[1]Ashutosh Mohite
[2]Akhilesh A. Waoo

Efficient Model for Enhancing Hand Gesture Identification Using Machine Learning with Principal Component Analysis (PCA) Feature Selection Technique

Journal of Electrical Systems

Abstract: - Hand gesture recognition and classification play a vital role in human-computer interaction, offering intuitive and natural means of communication and control in various applications. This research investigates an innovative approach to build a model for the hand gesture image classification and enhance the accuracy and efficiency of hand gesture recognition systems using a machine learning-based method coupled with Principal Component Analysis (PCA). The proposed methodology aims to address the challenges associated with variability in hand gestures, environmental conditions, and computational complexity. The research leverages different machine learning techniques like Stochastic Gradient Descent (SGD), K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF) and Gaussian Naïve Bayes (GNB) algorithm sand compares the classification performance with each other. Additionally, Principal Component Analysis (PCA) is employed for dimensionality reduction, extracting the most salient features from high-dimensional hand gesture data while preserving crucial information. The proposed model is trained and evaluated on the dataset, comparing its performance with different machine learning based classifiers which are used with the PCA feature selection technique. Furthermore, the impact of PCA-based feature selection on classification accuracy and computational efficiency is thoroughly analyzed. Using the Leap gesture dataset, the suggested method achieves remarkable results with an accuracy of 97.48% by KNN and also RFT and GNB achieve 99.99% and 82.09%, respectively. According to the experimental findings, the recommended PCA-based feature selection technique beat other classifiers in terms of F-measure, accuracy, recall, and precision.

Keywords: Hand Gesture Recognition, Principal Component Analysis (PCA), K-Nearest Neighbor, Human-Computer Interaction (HCI).

1. INTRODUCTION

Hand gestures play a significant role in Human-Computer Interaction (HCI) by providing an intuitive and natural means of communication between users and computing systems. Hand gestures are essential whenever there is a need to establish communication between a normal human and a deaf human [17]. Hand gestures are often used in combination with other input modalities, such as voice commands, touchscreens, or eye tracking, to create multimodal interfaces. Integrating multiple input modalities enhances the richness and flexibility of user interactions, catering to diverse user preferences and abilities. Gesture-based interfaces can increase accessibility for users with disabilities, particularly those with mobility impairments or conditions that limit their ability to use traditional input devices. Gesture recognition systems can enable hands-free interaction, empowering users to access and interact with computing devices more independently. Hand gesture recognition is a dynamic field within computer vision and human-computer interaction that focuses on interpreting and understanding hand movements in real-time. It involves converting hand movements which are recorded by different input devices, including cameras or sensors, into meaningful instructions or actions. Hand gesture recognition is a dynamic field within computer vision and human-computer interaction that focuses on interpreting and understanding hand movements in real-time. It enables machines to understand and respond to gestures made by humans, facilitating intuitive interaction between humans and computers, particularly in applications such as sign language recognition, virtual reality, human-computer interaction, and robotics. The primary objective of hand gesture classification and recognition is to enable natural and intuitive interaction between humans and machines, facilitating tasks ranging from controlling electronic devices to virtual reality environments. This technology finds applications in diverse domains including sign language recognition, gaming, robotics, healthcare, and user interface design. Gestures allow users to convey rich and nuanced information beyond simple commands. Hand movements can express emotions, convey spatial relationships,

[1] [1]Research Scholar, Department of Computer Science Engineering & IT, AKS University, Satna, M.P., India

[2]Department of Computer Science Engineering & IT, AKS University, Satna, M.P., India

and provide context to interactions in ways that traditional input methods cannot. This can enhance communication and collaboration in various applications, such as virtual environments and design software.

The primary objective of hand gesture classification and recognition is to enable natural and intuitive interaction between humans and machines, facilitating tasks ranging from controlling electronic devices to virtual reality environments. This technology finds applications in diverse domains including sign language recognition, gaming, robotics, healthcare, and user interface design. In the proposed work Principal Component Analysis (PCA) is frequently used as a preprocessing step before implementing machine learning techniques. By lowering data dimensionality and feature selection, PCA can boost the efficiency and effectiveness of subsequent modeling techniques. It also helps to minimize the curse of dimensionality, which occurs when the number of features increases compared to the number of observations, causing machine learning algorithms' performance to decrease. PCA is significant in feature selection because it allows you to reduce the dimensionality of datasets while maintaining important information, improve computing efficiency, reduce noise, handle multicollinearity, and make the data more interpretable. The proposed system perform the different steps of the data preprocessing and classification using the concepts of machine learning techniques on the different 10 phases of hand gesture. The work's initial stage illustrates the image pre-processing, which includes procedures for image reshaping, image normalization, and image flattening. Principal Component Analysis (PCA), a feature extraction/selection method, is also used to select a fix number of features from the dataset.

The contributions of this paper are as follows. Firstly a hand gesture image dataset is collected namely “Leap Hand Gesture Dataset” which contains 10 classes of hand gesture where each class represents different gesture shape i.e. Fist, I, Palm, Fist moved, Palm moved, Thumb, Index, OK, C, Down. The dataset has also been made publically available. Secondly the data preprocessing has been performed in three steps i.e. Reshaping, Normalization and Flattening the image data. Thirdly, for the feature selection, Principal Component Analysis (PCA) is employed. PCA selects the optimal features from the total 9600 features exist in the image data of the dataset.

2. RELATED WORKS

Related works helps to construct a conceptual framework or theoretical framework for the study. By integrating insights from prior research, researchers develop a theoretical framework that guides their understanding of the topic and informs the interpretation of their findings.

