



# Optimized Hybrid Convolution Neural Network with Machine Learning for Arabic Sign Language Recognition



Ahmed Osman Mahmoud<sup>\*</sup>, Ibrahim Ziedan<sup>†</sup>, Amr A. Zamel<sup>‡</sup>

Computer and Systems Engineering Department, Faculty of Engineering, Zagazig University, Zagazig 44519, Egypt

Corresponding Author Email: [aoeid@eng.zu.edu.eg](mailto:aoeid@eng.zu.edu.eg)

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.410415>

## ABSTRACT

**Received:** 27 November 2023

**Revised:** 25 March 2024

**Accepted:** 15 June 2024

**Available online:** 31 August 2024

### Keywords:

*classification, convolution neural network, hybrid CNN, hybrid ML, machine learning algorithms, optimization, sign language*

The World Health Organization stated that the global population of hard of hearing individuals is estimated to exceed 360 million people, and this number is continuously increasing. Communication barriers between these individuals and hearing individuals pose significant challenges in many areas of life, including education and employment. Therefore, developing methods to facilitate communication and bridge communication gaps is essential. In this paper, a novel approach is presented to Arabic sign alphabets recognition using optimized hybrid techniques. The proposed approach combines a convolutional neural network with five traditional machine learning algorithms: Feedforward Neural network, Decision Tree, Random Forest, Support Vector Machine and K-Nearest Neighbors. The proposed approach was evaluated on a large dataset called ArSL2018, which consists of 32 different classes of Arabic alphabet letters. Six different optimization techniques were investigated to find the optimal multipliers for the outputs of the Hybrid CNN models to achieve high classification accuracy. These techniques included the Genetic Algorithm, Particle Swarm, Firefly Algorithm, Differential Evolution, Sine Cosine, and Harris Hawks Optimization. The experiments demonstrated that the optimized hybrid techniques achieved the highest accuracy, approaching 99%, surpassing their counterparts. This demonstrates the efficiency of the proposed model in accurately recognizing Arabic alphabets.

## 1. INTRODUCTION

The World Health Organization declared that the percentage of people who suffer from speech and hearing problems exceeds five percent of the population, which is a significant proportion. It is expected that by 2050, the percentage will increase to one in ten people experiencing hearing loss [1, 2]. People with hearing and speech impairments, who are considered as individuals with special needs, face challenges in communicating with each other and with others. Consequently, they encounter difficulties in the learning process. Therefore, it is essential to find solutions that facilitate their learning, particularly when it comes to learning a difficult language such as Arabic (the language of the Holy Quran). These individuals communicate using sign language [3].

Deaf and hearing-impaired individuals utilize sign language as a means of visual communication to interact with one another and with the general population. The language relies on a set of movements that convey meanings and ideas. Similarly, some signs in Arabic sign language are used to represent letters in the Arabic language, expressing written sounds and characters. However, not all signs in Arabic sign language represent letters; some signs may convey words, phrases, or different concepts [4]. It is crucial to note that sign language is an independent and evolving communication system with its own rules and vocabulary, which develops and

changes over time. Arabic sign language may vary in certain signs or gestures from one country to another, influenced by local culture and developments [5].

Sign language is divided into two types: fingerspelling and symbolic sign language. Fingerspelling is primarily used among deaf individuals for communication and learning. This type of sign language relies on using a single sign to represent each letter or word, making the learning and communication process easier [6]. On the other hand, Symbolic sign language encompasses the expression of words and sentences by combining various movements, such as hand gestures, lip movements, facial expressions, and additional gestures. This type of sign language relies on encoding words and phrases using a variety of movements, allowing for more complex and detailed concepts to be expressed [7]. Both fingerspelling and symbolic sign language are used as means of communication and expression among deaf individuals, but they differ in the way expression and communication are carried out in each of them.

Two distinct categories of sign language recognition methods exist: device-based systems and vision-based systems. In device-based systems, the user wears a device such as gloves or uses a device like Microsoft Kinect. These devices capture the user's hand movements or body gestures and translate them into sign language recognition [8, 9]. In vision-based systems, the recognition relies on gathering images or videos using cameras that are commonly used by individuals.

These images or videos are then analyzed, classified, and processed using artificial intelligence and image processing techniques to recognize and interpret sign language gestures [10, 11]. Both device-based and vision-based systems are utilized for sign language recognition. However, these methods vary in their equipment or technological implementations employed to capture and interpret the gestures of sign language.

## 1.1 Key contributions

This research presents a novel approach to Sign Language Recognition (SLR) that addresses several limitations identified within the current literature. We propose a hybrid deep learning and machine learning model that leverages the strengths of both paradigms to achieve improved performance. This work offers the following key contributions:

**Hybrid Deep Learning and Machine Learning Approach:** a novel approach is proposed to combine deep learning and machine learning algorithms. This hybrid model leverages the strengths of both paradigms to achieve improved performance in SLR compared to using them in isolation.

**Optimized Multi-Algorithm Integration:** this work goes beyond simply combining algorithms. We meticulously optimize the integration of multiple algorithms within the model, leading to superior results compared to individual approaches.

**Large and Diverse Dataset Utilization:** this work employed a well-established dataset encompassing 32 distinct classes. This surpasses prior research that often utilizes limited datasets or requires data manipulation. This extensive and diverse dataset allows our model to capture a broader range of variations and enhances its overall robustness.

**Real-World Applicability through raw data:** in this work dataset is directly utilized without resorting to image processing or data augmentation techniques (even when dealing with an imbalanced dataset). This approach ensures the model's ability to handle the complexities and challenges of real-world data, leading to increased practicality and reliability in real-life scenarios.

In this research paper, a novel sign language recognition approach that leverages the strengths of both deep learning and machine learning algorithms through a meticulously optimized hybrid model is presented. This approach surpasses prior research by utilizing a large and diverse dataset (32 classes) without data manipulation, leading to a more robust model that generalizes well. Additionally, by working directly with the raw dataset, our model gains real-world applicability by learning to handle the inherent complexities and challenges of unprocessed data, ultimately resulting in increased practicality and reliability in real-life scenarios.

The subsequent section of this paper is structured as follows: Section 2 provides an overview of pertinent literature, encompassing prior studies and other research conducted within the corresponding domain. The aim of this section is to place the current research in the context of previous work and identify knowledge gaps that the research aims to fill. Section 3: This section presents the method used to find a solution to the problem addressed in the research. It includes a description of the tools and techniques used in the research, as well as how the data was prepared and analyzed. Section 4: This section presents the results obtained from the research. It includes a presentation of the collected data and information, their analysis, and interpretation based on the method used in

Section 3. Section 5: This section provides a general summary of the research, including an explanation of the main contributions and key findings achieved in the study and Future Work: This section discusses future research opportunities and potential directions for further investigation related to the topic. It may include suggestions for future studies or directions to expand the scope of the current research.

## 2. RELATED WORK

The deaf community in Arabic-speaking countries employs Arabic Sign Language, which is a visual-gestural communication system. ArSL allows deaf individuals to themselves and interact with others effectively through hand gestures, facial expressions, and body movements. It plays a vital role in facilitating meaningful connections and promoting equal access to information and opportunities for deaf individuals in Arabic-speaking regions. Some researchers use deep learning such as Convolution Neural Network in their research, while others use transfer learning to develop classification models for sign language, aiming to enhance the learning experience of individuals with hearing impairments.

