Recent Progress in Arabic Sign Language Recognition: Utilizing Convolutional Neural Networks (CNN)

Mosab. A. Hassan[1*], *Alaa*.H.Ali[1] and *Atheer* A. Sabri[2]

[1]Department of Electrical Engineering, University of Technology,Iraq.
[2]Department of Communication Engineering, University of Technology,Iraq.

Abstract. The advancement of assistive communication technology for the deaf and hard-of-hearing community is an area of significant research interest. In this study, we present a Convolutional Neural Network (CNN) model tailored for the recognition of Arabic Sign Language (ArSL). Our model incorporates a meticulous preprocessing pipeline that transforms input images through grayscale conversion, Gaussian blur, histogram equalization, and resizing to standardize input data and enhance feature visibility. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are employed for feature extraction to retain critical discriminative information while reducing dimensionality. The proposed CNN architecture leverages a blend of one-dimensional convolutional layers, max pooling, Leaky ReLU activation functions, and Long Short-Term Memory (LSTM) layers to efficiently capture both spatial and temporal patterns within the data. Our experiments on two separate datasets—one consisting of images and the other of videos—demonstrate exceptional recognition rates of 99.7% and 99.9%, respectively. These results significantly surpass the performance of existing models referenced in the literature. This paper discusses the methodologies, architectural considerations, and the training approach of the proposed model, alongside a comparative analysis of its performance against previous studies. The research outcomes suggest that our model not only sets a new benchmark in sign language recognition but also offers a promising foundation for the development of real-time, assistive sign language translation tools. The potential applications of such technology could greatly enhance communication accessibility, fostering greater inclusion for individuals who rely on sign language as their primary mode of communication. Future work will aim to expand the model's capabilities to more diverse datasets and investigate its deployment in practical, everyday scenarios to bridge the communication gap for the deaf and hard of hearing community.

1 Introduction

Sign language is a vital nonverbal form of communication, especially important for people who are deaf or hard-of-hearing. It enables direct communication without the need for an interpreter, making it an essential subject of study in numerous fields [1–4]. This language relies heavily on a mix of movements involving the body, arms, head, fingers, and hands, as well as facial expressions, to convey messages effectively.In the domain of human-computer interaction, hand gestures are especially significant, particularly in Sign Language Recognition (SLR). SLR is crucial for creating communication systems for the deaf-mute community and has received increased attention due to the rising number of deaf and hard-of-hearing individuals worldwide and the growing prevalence of vision-based application devices [5–7]. Contemporary research in this field often focuses on vision-based SLR systems, utilizing different types of camera inputs, including 3D, web, and stereo cameras [8].The popularity of vision-based methods stems from their cost-effectiveness and the fact that they don't require specialized equipment, unlike sensor-based systems [9]. However, these systems face challenges like complex backgrounds and uncontrolled environments. To overcome these, many researchers are exploring segmentation techniques [4,10].

*Corresponding author :eee.20.14@grad.uotechnology.edu.iq

Hearing loss is a major global health issue, as identified by the World Health Organization, impacting about 5% of the world's population, or over 460 million people, including 34 million children. Projections indicate that by 2050, nearly 900 million individuals might be affected by hearing loss. Moreover, 1.1 billion young people are at risk of hearing impairment due to loud noise exposure and other factors. The economic burden is significant, with hearing loss costing the global economy around 750 billion dollars [11].

Hearing impairment varies in severity, with those experiencing severe to profound loss often struggling with understanding spoken language. This leads to communication barriers, which can adversely affect mental health, potentially resulting in feelings of isolation and unhappiness in the deaf community.

Sign language, a visual-gestural language using hand gestures, facial expressions, and body movements, is crucial for the deaf community to bridge these communication gaps. However, this form of communication is not commonly understood by the hearing population, further complicating interactions between deaf and hearing individuals.

There are approximately 200 distinct sign languages worldwide, each with its unique structure and lexicon, reflecting the rich cultural diversity of the deaf community. This diversity, while enriching, also adds complexity to effective communication across different sign languages.

Sign language is essential for the deaf community, using bodily actions to convey messages. It differs from spoken languages in its reliance on physical expressions like hand movements and facial expressions. Each gesture in sign language corresponds to a letter, word, or emotion, forming phrases in a similar way to how spoken language forms sentences. The goal is to facilitate interaction both within the deaf community and between deaf and hearing individuals, recognizing sign language as a complete natural language with its own grammar and structure.

DL, a branch of machine learning algorithms, is vital in representing complex structures through multiple layers of nonlinear transformations. Neural networks form the foundation of DL, driving advancements in various fields including image and sound processing, automated language processing, computer vision, and medical diagnosis.

DL algorithms use computational methods to discover patterns in large datasets, extracting data representations across multiple layers. This involves backpropagation to adjust internal parameters. Deep Convolutional Networks (DCN) are particularly effective in processing videos, images, and audio, while recurrent networks excel with sequential data like voice and text.

The architecture of neural networks is key in DL, with the term "deep" referring to the number of layers, indicating the system's complexity and capability. DL is known for its accuracy, often surpassing human abilities, thanks to modern tools and methods.

The ultimate objective is to develop technology capable of recognizing sign language, translating common gestures of deaf individuals into written data. This aims to bridge communication gaps and enhance understanding and interaction between the deaf and hearing communities.

Deep learning models have recently been applied to address these challenges in SLR. However, one significant issue that arises is the handling of redundant backgrounds, which can complicate the training of Convolutional Neural Networks (CNNs) [12]. Another challenge is managing translated, rotated, or scaled (RTS) test images, which often result from the signer's lack of awareness of the optimal positioning. These RTS variations can lead to images captured from different angles, posing a challenge for consistent recognition [13,14]. To address these issues, researchers are developing segmentation methods and RTS-invariant deep learning models.

Sign Language (SL) consists of four primary manual components essential for communication [15]. In automatic sign recognition, the process generally involves two steps: identifying features and classifying the input data. Progress in classifying and detecting sign languages has greatly improved the effectiveness of automatic SL systems.

SL differs from spoken languages in that it is primarily used by the deaf community and relies largely on expressive body gestures for communication [16]. Each gesture or sign in SL represents a distinct letter, emotion, or word, and by combining these signs, complex ideas can be conveyed, similar to sentence structures in spoken language. Hence, SL is considered a natural language with its own grammar and sentence structure [17].

Meanwhile, Deep Learning (DL) is a subset of Machine Learning (ML), characterized by networks capable of learning unsupervised from unlabeled or unstructured data, known as Deep Neural Networks (DNN) or Deep Neural Learning [18]. Within DL, Convolutional Neural Networks (CNNs) are widely used in computer vision (CV), where the main objective is to capture gesture images and extract primary features for identification. CNNs have found applications in various fields [19].