The first step in any system that recognizes hand gestures is hand detection and tracking. A color histogram-based hand tracking model was proposed by Comaniciu et al. (2003) [5]. To find and track the hand in the video sequences, they employed the color histogram of the detected hand as the mean shift input. However, the model's shortcoming was that it could not identify the hand when the background and the item had the same hue. Similar to this, skin-colored information was employed by Chai et al. (1999) and Wang et al. (1997) to identify hands[6][7]. For segmentation, the YCbCr color space model was employed. In order to track hands in a variety of demanding environments, Asaari et al. incorporated an adaptive Kalman filter with eigenhand[8]. However, the algorithms failed in the presence of large-scale fluctuations and position changes. Kakizaki et al. (2024) proposed a dynamic JSL recognition system using machine learning, efficient feature extraction, and hand pose estimation. They used RGB cameras to collect video of JSL movements and MediaPipe to estimate hand pose. Four different feature types were suggested, allowing the same feature generating technique to be applied regardless of frame count or dynamic nature. A Random Forest (RF)-based feature selection method was used to choose features, and the reduced features were sent into a Support Vector Machine (SVM) classification algorithm [3]. As hand gesture recognition is a challenging problem for research, Malima et al. (2024) developed a straightforward, efficient procedure for gesture recognition by taking into account a fixed set of manual commands and a reasonably structured environment. They demonstrated the efficiency of their method by classifying the gesture, segmenting the hand region, and locating the fingers. The method is also invariant to translation, rotation, and hand scale [4]. Hand and finger gestures for the natural user interface are described by Lee et al. (2020)[9]. The author points out that while hand gestures are a basic form of communication, they are not organic. He concentrates more on tracking and identifying fingers. The user is able to monitor and recognize hand motions and finger movements with Kinect and Depth-Sense, even in situations where there is insufficient lighting or a strong background. They also recognized hand gestures and identified fingers using Kinect depth data. Thus, the model developed by the authors can offer an interface and natural communication. Lee, K.H. et al. (2022) use machine learning to build EMG-based hand/finger motion classifiers with fixed electrode location. Customized classifiers for 10 gestures were created using an artificial neural network (ANN), a support vector machine (SVM), a random forest (RF), and a logistic regression. To create the classifiers and identify the ten gestures in each subject dataset, the four machine learning algorithms stated above were used: ANN, SVM, RF, and LR. One of the objectives of this model was to determine whether typical TD characteristics might be used in an EMG-based hand/finger gesture detection system to create a multi-class classification model using an ANN. The accuracy of the ANN-based classifiers had the lowest inter-subject variance, indicating that this method was least affected by individual variability [14]. Gupta R, et al. (2021) He demonstrated a project that uses data from various surface electromyograms, as well as gyroscopes and accelerometers on signers' forearms, to classify 50 signs from the Indian sign language [11]. The use of ensemble machine learning methods to build a novel multistage classification of signs is proposed. To begin, a binary classifier is used to detect whether the sign is in a static position or has dynamic hand motion. The sign is then classified using one of two multi-class classifiers, both of which have been trained to differentiate between static and dynamic signals. The random forest (RF) and extreme gradient boosting machine were applied to select key features for categorizing signs from the two groups. The proposed RF-based classification technique enhances the accuracy of static sign classification to 98%, surpassing the performance of a single classifier.

To discover the most efficient model, Habib, A. et al. (2021) investigated a number of classical classification methods and deep learning classification techniques with a variety of parameters. The results show that tree-based classification techniques and LSTM are more effective in classifying EMG data. In this study, they used multiple supervised machine learning approaches to classify electromyography data into seven different human hand movements. They discovered that, when all aspects are taken into account, random forest is the most effective machine learning technique. Furthermore, LSTM is the most effective deep learning technique. With 99.43% accuracy and the lowest misclassification error when all criteria are considered, the Random Forest classifier is the best model. Furthermore, LSTM has a high accuracy of 99.19% and a minimal misclassification error [12]. Amin, P. et al. (2021) investigate the creation of a classifier that may be used to categorize motions based on Myoelectric data obtained from the Myo-armband. This study demonstrates how to use an Artificial Neural Network to collect data, extract characteristics, and classify motions offline. Finally, the findings are compared to a Support Vector Machine Classifier. The suggested approach obtains an accuracy of more than 94% on the validation data set for categorizing five different hand motions. This technology might be used in human-machine interfaces with five different control signals, including rest, it was discovered [15].Luo, Y. et al. (2021) proposed a segmentation method for gestures. This method on image object by using skin color detection, marker-based watershed algorithm, and seed filling algorithm for the preprocessing of the data and they achieve the 96.46% of recognition success rate by using AlexNet convolutional neural network. In other words the accuracy shows that the proposed method can recognize all kind of gestures accurately with very less time [16].

3. MATERIALS & METHODOLOGY

In this proposed work multiple machine learning algorithms are used for the classification operation and compared with and without the feature selection method i.e. Principal Component Analysis (PCA). The general workflow for the suggested hand gesture identification system is displayed in Figure 1.It includes hand gesture acquisition, gesture image processing, feature selection and classification of sign language. The ensuing subsections include information on each step. Here the entire workflow is represented by Algorithm-I and Algorithm-II. Algorithm-I showing the process without feature selection whereas Algorithm-II showing the process with feature selection method i.e. PCA.

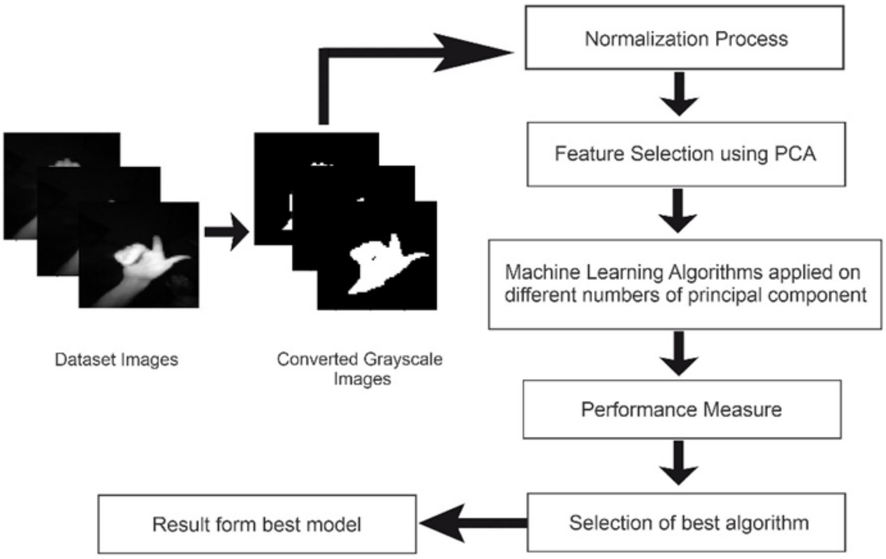


Figure-1 General workflow of the proposed work

Algorithm-I

```
// Training Phase
Input: Leap Gesture Dataset (Training Data)
Output: Knowledge Acquisition

Start
    Preprocess the images:
        Reshaping the image data;
        Normalize the data;
        Flattening the data;
    Perform the process with all existing features of image data;
    For all preprocessed training data: do
        Feed the data to train the classifiers;
    End;
End;

// Testing Phase
Input: Leap Gesture Dataset (Test Data)
Output: Gesture Image Classification
Start
    Preprocess the images:
        Reshaping the image data;
        Normalize the data;
        Flattening the data;
    Perform the process with all existing features of image data;
    For all preprocessed test data: do
        Perform classification task using classifiers;
        Collect the classification results from the classifiers;
        Compare training and test data samples:
         Select the most prevalent outcome and evaluate the performance;
          End;
End;
```

