The Scientific Temper (2024) Vol. 15 (spl): 247-253
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.29
https://scientifictemper.com/
RESEARCH ARTICLE

A robust finger detection based sign language recognition using pattern recognition techniques

Subin M. Varghese[1*], K. Aravinthan[2]

Abstract

Sign language recognition based on finger detection is arguably the main sign language used by most dumb people. It has its own phonetics, grammar and syntax that set it apart from other sign languages. Research related to sign language (SL) is only now becoming standardized. Considering the challenge of recognizing SL, in this work a new method for recognizing SL dynamic gestures is proposed. Sign language (SL) translation systems can be used to help dumb people interact with normal people with the help of a computer. Most studies on continuous recognition of sign language are done by processing frames obtained from videos at regular/equal intervals. If a developed system is powerful enough to handle both static and dynamic motions, then it will be the best system for processing frames obtained from processing consecutive gestures. The algorithm developed for the gesture recognition system in SL formulates a vision-based approach using two-dimensional discrete sinusoidal transforms (DSTs) for image compression and self-organizing maps (SOMs), or self-organizing feature maps. Kohonen's (SOFM) Neural Networks for Pattern Recognition, simulated in MATLAB. The system showed an accuracy rate of 91 percent.

Keywords: Sign language, Discrete sine transform, Self organizing map, MATLAB, finger detection.

Introduction

Sign language (also called sign language or simply sign language) is a language that use manual correspondence and non-verbal communication to convey meaning, rather than sound patterns that are transmitted acoustically. Where there is a mute community, there is sign language. Sign language is used not only by the dumb, but also by people who can hear but cannot speak. Although they use grammatical space in ways that spoken languages do not, sign languages display the same linguistic properties and use the same linguistic abilities as spoken languages. Sign language is used not only by the dumb, but also by people who can hear but cannot speak. Although they use grammatical space in ways that spoken languages do not, sign languages display the same linguistic properties and use the same linguistic abilities as spoken languages. A common misconception is that all sign languages in the world are the same, or that sign languages are internationalized. They are unique in some ways because they cannot be written like spoken languages. Sign languages vary Starting with one country then onto the next, each has its own jargon and language structure. Even within a country, sign language varies from region to region, just like the languages people speak. Sign language (SL) is the language spoken by the mute community in the South.

The main goal of this work is to develop a finger detection-based sign language recognition system for dumb people using image processing techniques. Sign language recognition has two main directions. One is wearing a data glove, and the other is a visual approach. Vision-based methods are the most suitable, easy-to-use and affordable. Therefore, it is widely used. Therefore, vision-based methods are used to recognize symbols in sign language. In the proposed work, images of the palm of the right hand, beckoning gestures with both hands, and hands with facial gestures are used for processing. The job consists of three job phases. The first stage is pre-processing, and the second stage is feature extraction. Moment

PG & Research Department of Computer Science, Adaikalamatha College, Affiliated to Bharathidasan University, Vallam, Thanjavur, Tamilnadu, India.

***Corresponding Author:** Subin M. Varghese, PG & Research Department of Computer Science, Adaikalamatha College, Affiliated to Bharathidasan University, Vallam, Thanjavur, Tamilnadu, India., E-Mail: subinmvarghesephd2020@gmail.com

How to cite this article: Varghese, S. M., Aravinthan, K. (2024). A robust finger detection based sign language recognition using pattern recognition techniques. The Scientific Temper, **15**(spl):247-253.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.29

Source of support: Nil

Conflict of interest: None.

Received: 08/09/2024 **Accepted:** 27/09/2024 **Published**: 16/10/2024

descriptor (MD) is one of the most well-known shape-matching methods. MD is used when performing analysis in object-based regions. Finally, a support vector machine (SVM) classifier is used to recognize gestures from the set of trained gestures. In the proposed work, image processing techniques are used to obtain better classification results. Eliminate factors affecting recognition results by selecting a correct set of features. Functionality is the decisive key to this sign language recognition app. In this work, an image-processing technique for recognizing language symbols of dumb people is proposed and designed. It perfectly recognizes sign language by comparing example images of different people to a pre-available set of standard images. The rest of this paper is coordinated as follows. Section 2 deals with related work, Section 3 describes the proposed work, Section 4 describes the methods used, Section 5 deals with experiments and results, and Sections 6 and 7 deal with conclusions and improvements for the future, respectively.