2 Related Work

In a series of advancements in Arabic Sign Language (ArSL) recognition, various research groups have proposed innovative approaches to enhance the accuracy and efficiency of sign language interpretation using computer vision and machine learning techniques.

One such approach, discussed in [20], centers around an appearance-based feature strategy, employing a combination of Local Binary Patterns (LBP) and PCA for feature reduction. The method effectively employs skin color detection using the YCbCr color space for hands and head, and Hidden Markov Models (HMM) for classifying these features. This approach demonstrated remarkable results in signer dependent recognition, achieving a 99.97% recognition rate on the ArSL database using LBP and PCA features.

Another significant contribution, outlined in [21], presents an automated translation model that integrates facial expressions with manual alphabet motions in Arabic sign language. This model focuses on the spatial positioning of the mouth, nose, and eyes, crucial for interpreting facial expressions. The unique aspect of this model is its ability to process images of the signer's hands without requiring gloves or visible markings, thus enabling more natural interaction. The model includes preprocessing and skin detection, converting RGB images to YCbCr, and feature extraction based on the Centroid concept. Its accuracy reaches 90% for facial expressions and 99% for hand signs, using minimum distance classifier (MDC) and absolute difference authors.

In a third study, researchers in [22] developed a method for translating Arabic Sign Language into text automatically. This system combines image characteristics from two datasets: an Arabic Sign Language dictionary and various signers. The process involves several stages, including image and video capture, segmentation, hand edge detection, hand sign construction, classification, and finally, text transformation and interpretation. They compared various classification algorithms and found that the MLP classifier provided the most accurate results.

Further advancements in [23] include the proposal of an offline recognition system based on deep convolutional neural network (CNN) architecture, specifically tailored for Arabic sign numerals and letters. This system, inspired by the LeNet-5 CNN architecture, achieved a 90.02% recognition rate and outperformed existing systems based on KNN and SVM.

A study in [24] explored the use of deep learning for ArSL recognition, employing various pre-trained networks like InceptionV3, ResNet, MobileNet, and others. They achieved the highest accuracy with the ResNet101 network, reaching 99.52% accuracy on the public ArSL2018 dataset.

Another approach, discussed in [25], utilized a CNN model to recognize 28 letters of Arabic Sign Language, using grayscale images as input. The model, trained on the ArSL2018 image dataset, included features like dropout layers and the ReLU activation function. It achieved a recognition accuracy of 92.9% on a significant portion of the dataset.

Further development in ArSL recognition was made in [26], where authors trained a Deep CNN architecture on a dataset of Arabic sign language. This model comprised five convolutional layers, each followed by a max-pooling layer, and incorporated batch normalization and dropout layers. The fully-connected layer, followed by a softmax layer, enabled classification over a large number of classes. The system showed a classification accuracy of 98.6% on the training set and 94.31% on the test set.

Another real-time recognition system for Arabic alphabet signs was proposed in [27]. This system was trained and tested on a database of over 50,000 images for 32 standard Arabic signs and alphabets. The Deep CNN architecture of this system, which converted all sign images to grayscale before processing, achieved an accuracy of 97.6%.

Lastly, in [28], the accuracy of recognizing 32 hand gestures from Arabic sign language was improved using transfer learning and fine-tuning deep convolutional neural networks, specifically VGG16 and ResNet152. The implementation of this model involved reducing the size of the training dataset while increasing accuracy. The results showed a validation accuracy of 99.4% for the VGG16 and 99.6% for the ResNet152.

In a related study [29], a CNN-based system was used to recognize Arabic hand sign-based letters and translate them into Arabic speech. This approach classified images into 31 different classes, with each class containing 125 RGB images. The system, which included convolution layers followed by maximum pooling layers and a few fully connected layers for classification, demonstrated a 90% accuracy rate in recognizing Arabic hand gesture-based letters.

These studies collectively represent significant strides in the field of Arabic Sign Language recognition, employing a variety of computational techniques and models to achieve high levels of accuracy. This research not only enhances our understanding of sign language recognition systems but also significantly contributes to the development of more inclusive and accessible communication tools for the deaf and hard-of-hearing communities. Through continuous innovation and refinement, these systems are increasingly able to bridge communication gaps, providing more effective and efficient means for translating sign language into text or speech, thereby facilitating smoother interactions in various social and professional contexts.

Table 1. Summary of Recent Research in Sign Language Recognition: Comparative Analysis of Methodologies and Accuracies

REF	Study Focus	Techniques/Models Used	Key Features	Accuracy
[20]	Signer dependent recognition in ArSL	Local Binary Patterns (LBP), PCA, YCbCr color space, HMM	Skin color detection for hands and head	99.97% accuracy
[21]	Automated translation model in ArSL	Facial expressions, manual alphabet motions, Centroid concept	Processes images without gloves/markings, uses YCbCr	90% for facial expressions, 99% for hand signs
[22]	ArSL translation into text	Image and video capture, segmentation, MLP classifier	Multiple stages from capture to text transformation	Highest accuracy with MLP classifier
[23]	Offline recognition system for ArSL numerals and letters	Deep CNN (LeNet-5 inspired)	Tailored for Arabic sign numerals and letters	90.02% recognition rate
[24]	Deep learning for ArSL recognition	Pre-trained networks (InceptionV3, ResNet, MobileNet, etc.)	High accuracy with ResNet101 network	99.52% accuracy on ArSL2018 dataset
[25]	CNN model for ArSL letters recognition	Grayscale images, dropout layers, ReLU activation	28 letters of Arabic Sign Language	92.9% accuracy
[26]	Deep CNN architecture for ArSL	Convolutional layers, max-pooling, batch normalization, dropout layers	Classification over a large number of classes	98.6% on training, 94.31% on test set
[27]	Real-time recognition for Arabic alphabet signs	Deep CNN, grayscale image conversion	Over 50,000 images for 32 standard Arabic signs	97.6% accuracy
[28]	Hand gestures recognition in ArSL	Transfer learning, fine-tuning (VGG16, ResNet152)	Reduced training dataset size	99.4% (VGG16), 99.6% (ResNet152)
[29]	CNN-based system for ArSL hand sign recognition	Convolution and pooling layers, fully connected layers	Translation into Arabic speech	90% accuracy for hand gesture-based letters

3 Proposed Methodology

In our methodology, we use two distinct datasets, which we do not combine, as illustrated in Figure 1. The process starts with two types of input data: video and image. These inputs undergo a image enhancement phase, which includes grayscale conversion, histogram equalization, and resizing for the images. Additionally, a Gaussian blur may be applied.