Algorithm-II

```
// Training Phase
Input: Leap Gesture Dataset (Training Data)
Output: Knowledge Acquisition

Start
    Preprocess the images:
        Reshaping the image data;
        Normalize the data;
        Flattening the data;
   Apply PCA for feature selection:
   Select the principal component from 10 to 100 (with the interval of 10);
    For all preprocessed training data: do
        Feed the data to train the classifiers;
    End;
End;

// Testing Phase
Input: Leap Gesture Dataset (Test Data)
Output: Gesture Image Classification
Start
    Preprocess the images:
        Reshaping the image data;
        Normalize the data;
        Flattening the data;
   Apply PCA for feature selection:
   Select the principal component from 10 to 100 (with the interval of 10);
    For all preprocessed test data: do
        Perform classification task using classifiers;
        Collect the classification results from the classifiers;
        Compare training and test data samples:
         Select the most prevalent outcome and evaluate the performance;
         End;
End;
```

3.1 Dataset

In the multidisciplinary fields of artificial intelligence and machine learning, computer vision deals with the autonomous extraction, interpretation, and analysis of meaningful information from images. The amount of digital content, particularly photos and videos, is growing at an exponential rate, thanks to recent technological breakthroughs. Understanding and analyzing images is a critical task in computer vision, where computers face greater challenges than humans. Therefore, human assistance will be used to help classify the images. Images with the dark background containing 10 different hand pose of the Leap Gesture Dataset are used into the proposed work.

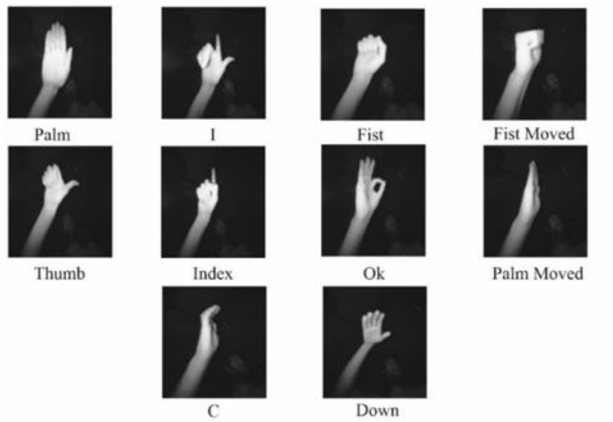


Figure-2 Sample images of the Dataset contains 10 different hand gesture

Dataset contained 10 static gestures (I, Index, Fist, Fist moved, Palm, Palm Moved, OK, Thumbs, C, Down) to recognize. Each class has 2000 images for training and 2000 images for testing purpose. So total number of images is 20,000. Sample of finalized dataset is provided on Figure-1, which shows the 10 different hand gesture phases. For the proposed work the dataset is split into training and test data samples in 70:30 ratios. Means that 70% of data is used to train the models and remaining 30% data used for the classification.

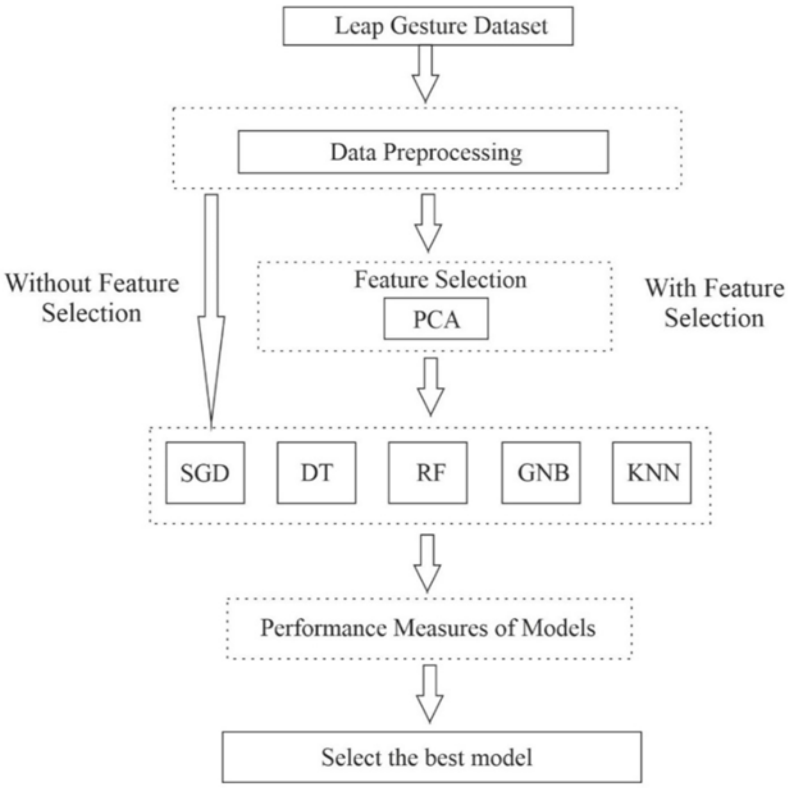


Figure-3 System Architecture of the Proposed Work

3.2 System Architecture

A supervised learning based classification strategy is used for the proposed work. Here in this work Principal Component Analysis (PCA) is also employed for the feature selection. Data preprocessing and normalization phase normalize the data and before training the models and the data was shuffled randomly. After normalization, classification task has performed using five different machine learning methods i.e. Stochastic Gradient Descent (SGD), Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB) and K-Nearest Neighbor (KNN). In the next phase of experiment classification task has performed with all the classifiers with feature selection technique i.e. Principal Component Analysis (PCA). PCA selects the feature components i.e. principal components from the dataset images. Here 70 components was selected from the total 9600 existing components that optimizes the desired metric of image data. As the 70 component gives the best result than the other number of components for which we perform the entire work from 10 to 100 number of components with each machine learning algorithms. Once we concatenated these features, machine learning approaches was applied for classification and analysis. The block diagram of the proposed work is shown in above Figure 3.