Abdelghfar et al. [12] presented a model for classifying 14 categories of Arabic language letters that represent the beginnings of Surahs in the Quran. They utilized a subset of a comprehensive dataset known as ArSL2018, which comprises 24,137 images. The model encompasses four distinct stages, namely data preparation, data preprocessing, feature extraction, and classification stages. The author to improve their accuracy used some resampling techniques such as random minority oversampling (RMO) and synthetic minority oversampling (SMOTE) and 97.79% test accuracy achieved. Although Alani and Cosma [13] proposed ArSL-CNN model for translating ArSL. It makes use of a dataset known as ArSL2018, which comprises 54,049 images capturing 32 different sign language gestures performed by forty participants. The initial experiments conducted with ArSL-CNN yielded 98.80% and 96.59% accuracies for training and testing respectively. Additionally, the research explores how imbalanced data can affect the accuracy of the model. To address the issue of imbalanced data, the researchers conducted additional experiments using various re-sampling methods. Among them, the synthetic minority oversampling technique showed significant improvement. Improving the test accuracy by ratio equal to 0.7%. This enhancement was statistically significant ( $p = 0.016$ ,  $\alpha < 0.05$ ), indicating a reliable improvement in accuracy.

Furthermore, Moustafa et al. [14] suggested approach involved utilizing a CNN with the Mediapipe model to classify 28 different classes representing Arabic sign language alphabets. A dataset was gathered for evaluating the model, comprising a total of 7,057 images. These images were divided into training, validation, and testing sets, accounting for 60%, 20%, and 20% respectively. The model was able to classification with true positive rate 97.1%. Also, Alawwad et al. [15] introduced a new ArSL recognition system using Faster R-CNN. The system localizes and recognizes Arabic sign language alphabets. Faster R-CNN extracts image features and learns hand positions. The approach addresses feature selection and hand segmentation challenges. VGG-16 and ResNet-18 models are exploited for implementation. A real ArSL image dataset is collected for training and testing

consisting of 18360 samples split into training, validation, and testing sets in the proportions of 60%, 20%, and 20% respectively. The proposed approach achieves 93% accuracy in sign recognition.

In other hand, some researchers employ transfer learning, which refers to the strategy of utilizing the knowledge and weights of pre-trained models on extensive datasets to improve the performance and efficiency of a new task that has limited labeled data. It involves using the learned features from one task to improve the learning of a related or different task, such as Zakariah et al. [16] developed a system to interpret visual hand gestures in Arabic Sign Language and convert them into textual information. The dataset used in this project is ArSL2018, which contains 1,500 images per class approximately representing different meanings conveyed by hand gestures. Several preprocessing and data augmentation methods were employed to improve the quality and diversity of the dataset. Multiple pretrained models were experimented with, and the EfficientNetB4 model was found to be the most suitable due to its complex architecture. Other lightweight models struggled with the complexity of the dataset. The top-performing model attained a training accuracy of 98% and 95% accuracy for testing.

Alharthi and Alzahrani [11] aimed to develop reliable transfer learning models for accurately classifying images of Arabic alphabets from an ArSL dataset. The study consisted of two parts: transfer learning and deep learning approaches. The transfer learning approach involved employing a range of pretrained models and vision transformers with diverse sizes and weight initialization methods. For comparison, convolutional neural networks were trained from scratch using the deep learning approach. The performance of these methods was evaluated using metrics such as accuracy, precision, recall, F1 score, and loss. Consistently, the transfer learning approach outperformed other CNN models on the ArSL2018 dataset, with ResNet and InceptionResNet exhibiting exceptional performance of 98%.

Also, Hmida and Romdhane [17] introduced a new method for recognizing static Arabic sign language using the deep learning model AlexNet and the categorical cross-entropy loss function. The emphasis of this approach was on accurately identifying 32 classes of sign language alphabets. To showcase its effectiveness, experimental evaluations were carried out using the ArSL2018 dataset. The experimental results revealed that the proposed method achieved an average F-measure of 97.38%.

Furthermore, Alnuaim et al. [18] created a paradigm for categorizing sign language using the Arabic alphabet. The hand position is used to identify sign language in images. The framework was proposed to contain two separately trained CNN models. To get better outcomes, these models' final predictions were integrated. The ArSL2018 dataset was used to train the model and assess its performance. Images undergo a variety of preparation steps, such as scaling to 64×64 pixels, converting grayscale to three-channel pictures, and lowpass filtering using a median filter to smooth and minimize noise. The goal of this preprocessing was to prevent overfitting and increase the model's robustness. Following preprocessing, the input images were fed into two distinct models, ResNet50 and MobileNetV2, which were combined for implementation. The accuracy obtained is almost 97% on the test set for the complete dataset after using several preprocessing approaches, different hyperparameters for each model, and different data augmentation strategies.

(1) From the study of related works, this research presents a novel approach to SLR that addresses several limitations identified within the current literature. Prior studies often: Limited Dataset Scope: Relied on self-collected datasets, neglecting the validation and generalizability offered by well-established benchmarks.

(2) Scalability Issues: Focused on analyzing subsets of datasets, hindering the scalability of the resulting models for broader application.

(3) Performance Bottlenecks: Achieved only average accuracy, failing to reach the potential for high performance in SLR.

(4) Limited Real-World Applicability: Employed image preprocessing and data augmentation techniques, restricting the models' applicability to the specific datasets they were trained on.

(5) Single Algorithm Dependence: Predominantly focused on utilizing single deep learning algorithms, overlooking the potential benefits of integrating multiple algorithms for enhanced performance.

To address these limitations, this work proposes a hybrid deep learning and machine learning model. This model leverages the strengths of both paradigms to achieve improved performance in SLR compared to using them in isolation. We further optimize the integration of multiple algorithms within the model, leading to superior results compared to individual approaches. Furthermore, we employ a well-established dataset encompassing 32 distinct classes. This extensive and diverse dataset allows our model to capture a broader range of variations and enhances its overall robustness. Notably, we directly utilize the dataset without resorting to image processing or data augmentation techniques. This approach ensures the model's ability to handle the complexities and challenges of real-world data, leading to increased practicality and reliability in real-life scenarios. By incorporating these key contributions, this research aims to advance the field of SLR by introducing a more robust, generalizable, and real-world applicable model.

### 3. METHODOLOGY

This section describes the framework designed to enhance the efficiency of the Arabic alphabet letter recognition model. The framework consists of three stages:

Stage 1: Model Design: In this stage, Arabic alphabets characteristics are to be learned by a convolutional neural network model. An extensive dataset of labelled Arabic alphabets is used to train the CNN model.

Stage 2: Hybrid Model Creation: In this stage, a hybrid model is created by combining the CNN model with traditional machine learning (ML) algorithms. The weights learned by the CNN model are transferred to the hybrid model.

Stage 3: Optimization: In this stage, the outputs of the CNN model and the hybrid models are subjected to optimization approaches in order to enhance the efficacy and precision of the Arabic alphabets recognition system.

The following subsections provide more detailed descriptions of each stage.

#### 3.1 CNN model design

Through the extraction of spatial and temporal information from the input pictures, the CNN model is trained to recognize

the characteristics of the Arabic alphabetic letters. Multiple convolutional layers, pooling layers, and fully connected layers make up the CNN model. The input pictures' spatial characteristics are extracted using the convolutional layers. Higher-level features are extracted from the feature maps and their dimensionality is decreased through the use of pooling layers. The collected characteristics are categorized into the various Arabic alphabet letters using the fully connected layers.

