

Article

An Experimental Analysis of Various Machine Learning Algorithms for Hand Gesture Recognition

Shashi Bhushan ^{1,*}, Mohammed Alshehri ^{2,*}, Ismail Keshta ³, Ashish Kumar Chakraverti ⁴,
Jitendra Rajpurohit ¹ and Ahed Abugabah ⁵

¹ School of Computer Science, University of Petroleum & Energy Studies, Dehradun 248001, India; jiten_rajpurohit@yahoo.com

² Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Majmaah 11952, Saudi Arabia

³ Computer Science and Information Systems Department, College of Applied Sciences, AlMaarefa University, Riyadh 12483, Saudi Arabia; imohamed@mcst.edu.sa

⁴ Department of Computer Science & Engineering, School of Engineering and Technology, Sharda University, Greater Noida 201310, India; ashish.me08@gmail.com

⁵ College of Technological Innovation, Zayed University, Abu Dhabi Campus, Abu Dhabi P.O. Box 144534, United Arab Emirates; ahed.abugabah@zu.ac.ae

* Correspondence: tyagi_shashi@yahoo.com (S.B.); ma.alshehri@mu.edu.sa (M.A.)

Abstract: Nowadays, hand gestures have become a booming area for researchers to work on. In communication, hand gestures play an important role so that humans can communicate through this. So, for accurate communication, it is necessary to capture the real meaning behind any hand gesture so that an appropriate response can be sent back. The correct prediction of gestures is a priority for meaningful communication, which will also enhance human–computer interactions. So, there are several techniques, classifiers, and methods available to improve this gesture recognition. In this research, analysis was conducted on some of the most popular classification techniques such as Naïve Bayes, K-Nearest Neighbor (KNN), random forest, XGBoost, Support vector classifier (SVC), logistic regression, Stochastic Gradient Descent Classifier (SGDC), and Convolution Neural Networks (CNN). By performing an analysis and comparative study on classifiers for gesture recognition, we found that the sign language MNIST dataset and random forest outperform traditional machine-learning classifiers, such as SVC, SGDC, KNN, Naïve Bayes, XG Boost, and logistic regression, predicting more accurate results. Still, the best results were obtained by the CNN algorithm.

Keywords: hand gesture recognition; machine learning; convolutional neural networks; sign MNIST



Citation: Bhushan, S.; Alshehri, M.; Keshta, I.; Chakraverti, A.K.; Rajpurohit, J.; Abugabah, A. An Experimental Analysis of Various Machine Learning Algorithms for Hand Gesture Recognition. *Electronics* **2022**, *11*, 968. <https://doi.org/10.3390/electronics11060968>

Academic Editors: Piyush Kumar Shukla, Manoj Kumar, Xiaochun Cheng and Prashant Kumar Shukla

Received: 1 February 2022

Accepted: 10 March 2022

Published: 21 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Every day, the need for and the level of services required by humans increase. In face-to-face communication, hand gestures are a key element. So, the body language of a human plays an important role in face-to-face communication, and so does making hand gestures. In communication, most things are explained with hand gestures, and analysis of this provides some insights into communication itself. However, current automation in this area does not focus on the use of hand gestures in daily activities, as is explained in [1,2]. Household items can also be controlled by hand gestures [3]. We are now moving towards an era in which everything can be controlled by hand gestures. Technology is advancing so much in this area that the complexity of operations of various computer programs and user interfaces is provided to the user. To make this system easy to understand and less complex, now, image processing is used. Whenever communication has to be established between a normal human and a deaf human [4], there is a strong need for hand gestures. To make a system smart, there is a need to enter hand gesture images into the system and conduct further analysis to determine their meaning [5]. For this, machine-learning algorithms

need to be applied. This can be undertaken by first training models or algorithms using the dataset (available images), and then conducting classification or categorization for the final prediction. Many state-of-the-art algorithms such as HOG, CNN, and Bagging are available, with which satisfactory results can be guaranteed. Algorithms such as KNN, logistic regression, SVC, Naive Bayes, and Stochastic Gradient Descent are also available for classification and results. So, here, the major focus is to analyze, implement, and compare the results of this kind of machine-learning algorithm—a comparison of results was conducted using confusion matrix and classification reports. We present and explain the stepwise process to extract hand features from the dataset so that results can be gathered, and classification can be performed [6].

The rest of the paper will be structured as follows: Section 2 will be a detailed discussion on the work done till now. Section 3 will be about the methodology used, including the study of different machine-learning algorithms, analysis terminology such as confusion matrix, and a detailed discussion about case studies from 1–6. Section 4 will be about the experimental results. The conclusion will be in Section 5.

2. Related Work

Tam et al. [7] discussed a real-time hand gesture recognition system. This author used an embedded convolutional neural network for classification. Using hand prosthesis leveraging HD-EMG and deep learning, the author improved reliability and execution, and reaction times were minimized. Ease of use is the major focus for the proposed model and is achieved by a reduction in infrastructure for classifier training. The upper limb's electro biography [8] was used for hand gestures. EMG measured electrical activity related to muscles. In this, the hand gesture was taken into consideration without training machine-learning algorithms. The results are compared with existing algorithms and techniques such as KNN, NB, Discriminant Analysis (DA), SVM, and random forest (RF). The created approach of arranging distinctive hand gestures will be valuable in human–PC cooperation, just as in controlling gadgets including prostheses, virtual items, and wheelchairs. The introduced approach of evaluating the muscle initiation design for particular hand gestures across subjects will give more prominent materialness in controlling the prosthetic devices.

Li et al. [9] introduced a spatial fuzzy matching (SFM) algorithm, with which the author achieved a fused gesture dataset. SFM is also used for capturing the dynamic behavior of hand gestures and even comparing this with the test dataset. This can run on a simple machine, as they have used a fused data set that does not require any special hardware/software requirements to run on. This author achieved a 94–100% accuracy in the case of static and around 90% for dynamic hand gestures. So, leap motion and fused dataset were used for the experimental setup. It has also been observed that this SFM will work better as data sets increase. Lee and Tanaka [10] explain hand and finger gestures for the natural user interface. The author mentions that hand gestures are a simple but not a natural way of interaction. He focuses more on finger identification and tracking. With the use of Kinect and Depth-Sense, the user can track and identify finger movement and hand gestures even if the surrounding does not have sufficient light or a robust background. They also used Kinect depth data for finger identification and hand gesture recognition. So, this authors' model can provide natural communication and an interface. Nogales et al. [11] discuss how present gesture recognition resembles an issue of component extraction and example recognition, in which development is marked as having a place with a given class. A gesture recognition framework's reaction could take care of various issues in different fields, such as medication, mechanical technology, communication through signing, human–PC interfaces, computer-generated reality, increased reality, and security. In this unique situation, this work proposes a methodical writing survey of hand gesture recognition dependent on infrared data and AI calculations. To foster this methodical writing survey, we utilized the Kitchenham strategy. This orderly writing survey recovers data about the models' structures, the carried-out methods in every module, the method of learning utilized (managed, solo, semi-regulated, and support learning), recognition

exactness grouping, and the preparation time. Likewise, it recognizes writing holes for future examination.