3.3 Preprocessing

In order to prepare raw data for the model to be trained from, preprocessing is an essential step in machine learning. The kind of data being utilized and the particular needs of the machine learning algorithm are major factors in the selection of preprocessing approaches. First, the dataset images are transformed into binary images at this step thresholding is a straightforward technique for image segmentation that works well. It changes a picture to a binary format, in which each pixel is either 255 or 0. The concept of thresholding is to choose a value, after which all pixel values are set to 255 (white) for all values above the threshold and 0 (black) for all values below it. To start, the picture is transformed onto a grayscale image that simplifies the process and cuts down on processing. This grayscale picture is represented as $(i,)$, where i stands for the coordinates of each pixel. This approach makes use of a single, fixed threshold value that may be determined automatically using Otsu's approach or manually established. In the proposed work, the threshold is automatically determined using Otsu's approach.

Using the image's histogram, Otsu's approach automatically determines the ideal threshold value. The main goal of Otsu's approach is to increase the inter-class variation as indicated by:

$$\sigma_b^2(t) = w_1(t) \cdot w_2(t) \cdot [\mu_1(t) - \mu_2(t)]^2$$

Where $\sigma_b^2(t)$ is the inter-class variance at threshold t, $w_1(t)$ and $w_2(t)$ are the probabilities of the two classes separated by the threshold t and $\mu_1(t)$and $\mu_2(t)$ are the means of the two classes separated by the threshold t.

After thresholding, the pixels are either 0 or 255:

$$if\ I(i,j) \leq threshold: I(i,j) = 0$$

$$if\ I(i,j) > threshold: I(i,j) = 255$$

The data then has to be reshaped and flattened. The dataset image is shrunk to a fixed size of 60×160 pixels from its original resolution of 240×640 pixels. Arrays are created from the grayscale source photos. Each image with 60 by 160 pixels is transformed into a one-dimensional array with the length 60 by 160 = 9600.

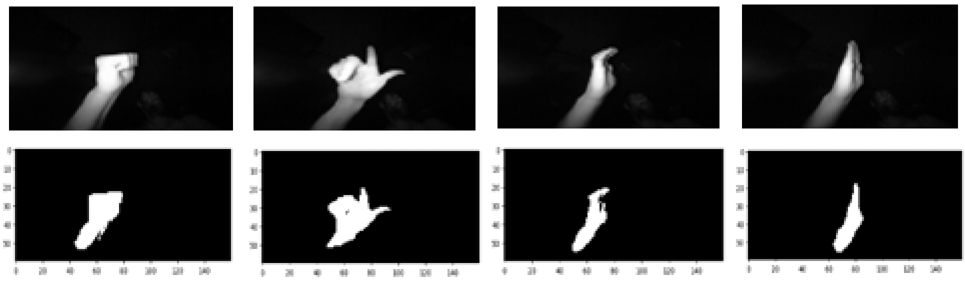


Figure-4Images showing different hand gestures along with the their normalized binary images

Preprocessing was carried out in three stages, viz., reshaping, normalization, and flattening the data. The raw data image is resized to a fixed size of 60×160 pixels, as was previously indicated. Each 60×160 pixel image is transformed into a one-dimensional array with a length of 60*160=9600 from the original grayscale image. This flattening is commonly used to transform photos into a format that can be fed into one-dimensional array-based machine learning techniques. Normalization is the last phase in the preprocessing process. The training picture's pixel value is normalized between 0 and 1 as a result of the normalization procedure, which reduces the pixel values on the image data to the range [0, 1].

3.4 Feature Selection using PCA

A subset of pertinent features, such as variables or predictors, is chosen during the feature selection process in order to construct a model. It minimizes training times, increases abstraction, and decreases overfitting, all of which contribute to the model to improve its performance. In this work, feature selection and feature space presentation were handled using PCA. The high-dimensional image data can be reduced in smaller dimensions by PCA. PCA increases the visibility of the image dataset by reducing the dimension of image characteristics to two or three dimensions, which facilitates a better understanding of the fundamental framework and relationships between images. By keeping just the top k principle components that are responsible for the majority of the variation in the data, it decreases dimensionality. It also removes some of the dataset's dimensions while keeping the most important information by mapping the data to this reduced-dimensional space. PCA serves to keep as much variety in a dataset as is possible while reducing its complexity. It may decrease the depth of data from n to k degrees while keeping most of the information by selecting the first k main components, where $k<n$.

3.5 Stochastic Gradient Descent (SGD)

Logistic regression is a type of regression analysis used for predicting the probability of a binary outcome. It models the relationship between a binary dependent variable and one or more independent variables by estimating probabilities using a logistic function. The logistic function, also known as the sigmoid function, is defined as:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-z}}$$

Here, $P(y = 1 \mid x)$is the probability that the dependent variable y is equal to 1 given the input features$x$, and $z$is a linear combination of the input features and their corresponding weights. Mathematically, $z = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$where w_0 is the intercept term and $w_1, w_2 \ldots w_n$are the coefficients (weights) associated with the input features $x_1, x_2 \ldots x_n$ respectively. At each iteration, SGD updates the parameters (weights) of the model by taking a small step in the direction of the negative gradient of the objective function with respect to the parameters. The update rule for the weights in logistic regression using SGD is given by:

$$w_i + 1 = w_i - \eta \cdot \nabla J(w_i)$$

Here, $w_i$are the current weights, η is the learning rate (step size), and $\nabla J(w_i)$ is the gradient of the logistic loss function with respect to the weights at iteration i. During training, the SGD algorithm iteratively updates the model parameters (weights) to minimize the logistic loss function. The logistic loss function for binary classification is defined as:

$$J(w) = -\frac{1}{N}\sum_{i=0}^{n}[y_i log(\hat{y_i}) + (1 - y_i)log(1 - \hat{y_i})]$$

Here, N is the number of samples, y_i is the true label (0 or 1) of sample i, $\hat{y_i}$ is the predicted probability that sample i belongs to class 1, and the summation is taken over all samples. The goal is to minimize this loss function by adjusting the model parameters. Once the model is trained, its performance is evaluated using metrics such as accuracy and Hamming loss.