In our previous work [9], we introduced an efficient CNN architecture for classifying multiple sign language alphabets. Our methodology focuses on selecting optimizing CNN hyper-parameters and achieves superior prediction time and accuracy compared to other contemporary models. This architecture may be used for a variety of datasets since it can work directly on raw data without requiring any preparation. We thoroughly examined the model's generalization

capabilities using five datasets and three sign language alphabets. In this work we use the same architecture that consists of the image input layer after resizing the image to be 64×64 in grayscale followed by four convolutional layers with number of filters with size 3×3 in sequential order: 32, 64, 128, 128 to produce number of feature maps equal to the same number of filters. This configuration is aimed at achieving higher efficiency in the model [9]. The Relu activation function and max pooling layer, which minimizes the feature map dimension to prevent over-fitting and the computation process, come after each convolutional layer. The pool size is equal to (2×2). After that there are three fully connected layers with size in sequence 1024, 512 and 32 which is the number of 32 classes that needed to be classified. Figure 1 shows the model description. The Adam learning algorithm is used to adjust weights for all layers.

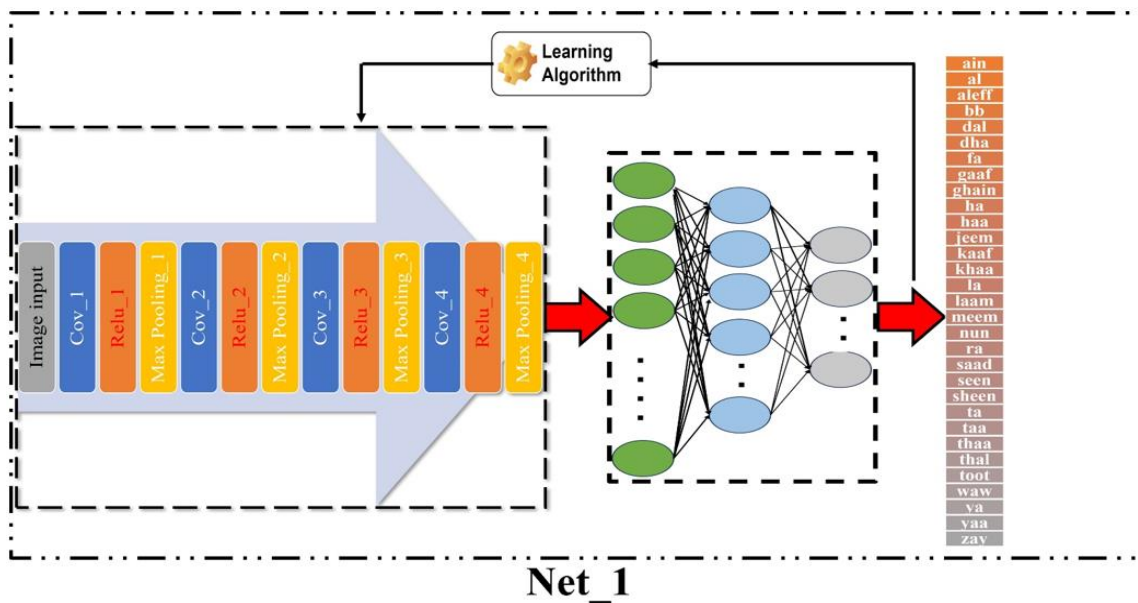


Figure 1. CNN model architecture

### 3.2 Proposed hybrid CNN model with traditional machine learning algorithms

In order to recognize Arabic alphabet letters, this research suggests a hybrid CNN model that combines a convolutional neural network with well-known machine learning techniques including Decision Trees, Random Forests, K-Nearest Neighbors, and Support Vector Machines. These techniques were selected due to their common usage in classification tasks, akin to Feedforward neural networks utilized in CNNs. However, they employ distinct algorithms and functions. For instance, K-Nearest Neighbor classification is based on nodes closest to each other, sometimes resulting in superior outcomes. They are extensively employed by researchers, which justifies their choice for experimentation. The CNN model's weights are passed to the hybrid model, which is trained using a dataset of images of Arabic alphabet letters. This enables the hybrid model to increase the accuracy of the Arabic alphabet letter identification system by utilizing the attributes that the CNN model learns.

#### 3.2.1 Support vector machines (SVMs)

Support vector machines (SVM) are a powerful class of supervised machine learning algorithms used for both

classification and regression tasks. The primary objective of SVM is to find an optimal hyperplane that effectively separates data points into distinct classes while maximizing the margin between these classes. The hyperplane is positioned in a high-dimensional space, and the data points closest to it are referred to as support vectors.

One of SVM's key strengths lies in its ability to handle high-dimensional data effectively. The algorithm achieves this by mapping the input data into a higher-dimensional feature space using a technique called the kernel trick. This transformation enables SVM to address complex relationships within the data that may not be linearly separable in the original space.

SVM is particularly well-suited for scenarios with clear class boundaries, making it effective in cases where the data exhibits a distinct separation between different classes. The margin, which represents the distance between the hyperplane and the nearest data points, is maximized during the training process, contributing to the model's robustness and generalization capabilities.

Furthermore, SVM exhibits resilience to overfitting, making it reliable even when dealing with datasets with noise. The adaptability of SVM to various kernels, such as linear, polynomial, and radial basis function (RBF), enhances its versatility and allows it to handle both linear and non-linear

data.

In summary, Support Vector Machines are widely used due to their ability to create effective decision boundaries in high-dimensional spaces, handle complex relationships through the kernel trick, and provide robust generalization with optimal margins. Their versatility makes SVM a valuable tool in various applications, including image classification, text categorization, and financial forecasting [19].

### 3.2.2 Random forests

Random Forest is a powerful ensemble learning algorithm widely used in machine learning for classification and regression tasks. It operates by constructing multiple Decision Trees during training, where each tree is built on a subset of the data and with a subset of features. The collective predictions of these trees are then aggregated to form the final output.

One key strength of Random Forest lies in its ability to mitigate overfitting. The algorithm introduces randomness by using bootstrapped samples and random feature selection for each tree, promoting diversity among the trees and enhancing the model's generalization capabilities.

Random Forest excels in handling high-dimensional data and is robust against noise and outliers. It is known for its versatility, adaptability to various types of data, and ease of use. The algorithm is effective in both classification and regression tasks and provides valuable insights into feature importance.

Due to its ability to maintain accuracy on diverse datasets and manage complex relationships within the data, Random Forest has found applications in areas such as finance, healthcare, and image recognition. Its popularity is attributed to its simplicity, resilience, and consistently high performance across various domains [20].

### 3.2.3 K-nearest neighbors (KNNs)

KNN is a basic machine learning technique for classification that utilizes the K nearest neighbors' majority class in the feature space to classify a sample. KNN is a straightforward and effective supervised machine learning algorithm used for both classification and regression tasks. The central concept behind KNN is to predict the class or value of a data point based on the classes or values of its neighboring points in the feature space. The "k" in KNN represents the number of nearest neighbors considered for the prediction.

During the training phase, KNN memorizes the entire dataset. When predicting the class or value for a new data point, it identifies the k-Nearest Neighbors based on a chosen distance metric, often using Euclidean distance. The prediction is then determined by majority voting for classification or averaging for regression among these neighbors.

KNN is non-parametric, meaning it doesn't make explicit assumptions about the underlying data distribution. It is versatile, simple to implement, and suitable for scenarios where the data distribution is not well-defined or changes over time. However, KNN's performance can be sensitive to the choice of distance metric and the value of k [19].

### 3.2.4 Decision trees

Decision Trees (DT) are a fundamental machine learning algorithm used for both classification and regression tasks. The algorithm employs a tree-like structure to make decisions based on input features. At each internal node of the tree, a decision is made by evaluating a specific feature, and the tree

branches into subtrees corresponding to different feature values. This process continues until reaching leaf nodes, where the final decision or prediction is made.

Decision Trees are attractive due to their interpretability and ease of understanding. They effectively capture decision-making processes by creating a hierarchical structure that represents the relationships between input features and the target variable. However, they are prone to overfitting, especially on complex datasets, which can be mitigated using techniques like pruning.