Allard et al. [12] has used deep learning for hand gesture recognition. Deep learning calculations have become progressively more popular for their unrivaled capacity to simultaneously process multiple features from a lot of information. Be that as it may, inside the area of electromyography-based gesture recognition, deep learning calculations are only sometimes utilized as they require a preposterous amount of effort from a solitary individual to produce a huge number of models. This current paper's speculation is that, in general, educational provisions can be gained from a lot of information produced by accumulating the signs of numerous clients, hence diminishing recording errors while improving gesture recognition. Thus, this paper proposes applying gesture learning on amassed information from various clients while utilizing the limit of deep-learning calculations to process multiple features from enormous datasets. Two datasets involving 19 and 17 physically fit members, respectively (the first is utilized for pre-preparing), were employed for this work, utilizing the Myo armband. A third Myo armband dataset was taken from the Nina Pro information base and included ten physically fit members. Three unique deep-learning networks utilizing three distinct modalities as information (crude EMG, spectrograms, and ceaseless wavelet change (CWT)) are tested on the second and third datasets. Finally, a contextual investigation utilizing eight physically fit members proposes that a constant input permits clients to adjust their muscle actuation methodology, which decreases the corruption in exactness ordinarily experienced over long periods of time. Table 1 represents chronological summary of various techniques in the domain.

Table 1. Chronological summary of various techniques in the domain.

S. No	Year	Author	Detection Technique	Dataset	Other Characteristics
1	2010	Heo et al. [13]	Binary open (stretching) and close (crooking)	Hand Gesture Dataset	Used for the game system, so only grabbing and not grabbing have been used in this.
	2011	Dardas and Georganas [14]	Support vector machine (SVM), scale invariance feature transform (SIFT), and K-means clustering	Real-time Dataset	Accuracy of 96.23% under variable scale
3	2012	Zhang et al. [15]	Three-axis accelerometer (ACC) and multi-channel electromyography (EMG) sensors, multistream hidden Markov models, and HMM classifiers	Work on 72 CSL words and Hand Gesture Dataset	Accuracies of 95.3% and 96.3% for two subjects and by HMM accuracy is increased by 2.5%
4	2012	Keskin et al. [16]	Shape Classification Forest (SCF)	American Sign Language (ASL) dataset and ChaLearn Gesture Dataset (CGD2011)	Achieved a success rate of 97.8% when using the ASL dataset
5	2013	Ren et al. [17]	Kinect sensor, Finger-Earth Mover's Distance (FEMD)	Hand Gesture Dataset, 10-gesture dataset	93.2% mean accuracy, efficiency: 0.0750 s per frame
6	2014	Bar and Trivedi [18]	A Multi-modal Vision-Based Approach and Evaluations	Real dataset (set of 19 gestures), RGBD fusion	Studied the feasibility of an in-vehicle vision-based gesture recognition system

Table 1. Cont.

S. No	Year	Author	Detection Technique	Dataset	Other Characteristics
7	2016	Plouffe and Cretu [19]	Kinect sensor, k-curvature algorithm, DTW algorithm	Real-time hand gesture.	Accuracy of 92.4% is achieved over 55 static and dynamic hand gestures.
8	2018	Wahid et al. [8]	Upper limb's electromyography (EMG). For classification: k-Nearest Neighbor (kNN), Discriminant Analysis (DA), Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM) and non-parametric Wilcoxon signed-rank test.	Three different hand gestures as a fist, wave in, and wave out	Accuracy of 96.4% is achieved by using the area under the averaged root mean square curve (AUC-RMS)
9	2021	Zhang et al. [20]	Convolutional neural network (ConvNet), short-term memory (LSTM) network	Jester dataset and Nvidia dataset.	An accuracy of 95.73% received by Jester dataset and 95.69% by the "zoomed-out" Jester dataset. In the Nvidia dataset, an accuracy of 85.13% has been achieved.
10	2018	Zhao and Wang [21]	Convolutional Neural Network (CNN)	American Sign Language (ASL) dataset and MNIST dataset	CNN gave the highest efficiency on parameter distribution on the ASL dataset.
11	2019	Allard et al. [12]	Raw EMG, spectrograms, and continuous wavelet transform (CWT)	Two datasets: 19 and 17 able-bodied participants out of the first one is for pre-training Myo armband, NinaPro database	An offline accuracy: 98.31% (7 gestures and 17 participants by CWT-based ConvNet) and 68.98% (18 gestures and 10 participants by raw EMG-based ConvNet)
12	2020	Li et al. [9]	Leap Motion gen.2, spatial fuzzy matching (SFM) algorithm	Hand gesture dataset	Static Gesture: accuracy ranges from 94 to 100%. Dynamic Gesture: More than 90% accuracy has been achieved.
13	2020	Tam et al. [7]	Convolutional neural network (CNN) and myoelectric control scheme	Nina database, real-time hand gesture	Accuracy of 98.2% was achieved.

3. Methodology

In this paper, the MNIST dataset is used for the analysis of different algorithms. The sign MNIST dataset can be downloaded from Kaggle.com, accessed on 1 January 2022. This dataset is very popular and valuable for gesture recognition. It contains 24,000 images, which are of 20 different gestures. For training, 70% were used, and the remaining 30% were used for testing. Three classes were used, considering only 3 postures. Sections 3.2–3.9 describe analysis using the SVC algorithm, KNN algorithm, logistic regression, Naïve Bayes Classifier algorithm, SGDC, CNN model, random forest, and XGBoost, respectively.

3.1. Feature Selection

Feature selection involves the selection of feature subsets. For some classifiers, the reduced feature set gives a better recognition accuracy than the original feature set. However, in other classifiers, they give equal and comparable accuracy. Generalized steps related to feature selection are:

1. **Starting Point Selection:** Selecting a starting point for a feature subset is important. It can be done by starting with an empty matrix and adding relevant features to it. This is a forward selection method. Another technique is by starting from the feature set and eliminating irrelevant features from it. This is a backward selection method. The selection can also be initiated from the middle, proceeding outwards.
2. **Evaluation Method:** The evaluation strategies for different feature selection algorithms vary. In the filter method, irrelevant and redundant features are eliminated before the learning algorithm begins. In the wrapper method, the bias of a particular induction algorithm that uses cross-validation to give final accuracy is taken into account for feature selection.
3. **Stopping Criteria:** The feature selection algorithm will stop if on adding or removing features, the accuracy does not improve any further. The feature subset is revised until the merit does not degrade. This is based on the evaluation method.
4. The training data set comprised 27,455 cases (80%), and the test data set comprised 7172 cases (20%). One-third of the data (threefold) were used for cross-validation, one-third of the samples were reserved from each subject to be trained, and the rest were used for testing. Figure 1 represents the mechanism of the entire network architecture of CNN, which was used for the experimental setup. Using this, images are classified into different classes, which are used for further analysis.