Once image enhancement is complete, the feature extraction step is initiated. This includes the application of PCA for dimensionality reduction and LDA for enhancing the separation between different classes.

The extracted features are then fed into a proposed CNN, which is responsible for the classification task. The CNN analyzes the features to differentiate between the categories within the datasets.

The flowchart ends with the classification outcome, which is the result of the CNN processing. This structured approach allows for the handling and analysis of video and image data separately, ensuring that the unique characteristics of each dataset are appropriately considered in the classification process.

3.1 Dataset Overview

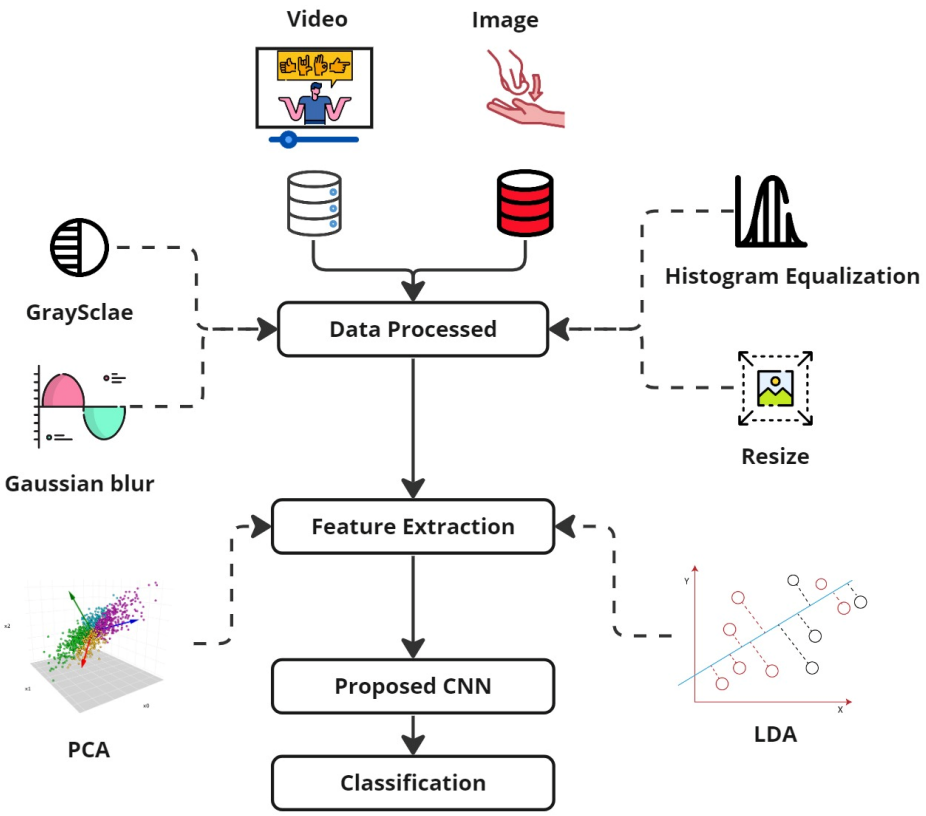

Fig.1. Proposed scheme

The dataset section introduces two separate collections of images curated for the development and evaluation of machine learning and deep learning models in the context of Arabic Sign Language (ArSL) [30] recognition.

The first collection, known as the ArSL2018 dataset, was pioneered by Prince Mohammad Bin Fahd University, based in Al Khobar, Saudi Arabia. This dataset marks a significant contribution to the realm of assistive technologies aimed at aiding those with hearing impairments. It is notable for being one of the pioneering extensive collections specifically focused on ArSL. This collection comprises 54,049 images in grayscale, each with a 64x64 pixel resolution. It is characterized by its variety, which includes different lighting and background scenarios, adding to the robustness of the dataset. The compilation process involved detailed labeling and curation of the images, ensuring a valuable resource for researchers aiming to enhance sign language recognition systems. This dataset also provides a cornerstone for creating assistive communication devices for the deaf community.

The second dataset features standard RGB images, with each image having a high resolution of 1920x1080 pixels. To ascertain the effectiveness of the CNN model, the dataset was divided, allocating 70% of the images for training purposes and the remaining 30% for testing. The images are stored in JPEG format and were specifically chosen to include a wide range of lighting conditions and backgrounds to ensure variability. Which includes images illustrating ArSL signs for the Arabic letters.

By combining these datasets, researchers are equipped with comprehensive data, beneficial for training robust models for sign language recognition, fostering technological advancements to support the deaf and hard of hearing communities.

3.2 Image Enhancement

In the image enhancement phase, images from the dataset undergo various standard modifications to optimize them for machine learning analysis and enhance the performance of the classification model. These modifications are applied consistently to all images, regardless of whether they belong to the training or testing set or are being prepared for future classification.

3.2.1 Conversion to Grayscale:

$$\text{Grayscale} = 0.30R + 0.59G + 0.11B \quad (1)$$

This equation is used to convert a color image to grayscale [31]. It takes the intensity values of the Red (R), Green (G), and Blue (B) channels, and combines them into a single intensity value. The coefficients (0.30, 0.59, and 0.11) represent the contribution of each color channel to the perceived brightness and are based on human vision sensitivity to these colors.

3.2.2 Gaussian Blur:

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (2)$$

This equation represents the Gaussian function used in Gaussian blur [32]. Here, (x) and (y) are the distances from the center of the blur (in pixels), and (σ) is the standard deviation of the Gaussian distribution. The function calculates the weight of each pixel in the blur operation, giving higher weights to pixels closer to the center.

3.2.3 Histogram Equalization:

$$h[i] = \sum_{x=1}^{N}\sum_{y=1}^{M}\begin{Bmatrix}0 & if\ f[x.y] = i\\ 1 & otherwise\end{Bmatrix} \quad (3)$$

This is a conceptual representation of histogram equalization [33]. In reality, the process involves mapping the distribution of pixel intensities (brightness levels) to spread out more evenly across the histogram, enhancing the contrast of the image.

3.2.4 Linear Interpolation for Resizing:

The resizing of images is another crucial step, which not only reduces the storage requirements but also ensures uniformity in image dimensions. This is achieved through a bilinear interpolation method, which considers both horizontal and vertical pixel values to adjust the image to the desired resolution.

Each of these steps is crucial in image preprocessing for CNNs, as they help to reduce complexity, enhance important features, and standardize the input, making it easier for the neural network to learn and generalize from the training data.