Ensemble methods, such as Random Forests, are often employed to enhance the robustness and accuracy of Decision Trees by combining the predictions of multiple trees. Despite their limitations, Decision Trees remain valuable tools in various domains, including finance, healthcare, and natural language processing [19].

### 3.2.5 Hybrid model architecture

The proposed hybrid CNN model for Arabic alphabet letter recognition is a novel approach that combines the strengths of CNNs and traditional machine learning algorithms. The CNN model possesses the ability to acquire intricate characteristics from images of Arabic alphabet letters, whereas conventional machine learning algorithms can grasp intricate connections between these characteristics and the corresponding class labels.

Figure 2 illustrates the hybrid model, which comprises multiple convolutional layers, a pooling layer, and a fully connected layer. In this hybrid model, the fully connected layer is substituted with traditional machine learning algorithms, serving as classifiers at the model's output. These traditional machine learning algorithms are employed as classifiers at the output of the hybrid model. The weights learned by the CNN model are transferred to the hybrid model, which allows the hybrid model to leverage the features learned by the CNN model.

### 3.2.6 Hybrid model training procedure

The proposed hybrid CNN model is trained using a two-step procedure:

(1) CNN model pre-training: the CNN model undergoes pre-training on a collection of Arabic alphabet letter images. This initial training enables the CNN model to acquire intricate features from the dataset.

(2) Traditional machine learning algorithm training: The traditional machine learning algorithms are then trained on the output features of the pre-trained CNN model to classify the Arabic alphabet letters.

## 3.3 Ensemble technique for improved accuracy

The final stage of our proposed hybrid CNN model focuses on combining the outputs from the five individual networks (Net\_1, Net\_2, Net\_3, Net\_4, and Net\_5) to achieve improved overall recognition accuracy. This technique is known as ensemble learning.

We employ a weighted averaging approach for ensemble learning. This involves assigning optimal weights to each network's output based on its performance. The goal of optimization is to find these weights that minimize the loss function on the validation data. Essentially, the optimization algorithm identifies how much "trust" to give to each network's prediction. Figure 3 (Optimized Net) illustrates this ensemble technique. Each network independently produces 32



probability values, corresponding to the 32 classes in the Arabic alphabet. The optimization process results in a  $5 \times 32$

matrix, where each row represents the optimal weight vector for the corresponding network in the ensemble.

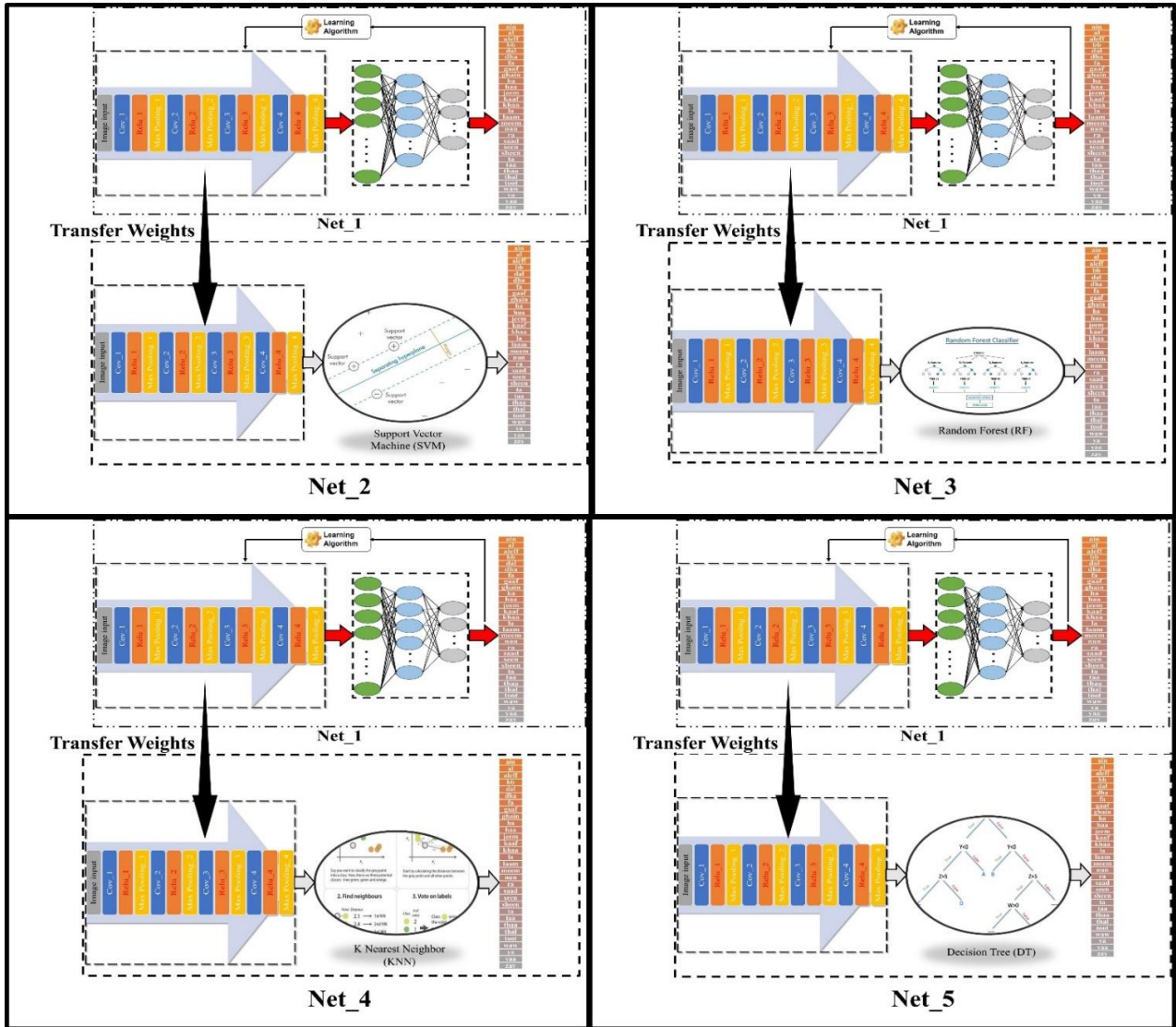


Figure 2. Hybrid CNN Model and traditional machine learning algorithms

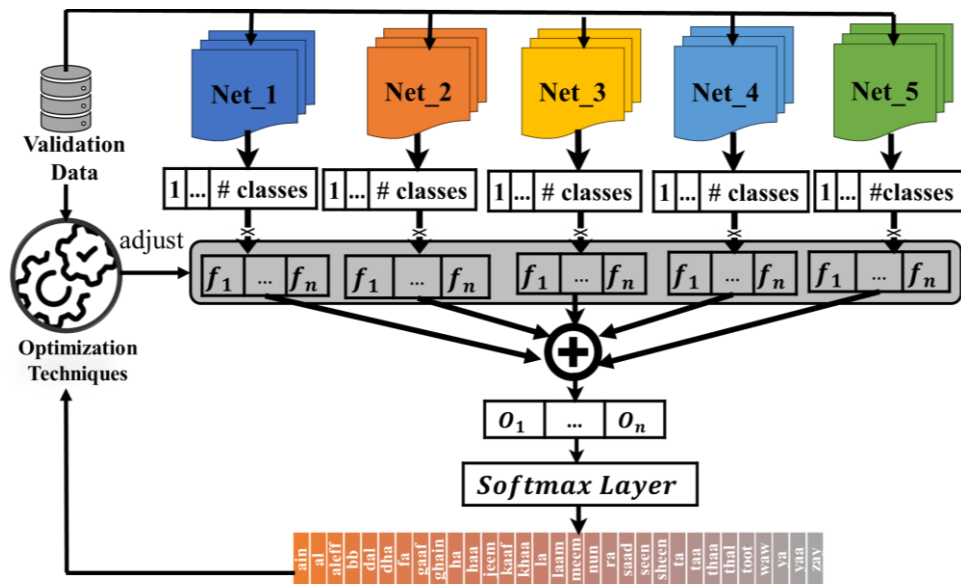


Figure 3. Optimization of CNN and hybrid ML techniques

The proposed ensemble technique operates as following:

**Weighted sum:** The output (probability vector) of each network is multiplied by the corresponding row weights from the optimization matrix. This creates a weighted sum, prioritizing the contribution of higher-performing networks.