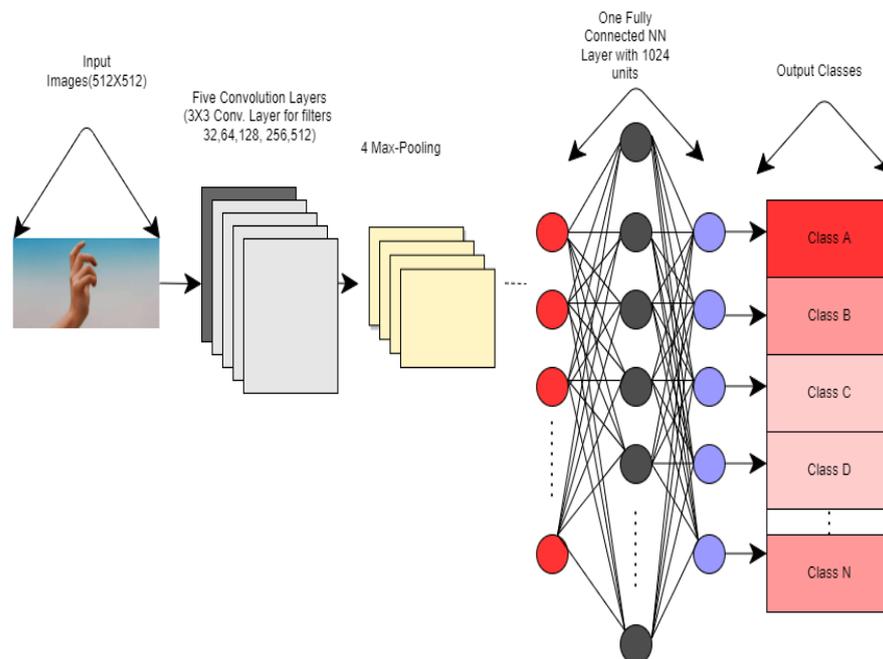


Figure 1. Internal Network Architecture of CNN for hand Gesture.

Features and Features Extraction

Feature extraction calculates various discriminating attributes of an image. Therefore, the selection of feature sets plays a vital role in the accurate classification of the images. These features also have invariance to translation, scale, and orientation. There are various shape-based feature extraction techniques that are categorized as contour-based, region-based, and gradient-based.

1. Contour-based: These features are extracted from shape boundaries. These are known as the local methods, in which rather than extracting features from the whole image, the features are only extracted from the ROI.
2. Region-based: These features are extracted from the whole image. These are known as global methods. Examples of tour-based features are geometric moments, Zernike moments, etc.
3. Gradient-based: In these features, images are split into blocks, and features are taken from these blocks only. This is local shape information, and objects are captured from these dense blocks such as HOG and Scale Invariant Feature Transform (SIFT).

An efficient feature descriptor is a set of numbers that represent a given shape or object. An apt feature descriptor must possess the properties listed below.

1. Distinguishable: An efficient descriptor must be able to distinguish the classes of hand gestures to be discriminated against. This must have inter-class variance.
2. Invariant: A feature extraction must be resistant to changes in rotation, translation, and scale. It should have small a variance within the class, even if there is a small change in the image.
3. Reliable: Features must be resistant to noise in the image. The feature set must be strong enough to handle noise and variations due to ambient conditions.
4. Statistically Independent: Two or more than two features must be independent of each other. A small change in the first feature must not affect the other feature set. The features must be statically independent. Otherwise, a small error in the first feature would affect the total system accuracy.

3.2. Analysis Using SVC Algorithm

A Support Vector Clustering (SVC) algorithm is a supervised machine-learning algorithm. SVC is based on kernel-based learning. To analyze a pattern, SVC uses both classification and regression techniques. The objective behind the use of SVC is to divide a data set into two classes of data so that it can find a hyperplane [8]. We use this hyperplane as a binary classifier. In this, first, searching for the nearest data points of the hyperplane has to be performed so that we can use these data points as support vectors. These support vectors play a crucial role in the development of hyperplanes [21]. Support vectors are also responsible for affecting the position and presentation of the hyperplane. Each point in support vectors represents n-dimensional space, where n represents several features. SVM's loss function is quite similar to logistic regression and is known as hinge loss. We can use this loss function to maximize the difference/margin in between data points and hyperplane. The loss function is mentioned below as Equation (1):

$$l(p, q, f(q)) = (1 - q * f(p)) \quad (1)$$

where l : loss function, q : true labels, p : sample and $f(p)$: predicted labels. if $q * f(p) \geq 1$ or hinge loss = 0; otherwise, it provides values as $1 - q * f(p)$.

3.3. Analysis Using KNN Algorithm

For gesture recognition, the K-Nearest Neighbor (KNN) algorithm is a supervised machine-learning algorithm. KNN is used for classification, by which a data point's classification is determined by how its neighbor is classified. Euclidean distance is used to find the nearest neighbor in KNN [8]. In this, the target is to achieve minimum Euclidean distance, and the calculation is performed based on several small distances [20]. As soon as the k value increases, accuracy also increases.

In general, the Euclidian distance formula is used. With the use of KNN, classification is performed using the threshold value, which is calculated by the average of the k data point that is nearest. The performance is totally based on the distance of the nearest neighbor, similarity measurement, and a threshold value.

To obtain a measurement of accuracy, hidden layer size is calculated by the number of neurons in the hidden layer. Weight optimization is conducted by the use of a solver, and the learning rate is calculated and represented by `learning_rate_initdouble`. This whole setup exists under scikit-learn.

3.4. Analysis Using Logistic Regression

Logistic regression involves gathering regression techniques and concerns portraying the connection between informative variables and a discrete reaction variable. The distinction between standard straight regression and logistic regression represents how the reaction variable Y is continuous for straight regression. In logistic regression, the reaction variable is discrete [22]. This distinction is shown in the selection of boundaries and suspensions. Logistic regression working is based on probability, and a linear equation is used to predict the values that lie between 0 and 1, and because of this, it is known as a predictive analysis algorithm. It uses the activation function as sigmoid so that outcome can be converted into categorical values. This sigmoid function can also be called a logistic function. It can be represented as below:

$$f(y) = \frac{1}{1 + e^{-(y)}} \quad (2)$$

where $-\infty \leq f(y) \leq +\infty$.

3.5. Analysis Using Naïve Bayes

Classification is an ordinarily utilized AI and information mining approach. Contingent upon the number of target classifications that are utilized to classify an informational index, various methodologies may be picked to conduct the classification work. For paired classifications, typically, choice trees and backing vector machines are normally embraced, yet these two methodologies are dependent on an imperative that the number of target classifications cannot exceed two. This unbending limitation makes them difficult to be considered to fit a wide range of genuine classification works in which the number of target classifications is generally more than two. The Naive Bayes Classifier is more reasonable for general classification assumptions [8]. There have been a decent assortment of effective genuine applications that depend on Naive Bayes classifier: for example, climate forecast administrations, client acknowledgment assessments, disease orders, etc. However, the configuration of an informational collection inside the issue area is preprocessed into an even arrangement [22]. This numerical classifier can proceed to determine the validities of squeezing a piece of new information into every conceivable classification. Thus, the classification with the most trustworthy fitness can be picked as the best-fitting classification of this piece of information.