3.6 K-Nearest Neighbor (KNN)

KNN is a simple yet effective algorithm used for classification and regression tasks. It operates based on the assumption that similar data points exist in close proximity to each other. In classification tasks, the algorithm assigns a class label to a data point based on the majority class among its k nearest neighbors, where k is a user-defined parameter. The KNN algorithm relies on measuring the distance between data points. In this work, Minkowski Distance which is a generalization of Euclidean and Manhattan distances, parameterized by the order 'p'. When 'p' is set to 2, Minkowski distance becomes Euclidean distance. During training, the algorithm computes distances between data points using chosen distance metrics. The algorithm identifies the k nearest neighbors by calculating their distances from the query point and selecting the k points with the smallest distances and applies the majority voting scheme to assign a class label.

3.7 Decision Tree

A Decision Tree is a tree structure similar to a flowchart, with an internal node representing a feature or attribute, a branch representing a decision rule, and each leaf node representing an outcome or class label. During the training phase, the Decision Tree algorithm chooses the best feature to split the data at each node using a criterion. Node purity is often assessed using measures such as Gini impurity or entropy. In this work, entropy is employed. Entropy is a measure of impurity or disorder in a set of data points.

Mathematically, the entropy of a set S with respect to a binary classification problem is given by:

$$Entropy(S) = -\sum_{i=1}^{c} p_i log_2(p_i)$$

where c is the number of classes, and $p_i$is the proportion of instances in class $i$in set S.

The algorithm traverses the decision tree from the root to a leaf node for each new data point using the data point's feature values. The shown class label represents the majority of the instances in the leaf node. Information gain is the reduction in entropy gained by separating a set of data points based on a specific attribute. It is determined as the difference between the parent node's entropy and the weighted average of the offspring nodes' entropies following the split. The feature with the greatest information gain is selected for splitting at each node.

Decision trees are used in image classification to divide images into many groups based on characteristics and pixel values. Based on the values of pixels, color channels, appearances, and patterns, decision trees identify the most important characteristics of an image. The algorithm chooses the optimal split to divide images into different categories. The algorithm classifies images by making a sequence of decisions at each node.

3.8 Gaussian Naïve Bayes (GNB)

Naive Bayes Algorithm is a machine learning technique is based on the Bayes theorem and supposing feature independence. This approach determines the likelihood that an image falls into a specific class in image classification by looking for specific features or patterns in the image data. The technique analyzes at an image's pixel values and calculates the probability that these values belong to several classes. Naive Bayes takes into account the likelihood of each attribute separately, which streamlines the computation and increases computing efficiency. The class having the highest probability is then chosen by the algorithm to be the image's predicted label. This algorithm determines the probability that each pixel value in an image belongs to a number of classes by analyzing its values. Because Naive Bayes considers the probability of each attribute independently, it simplifies calculations and boosts computational performance. This algorithm then selects the class with the greatest probability to be predicted label for the image.

3.9 Random Forest (RF)

The random forest method is a well-known and effective technique for image classification problems in the fields of artificial intelligence and machine learning. The random forest algorithm, which builds several decision trees during training and outputs the mode of the classes providing the prediction. Every tree in the forest is trained separately using a random subset of the training data according to the bagging principle. The wide range and unpredictability of the trees contribute to a decrease in overfitting and an increase in the model's overall accuracy. In this work, total 100 numbers of tree defines in the random forest algorithm and the maximum depth of each tree is 15.A node's "purity" is frequently assessed using metrics like entropy (information gain) or Gini impurity. Both entropy (information gain) and Gini impurity are defined by

$$Gini\ (D) = 1 - \sum_{i-1}^{C} (Pi)^2$$

where $Pi$is the proportion of instances of class i in the dataset$D$.

$$Entropy\ (D) = -\sum_{i-1}^{C} Pi\ log_2\ (Pi)$$

Several subgroups of the training data are created using random forests using a method called bagging and each subset is then used to train a different decision tree. The random forest generates predictions by combining the predictions of each individual tree once they have all been trained. This is often accomplished by majority vote in categorization work. The classifier gains knowledge from the training data by fitting the model to the data. The judgments made by the 100 trees are combined to provide predictions. Suppose $T_1, T_2, \ldots. T_k$ are the key decision tree in the forest and for a new input x, each treeT_i, outputs a class prediction $\hat{y_i}$ and finally the final prediction $\hat{y}$ is given by the class prediction$\hat{y} = (\widehat{y_1}, \widehat{y_{2,}} \ldots. \widehat{y_k})$.

Five Different classifiers employed here with certain parameters to design the model. These model parameters are shown in the Table-1.

Table-1Parameters of different classifiers used in the work

Method	Parameters
Stochastic Gradient Descent	Loss : log, Random state : 201
K-Nearest Neighbor	Number of neighbors : 60
Decision Tree	Depth : 8
Random Forest	Depth : 6, Estimators : 50
Gaussian Naïve Bayes	------

4. EXPERIMENTAL WORK & RESULTS

After the preprocessing steps applied on the dataset, the entire dataset is divided into training and testing datasets in 70:30 ratios. Which means training set contains approx. 14,000 images and test data set contains approx. 6,000 images. In the current work, the entire experiment was performed in two different phases. These phases are described in experiment-1 and experiment-2.

Experiment-1: In the first phase, classification task has performed on the image data without any feature selection technique. The dataset have total 9600 features so that the classification performed on all 9600 features. The parameters for the performance evaluation are shown in the Table-2. Here from the Table-2, we can observe that all the classifiers perform well except the Gaussian Naïve Bayes method. The highest accuracy achieved by random forest i.e. 99.93% with the 0.06% loss. Classification performance is evaluated by precision, recall and f1-score parameters. These parameters of all classifiers shown in Figure-4 where color blue represents precision, green represents recall and yellow represents f1-score.