**SoftMax layer:** The weighted sum is then fed into the SoftMax layer. This layer normalizes the weighted outputs into probabilities, ensuring they sum to one. Finally, the SoftMax layer provides the final classification output, indicating the most likely class for the given sign language gesture.

By combining the strengths of individual networks through weighted averaging, this ensemble approach aims to achieve superior recognition accuracy compared to relying on any single network alone.

### 3.4 Optimized hybrid machine learning techniques

A crucial step is finding the optimal weights for each network within the ensemble. This optimization process aims to minimize the loss function on a dedicated validation dataset. The validation set helps identify the best weight combination that generalizes well to unseen data (data not used for training). Six optimization techniques are used in this stage: Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, Sine Cosine Algorithm, Firefly Algorithm, and Harris Hawk's Optimization. These six algorithms were chosen for their widespread use and proven effectiveness across multiple fields. They are commonly employed by researchers in various domains, including medical and engineering fields, to optimize variables effectively. In this model, we aimed to determine the optimal weights that achieve the best classification for sign language alphabet letters. Additionally, factors of optimization techniques selection such as problem structure, problem size, problem uncertainty, and others were all applicable to our classification problem. These techniques are all population-based algorithms, meaning that they work by maintaining a population of candidate solutions and iteratively improving them over time. The optimization techniques are used in this stage are listed as follows:

**Genetic algorithm (GA):** GA is a computational technique that draws inspiration from natural selection. It tackles intricate optimization problems by emulating the evolution of a population consisting of potential solutions. Each solution, represented as an individual, undergoes assessment and undergoes genetic operations such as crossover and mutation. This iterative process continues until either a satisfactory solution is found or a termination condition is met. GA finds extensive application across various domains owing to its capacity to explore vast solution spaces and adapt to dynamic environments, rendering it a powerful tool for addressing complex problems [21].

**Particle swarm optimization (PSO):** PSO is a computational approach that emulates the collective behavior of natural swarms to address optimization problems. It employs a population of particles that traverse through a search space, adapting their positions and velocities based on individual experiences and the shared knowledge of the swarm. The iterative adjustment of particle positions enables PSO to gradually converge towards an optimal solution. Renowned for its simplicity and rapid convergence to near-optimal solutions, PSO has gained popularity as an efficient technique for tackling complex optimization problems [22].

**Differential evolution (DE):** DE is an algorithm for optimization that enhances potential solutions through mutation, recombination, and selection. It randomly initializes a population and generates new solutions by mutating and recombining individuals. Fitness evaluation determines the best solutions for replacement. DE iterates until a termination condition is met. It is widely used in different domains and is valued for its simplicity, versatility, and effectiveness in solving complex optimization problems [23].

**The sine cosine algorithm (SCA):** SCA is an optimization method that employs sine and cosine functions to navigate the exploration of optimal solutions. It commences with a random population and iteratively updates solution positions based on the sine and cosine functions. By combining exploration and exploitation, the algorithm efficiently searches for the most favorable solutions. SCA incorporates techniques such as randomization and local search, while dynamically adapting its parameters. The iterative process continues until a termination criterion is satisfied, at which point the best solution discovered is considered the output. The SCA algorithm is characterized by its simplicity, ease of implementation, and its ability to yield promising results when applied to a wide range of optimization problems [24].

**Firefly algorithm (FA):** FA is an optimization technique inspired by the captivating flashing behavior of fireflies. It endeavors to address optimization problems by simulating the movement and attraction observed among fireflies. FA initiates with an initial population of fireflies and iteratively adjusts their positions, taking into account their brightness and attractiveness. By incorporating the principles of attraction and randomness, the algorithm adeptly explores and exploits the search space. FA has proven its efficacy in solving diverse optimization problems and is highly regarded for its simplicity, adaptability, and capacity to handle both continuous and discrete search spaces [25].

**Harris hawks optimization (HHO):** HHO is an optimization algorithm that draws inspiration from the cooperative hunting behavior of Harris hawks. It emulates the collaborative hunting strategy of these birds to address optimization problems. HHO begins with a population of potential solutions and iteratively enhances them by employing a blend of exploration and exploitation techniques. The algorithm integrates diverse search operators, including prey capture, leadership, and knowledge sharing, to effectively traverse the search space. HHO has demonstrated encouraging outcomes in solving intricate optimization problems and is esteemed for its capacity to strike a balance between exploration and exploitation while harnessing the power of collaboration among individuals [26].

### 3.5 Fitness function design

The fitness function is a crucial component of optimization algorithms, guiding the search process towards optimal solutions. It is a metric that quantifies the quality of a candidate solution and is tailored to the specific problem being solved. In machine learning, accuracy is often used as the fitness function, but other metrics, such as error measurement, may be more appropriate depending on the task.

The objective of the fitness function employed in this study is to identify the optimum weights for each network within the ensemble of hybrid machine learning methods, with the goal of minimizing the loss on the validation dataset. The fitness function encompasses the following steps:

1-Multiply the output of each network by the corresponding optimization factors and then sum the resulting values for each class output. This is represented by Eq. (1).

$$PT_j = \sum_{i=1}^n f_{ij}P_{ij} \quad (1)$$

where,  $f_{ij}$  is the optimization factor for each output  $j$  in the network  $i$ ,  $P_{ij}$  is the predict output of the corresponding network  $i$  for each class  $j$ ,  $i$  is the network model and  $j$  is the output class number, and  $n$  is the quantity of the categories.

2-Utilize the SoftMax function on the output acquired in Step 1. This is represented by Eq. (2).

$$\sigma_j = \frac{e^{PT_j}}{\sum_{j=1}^n PT_j} \quad (2)$$

where,  $\sigma_j$  is the total output of the final SoftMax layer.

3-Compute the fitness equation employed, which is referred to as the average absolute difference between the predicted output ( $\sigma_{kj}$ ) and the target output ( $T_{kj}$ ) for all images within the validation dataset. This is represented by Eq. (3).

$$Fitness\ Function = \frac{1}{n \times m} \sum_{k=1}^m \sum_{j=1}^n |T_{kj} - \sigma_{kj}| \quad (3)$$

where, 'n' represents the total number of classes and 'm' denotes the count of images within the validation datasets.

#### 4. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the results of the proposed ensemble of hybrid CNN and machine learning models with optimization techniques, compared against five other models, (1) CNN model: This model relies on integrated layers that enable it to extract distinctive features from images and improve classification performance. (2) Hybrid CNN with SVM model: The CNN model is combined with the Support Vector Machine model into a single model. This model benefits from CNN's ability to extract features and represent images, while leveraging SVM's accuracy in classification. (3) Hybrid CNN with RF model: This model combines CNN with Random Forest, a technique used in machine learning to enhance performance. RF is used to improve the classification process based on the features extracted by CNN. (4) Hybrid CNN with DT model: This model integrates CNN with Decision Tree, a technique used in classification and decision-making based on a set of predefined rules. DT is employed here to enhance the classification process using the features extracted by CNN. (5) Hybrid CNN with KNN model: This model combines CNN with K-Nearest Neighbors, a technique that relies on the concept of proximity for data classification. The model benefits from CNN's ability to extract good features and utilizes KNN to improve the classification process.

