.

Figure 19 indicates the confusion matrix of the hand gesture classes for the SJM method. The confusion percentage using the SJM for the data reduction method is 7.8% for hand-grab and hand-open, and 1.1% confusion for hand-close and hand-grab. As a result, the uncertainty using the PCA method occurs in all classes where the SJM only occurs between hand-close and hand-grab.

E. T-TEST PCA AND SJM

The T-Test result of  $PC_1$ ,  $PC_2$  and  $PC_3$  between PCA and SJM rejects  $H_0$ , where P-value as in Table 8. The result  $P < 0.05$  concludes that there is a significant difference between these two methods for every hand gesture class.

TABLE 8. T-Test result for PCA and SJM.

Hand gesture class		T-Value
Hand-open	$PC_1$	-50.4554
	$PC_2$	-5.4809
	$PC_3$	-3.3887
Hand-close	$PC_1$	218.6887
	$PC_2$	-50.1723
	$PC_3$	64.3675
Hand-point	$PC_1$	127.0675
	$PC_2$	5.3621
	$PC_3$	-77.7007
Hand-grab	$PC_2$	-13.1011
	$PC_2$	41.8416
	$PC_3$	24.2881
p value		0.05

VI. LIMITATION AND FUTURE WORKS

The SJM technique empowers machine learning to apply incremental learning without computing principal components using the covariance matrix to reduce dimension. Its focus is only on the rigged skeleton where the joints are invariant to transformation: scaling, rotating and translating. The classification applied in the SJM k-NN used fixed data samples. This study will continue with a functional classification of incremental learning for future works to calculate the arriving data stream in real-time based on the user hand gesture. The incremental approach will increase the accuracy of hand tracking because different users have different hand movement practices when they perform the gesture, and the hand gesture might be affected by the user’s pose. For example, a user with longer fingers might hold a different pose than someone with a shorter finger.

VII. CONCLUSION

The problem highlighted on the PCA for hand gestures is the requirement to recalculate new arriving data caused by the covariance. Hence, this paper proposes the SJM method to prevent the covariance recalculation of the whole dataset. Next, the experiment continued with the SJM k-NN algorithm to prove the accuracy of the SJM dataset. The SJM k-NN algorithm uses the k nearest neighbour to classify the hand gesture based on the corresponding decision distance. The statistical analysis shows that the SJM method reduces computational complexity for continuous learning than a standard PCA method, where the PCA  $O(\min(p^3, n^3))$  and SJM  $O(n/d)$ . Results show that SJM is faster by 82% than PCA. However, the detection accuracy of the SJM and PCA is similar at 96% with a significant level of  $P < 0.05$  when testing with a two-tailed paired t-test. The experimental results indicate that there are significant differences between both methods. In conclusion, the SJM method reduces the data dimension and retains the information without recalculating the new data’s covariance, which improves the computational speed. Hence, this study has successfully fulfilled the first objective of developing a data reduction method for hand gesture classification that supports incremental learning.