Table-2 Accuracy and loss values of classifiers without feature selection (with all 9,600 features)

Classifier	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss
Stochastic Gradient Descent(SGD)	99.10%	0.89%	97.88%	2.11%
K-Nearest Neighbor (KNN)	95.84%	4.15%	95.43%	4.56%
Decision Tree (DT)	94.10%	5.89%	92.48%	7.51%
Random Forest (RFT)	100%	0%	99.93%	0.06%
Gaussian Naïve Bayes (GNB)	39.79%	60.20%	39.08%	60.91%

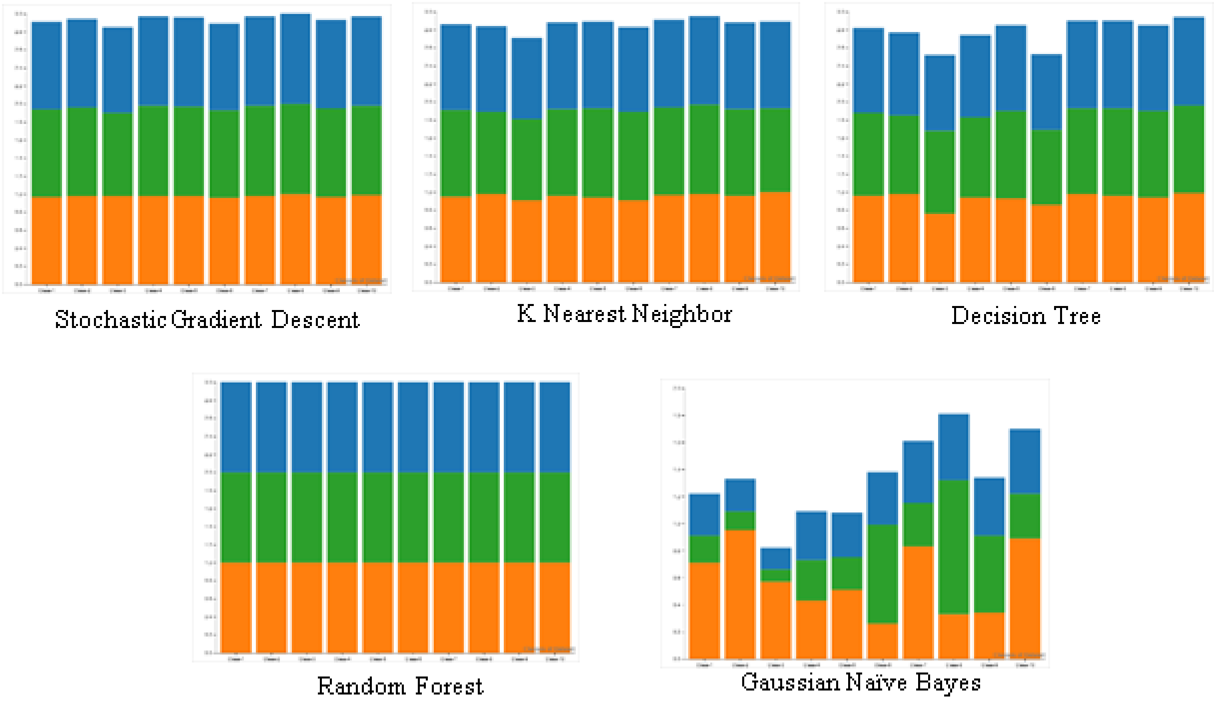


Figure 4 Plot for the precision, recall & f1-score values of different classifiers

Experiment-2: On the other hand, in the second phase of the experiment, PCA has applied as feature selection method. Here, PCA extracts the certain numbers of principal components from the image data without affecting its internal features and also reduce the dimensionality of the data. Here, the classification task has performed with the different number of features (principal components). Gesture image data has total 9,600 features. So we start process by selecting the number of components as 10, which means that using PCA with 10 component we have reduced 9,600 predictors to 10 without compromising on explained variance. This feature selection is applied on the training dataset. After we've performed PCA on the training set, then predicting test data using these components. Just like PCA components obtained on the training set, another bunch of components will get on the testing set. Finally, we train the model. This entire process has been performed in the iteration up to 100 components. Which means that, PCA was utilized with all the five different above discussed classifiers? Classification task has been performed multiple times with the different principal component 10, 20, 30 till 100. Performance of all classifiers measured in each step and the highest accuracy achieved at the component 70. Finally, here performance was evaluated by using the hyper parameters like training and testing accuracy with loss value. Table-3 represents results achieved at the 70 principal components.

Precision, recall, and f1-score metrics are used to assess classification performance. All of the classifiers' parameters are shown in Figure 5.Precision, recall, and f1-score metrics represented by blue, green and yellow respectively.

Table-3 Accuracy and loss values of classifiers with feature selection method PCA with only 70 components

Classifier	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss
Stochastic Gradient Descent+PCA	94.99%	5.00%	93.83%	6.16%
K-Nearest Neighbor (KNN)+PCA	97.65%	2.34%	97.48%	2.51%

Decision Tree +PCA	92.38%	7.61%	91.35%	8.65%
Random Forest +PCA	100%	0%	99.99%	0.01%
Gaussian Naïve Bayes +PCA	83.97%	16.02%	82.90%	17.10%

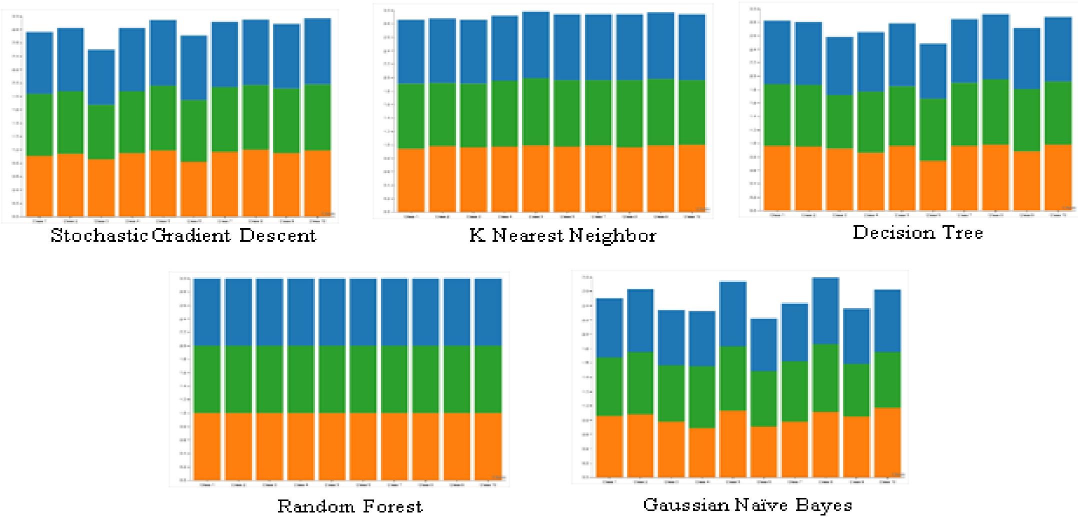


Figure 5 Plot for the precision, recall & f1-score values of different classifiers using PCA

From both the experiments we can conclude that when feature selection technique applied into the image data, our three models K-Nearest Neighbor, Random Forest and Gaussian Naïve Bayes classifiers performs batter than previous experiment in which no feature selection method is used. K-Nearest Neighbor and Random forest model already classified the gesture images with great accuracy but with PCA increases theirs accuracy values in second experiment. But Gaussian Naïve Bayes model achieved only 39.08% accuracy only without feature selection. Whereas in the experiment-2 Gaussian Naïve Bayes performed well and achieved 82.90% accuracy with 17.01% loss with PCA. K-Nearest Neighbor and Random Forest models little bit improve the results from the previous experiment and perform the classification on the gesture image dataset with 97.48% and 99.99% accuracy respectively which is a highest accuracy values achieved in this experiment. Now here is the comparison between these three models shown in the Figure-6. Performance parameters like precision, recall and f-1 score are graphically represented and measured by confusion matrix. Performance parameters are represented through confusion matrix shown in the Figure-7 for all three models.