##### 4.1 Dataset description

The dataset used in this work is called ArSL 2018 [27], which consists of 54,049 images divided into 32 characters of the Arabic alphabet. Each image represents a sign indicating a letter of the Arabic language. It is a publicly available, free,

and static dataset that was collected by 40 volunteers to represent the 32 characters in different positions and backgrounds. The dataset is unbalanced because the number of images within each class is not equal. The data was divided as follows: 70% for training, which is equal to 37,835 images, and the remaining data was equally divided between validation and testing. This means that 8,110 images were allocated for validation and 8,114 images for testing. In each training iteration, the data was randomly shuffled to ensure that the model is protected against the drawbacks of overfitting. Figure 4 displays the distribution of images across individual classes, while Figure 5 presents a representative sample of images from each class.

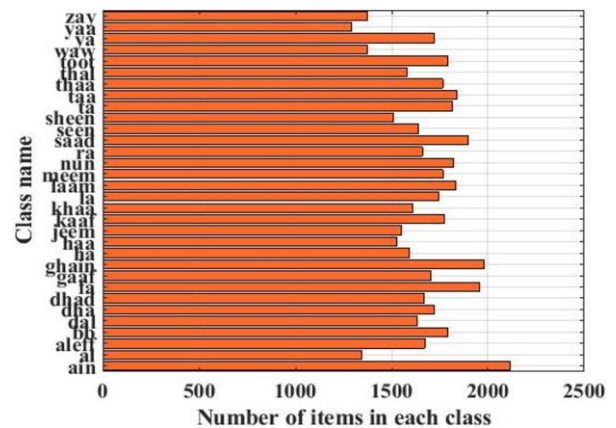


Figure 4. Number of samples in each class



Figure 5. Samples of ArSL dataset

##### 4.2 Experimental setup

The experiment employed the MATLAB® 2020 Deep Learning Toolbox on a system equipped with a 2.60 GHz Intel i5 CPU, 8 GB RAM, Intel 4K graphics, and AMD Radeon 7500M/7600M series. To ensure the model's robustness and minimize parameter bias, the experiment was repeated 10 times. Each trial involved randomly splitting the dataset into training and testing sets, and the results were averaged across the trials [13]. This approach aimed to assess the model's performance under various randomization conditions, yielding more reliable and unbiased outcomes. To address the issue of overfitting, a dropout layer was incorporated into the CNN model [28].

For evaluating the models, the following metrics were utilized: accuracy, precision, recall, and F-score. Accuracy measures the ratio of correctly classified instances to the total number of instances and is calculated using Eq. (4). Precision measures the ratio of correctly predicted positive instances to the total predicted positive instances and is calculated using



Eq. (5). Recall, also referred to as sensitivity or true positive rate, measures the ratio of correctly predicted positive instances to the total actual positive instances, calculated using Eq. (6). The F-score, also known as the F1 score, is a balanced measure that combines precision and recall through harmonic means. It is calculated using Eq. (7).

$$Accuracy = \frac{TP + TN}{P + N} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{P} \quad (6)$$

$$F - score = \frac{2 \times precision \times recall}{precision + recall} \quad (7)$$

### 4.3 Performance evaluation of the proposed model

In this part of this section, the results obtained from the five networks (Net\_1, Net\_2, Net\_3, Net\_4, Net\_5 and Optimized Net) be presented in section 3 (methodology) are presented.

Net\_1: A CNN with a fully connected layer is training on the training dataset. Each experiment was trained for 15 epochs, and the highest accuracy was achieved in epoch 15. Figure 6 illustrates the training and validation of the CNN model.

Net\_2, Net\_3, Net\_4 and Net\_5: Hybrid networks between CNN and SVM, RF, KNN and DT, consequently, trained with the same data for Net\_1. Figure 7 shows the losses for Net\_3.

Propose Optimized Net: Six optimization techniques were conducted on the validation dataset, and the experiment was run 10 times with 500 iterations each time. The average results were taken. Figure 8 illustrates the losses during the experiment, showing that the FA technique performed the best in terms of accuracy and converging speed. Afterward, the minimum, maximum and average fitness values were calculated on the validation data using the best solution from each algorithm, and it was presented in Table 1. It also confirms that the FA technique is the best and also the minimum, maximum and standard deviation for each experiment.

Table 2 displays the accuracy, average precision, average recall, and average F-score achieved by all the preceding methods on the test data following the training phase. The results demonstrate that the latest method, the Optimized Net using FA, achieves the best performance by a significant margin. Figure 9 also illustrates the confusion matrix of the optimized Net using FA with a summary of rows and columns, indicating the number of correctly classified instances for each class.

Figure 10 illustrates the Box Plot, a statistical method used

to compare the results obtained from the 6 optimization techniques utilized in the proposed model. It is evident that DE and PSO have the lowest Interquartile Range (IQR), indicating a smaller spread of results. However, they are not the best performers as their median values are not the highest. Additionally, it can be observed that FA is the best performer overall, with the smallest IQR, indicating a smaller spread of results, and also having the lowest median among the 6 algorithms used. Hence, it was employed in the proposed model for result validation.