## REFERENCES

- [1] K. G. Ramsundar, "Interactive touch board using IR camera," in *Proc. 2nd Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2015, pp. 1690–1695.
- [2] G. T. Reddy, M. P. K. Reddy, K. Lakshmana, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.
- [3] M.-T. Yang and W.-C. Liao, "Computer-assisted culture learning in an online augmented reality environment based on free-hand gesture interaction," *IEEE Trans. Learn. Technol.*, vol. 7, no. 2, pp. 107–117, Apr. 2014.
- [4] S. S. Sarwar, A. Ankit, and K. Roy, "Incremental learning in deep convolutional neural networks using partial network sharing," *IEEE Access*, vol. 8, pp. 4615–4628, 2020.
- [5] C. Lv, P. Zhang, and D. Wu, "Gear fault feature extraction based on fuzzy function and improved hu invariant moments," *IEEE Access*, vol. 8, pp. 47490–47499, 2020.
- [6] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1659–1673, Sep. 2016.
- [7] P. Rosedale, "Virtual reality: The next disruptor: A new kind of worldwide communication," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 48–50, Jan. 2017.
- [8] S. P. More and A. Sattar, "Hand gesture recognition system using image processing," in *Proc. Int. Conf. Electr., Electron., Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 671–675.
- [9] Y. Yao and Y. Fu, "Contour model-based hand-gesture recognition using the Kinect sensor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 11, pp. 1935–1944, Nov. 2014.
- [10] R. Xiao, J. Schwarz, N. Throm, A. D. Wilson, and H. Benko, "MRTouch: Adding touch input to head-mounted mixed reality," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 4, pp. 1653–1660, Apr. 2018.
- [11] J. S. Sonkusare, N. B. Chopade, R. Sor, and S. L. Tade, "A review on hand gesture recognition system," in *Proc. 1st Int. Conf. Comput., Commun., Control Autom., (ICCUBEA)*, 2015, pp. 790–794.
- [12] C. Quan and J. Liang, "A simple and effective method for hand gesture recognition," in *Proc. Int. Conf. Netw. Inf. Syst. Comput. (ICNISC)*, Apr. 2016, pp. 302–305.
- [13] E. Pasolli, F. Melgani, and Y. Bazi, "Support vector machine active learning through significance space construction," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 431–435, May 2011.
- [14] J. Zhang, X.-D. Zhang, and S.-W. Ha, "A novel approach using PCA and SVM for face detection," in *Proc. 4th Int. Conf. Natural Comput.*, vol. 3, 2008, pp. 29–33.
- [15] W. Lin and G. Chen, "Large memory capacity in chaotic artificial neural networks: A view of the anti-integrable limit," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1340–1351, Aug. 2009.
- [16] C. Mendes, A. Barcelos, and S. J. Rigo, "MLtool: A tool to automate the construction, evaluation, and selection of machine learning models," *IEEE Latin Amer. Trans.*, vol. 17, no. 7, pp. 1163–1170, Jul. 2019.
- [17] A. Bouchachia, B. Gabrys, and Z. Sahel, "Overview of some incremental learning algorithms," in *Proc. IEEE Int. Fuzzy Syst. Conf.*, Jun. 2007, pp. 1–6.
- [18] C. Wu, X. Wang, D. Bai, and H. Zhang, "Fast SVM incremental learning based on the convex hulls algorithm," in *Proc. Int. Conf. Comput. Intell. Secur.*, vol. 1, Dec. 2008, pp. 249–252.
- [19] S. Guo, H. He, and X. Huang, "A multi-stage self-adaptive classifier ensemble model with application in credit scoring," *IEEE Access*, vol. 7, pp. 78549–78559, 2019.
- [20] L. Jie, X. Yaxu, and Y. Yadong, "Incremental learning algorithm of data complexity based on KNN classifier," in *Proc. Int. Symp. Community-Centric Syst. (CCS)*, Sep. 2020, pp. 1–3.
- [21] T. Rzayev, S. Moradi, D. H. Albonesi, and R. Manchar, "DeepRecon: Dynamically reconfigurable architecture for accelerating deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 116–124.
- [22] M. Mussetta, D. Caputo, A. Pirisi, F. Grimaccia, L. Valbonesi, and R. E. Zich, "Neural networks and evolutionary algorithm application to complex EM structures modeling," in *Proc. Int. Conf. Electromagn. Adv. Appl.*, Sep. 2009, pp. 1020–1023.
- [23] R. Li, S. Ji, S. Shen, P. Li, X. Wang, T. Xie, X. Zhang, and Z. Wang, "Arrhythmia multiple categories recognition based on PCA-KNN clustering model," in *Proc. 8th Int. Symp. Next Gener. Electron. (ISNE)*, Oct. 2019, pp. 6–8.
- [24] N. Pattanadech and P. Nimsanong, "Effect of training methods on the accuracy of PCA-KNN partial discharge classification model," in *Proc. IEEE Region Conf. (TENCON)*, Oct. 2014, pp. 1–5.
- [25] Y. J. Zhu, W. B. Liu, F. Y. Jin, and Y. Wu, "The research of face recognition based on kernel function," *Appl. Mech. Mater.*, vols. 321–324, pp. 1181–1185, Jun. 2013.
- [26] J. Weng, Y. Zhang, and W.-S. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1034–1040, Aug. 2003.
- [27] S. A. Stuttaford, A. Krasoulis, S. S. G. Dupan, K. Nazarpour, and M. Dyson, "Automatic myoelectric control site detection using candid covariance-free incremental principal component analysis," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 3497–3500.
- [28] S. Taneja, C. Gupta, K. Goyal, and D. Gureja, "An enhanced K-nearest neighbor algorithm using information gain and clustering," in *Proc. 4th Int. Conf. Adv. Comput. Commun. Technol.*, Feb. 2014, pp. 325–329.
- [29] I. Menhour, M. B. Abidine, and B. Fergani, "A new framework using PCA, LDA and KNN-SVM to activity recognition based Smartphone's sensors," in *Proc. 6th Int. Conf. Multimedia Comput. Syst. (ICMCS)*, May 2018, pp. 1–5.
- [30] N. Suo, X. Qian, and J. Zhao, "Gait recognition based on KPCA and KNN," in *Proc. 2nd Conf. Environ. Sci. Inf. Appl. Technol.*, Jul. 2010, pp. 432–435.
- [31] S. B. Dabhade, N. Bansod, M. Naveena, K. Khobragade, Y. S. Rode, M. M. Kazi, and K. V. Kale, "Double layer PCA based hyper spectral face recognition using KNN classifier," in *Proc. Int. Conf. Current Trends Comput., Electr., Electron. Commun. (CTCEEC)*, Sep. 2017, pp. 289–293.
- [32] P. Kamencay, R. Hudec, M. Benco, and M. Zachariasova, "Feature extraction for object recognition using PCA-KNN with application to medical image analysis," in *Proc. 36th Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2013, pp. 830–834.
- [33] S. Taneja, C. Gupta, S. Aggarwal, and V. Jindal, "MFZ-KNN—A modified fuzzy based K nearest neighbor algorithm," in *Proc. Int. Conf. Cogn. Comput. Inf. Process. (CCIP)*, 2015, pp. 1–5, doi: 10.1109/CCIP.2015.7100689.
- [34] Z. Lai, Z. Yao, C. Wang, H. Liang, H. Chen, and W. Xia, "Fingertips detection and hand gesture recognition based on discrete curve evolution with a Kinect sensor," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2016, pp. 1–4.
- [35] M. Maisto, M. Panella, L. Liparulo, and A. Proietti, "An accurate algorithm for the identification of fingertips using an RGB-D camera," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 3, no. 2, pp. 272–283, Jun. 2013.
- [36] Y. Liu, Y. Yin, and S. Zhang, "Hand gesture recognition based on HU moments in interaction of virtual reality," in *Proc. 4th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, vol. 1, Aug. 2012, pp. 145–148.
- [37] T. Chehlarova and K. Chehlarova, "Managing Pepper's Ghost illusion using intelligent methods," in *Proc. IEEE 10th Int. Conf. Intell. Syst. (IS)*, Aug. 2020, pp. 415–420.