3.5.1. Analysis Using Naïve Bayes for Multinomial Models

This Naive Bayes model is utilized for discrete checks, and it considers and presumes the freedom of the multitude of provisions and factors in the dataset [23]. It can deal with text classification issues, and its precision can be further developed while applying some change strategies such as element development, including choice, and cleaning the dataset [24]. It centers on the word as well as the recurrence of the word in the dataset.

3.5.2. Analysis Using Naïve Bayes for Gaussian Models

It supports Gaussian normal distribution and uses only continuous data. It is also a supervised machine-learning algorithm based on the Bayes theorem [25]. This one is easy to use, as the only estimation that needs to be made is of mean and standard deviation from the training dataset. Calculation of probability for input values will be performed using each class frequency. It is also required to store the mean and standard deviation of each class.

3.6. Analysis Using SGDC

Stochastic Gradient Descent (SGD) is a straightforward yet proficient optimization calculation used to discover the upsides of boundaries/coefficients of capacities that limit expense work. It is utilized for the discriminative learning of direct classifiers under arched misfortune capacities such as SVM and logistic regression. It has been effectively applied to enormous scope datasets because the coefficients are updated for each preparation occasion instead of at the end of instances. If the selection of hyper parameters does not carry properly, then SGDC will provide the worst results.

Summation of finite numbers is performed by objective functions and is calculated as

$$\Phi_{Ffj}(y) = \frac{1}{m} \sum_{j=1}^m \Phi_{Ffj}(y) = \Phi_{Ff}(y) \quad (3)$$

where m is the maximum number of iterations, and $\Phi_{Ffj}(y)$ is an unbiased function.

3.7. Analysis Using CNN

CNN is famous for recognition and has preferable outcomes over different strategies, primarily because it can obtain the necessary element esteems from the information picture and become familiar with the contrast between various examples by utilizing countless examples in its preparation [7,20]. In any case, previously, its advancement has been restricted because of the speed of equipment computing. Lately, because of the progression of semiconductor fabricating, the computing pace of illustrations preparing units is becoming quicker, and the bottleneck of equipment handling speed has permitted the CNN network to grow quickly [21]. The steps followed in applying this CNN are as follows: first, a picture is inputted (interpreted as an array of pixels); second, processing and filtering have to be conducted; and third, the results are obtained after the classification. Every model has to be trained and then tested so that it can be used in a layered architecture in which many convolutional layers involve kernels (or filters) and a Pooling operation.

3.8. Analysis Using Random Forest

A random forest [22,23,26] classifier uses a series of decision trees to label the sample data. A number of decision trees exist to determine the performance of the algorithm. Additionally, all these trees have been formed by the algorithm to make a prediction.

Random forest methods are selected due to their advantages; for example, their performance is good in large datasets, they have no overfit issue, variables can be used as numeric and categorical, they can be easily used in a multi-class environment, and fewer parameters are required as compared to other state-of-the-art methods.

3.9. Analysis Using XGBoost

Parameter tuning was used to train an XGBoost model [23,25,27]. Following are the two steps that were used:

- i. Training to a baseline model took place to check the performance of the model in general.
- ii. A second model was used to train by parameter tuning, and the results are compared with the baseline model.

It is not only used for classification; it can also be used for regression. It is faster than other state-of-the-art methods due to qualities such as parallel, distributed, and cache-aware computing. Some of the other qualities are optimization and scalability.

Confusion Matrix of Analysis:

Analysis using algorithms such as SVC, KNN, logistic regression, Naïve Bayes Multinomial, Naïve Bayes Gaussian, SGDC, CNN, random forest, and XGBoost [28–32] is represented by Figure 2.