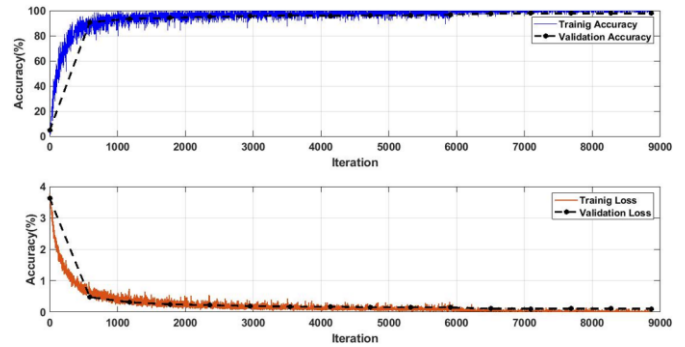


Figure 6. Training and validation accuracies and losses for Net\_1

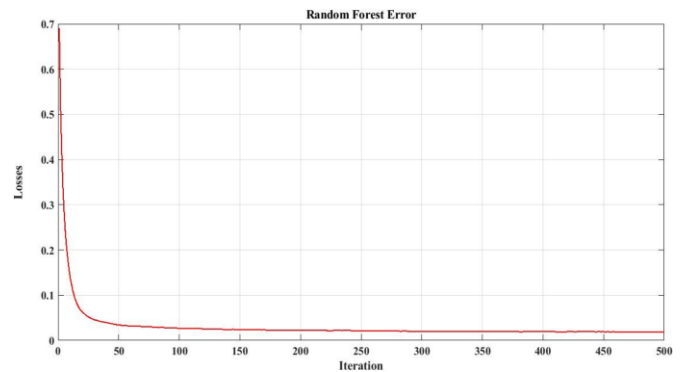


Figure 7. Net\_3 losses

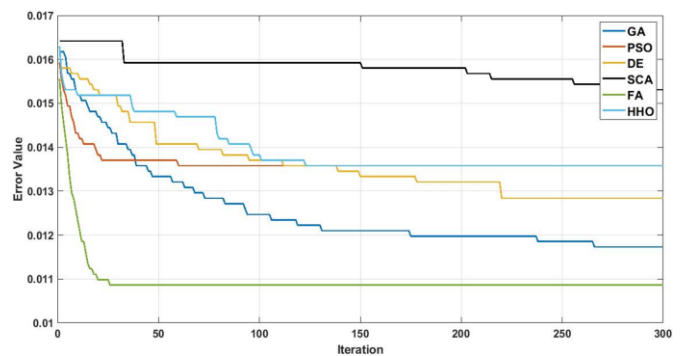


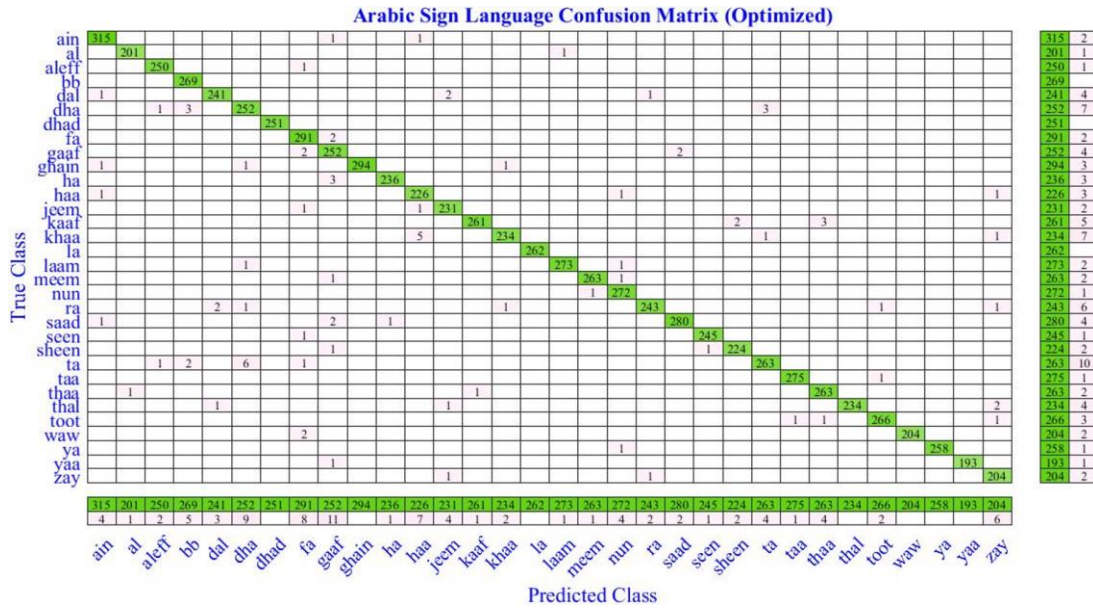
Figure 8. Optimization losses

Table 1. Mean, minimum, and maximum fitness on validation data from 10 runs of optimization techniques

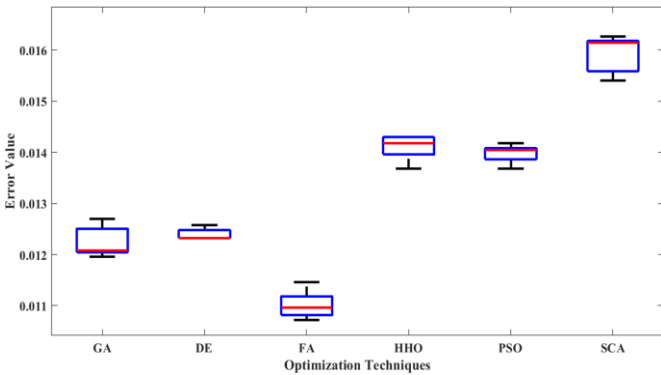
Method	Average Fitness	Min. Fitness	Max. Fitness	Std.
GA	0.012519	0.012222	0.012716	0.00022427
PSO	0.013975	0.013704	0.014321	0.00023747
DE	0.012617	0.012469	0.01284	0.00013524
SCA	0.016049	0.015679	0.016543	0.00038052
FA	0.011235	0.010988	0.011481	0.0001952
HHO	0.015185	0.014568	0.015556	0.00037037

**Table 2.** Accuracy, precision, recall and f-score for 6 networks

	Net 1	Net 2	Net 3	Net 4	Net 5	Optimized Net	FA
Accuracy	98.57%	98.48%	97.72%	96.42%	86.64%	98.90%	
Precision (avg.)	98.57%	98.48%	97.69%	96.4%	86.53%	98.89%	
Recall (avg.)	98.59%	98.5%	97.78%	96.6%	86.76%	98.92%	
F-score (avg.)	98.58%	98.5%	97.74%	96.5%	86.65%	98.91%	



**Figure 9.** Confusion matrix of ArSL using optimized net



**Figure 10.** Box Plot to illustrate the statistical differences in the results of optimization techniques

**4.4 Comparison with other models**

In this subsection, A comparison was made between the proposed model and the models that have been recently

published in various articles. It was evident that the proposed model outperformed its counterparts because it incorporated several techniques such as hybrid techniques and the utilization of optimization techniques to adjust the ensemble parameters. This allowed the proposed model to operate directly on the raw data without requiring any preprocessing, unlike the other counterparts. As a result, it can be applied to any data. Table 3 illustrates the comparison between the proposed algorithm and the other counterparts.

Table 3 also highlights those references [9, 13] utilized the same technique and dataset as the proposed model. However, they obtained lower results. In some cases, like reference [12], a portion of the same data was used along with the same technique, yet they achieved inferior results. Thus, the proposed model outperformed them by at least 0.5%. Additionally, references [11, 16-18] employed transfer learning techniques, which are modern applications, on the same dataset used in the proposed model. However, they also achieved results inferior to the presented model by at least 1%.

**Table 3.** Comparison between the proposed model and other articles

Ref. No.	Dataset	No. of Samples	Classification Method	Accuracy
[13]	ArSL2018	54059	CNN	97.29%
[15]	Collected	18360	CNN	93%
[16]	ArSL2018	54059	Efficient B4	95%
[18]	ArSL2018	54059	ResNet50, MobileNetV2	97%
[17]	ArSL2018	54059	Alex Net	97.39%
[9]	ArSL2018	54059	CNN	98.47%
[12]	Part of ArSL2018	24137	CNN with sampling techniques	97.79%
[14]	Collected	7057	CNN with Media pipe	97.1%
[11]	ArSL2018	54059	ResNet, InceptionResNet	98%
Proposed model	ArSL2018	54059	Optimized hybrid CNN and ML	98.9%

## 5. CONCLUSION AND FUTURE WORK

In this research, a novel hybrid CNN model is introduced for the recognition of Arabic alphabet letters. The model is trained on a comprehensive dataset called ArSL2018 and has achieved remarkable outcomes, setting a new benchmark. By combining the advantages of CNNs and traditional machine learning algorithms, this hybrid model effectively captures intricate features and relationships within the data. Furthermore, the model demonstrates efficiency and robustness, making it well-suited for real-world applications.

The suggested model amalgamated the CNN model with four traditional machine learning algorithms (Support Vector Machine, Random Forest, Decision Tree, and K-Nearest Neighbors). These models were integrated and subjected to optimization techniques to enhance their performance. The outcomes demonstrated that the optimized hybrid techniques achieved exceptional accuracy, reaching nearly 99%, surpassing the performance of the individual models. The proposed model outperformed both the individual CNN and machine learning models, highlighting the effectiveness of the hybrid approach and the significance of optimization in attaining state-of-the-art results.

For future developments, the suggested model could broaden its scope to encompass real-world applications through the creation of a mobile app or a Raspberry Pi-based application. This expansion would facilitate testing in practical contexts and allow for user evaluations. Furthermore, the model could undergo assessment on more varied datasets comprising diverse words and sentences, enabling a thorough examination of its capacity to handle a broader array of data and generalize to new inputs.