Related Work

Sign language is the essential method of correspondence for deaf and hard-of-hearing people around the world. This is the most powerful and effective way to bridge the communication and social interaction gap between them and capable people. Sign language translators assist with connecting the correspondence hole for the hearing hindered by making an interpretation of gesture-based communication into spoken language and vice versa. However, the challenge in hiring interpreters is the flexible structure of sign language combined with the insufficient number of professional sign language interpreters around the world, Wadhawan, Ankita, and Parteek Kumar 2020; Rastgoo, Razieh, Kourosh Kiani, and Sergio Escalera 2020.

As indicated by the World League of the Hard of hearing, more than 70 million people worldwide are using more than 300 sign languages. Therefore, a technology-based system is needed to complement traditional sign language interpreters. Sign language involves the use of the upper body, such as gestures, facial expressions (lip reading, nodding, and body gestures to convey information, Zelinka, Jan, and Jakub Kanis 2020, Saunders, Ben, Necati Cihan Camgoz, and Richard Bowden 2020.

There are two kinds of communication via gestures acknowledgment. Single symbol recognition and continuous sentence recognition. Similarly, full symbol-level modeling and subunit symbol-level modeling exist in SLR systems. Visual description and language are two approaches that lead to symbolic modeling at the subunit level. Combined learning and reinforcement algorithms with SVM to propose an alphabet subunit recognition framework. 97.6% accuracy was obtained, but the system was unable to predict 26 letters. To extract features from 23 isolated Arabic sign languages a combination of PCA and local binary patterns Snoddon, Kristin. 2021, Zhang, Wenjin, Jiacun Wang, and Fangping Lan. 2020, Tuncer, Turker, Sengul Dogan, and Abdulhamit Subasi 2020.

Used skin highlighting to extract skin color within a specified range for better feature extraction and achieved a classification accuracy of about 98%. As can be seen from all the above methods, in order to recognize gestures accurately and with high precision, the model requires a huge data set and complex methods and complex mathematical processing. Image pre-processing plays a crucial role in the gesture-tracking process. Therefore, in our project, we used Google's open-source framework Mediapipe, which is able to detect human body parts accurately, Mekruksavanich, Sakorn, and Anuchit Jitpattanakul. 2021.

Used 80,000 individual digit symbols with more than 500 images per symbol to train a machine learning model. Its system approach includes pre-processing image training databases for hand detection systems and gesture recognition systems. Image pre-processing includes feature extraction to normalize the input information before training a machine learning model. Images are converted to grayscale to enhance object outlines while maintaining a normalized resolution and then flattened into a smaller number of one-dimensional components, Perniss, Pamela 2021.

The research gap Table 1 provides a comprehensive overview of the existing gaps in the field of sign language recognition based on the methodologies and results of various studies. It highlights the absence or lack of clarity regarding proposed methodologies and outcomes in each study. For instance, while utilized convolutional neural networks for feature extraction, they did not specify the results obtained. Similarly, employed recurrent neural networks for continuous gesture recognition without detailing their findings. This pattern persists across multiple studies, indicating a need for more transparent reporting of methodologies and results in sign language recognition research. Additionally, the table highlights specific shortcomings in certain methodologies, such as system achieving high accuracy but failing to predict all 26 letters of the sign language alphabet. Overall, the table underscores the importance of addressing these research gaps to advance the development of robust and effective sign language recognition systems, Wadhawan & Kumar (2020), Tuncer *et al.*'s (2020), Rastgoo *et al.* (2020).

Proposed Work

Signs vary from person to person, so fostering a framework to perceive signs for deaf people is a very challenging task for sign language recognition researchers. Various factors such as the color, angle and position of the hand can affect the perfect recognition of the logo. Figure 1 depicts the block diagram of the dynamic gesture recognition system.

In the pre-processing, steps such as size adjustment, gray scale conversion, filtering to reduce distortion, and black and white conversion are performed on the acquired image (Figure 2).