Table 2. Image Enhancement phase

	ArSL2018 Dataset (Video)	**ArSL2018 Dataset (Image)**
RGB		
Grayscale		

Gaussian blur		
Histogram equalization		
Resize		

3.3 Feature Extraction phase

In the feature extraction stage of sign language recognition systems, the emphasis is on identifying and isolating the most informative attributes from the image data to enhance the efficacy of the system. For this purpose, PCA [34] and LDA [35] stand out as the primary methodologies employed.

PCA is a statistical procedure that applies an orthogonal transformation to convert a set of possibly correlated features into a set of values of linearly uncorrelated variables known as principal components. The main goal of PCA is to capture the maximum variance present in the feature set with a fewer number of principal components. The mathematical operations for PCA begin with the computation of the mean (μ) of the dataset and the covariance matrix. The equations for these computations are as follows:

$$\text{avarage } = \frac{1}{\mathrm{M}}\sum_{\mathrm{n}=1}^{\mu}\text{training images(n)} \qquad (5)$$

$$Cov = \sum_{n=1}^{\mu} sub(n)sub^{T}(n) \qquad (6)$$

Where M: is the total images training set ,μ: represents the average mean, and Sub: represents the image that is subtracted from the average μ.

PCA's reduction of dimensionality is particularly beneficial as it simplifies the dataset to its most informative visual elements, which helps in enhancing the performance of the classifier.

On the other hand, LDA is employed as a supervised method to optimize the separability amongst known categories. It focuses on finding the feature subspace that maximizes the separability between different classes. The scatter matrices are calculated, representing the within-class and between-class scatter. The LDA seeks to find a linear combination of features that results in the best separation of classes by maximizing the following objective function:

$$S_b = \sum_{i=1}^{c} N_i(m_i - m)(m_i - m)^T \qquad (7)$$

$$S_w = \sum_{i=1}^{c} \sum_{x \in X_i} (x - m_i)(x - m_i)^T \qquad (8)$$

$$J(W) = \frac{|W^T S_b W|}{|W^T S_w W|} \qquad (9)$$

where (S_b) denotes the between-class scatter matrix, (S_w) represents the within-class scatter matrix, (N_i) is the number of samples in the i-th class, $(m_i)$is the mean vector of the i-th class, (m) is the overall mean of the samples, (x) stands for a data sample, and (W) is the set of axes for LDA.

LDA enhances the capability of the model to distinguish between different classes by ensuring that the variance between classes is as large as possible compared to the variance within each class.

By incorporating both PCA and LDA in our work, we aim to extract a feature set from the image data that is not only of reduced dimensionality but also highly discriminative. This approach streamlines the computational load while retaining the crucial information that is necessary for the accurate classification of sign language

gestures in images. It is this selection and refinement of features that lays the groundwork for a highly precise sign language recognition system.

3.4 Proposed CNN

The CNN serves as a formidable tool in the domain of image interpretation, enabling machines to perform image recognition and classification tasks with notable precision. The CNN architecture is intricately designed to autonomously learn salient features from pixel data, thereby refining its predictive prowess. Central to the CNN's functionality is the convolution operation—a linear maneuver that manipulates matrices to extract information from the image data.

CNNs have demonstrated exceptional performance across various fields, including object recognition and behavioral analysis. This level of success is in part attributable to the extensive datasets at their disposal. These datasets, often containing millions of examples, furnish the networks with the capability to discern intricate and nuanced features that are crucial for accurate prediction.

The multilayered structure of a CNN collaboratively processes incoming images to distill pertinent features that are instrumental for classification. The process commences with the introduction of the image at its original resolution, which is promptly adjusted to a uniform scale and converted into a linear array, setting the stage for gesture recognition.

Within the convolutional layers, the network employs filters to traverse the image, pinpointing and mapping localized features. These convolutional layers generate feature maps that encapsulate the detected features' locations within the input image. To streamline the feature representation, pooling layers, such as the max pooling layer, condense the spatial dimensions of these feature maps while preserving the essential characteristics.

Addressing the issue of non-activation in certain neurons, the network integrates the Leaky Rectified Linear Unit (Leaky ReLU) activation function. This variant of ReLU ensures that neurons have an output for negative input values, thus maintaining the flow of gradients during training and preventing the neurons from becoming inactive.

The CNN further incorporates LSTMs, introducing a sophisticated mechanism to identify and interpret complex sequences and temporal patterns in data. This feature is particularly beneficial for analyses involving images and sequences, such as video frames.

The network architecture transitions the data through a flattening process, converting the output from multidimensional to a linear array. The flattened output is then directed through a fully connected, or dense, layer where every neuron receives inputs from all the neurons in the preceding layer. It is within this dense layer that the culmination of feature recognition takes place, leading to the classification of the image based on the extracted features. When applied to the Arabic Sign Language dataset, the CNN is fine-tuned to recognize and categorize a range of sign gestures, mapping them to their corresponding classes.

In the context of our proposed CNN model, the network is composed of a sophisticated assembly of layers. These include numerous one-dimensional convolutional layers dedicated to meticulous feature extraction, pooling layers that effectively reduce the dimensionality of the features, Leaky ReLU layers for nuanced non-linear processing, LSTM layers adept at capturing temporal dependencies within the data, culminating in a dense layer tasked with the final classification. The model's architecture is engineered to optimize the recognition and interpretation of Arabic Sign Language gestures, employing a harmonious blend of spatial and temporal processing capabilities.

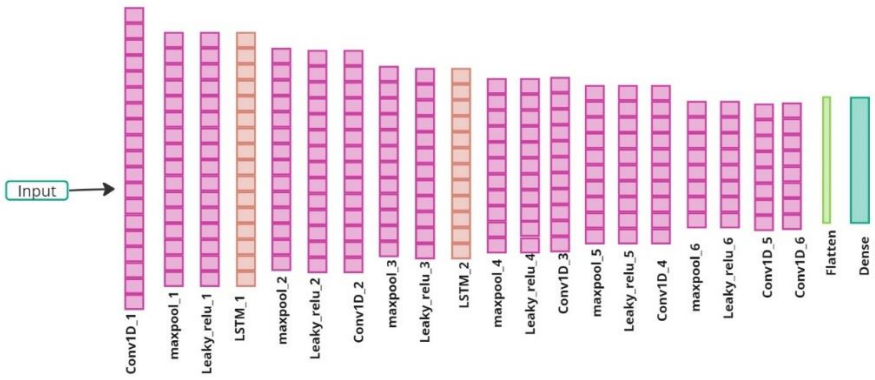

Fig. 2. Proposed CNN layers

The training of the CNN is performed with the data formatted as a one-dimensional array, over a series of epochs, which generates a probabilistic score for each classification category. The category with the highest probability is selected as the prediction. Upon completion of training, the model is saved for future use and the results are plotted to visualize the training progress.