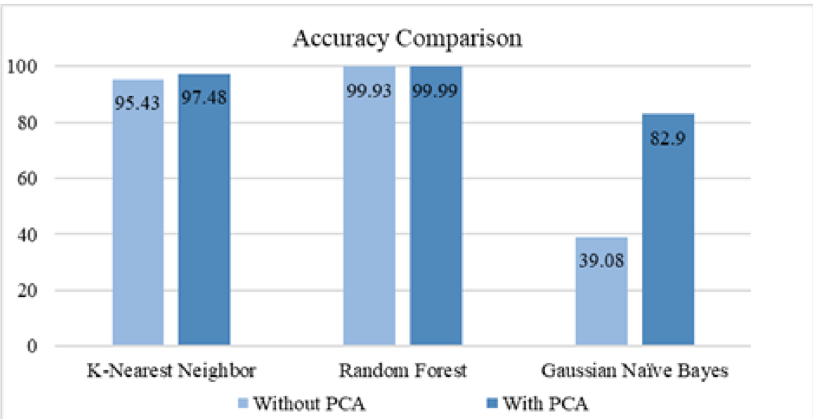
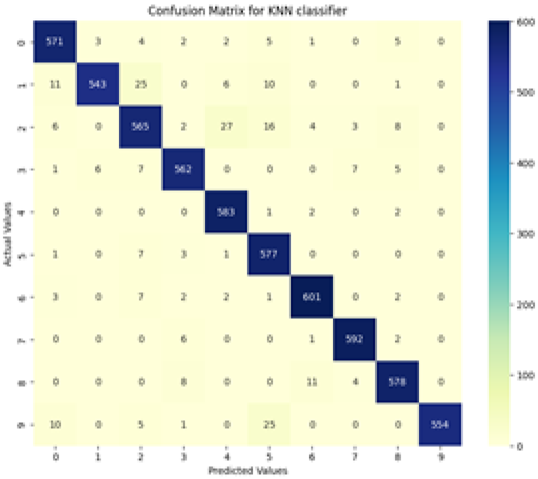
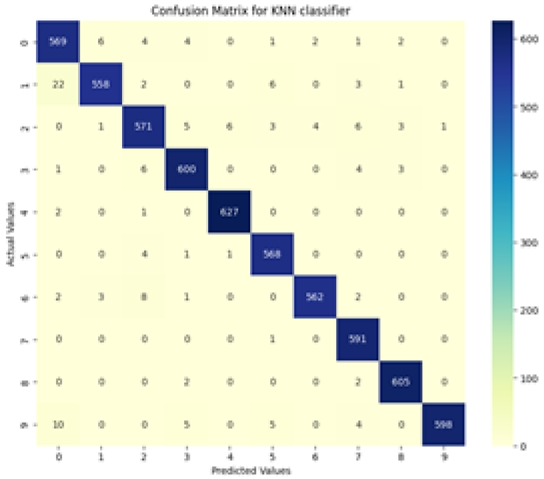


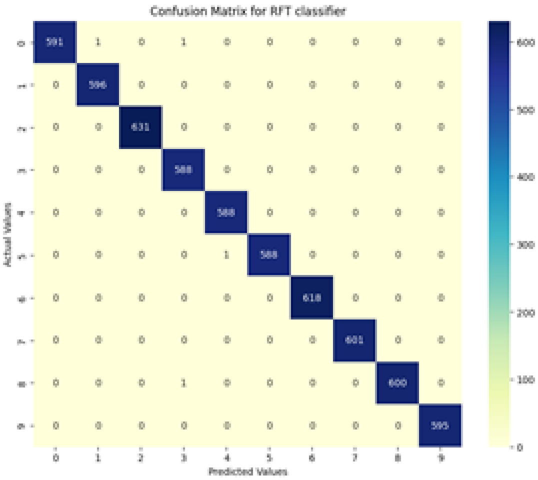
Figure 6 Plot showing comparison the accuracy values of different classifiers