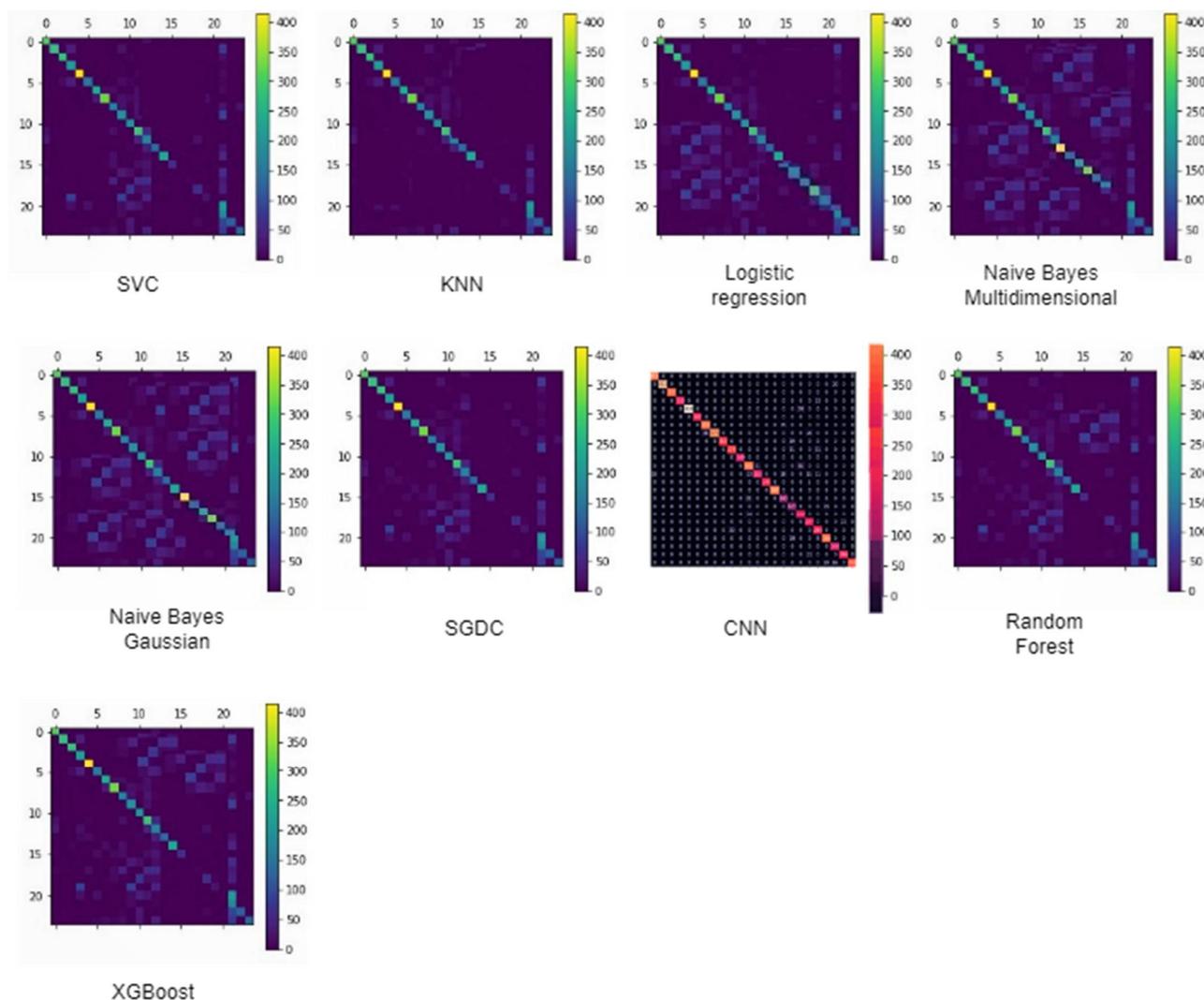


Figure 2. Confusion matrix of analysis using SVC, KNN, logistic regression, Naïve Bayes Multinomial, Naïve Bayes Gaussian, SGDC, CNN, random forest, and XGBoost.

4. Results

SVC, KNN, Naïve Bayes Multinomial, Naïve Bayes Gaussian, Stochastic Gradient Descent Classifier (SGDC), Convolutional Neural Network (CNN), random forest, and XGBoost were applied to the sign MNIST dataset, and accuracies of 66.85, 80.46, 68.21, 46.85%, 38.9%, 59.80%, 91.41%, 84.43%, and 81.35% were achieved, respectively. Observation was conducted based on factors such as accuracy, classification report, and confusion matrix. KNN [33–41] has benefits such as straightforwardness of execution, simplicity of debugging, and robustness; this method also showed a few limits, such as a poor computational time for enormous datasets and affectability towards duplicity, as we found that the least precision was found with the NB model [42]. These results could be better with some adjustment, smoothing, and preprocessing in the dataset. Presumptions of the freedom of the factors are a solid restriction, because of which the probability yields are not helpful in determining dependable results. Currently, CNN is an exceptional method for best classification results, and different upgrades and improvements are likewise being proposed in different papers.

Figure 3 represents the accuracy of Naïve Bayes, K-Nearest Neighbor (KNN), random forest, XGBoost, Support vector classifier (SVC), logistic regression, Stochastic Gradient Descent Classifier (SGDC), and Convolution Neural Networks (CNN). Precision is a measure of a model's accuracy in classifying a sample as positive. Recall measures the ability

to detect positive samples, and F1-Score is used to balance precision and recall [43–45]. SVM [46] is used to find a hyperplane, and a classification report for SVC is generated. Figure 4 is a graphical representation of the classification report, which shows precision values for SVC, and average values have been calculated.

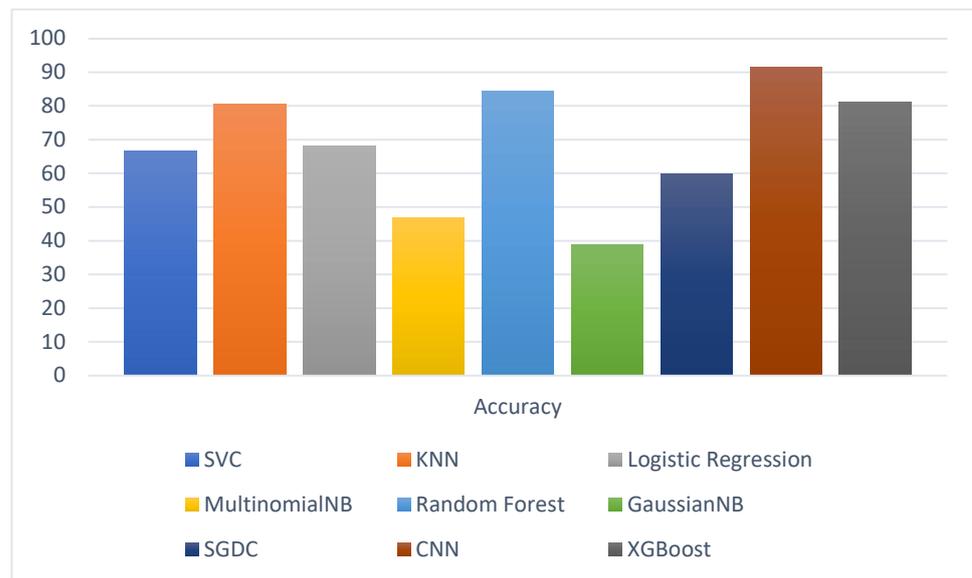


Figure 3. Selected machine learning algorithms vs. accuracy.

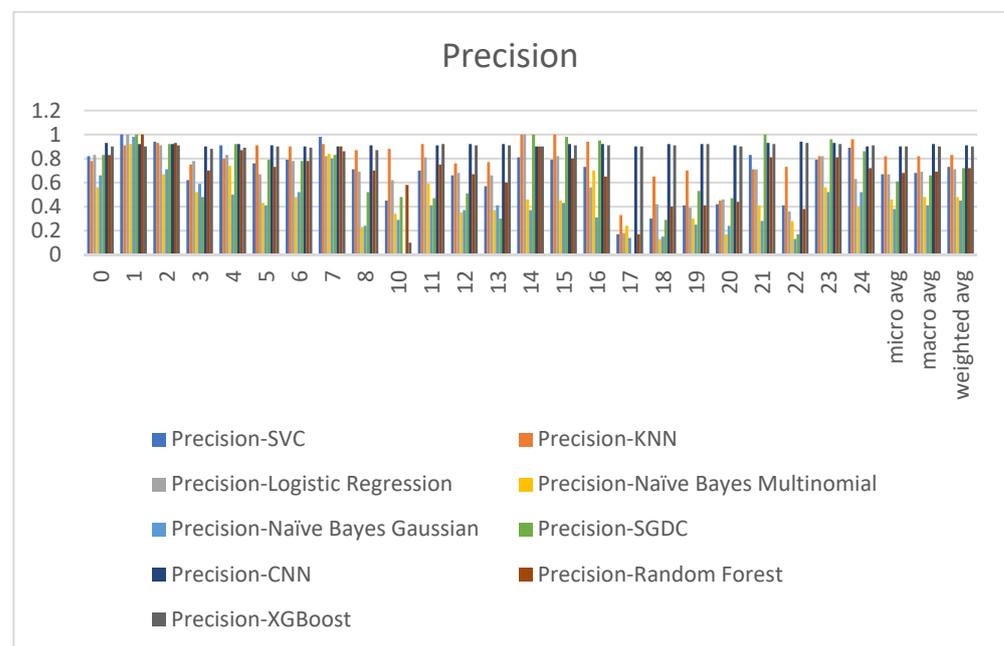


Figure 4. Precision.