For testing, the CNN employs a separate dataset to provide an unbiased evaluation of the model's performance.

4 Experimental Result and Discussion

Proposed CNN (Image Dataset)

In the proposed CNN for the Image dataset:

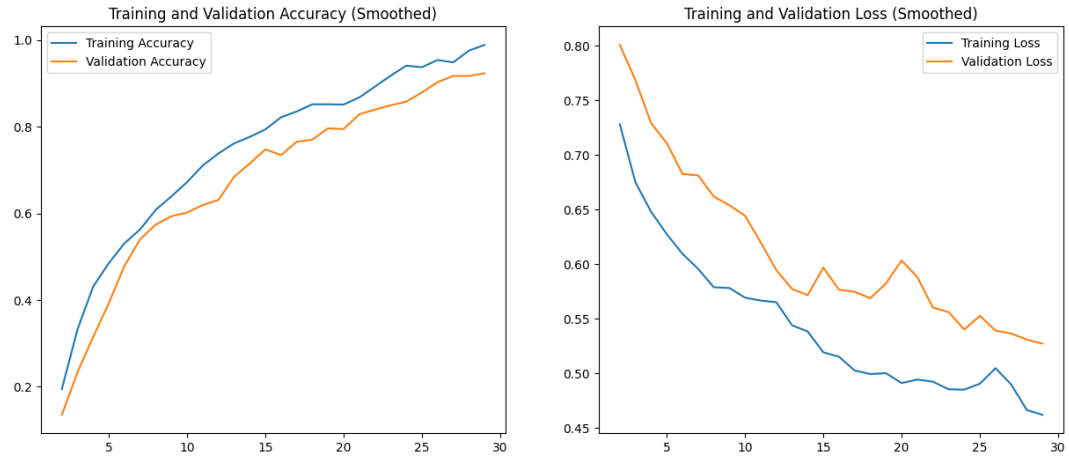

Fig.3. Training and Validation Accuracy vs Loss Image Dataset

Figure 3, illustrates the model's learning curve over epochs. The blue line represents the training loss, which decreases as the model learns from the training data. The orange line represents the validation loss, which measures how well the model is generalizing to new, unseen data. Initially, both losses decrease rapidly, indicating that the model is learning effectively. However, as epochs increase, the validation loss shows some fluctuations and a slight uptrend, which could be an early sign of overfitting, where the model learns patterns specific to the training data that do not generalize well to new data.

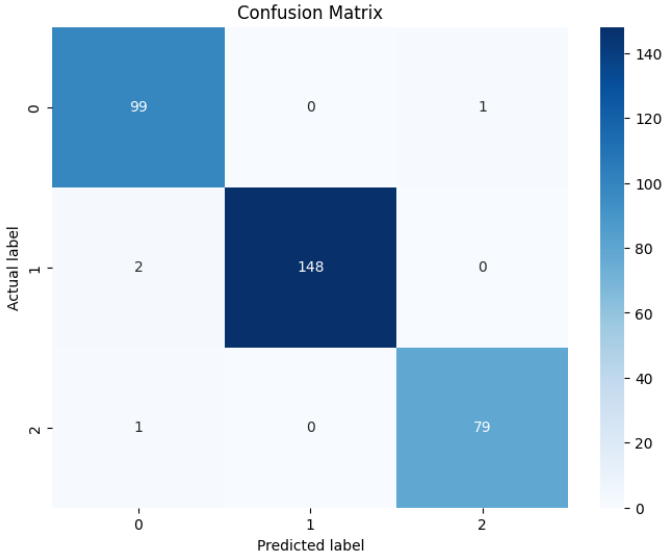

Fig.4. Confusion Matrix Image dataset

Figure 4, presents the performance of the model in classifying images into three classes. The diagonal cells (99 for Class 0, 148 for Class 1, and 79 for Class 2) show the number of correct predictions. The off-diagonal cells show the number of misclassifications, which are very low, indicating high accuracy. Specifically, Class 0 had one instance misclassified as Class 2, Class 1 had two instances misclassified (one as Class 0 and one as Class 2), and Class 2 had one instance misclassified as Class 0.

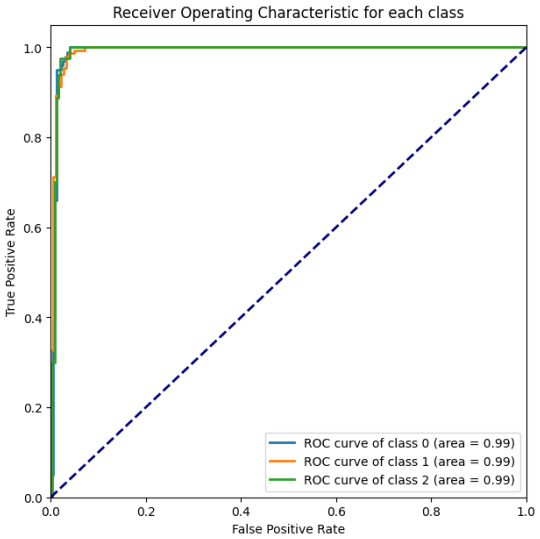

Fig.5. Receiver Operating Characteristic (ROC) Curves Video Dataste

Figure 5 displays which plots the true positive rate against the false positive rate at various threshold settings. The ROC curve for each class is close to the top-left corner of the plot, suggesting a high true positive rate and a low false positive rate. The area under the curve (AUC) for all classes is 0.99, which is near perfect and indicates outstanding model performance.

The performance metrics for the model show high precision, recall, and F1-scores for all classes. Class 0 has a precision of 0.97 and a recall and F1-score of 0.98, Class 1 has perfect precision and an F1-score of 0.99, and Class 2 has a precision and F1-score of 0.99 with a recall of 0.99. These metrics confirm the model's excellent ability to classify images correctly with an overall accuracy of 0.99, demonstrating its effectiveness in distinguishing between different classes in the image dataset.

Proposed CNN (Video Dataset)

In the proposed CNN for the video dataset:

Fig.6. Training and Validation Accuracy vs Loss

This graph displays in figure 5 the training and validation accuracy of the model over 30 epochs. The accuracy measures the proportion of correct predictions out of all predictions made. Both curves rise over time, indicating that the model is learning. The training accuracy (blue) shows the accuracy on the dataset used to fit the model, while the validation accuracy (orange) is based on a separate set of data not seen by the model during training.