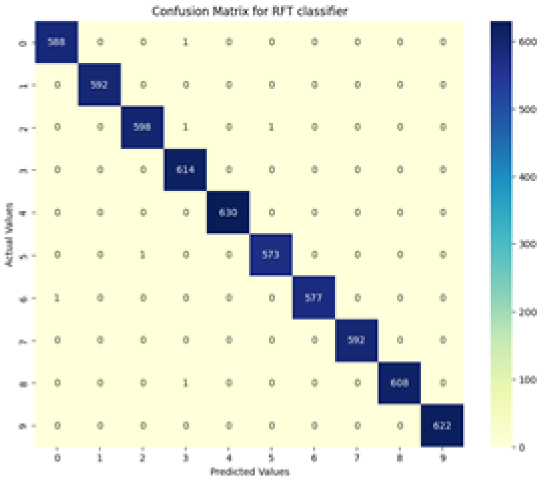
KNN without PCA



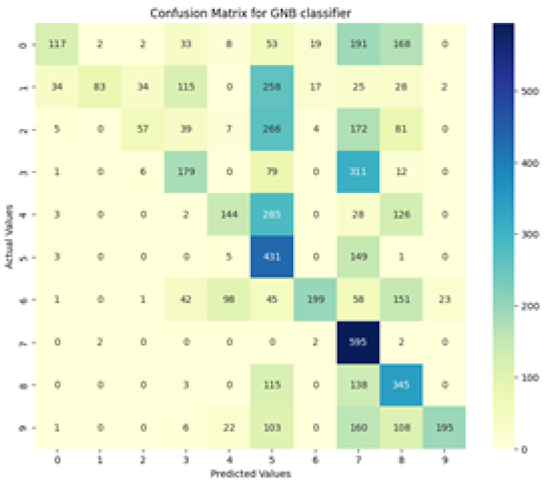
KNN with PCA



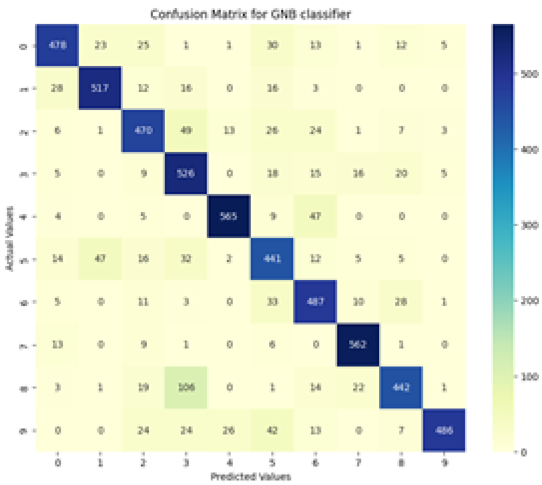
Random Forest without PCA



Random Forest with PCA



Gaussian Naïve Bayes without PCA



Gaussian Naïve Bayes with PCA

Figure 7Confusion Matrix of the models

5. CONCLUSION

In order to extract the specific features needed for human hand gesture detection from each group of sequential image frames, we presented a PCA-based classification model. The classification method was incorporated with these enhanced features. Interestingly, when employing the feature selection technique (PCA) for the Gaussian Naïve Bayes classifier, the models showed remarkable accuracy. PCA, K-Nearest Neighbor, Random Forest, and Gaussian Naïve Bayes all produced accuracy scores of 97.48%, 99.99%, and 82.90%, in that order. In conclusion, developing an efficient model to enhance hand gesture identification, particularly using feature selection techniques, holds significant promise for enhancing gesture-based interaction systems. This method improves the accuracy and efficiency of gesture detection by feature selection optimization, and it may be applied practically in various HCI domains.

6. FUTURE WORKS

The creation of a real-time gesture detection system is one of our future goals. More sophisticated feature selection methods, such as recursive feature deletion or evolutionary algorithms, as well as the incorporation of multimodal data sources, such as noise or depth information, could be future advances for the proposed study in order to achieve more accurate recognition. To improve the model for real-time use on devices with limited resources and implement it to dynamic scenarios, more research is required.

References

[1] Gates, G. (1972). The reduced nearest neighbor rule (corresp.). IEEE transactions on information theory, 18(3), 431-433.

[2] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[3] Kakizaki, M., Miah, A. S. M., Hirooka, K., & Shin, J. (2024). Dynamic Japanese Sign Language Recognition Throw Hand Pose Estimation Using Effective Feature Extraction and Classification Approach. Sensors, 24(3), 826.

[4] Asanterabi Malima, Erol Ozgur, and Mujdat Çetin, "A_Fast_Algorithm_For_Vision-Based Hand Gesture Recognition for Robot Control", available at., http://people.sabanciuniv.edu/mcetin/publications/malima_SIU06.pdf.

[5] Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):234–240

[6] Chai D, Ngan KN (1999) Face segmentation using skin-color map in videophone applications. IEEE Trans Circuits Syst Video Technol 9:551–564

[7] Wang H, Chang S-F (1997) A highly efficient system for automatic face region detection in MPEG video. IEEE Trans Circuits Syst Video Technol 7:615–628.

[8] Asaari MSM, Rosdi BA, Suandi SA (2014) Adaptive Kalman filter incorporated eigenhand (AKFIE) for real-time hand tracking system. Multimed Tools Appl 70(3):1869–1898.

[9] Li, H.;Wu, L.;Wang, H.; Han, C.; Quan,W.; Zhao, J. Hand Gesture Recognition Enhancement Based on Spatial Fuzzy Matching in Leap Motion. IEEE Trans. Ind. Inform. 2020, 16, 1885–1894. [CrossRef]

[10] Devineau, G.; Moutarde, F.; Xi, W.; Yang, J. (2018). Deep learning for hand gesture recognition on skeletal data. In: 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 106-113.

[11] Gupta, R.; Bhatnagar, A. S. (2021). Multi-stage Indian sign language classification with Sensor Modality Assessment. In: 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore, India. Vol. (1). pp. 18-22.

[12] Habib, A. B.; Ashraf, F. B.; Shakil, A. (2021). Finding Efficient Machine Learning Model for Hand Gesture Classification Using EMG Data. In: 2021 5th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT); IEEE Press: Dhaka, Bangladesh. pp. 1-6.

[13] Côté-Allard, U.; Fall, C. L.; Drouin, A.; Campeau-Lecours, A.; Gosselin, C.; Glette, K.; Gosselin, B. (2019). Deep learning for electromyographic hand gesture signal classification using transfer learning. IEEE transactions on neural systems and rehabilitation engineering, Vol. 27(4), pp.760-771.

[14] Lee, K.H.; Min, J.Y.; Byun, S. (2022). Electromyogram-Based Classification of Hand and Finger Gestures Using Artificial Neural Networks. Sensors, Vol.22 (1), pp.-225.

[15] Amin, P.; Khan, A. M.; Bhat, A. R.; Rao, G. (2021). Feature Extraction and Classification of Gestures from Myo-Electric Data Using a Neural Network Classifier. In Evolution in Computational Intelligence. Springer.Singapore.pp-65-80.

[16] Luo, Y.; Cui, G.; Li, D. (2021). An Improved Gesture Segmentation Method for Gesture Recognition Based on CNN and YCbCr. Journal of Electrical and Computer Engineering, 2021.

[17] Chakraborty, D.; Garg, D.; Ghosh, A.; Chan, J.H. Trigger Detection System for American Sign Language using Deep Convolutional Neural Networks. In Proceedings of the 10th International Conference on Advances in Information Technology, Bangkok, Thailand, 10–13 December 2018; Association for Computing Machinery: New York, NY, USA, 2018; p. 4.

[18] Mohite, A.; Waoo, A. A. (2022). Analysis of Supervised Machine Learning Classifier Techniques used in Gesture Classification. In THEETAS 2022: In: Proceedings of The International Conference on Emerging Trends in Artificial Intelligence and Smart Systems, THEETAS 2022, 16-17 April 2022, Jabalpur, India (p. 174). European Alliance for Innovation.