K-nearest neighbors, or simply KNN, is a classification technique that falls under the category of supervised machine-learning algorithms. Figure 5 is a graphical representation of the classification report, which shows recall values and their average for KNN, where recall is recorded as high compared to precision and F1-Score at the beginning of classification and achieves its highest value at 16; after that, the value of precision is higher than this value till the end.

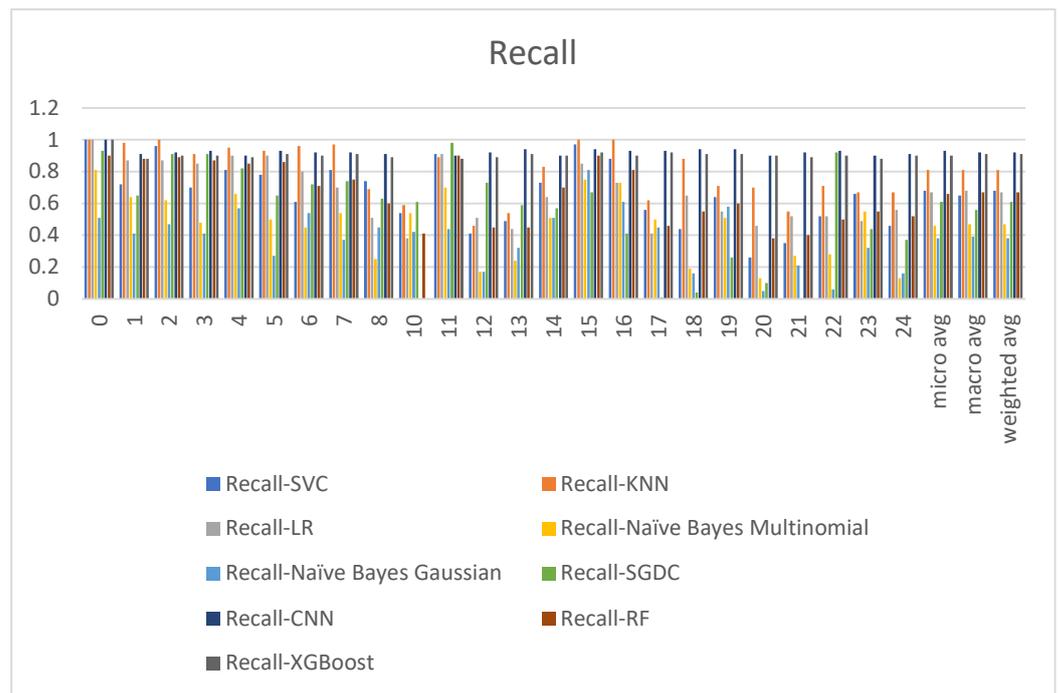


Figure 5. Recall.

Logistic regression is widely used to solve the classification problem. It is based on the concept of probability. Logistic regression uses a linear equation to predict a value where predictors are independent. Figures 4–6 are the graphical representations of the classification report, which show precision, recall, and F1-Score values for logistic regression.

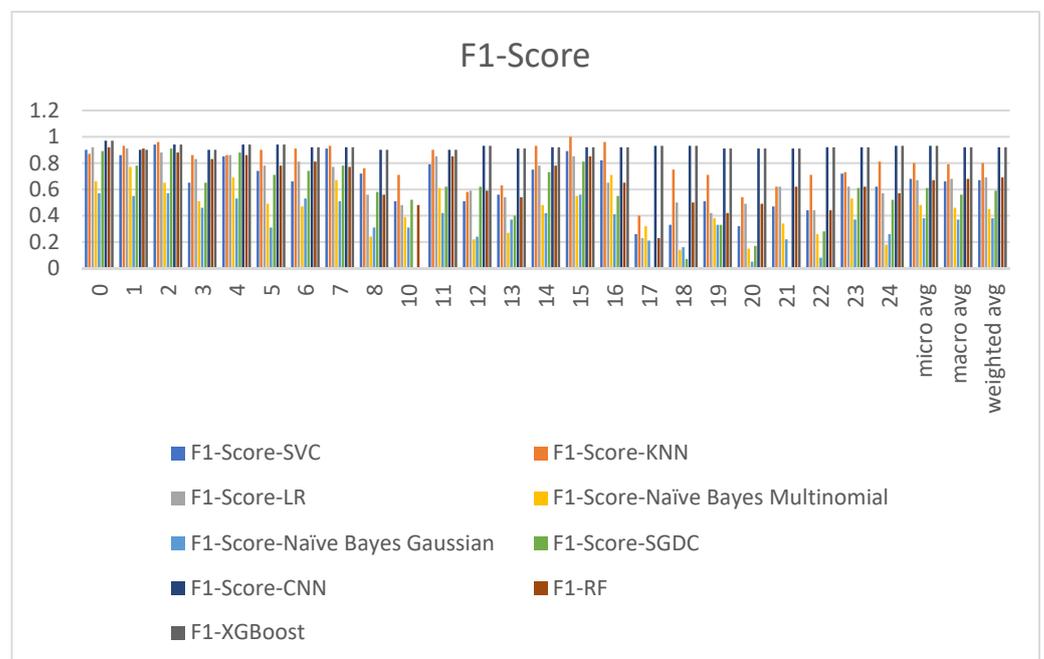


Figure 6. F1 Score.

This Naïve Bayes model is used for discrete counts, and it considers an assumption of the independence of all the features and variables in the dataset. It can handle text classification problems, and the accuracy can be improved when applying transformation techniques such as feature construction, feature selection, and dataset cleaning. Figures 4–6

are the classification report's graphical representation, which show precision, recall, and F1-Score values for Naïve Bayes Multinomial.

This model is useful when the feature vectors are assumed to be binary, that is, containing zeros and ones or true and false. Figures 4–6 show the classification report using Naïve Bayes Gaussian. There are various reasons to use this algorithm, such as it being easy and fast in implementing and predicting the class in a multiclass dataset.

The reason to use SGDC is its efficiency for a linear classifier and that it is easily implemented and overcomes the issue of gradient descent, which is also an expensive way to perform classification. Figures 4–6 are the representation of the classification report using SGDC. CNN is famous for its grid pattern and is used to extract unique features from gestures and feed the same to the classifier. Figures 4–6 are pictorial representations of the classification report based on factors such as precision, recall, and F1-Score, and the average values of these are also calculated.

Figures 4–6 show a comparison of the implementation of different algorithms such as Naïve Bayes, K-Nearest Neighbor (KNN), random forest, XGBoost, support vector classifier (SVC), logistic regression, Stochastic Gradient Descent Classifier (SGDC) and Convolution Neural Networks (CNN) to the MNIST dataset based on precision, recall, and FI-score. So, using CNN, the average value of precision is 0.91, recall is 0.92, and FI-score is 0.92, which is far better than other mentioned algorithms.