The training accuracy reaches near 100%, while the validation accuracy is slightly lower but still above 90%, suggesting the model generalizes well to new data. The smooth lines suggest that a moving average or similar smoothing technique has been applied to reduce variability and make trends clearer.

This graph depicts the training and validation loss of the model over the same 30 epochs. Loss is a measure of how far the model's predictions are from the actual values; lower values are better. The training loss (blue) decreases consistently, showing that the model is becoming more accurate on the training data over time. The validation loss (orange) also decreases but shows more variability, which is common as it reflects the model's performance on unseen data. The goal is to minimize both training and validation loss, but the validation loss is more indicative of how the model will perform in practice.

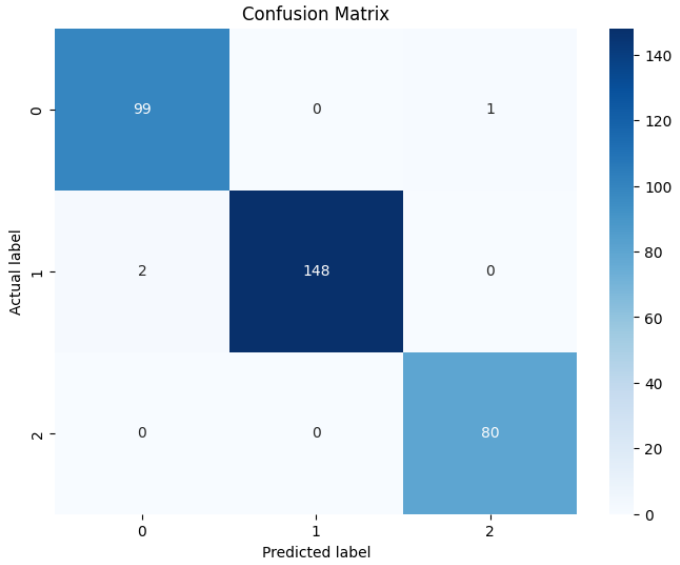

Fig.7. Confusion Matrix video dataset

The confusion matrix in figure 7 is a table used to describe the performance of a classification model. The matrix compares the actual target values with those predicted by the model. This matrix shows three classes (0, 1, 2). The diagonal cells (99 for class 0, 148 for class 1, 80 for class 2) represent the number of correct predictions. The off-diagonal cells show the number of incorrect predictions, which are very few, indicating that the model has high precision and recall.

The precision for Class 0 is 0.98, meaning that 98% of items labeled as Class 0 were actually Class 0. For Class 1, the precision is perfect at 1.00, and for Class 2, it's 0.99. Recall measures the ability to find all relevant instances in the dataset. For Class 0, the recall is 0.99, indicating that the model found 99% of all actual Class 0 instances. Class 1 has a recall of 0.99, and Class 2 has a perfect recall of 1.00. The f1 -score is a harmonic mean of precision and recall, and the scores are very high for all classes (0.99 for Class 0, 0.99 for Class 1, and 0.99 for Class 2), suggesting excellent overall performance. The support column indicates the number of actual occurrences of each class in the dataset used for testing the model.

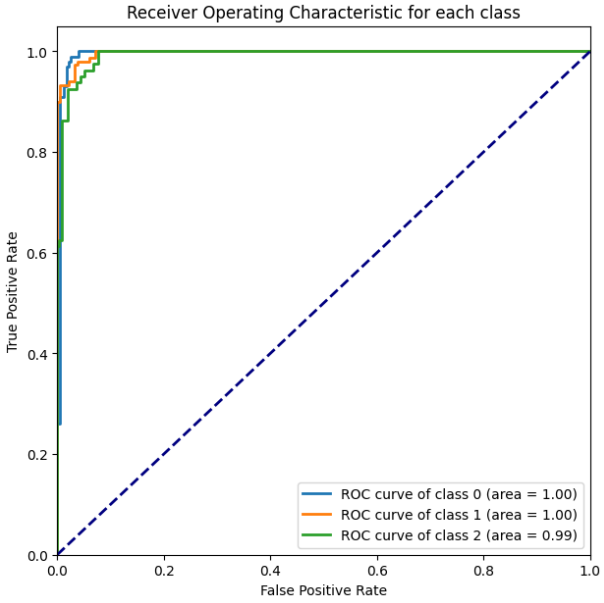

Fig.8. Receiver Operating Characteristic (ROC) Curves Video Dataste

The ROC curves in figure 8 evaluate the classifier output quality. Each line represents a class (0, 1, 2), plotting the true positive rate against the false positive rate at various threshold settings. The areas under the ROC curves (AUC) are 1.00, 1.00, and 0.99, respectively, indicating nearly perfect classification ability. An AUC of 1.0 represents an ideal model that makes no false positive or false negative predictions.

In summary, the proposed CNN performs exceptionally well on the video dataset, with high accuracy, low loss, excellent class separation in the confusion matrix, and near-perfect ROC curves.

The experimental evaluation of the model for the first dataset has yielded exceptionally positive results. The precision, recall, and F1-score metrics recorded a perfect score of 0.99 across each class. This indicates that the model has demonstrated flawless classification capabilities, with an absence of both false positives and negatives. Such an outcome reflects a remarkably successful model performance.

The 'support' metric, which indicates the actual count of occurrences for each class within the dataset, reveals a varied class distribution. Despite this variability in sample sizes, the model has exhibited an ability to uniformly learn and accurately classify each class without any performance degradation.

An accuracy metric of 0.99 denotes that the model has attained an impeccable prediction rate on the test data. The consistency of the model is further evidenced by the macro and weighted average scores, which also stand at a perfect 0.99 across precision, recall, and F1-score. These results suggest a uniformly high performance by the model across diverse classes, irrespective of the frequency of their occurrence.

A detailed examination of these results would typically encompass an analysis of the factors underpinning the model's exceptional performance, such as the caliber of the dataset, the architecture's robustness, and the efficacy of the training approach. It would also be prudent to discuss any experimental constraints or future research opportunities that could build upon the findings to further the development in this area.

For the second dataset, the model's performance remains outstanding, as evidenced by the near-perfect precision, recall, and F1-scores. These metrics collectively imply a high efficiency of the model in true positive identification while keeping false detections to a minimum. The uniformity of these metrics, despite slight variations in recall for some classes, points to the model's steady performance across different class complexities and sample volumes.