Moreover, the current implementation of the model relies on static images. However, future enhancements may involve adapting it to process dynamic images, including the utilization of video datasets or the integration of sensors to capture real-time imagery.

## REFERENCES

- [1] McPhillips, E. (2022). World wide hearing loss: Stats from around the world. Audicus. <https://www.audicus.com/world-wide-hearing-loss-stats-from-around-the-world/>, accessed on September 10, 2022.
- [2] World Health Organization. (2023). Deafness and hearing loss. <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>.
- [3] International House Cairo. (2023). Is Arabic the Hardest Language in the World? <https://ihcairoeg.com/arabic/arabic-the-hardest-language/>.
- [4] Núñez-Marcos, A., Perez-de-Viñaspre, O., Labaka, G. (2023). A survey on sign language machine translation. *Expert Systems with Applications*, 213: 118993. <https://doi.org/10.1016/j.eswa.2022.118993>
- [5] Ronchetti, F., Quiroga, F. M., Estrebou, C., Lanzarini, L., Rosete, A. (2023). LSA64: An Argentinian sign language dataset. arXiv preprint arXiv:2310.17429. <https://doi.org/10.48550/arXiv.2310.17429>
- [6] Adeyanju, I.A., Bello, O.O., Adegboye, M.A. (2021). Machine learning methods for sign language recognition: A critical review and analysis. *Intelligent Systems with Applications*, 12:200056. <https://doi.org/10.1016/j.iswa.2021.200056>
- [7] Mohammed, R.M., Kadhem, S.M. (2021). A review on Arabic sign language translator systems. In *Journal of Physics: Conference Series*, 1818(1): 012033. <https://doi.org/10.1088/1742-6596/1818/1/012033>
- [8] Al-Obodi, A.H., Al-Hanine, A.M., Al-Harbi, K.N., Al-Dawas, M.S., Al-Shargabi, A.A. (2020). A Saudi sign language recognition system based on convolutional neural networks. *International Journal of Engineering Research and Technology*, 13(11): 3328-3334. <https://doi.org/10.37624/IJERT/13.11.2020.3328-3334>
- [9] Mahmoud, A.O., Ziedan, I., Zamel, A.A. (2023). An efficient convolutional neural network classification model for several sign language alphabets. *International Journal of Advanced Computer Science and Applications*, 14(9): 1040-1050. <https://doi.org/10.14569/IJACSA.2023.01409108>
- [10] Obi, Y., Claudio, K. S., Budiman, V.M., Achmad, S., Kurniawan, A. (2023). Sign language recognition system for communicating to people with disabilities. *Procedia Computer Science*, 216: 13-20. <https://doi.org/10.1016/j.procs.2022.12.106>
- [11] Alharthi, N.M., Alzahrani, S.M. (2023). Vision transformers and transfer learning approaches for Arabic sign language recognition. *Applied Sciences*, 13(21): 11625. <https://doi.org/10.3390/app132111625>
- [12] AbdElghfar, H.A., Ahmed, A.M., Alani, A.A., et al. (2023). A model for qur'anic sign language recognition based on deep learning algorithms. *Journal of Sensors*, 2023(1): 9926245. <https://doi.org/10.1155/2023/9926245>
- [13] Alani, A.A., Cosma, G. (2021). ArSL-CNN: A convolutional neural network for Arabic sign language gesture recognition. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2): 1096-1107. <https://doi.org/10.11591/ijeecs.v22i2.pp1096-1107>
- [14] Moustafa, A.M.A., Mohd Rahim, M.S., Bouallegue, B., Khattab, M.M., Soliman, A.M., Tharwat, G., Ahmed, A.M. (2023). Integrated mediapipe with a CNN model for Arabic sign language recognition. *Journal of Electrical and Computer Engineering*, 2023(1): 8870750. <https://doi.org/10.1155/2023/8870750>
- [15] Alawwad, R.A., Bchir, O., Ismail, M.M.B. (2021). Arabic sign language recognition using Faster R-CNN. *International Journal of Advanced Computer Science and Applications*, 12(3): 692-700. <https://doi.org/10.14569/IJACSA.2021.0120380>
- [16] Zakariah, M., Alotaibi, Y.A., Koundal, D., Guo, Y., Mamun Elahi, M. (2022). Sign language recognition for Arabic alphabets using transfer learning technique. *Computational Intelligence and Neuroscience*, 2022(1): 4567989. <https://doi.org/10.1155/2022/4567989>
- [17] Hmida, I., Romdhane, N.B. (2022). Arabic sign language recognition algorithm based on deep learning for smart cities. In *The 3rd International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2022)*, Hybrid Conference, Sharjah, United Arab Emirates, pp. 119-127. <https://doi.org/10.1049/icp.2022.2426>
- [18] Alnuaim, A., Zakariah, M., Hatamleh, W.A., Tarazi, H., Tripathi, V., Amoatey, E. T. (2022). Human-computer interaction with hand gesture recognition using ResNet and MobileNet. *Computational Intelligence and Neuroscience*, 2022(1): 8777355.

- <https://doi.org/10.1155/2022/8777355>
- [19] Bansal, M., Goyal, A., Choudhary, A. (2022). A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decision Analytics Journal*, 3: 100071. <https://doi.org/10.1016/j.dajour.2022.100071>
- [20] Das, S., Imtiaz, M.S., Neom, N.H., Siddique, N., Wang, H. (2023). A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. *Expert Systems with Applications*, 213: 118914. <https://doi.org/10.1016/j.eswa.2022.118914>
- [21] Yilmaz, A.A. (2022). A novel hyperparameter optimization aided hand gesture recognition framework based on deep learning algorithms. *Traitement du Signal*, 39(3): 823-833. <https://doi.org/10.18280/ts.390307>
- [22] Lakumalla, N., Kumar, P.K. (2023). Enhanced single-snapshot 1-D and 2-D DOA estimation using particle swarm optimization. *Traitement du Signal*, 40(3): 1267-1273. <https://doi.org/10.18280/ts.400345>
- [23] Rugmini, S., Linsely, J.A. (2023). Diagnosis of melanoma using differential evolution optimized artificial neural network. *Traitement du Signal*, 40(3): 1203-1209. <https://doi.org/10.18280/ts.400337>
- [24] Saka, M., Coban, M., Eke, I., Tezcan, S.S., Taplamacioğlu, M.C. (2021). A novel hybrid global optimization algorithm having training strategy: HybridTaguchi-vortex search algorithm. *Turkish Journal of Electrical Engineering and Computer Sciences*, 29(4): 1908-1928. <https://doi.org/10.3906/elk-2004-193>
- [25] Ghasemi, M., kakhoda Mohammadi, S., Zare, M., Mirjalili, S., Gil, M., Hemmati, R. (2022). A new firefly algorithm with improved global exploration and convergence with application to engineering optimization. *Decision Analytics Journal*, 5: 100125. <https://doi.org/10.1016/j.dajour.2022.100125>
- [26] Tripathy, B.K., Reddy Maddikunta, P.K., Pham, Q.V., Gadekallu, T.R., Dev, K., Pandya, S., ElHalawany, B.M. (2022). Harris hawk optimization: A survey on variants and applications. *Computational Intelligence and Neuroscience*, 2022(1): 2218594. <https://doi.org/10.1155/2022/2218594>
- [27] Latif, G., Mohammad, N., Alghazo, J., AlKhalaf, R., AlKhalaf, R. (2019). ArASL: Arabic alphabets sign language dataset. *Data in Brief*, 23: 103777. <https://doi.org/10.1016/j.dib.2019.103777>
- [28] Alzubaidi, L., Zhang, J., Humaidi, A.J., et al. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8: 53. <https://doi.org/10.1186/s40537-021-00444-8>