In this article, CNN achieved the highest accuracy of 91.41%, which is also higher when compared with a deep convolutional neural network [34], in which hand gestures are directly classified in images without any segmentation [35–37] or detection stage that could discard irrelevant not-hand areas, and it achieved an accuracy of 85.3% in the dataset with complex backgrounds [38,39].

In this work, a comparison has been made between SVC, KNN, Naïve Bayes Multinomial, Naïve Bayes Gaussian, Stochastic Gradient Descent Classifier (SGDC), and Convolutional Neural Network (CNN), which were applied to the sign MNIST dataset, and accuracies of 66.85%, 80.46%, 68.21%, 46.85%, 38.9%, 59.80%, and 91.41% are achieved, respectively.

In [40,41], the left and the right datasets are used to perform a comparison between SVM, CNN, KNN, and RFC and achieve the highest accuracy of 77.82% using CNN (which took maximum execution time) on left and right gesture datasets.

A comparison of the proposed model with other state-of-the-art techniques is represented in Table 2.

Table 2. Comparison between proposed and state-of-the-art work.

Reference	Model	Accuracy
Saad et al. [47]	Random Forest (RF) and Boosting Algorithms, Decision Tree Algorithm	63%
Fong et al. [47]	Model Induction Algorithm, K-star Algorithm, Updated Naïve Bayes Algorithm, Decision Tree Algorithm	76%
Yan et al. [48]	AdaBoost Algorithm, SAMME Algorithm, SGD Algorithm, Edgebox Algorithm	81.25%
Proposed Model	Convolutional Neural Networks (CNN) model applied to sign language MNIST dataset	91.41%

5. Conclusions

To make human–computer interactions better, it is required to predict the correct hand gestures of humans. This paper includes some experimental data to improve gesture recognition by implementing some existing algorithms such as Support Vector Classifier (SVC) algorithm, K-Nearest Neighbor (KNN) algorithm, logistic regression, Naïve Bayes (Multinomial NB and Gaussian NB), Stochastic Gradient Descent Classifier (SGDC), and Convolutional Neural Networks (CNN) model on the sign language MNIST dataset. The

best results are achieved using Convolutional Neural Networks (CNN), acquiring an accuracy of 91.41%. Apart from CNN, random forest results are better than other traditional algorithms, as the model acquires an accuracy of 84.43%. So, this proposed objective can be used to predict more accurate results in terms of better communication between humans and machines; deaf and mute people will also get used to that in their normal communication, and these data can even be used for researchers who are working in the same field.

Author Contributions: Conceptualization, S.B., M.A. and J.R.; methodology, I.K. and A.A.; software, A.A. and S.B.; validation, A.A., I.K. and J.R.; formal analysis, J.R.; investigation, M.A., A.K.C. and I.K.; resources, I.K.; data curation, S.B. and A.K.C.; writing—original draft preparation, S.B. and J.R.; writing—review and editing, M.A. and A.A.; visualization, M.A.; supervision, M.A.; project administration, A.A.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Researchers Supporting Program (TUMA-Project-2021-14), AlMaarefa University, Riyadh, Saudi Arabia.

Data Availability Statement: Not Applicable.

Acknowledgments: Mohammed Alshehri would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project No. R-2022-83. The authors deeply acknowledge the Researchers Supporting Program (TUMA-Project-2021-14), AlMaarefa University, Riyadh, Saudi Arabia for supporting steps of this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dardas, N.H.; Georganas, N.D. Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 3592–3607. [[CrossRef](#)]
2. Skaria, S.; Al-Hourani, A.; Evans, R.J. Deep-Learning Methods for Hand-Gesture Recognition Using Ultra-Wideband Radar. *IEEE Access* **2020**, *8*, 203580–203590. [[CrossRef](#)]
3. Keskin, C.; Kirac, F.; Kara, Y.E.; Akarun, L. Randomized decision forests for static and dynamic handshape classification. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 31–36.
4. 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.
5. Wang, S.B.; Quattoni, A.; Morency, L.P.; Demirdjian, D.; Darrell, T. Hidden conditional random fields for gesture recognition. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1521–1527.
6. Just, A.; Marcel, S. A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition. *Comput. Vis. Image Underst.* **2009**, *113*, 532–543. [[CrossRef](#)]
7. Tam, S.; Boukadoum, M.; Campeau-Lecours, A.; Gosselin, B. A Fully Embedded Adaptive Real-Time Hand Gesture Classifier Leveraging HD-sEMG and Deep Learning. *IEEE Trans. Biomed. Circuits Syst.* **2020**, *14*, 232–243. [[CrossRef](#)]
8. Wahid, M.F.; Tafreshi, R.; Al-Sowaidi, M.; Langari, R. Subject-independent hand gesture recognition using normalization and machine learning algorithms. *J. Comput. Sci.* **2018**, *27*, 69–76. [[CrossRef](#)]
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. Lee, U.; Tanaka, J. Finger identification and hand gesture recognition techniques for natural user interface. In Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction, Bangalore, India, 24–27 September 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 274–279. [[CrossRef](#)]
11. Nogales, R.E.; Benalcázar, M.E. Hand gesture recognition using machine learning and infrared information: A systematic literature review. *Int. J. Mach. Learn. Cyber.* **2021**, *12*, 2859–2886. [[CrossRef](#)]
12. Cote-Allard, U.; Fall, C.L.; Drouin, A.; Campeau-Lecours, A.; Gosselin, C.; Glette, K.; Laviolette, F.; Gosselin, B. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 760–771. [[CrossRef](#)]
13. Heo, H.; Lee, E.C.; Park, K.R.; Kim, C.J.; Whang, M. A realistic game system using multi-modal user interfaces. *IEEE Trans. Consum. Electron.* **2010**, *56*, 1364–1372. [[CrossRef](#)]