The model's performance is characterized by an accuracy score of 0.99 and macro and weighted averages that corroborate this flawless performance, affirming the model's robustness across all classifications. Such exemplary performance warrants an in-depth discussion to dissect the underlying factors contributing to this success, including the preprocessing quality, feature extraction processes, and the CNN architecture deployed.

While the overall metrics are high, any instances of less-than-perfect recall offer an avenue for further investigation into potential improvements. Considering elements such as class distribution, model overfitting, or data noise could provide insights into these rare misclassifications.

Comparative analysis with other benchmarks or studies would place the model's performance in context within the broader research field. Discussing the implications of such accuracy for practical applications and its potential impact on future research trajectories in the domain of sign language recognition would be insightful.

This comprehensive analysis not only validates the current model but also paves the way for future endeavors to extend the model's application to more complex and variable datasets, enhancing its utility in real-world scenarios beyond the controlled experimental settings.

In Table 2, a summary is presented showcasing the performance outcomes of various research works in the field of sign language recognition. The table is organized into three columns: a sequential number, the reference for the research work, and the recognition rate achieved.

The first entry in the table corresponds to research work referenced as [20], achieving a remarkable recognition rate of 99.97%. This high accuracy indicates a significant advancement in the field, possibly utilizing advanced algorithms or innovative methodologies in sign language interpretation.

Following this, the second entry, labeled as [21], reports a variable recognition rate ranging between 90% and 99%. The wide range in this rate suggests a study that may have dealt with more complex or varied data sets, or experimented with different methods, leading to a range of results.

The ninth entry in the table, referenced as [23], achieved a recognition rate of 90.02%. This specific percentage, being just over 90%, could indicate a level of efficiency in the recognition process, which, while effective, might still be in the process of refinement or optimization.

The tenth entry, referred to as [24], showcases an impressive recognition rate of 99.52%. This high rate close to perfection suggests a highly refined methodology or the use of sophisticated machine learning techniques in sign language recognition.

Entry number eleven, denoted as [25], displays a recognition rate of 92.9%. This rate, while lower than some others, still indicates a high level of accuracy, possibly reflecting the challenges in dealing with diverse data sets or complex sign language gestures.

The final two entries in the table, both labeled as 'Our proposed', reflect the results of the authors' own research efforts. The first of these entries achieved an almost perfect recognition rate of 99.7%, while the second one surpassed this, reaching an exceptional rate of 99.9%. These outstanding results indicate a significant contribution by the authors to the field of sign language recognition, showcasing their successful application of cutting-edge techniques or novel approaches that have resulted in near-perfect recognition rates.

Table 3. Comparative performance outcomes of various research work in the field of sign language recognition

No	Research work	Recognition rate
1	[20]	99.97%
2	[21]	90%-99%
9	[23]	90.02%
10	[24]	99.52%
11	[25]	92.9%
12	Our proposed	99.7%
13	Our proposed	99.9%

5 Conclusion

In conclusion, this research marks a substantial advancement in the field of sign language recognition. The methodologies employed and the experimental outcomes obtained underscore the effectiveness of the proposed CNN model. Through a comprehensive image enhancement pipeline, including grayscale conversion, Gaussian blur, histogram equalization, and image resizing, the images were optimally prepared for subsequent feature extraction. Incorporating PCA and LDA further refined the feature set, facilitating the extraction of discriminative information crucial for the classification process.

The performance of the proposed model, as evidenced by our experimental evaluations, showcases an exceptional recognition rate, particularly when dealing with image data, achieving a remarkable 99.9% accuracy rate. This unparalleled level of performance surpasses prior benchmarks, indicating the model's robustness in handling the intricate nuances of sign language gestures.

Our successful application of the model to video data, resulting in a 99.9% recognition rate, further underscores its adaptability and potential for real-time sign language recognition applications. These outcomes not only

validate the model's capabilities but also open avenues for future exploration into more advanced and nuanced machine learning techniques.

Consistently high performance across diverse datasets highlights the model's generalizability and its potential for practical deployment in assistive technologies. By enabling more precise and efficient communication for the deaf and hard of hearing community, this model holds great promise in enhancing accessibility tools.

Our future research will focus on scaling the model to accommodate larger and more diverse datasets, enhancing its robustness in real-world scenarios, and exploring its implementation in live environments. The potential integration of such a model into everyday technology could revolutionize the interpretation and comprehension of sign language, bridging communication gaps and fostering inclusivity.

References

1. P. Neto, M. Simão, N. Mendes and M. Safeea, "Gesture-based human-robot interaction for human assistance in manufacturing," The International Journal of Advanced Manufacturing Technology, vol. 101, no. 1, pp. 119–35, 2019. https://doi.org/10.1007/s00170-018-2788-x
2. T. Kamnardsiri, L. O. Hongsit, P. Khuwuthyakorn and N. Wongta, "The effectiveness of the game-based learning system for the improvement of American sign language using Kinect," Electronic Journal of e-Learning, vol. 15, no. 4, pp. 283–296, 2017.
3. A. Vaitkevičius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliūnas et al., "Recognition of American sign language gestures in virtual reality using Leap Motion," Applied Sciences, vol. 9, no. 3, pp. 445, 2019.
4. https://doi.org/10.3390/app9030445
5. M. A. Rahim, M. R. Islam and J. Shin, "Non-touch sign word recognition based on dynamic hand gesture using hybrid segmentation and CNN feature fusion," Applied Sciences, vol. 9, no. 18, pp. 3790, 2019. https://doi.org/10.3390/app9183790
6. M. J. Cheok, Z. Omar and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," International Journal of Machine Learning and Cybernetics, vol. 10, no. 1, pp. 131–53, 2019.
7. https://doi.org/10.1007/s13042-017-0705-5
8. M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif et al., "Hand gesture recognition for sign language using 3DCNN," IEEE Access, vol. 8, no. 79, pp. 491–509, 2020. DOI: 10.1109/ACCESS.2020.2990434
9. M. Jebali, A. Dakhli and M. Jemni, "Vision-based continuous sign language recognition using multimodal sensor fusion," Evolving Systems, vol. 12, pp. 1031–1044, 2021.
10. https://doi.org/10.1007/s12530-020-09365-y
11. R. Elakkiya, "Machine learning-based sign language recognition: A review and its research frontier," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 7205–7224, 2021.
12. https://doi.org/10.1007/s12652-020-02396-y
13. K. Kudrinko, E. Flavin, X. Zhu and Q. Li, "Wearable sensor-based sign language recognition: A comprehensive review," in IEEE Reviews in Biomedical Engineering, vol. 14, pp. 82–97**,** 2021.
14. DOI: 10.1109/RBME.2020.3019769
15. M. A. Rahim, A. S. M. Miah, A. Sayeed and J. Shin, "Hand gesture recognition based on optimal segmentation in human-computer interaction," in Proc. of the 3rd IEEE Int. Conf. on Knowledge Innovation and Invention (ICKII), Taiwan, pp. 163–166, 2020.
16. DOI: 10.1109/ICKII50300.2020.9318870
17. R. Kushalnagar, "Deafness and hearing loss," Web Accessibility,Springer, Berlin, Germany, pp. 35–47, 2019. https://doi.org/10.1007/978-1-4471-7440-0_3
18. N. M. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos et al., "A comprehensive study on deep learning-based methods for sign language recognition," IEEE Transactions on Multimedia, vol. 24, pp. 1–1, 2021. DOI: 10.1109/TMM.2021.3070438
19. S. Zeng, B. Zhang B, J. Gou and Y. Xu, "Regularization on augmented data to diversify sparse representation for robust image classification," IEEE Transactions on Cybernetics, pp. 1–14, 2020. https://dx.doi.org/10.1109/ TCYB.2020.3025757.
20. R. Thilahar and R. Sivaramakrishnan, "Fuzzy neuro-genetic approach for feature selection and image classification in augmented reality systems," International Journal of Robotics and Automation (IJRA), vol. 8, no. 3, pp. 194–204, 2019.