**ZAINAL ABDUL KAHAR** received the Diploma and B.Sc. degrees in computer science from Universiti Teknologi Malaysia (UTM), in 2002 and 2009, respectively, and the master's degree in computer science from Universiti Putra Malaysia (UPM), in 2015, where he is currently pursuing the Ph.D. degree in multimedia computing. From 2002 to 2015, he worked as a Network Engineer with Telekom Malaysia. He is also an App and Game Developer with Game Taiko and has

been involved in designing and developing multimedia content for events, mobile, and computer games for more than six years. His research interests include machine learning, hologram, virtual reality, augmented reality, game engines, and mobile application development.





**PUTERI SUHAIZA SULAIMAN** received the degree in science (computer) and the M.Sc. degree in computer science from Universiti Teknologi Malaysia (UTM) and the Ph.D. degree in computer graphics from Universiti Putra Malaysia (UPM). She is currently a Senior Lecturer and an Associate Professor with UPM. She has more than 15 years of experience in the field of computer graphics, computer vision, information visualization, and advanced interaction. Her current research explores the realm of virtual and mixed reality, focusing on the interaction and visualization elucidation. She has been involved in designing and developing computer games for children and people with special needs.



**FATIMAH KHALID** received the master's degree from Universiti Kebangsaan Malaysia (UKM), in 1977, the B.Sc. degree in computer science from Universiti Teknologi Malaysia (UTM), in 1992, and the Ph.D. degree in system science and management from UKM, in 2008. From 1993 to 1995, she worked as a System Analyst with UKM. After 1977, she started involved in teaching with Sal College, until 1999, and continued with Universiti Putra Malaysia, in June 1999, where she is currently working with the Faculty of Computer Science and Information Technology, as a Lecturer and an Associate Professor.



**AZREEN AZMAN** (Member, IEEE) received the Diploma degree in software engineering from the Institute of Telecommunication and Information Technology, in 1997, the Bachelor of Information Technology degree in information systems engineering from Multimedia University, Malaysia, in 1999, and the Ph.D. degree in computing science from the University of Glasgow, Scotland, in September 2007, specializing in information retrieval. After serving in the industry for a few years, he enrolled for the Ph.D. degree. He is currently an Associate Professor with the Universiti Putra Malaysia. His current research interests include information retrieval, text mining, natural language processing, and intelligent systems. He serves as a Committee Member for the Malaysian Information Technology Society (MITS) and the Malaysian Society of Information Retrieval and Knowledge Management (PECAMP). He is an Active Member of ACM.

...