14. Dardas, N.; Chen, Q.; Georganas, N.D.; Petriu, E.M. Hand gesture recognition using Bag-of-features and multi-class Support Vector Machine. In Proceedings of the 2010 IEEE International Symposium on Haptic Audio Visual Environments and Games, Phoenix, AZ, USA, 16–17 October 2010; pp. 1–5.
15. Zhang, X.; Chen, X.; Li, Y.; Lantz, V.; Wang, K.; Yang, J. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Trans. Syst. Man Cyber.-Part A Syst. Hum.* **2011**, *41*, 1064–1076. [[CrossRef](#)]
16. Keskin, C.; Kiraç, F.; Kara, Y.E.; Akarun, L. Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests. In *Computer Vision—ECCV 2012. ECCV 2012. Lecture Notes in Computer Science*; Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7577. [[CrossRef](#)]
17. Ren, Z.; Yuan, J.; Meng, J.; Zhang, Z. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. *IEEE Trans. Multimed.* **2013**, *15*, 1110–1120. [[CrossRef](#)]
18. Ohn-Bar, E.; Trivedi, M.M. Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2368–2377. [[CrossRef](#)]
19. Plouffe, G.; Cretu, A. Static and Dynamic Hand Gesture Recognition in Depth Data Using Dynamic Time Warping. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 305–316. [[CrossRef](#)]
20. Zhang, W.; Wang, J.; Lan, F. Dynamic hand gesture recognition based on short-term sampling neural networks. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 110–120. [[CrossRef](#)]
21. Zhao, Y.; Wang, L. The Application of Convolution Neural Networks in Sign Language Recognition. In Proceedings of the 2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP), Wanzhou, China, 9–11 November 2018; pp. 269–272.
22. Gajowniczek, K.; Grzegorzczak, I.; Ząbkowski, T.; Bajaj, C. Weighted Random Forests to Improve Arrhythmia Classification. *Electronics* **2020**, *9*, 99. [[CrossRef](#)] [[PubMed](#)]
23. Yu, J.-W.; Yoon, Y.-W.; Baek, W.-K.; Jung, H.-S. Forest Vertical Structure Mapping Using Two-Seasonal Optic Images and LiDAR DSM Acquired from UAV Platform through Random Forest, XGBoost, and Support Vector Machine Approaches. *Remote Sens.* **2021**, *13*, 4282. [[CrossRef](#)]
24. Aggarwal, A.; Kumar, M. Image surface texture analysis and classification using deep learning. *Multimed. Tools Appl.* **2021**, *80*, 1289–1309. [[CrossRef](#)]
25. Ding, X.; Jiang, T.; Xue, W.; Li, Z.; Zhong, Y. A New Method of Human Gesture Recognition Using Wi-Fi Signals Based on XGBoost. In Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops), Chongqing, China, 9–11 August 2020; pp. 237–241. [[CrossRef](#)]
26. Li, T.; Zhou, M. ECG Classification Using Wavelet Packet Entropy and Random Forests. *Entropy* **2016**, *18*, 285. [[CrossRef](#)]
27. Paleczek, A.; Grochala, D.; Rydosz, A. Artificial Breath Classification Using XGBoost Algorithm for Diabetes Detection. *Sensors* **2021**, *21*, 4187. [[CrossRef](#)]
28. Jin, J.; Fu, K.; Zhang, C. Traffic Sign Recognition with Hinge Loss Trained Convolutional Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1991–2000. [[CrossRef](#)]
29. Alshehri, M.; Kumar, M.; Bhardwaj, A.; Mishra, S.; Gyani, J. Deep Learning Based Approach to Classify Saline Particles in Sea Water. *Water* **2021**, *13*, 1251. [[CrossRef](#)]
30. Badi, H. Recent methods in vision-based hand gesture recognition. *Int. J. Data Sci. Anal.* **2016**, *1*, 77–87. [[CrossRef](#)]
31. Patwary, M.J.A.; Parvin, S.; Akter, S. Significant HOG-Histogram of Oriented Gradient Feature Selection for Human Detection. *Int. J. Comput. Appl.* **2015**, *132*, 20–24.
32. Devineau, G.; Moutarde, F.; Xi, W.; Yang, J. Deep Learning for Hand Gesture Recognition on Skeletal Data. In Proceedings of the 13th IEEE Conference on Automatic Face and Gesture Recognition (FG'2018), Xi'an, China, 15–19 May 2018.
33. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Bencherif, M.A.; Mekhtiche, M.A. Hand Gesture Recognition for Sign Language Using 3DCNN. *IEEE Access* **2020**, *8*, 79491–79509. [[CrossRef](#)]
34. Bao, P.; Maqueda, A.I.; Del-Blanco, C.R.; García, N. Tiny hand gesture recognition without localization via a deep convolutional network. *IEEE Trans. Consum. Electron.* **2017**, *63*, 251–257. [[CrossRef](#)]
35. Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M.P.; Shyu, M.L.; Chen, S.C.; Iyengar, S.S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.* **2018**, *51*, 1–36. [[CrossRef](#)]
36. Rida, I.; Al-Maadeed, N.; Al-Maadeed, S.; Bakshi, S. A comprehensive overview of feature representation for biometric recognition. *Multimed. Tools Appl.* **2018**, *79*, 4867–4890. [[CrossRef](#)]
37. Yang, M.; Kidiyo, K.; Joseph, R. A survey of shape feature extraction techniques. *Pattern Recognit.* **2008**, *15*, 43–90.
38. Ping Tian, D. A review on image feature extraction and representation techniques. *Int. J. Multimed. Ubiquitous Eng.* **2013**, *8*, 385–396.
39. Rida, I.; Herault, R.; Marcialis, G.L.; Gasso, G. Palmprint recognition with an efficient data driven ensemble classifier. *Pattern Recognit. Lett.* **2019**, *126*, 21–30. [[CrossRef](#)]
40. Hamid, N.A.; Sjarif, N.N.A. Handwritten recognition using SVM, KNN, and Neural networks. *arXiv* **2017**, arXiv:1702.00723.
41. Kumar, M.; Sriastava, S.; Hensman, A. A Hybrid Novel Approach of Video Watermarking. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2016**, *9*, 365–376. [[CrossRef](#)]
42. Chakradar, M.; Aggarwal, A.; Cheng, X.; Rani, A.; Kumar, M.; Shankar, A. A Non-invasive Approach to Identify Insulin Resistance with Triglycerides and HDL-c Ratio Using Machine learning. *Neural Process. Lett.* **2021**, 1–21. [[CrossRef](#)]

43. Kumar, M.; Aggarwal, J.; Rani, A.; Stephan, T.; Shankar, A.; Mirjalili, S. Secure video communication using firefly optimization and visual cryptography. *Artif. Intell. Rev.* **2021**, 1–21. [[CrossRef](#)]
44. Bhushan, S.; Alshehri, M.; Agarwal, N.; Keshta, I.; Rajpurohit, J.; Abugabah, A. A Novel Approach to Face Pattern Analysis. *Electronics* **2022**, *11*, 444. [[CrossRef](#)]
45. Singh, A.K.; Kumar, S.; Bhushan, S.; Kumar, P.; Vashishtha, A. A Proportional Sentiment Analysis of MOOCs Course Reviews Using Supervised Learning Algorithms. *Ingénierie Syst. D'inf.* **2021**, *26*, 501–506. [[CrossRef](#)]
46. Albawi, S.; Bayat, O.; Al-Azawi, S.; Ucan, O.N. Social Touch Gesture Recognition Using Convolutional Neural Network. *Comput. Intell. Neurosci.* **2018**, *2018*, 6973103. [[CrossRef](#)]
47. Fong, S.; Liang, J.; Fister, I.J.; Mohammed, S. Gesture Recognition from Data Streams of Human Motion Sensor Using Accelerated PSO Swarm Search Feature Selection Algorithm. *J. Sens.* **2015**, *2015*, 205707. [[CrossRef](#)]
48. Yan, S.; Xia, Y.; Smith, J.S.; Lu, W.; Zhang, B. Multiscale Convolutional Neural Networks for Hand Detection. *Appl. Comput. Intell. Soft Comput.* **2017**, *2017*, 9830641. [[CrossRef](#)]