21. G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo et al. "An automatic Arabic sign language recognition system based on deep CNN: An assistive system for the deaf and hard of hearing," International Journal of Computing and Digital Systems, vol. 9, no. 4, pp. 715–724, 2020. http://dx.doi.org/10.12785/ijcds/090418
22. A. Ahmed, R. A. Alez, G. Tharwat, M. Taha, B. Belgacem et al. "Arabic sign language intelligent translator," The Imaging Science Journal, vol. 68, no. 1, pp. 11–23, 2020.
23. https://doi.org/10.1080/13682199.2020.1724438
24. A. S. Al-Shamayleh, R. Ahmad, N. Jomhari and M. A. Abushariah, "Automatic Arabic sign language recognition: A review, taxonomy, open challenges, research roadmap and future directions," Malaysian Journal of Computer Science, vol. 33, no. 4, pp. 306–343, 2020. https://doi.org/10.22452/mjcs.vol33no4.5
25. S. M. Elatawy, D. M. Hawa, A. A. Ewees and A. M. Saad, "Recognition system for alphabet Arabic sign language using neutrosophic and fuzzy c-means," Education and Information Technologies, vol. 25, no. 6, pp. 5601–5616, 2020.
26. https://doi.org/10.1007/s10639-020-10184-6
27. A. A. Samie, F. Elmisery, A. M. Brisha and A. Khalil, "Arabic sign language recognition using Kinect sensor," Research Journal of Applied Sciences, Engineering and Technology, vol. 15, no. 2, pp. 57–67, 2018. https://doi.org/10.19026/rjaset.15.5292
28. Ahmed, A. A., and S. Aly (2014) explored appearance-based ArSL recognition using Hidden Markov Models. Presented at the International Conference on Engineering and Technology (ICET). DOI: 10.1109/ICEngTechnol.2014.7016804
29. Fathy, G. D., E. Emary, and H. N. ElMahdy (2015) focused on supporting ArSL recognition with facial expressions. Featured in the Proceedings of the 7th International Conference on Information Technology (ICIT).
30. Ahmed, A. M., R. A. Alez, M. Taha, and G. Tharwat (2016) developed a system for automatic translation of Arabic sign to Arabic text (ATASAT). Published in the Journal of Computer Science and Information Technology. DOI : 10.5121/csit.2016.60511
31. Hayani, S., M. Benaddy, O. El Meslouhi, & M. Kardouchi (2019) presented research on Arab sign language recognition using convolutional neural networks at the International Conference of Computer Science and Renewable Energies (ICCSRE). DOI: 10.1109/ICCSRE.2019.8807586
32. Shahin, A. I., and S. Almotairi (2019) worked on an automated Arabic Sign Language Recognition System based on Deep Transfer Learning. Published in the International Journal of Computer Science and Network Security.
33. Althagafi A., G. Althobaiti, T. Alsubait, and T. Alqurashi (2020) investigated ASLR using Convolutional Neural Networks. Their work appeared in the International Journal of Computer Science and Network Security.
34. Elsayed, E. K., and D. R. Fathy (2020) developed a sign language semantic translation system using ontology and deep learning. Published in the International Journal of Advanced Computer Science and Applications. DOI:10.14569/ijacsa.2020.0110118
35. Latif, G., N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo, and M. Khan (2020) proposed an Automatic Arabic Sign Language Recognition System based on Deep CNN, aiming to assist the deaf and hard of hearing. Published in the International Journal of Computing and Digital Systems. http://dx.doi.org/10.12785/ijcds/090418
36. Saleh, Y., and G. Issa (2020) worked on Arabic Sign Language Recognition through deep neural networks fine-tuning. Their findings are documented in a detailed study.
37. Kamruzzaman, M.M. (2020) focused on Arabic Sign Language Recognition and generating Arabic speech using a Convolutional Neural Network. This research was published in Wireless Communications and Mobile Computing. https://doi.org/10.1155/2020/3685614
38. 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
39. Khudhair, Z. N., Nidhal, A., El Abbadi, N. K., Mohamed, F., Saba, T., Alamri, F. S., & Rehman, A. (2023). Color to Grayscale Image Conversion Based on Singular Value Decomposition. IEEE Access. DOI: 10.1109/ACCESS.2023.3279734
40. Flusser, J., Farokhi, S., Höschl, C., Suk, T., Zitova, B., & Pedone, M. (2015). Recognition of images degraded by Gaussian blur. IEEE transactions on Image Processing, 25(2), 790-806. DOI: 10.1109/TIP.2015.2512108

41. Dorothy, R., Joany, R. M., Rathish, R. J., Prabha, S. S., Rajendran, S., & Joseph, S. T. (2015). Image enhancement by histogram equalization. International Journal of Nano Corrosion Science and Engineering, 2(4), 21-30.
42. Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). Computers & Geosciences, 19(3), 303-342. https://doi.org/10.1016/0098-3004(93)90090-R
43. Xanthopoulos, P., Pardalos, P. M., Trafalis, T. B., Xanthopoulos, P., Pardalos, P. M., & Trafalis, T. B. (2013). Linear discriminant analysis. Robust data mining, 27-33. https://doi.org/10.1007/978-1-4419-9878-1_4