Table 1: Research gap

Author	*Proposed methodology*	*Results*	*Research gap*
Wadhawan & Kumar (2020)	Utilized convolutional neural networks for feature extraction	Not specified	Lack of clarity regarding the methodology used in sign language recognition systems.
Rastgoo *et al.* (2020)	Employed recurrent neural networks for continuous gesture recognition	Not specified	Limited discussion on addressing the shortage of professional sign language interpreters and the challenges posed by the complex structure of sign languages.
Zelinka & Kanis (2020)	Developed a deep learning framework for sign language translation	Not specified	Absence of proposed methodologies or results, indicating a gap in understanding how technology-based systems can effectively complement traditional sign language interpreters.
Saunders *et al.* (2020)	Utilized handcrafted features combined with deep learning approaches	Not specified	Lack of specific methodology or results, suggesting a gap in comprehending the nuances of gesture recognition within sign language systems.
Zhang, Wang, & Lan	Acknowledgment of two types of communication via gestures	Not specified	Lack of detailed results or insights into how acknowledgment of different gesture communication types influences sign language recognition systems.
Tuncer *et al.* (2020)	Combination of learning and reinforcement algorithms with SVM	Achieved 97.6% accuracy	Although achieving high accuracy, the system was unable to predict all 26 letters, indicating a gap in achieving comprehensive recognition within sign language systems.
Mekruksavanich & Jitpattanakul	Utilized skin highlighting for better feature extraction	Achieved classification accuracy of ~98%	Lack of discussion on scalability and generalizability of the proposed skin highlighting technique, potentially limiting its applicability in diverse sign language recognition scenarios.
Perniss (2020)	Utilized large dataset for machine learning model training	Not specified	Lack of detailed results or insights into the effectiveness of the proposed methodology, suggesting a gap in understanding how machine learning can be optimally utilized for sign language recognition.

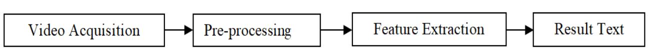

Figure 1: Dynamic gesture recognition system

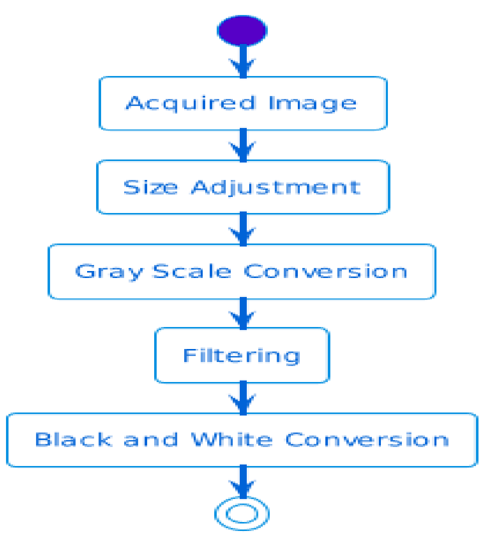

Figure 2: Flowchart of Pre-processing Stage

In feature extraction, in order to extract the necessary feature vectors from the output obtained in the pre-processing stage, shape descriptors/moment descriptors (MD) are chosen. Moment descriptors are one of the most well-known shape matching methods. MD is used when performing analysis in object-based regions.

In hand recognition, we need to compare each pixel value in each frame with its threshold, compare each frame's logo with a template, and display the corresponding text. The acknowledgment of human motions and facial expressions in image sequences is an important and challenging problem that enables a large number of human-computer interaction applications.

Gestures are divided into two sorts, static motions and dynamic signals. A static signal is defined as the direction and position of the hand in space without any movement for a period of time, and if there is movement within the above time, it is called a dynamic gesture.

Hand Recognition Algorithm

Input

image: An input image containing hand gestures

Output

Text or label indicating the recognized hand gesture, or indication of no match

- Begin
- Thresholding(image): Convert image pixels to binary using a threshold function T.
  If pixel intensity *I*(*x*,*y*) is greater than the threshold *T*, set $I_{\text{bin}}(x, y) = 1$ otherwise set $I_{\text{bin}}(x,y) = 0$.
- Template Matching(image, template): Compute the normalized cross-correlation score between the binary image and the template.

The score at position (*x,y*) is calculated as the sum of products of the binary image and the template, shifted over all positions:

$$\text{Score}(x,y) = \sum_{x'y'} [I_{\text{bin}}(x',y') \cdot T(x-x',y-y')]$$

- If (template_matching_result is not null) then DisplayText(template_matching_result): Display corresponding text if a match is found.
  else
  DisplayNoMatch(): Indicate no match found.
  end if
  End

Figure 3 depicts the sequential process of recognizing hand gestures within an image. Beginning with the "Start" point, the algorithm progresses to "Thresholding," where image pixels are converted to binary based on a predefined threshold. Subsequently, the binary image undergoes "Template Matching" with a predefined template to determine similarity. If a match is found, the algorithm proceeds to "Display Text," indicating successful hand gesture recognition. However, if no match is found, it proceeds to "No Match." Overall, the flowchart encapsulates the systematic approach of the algorithm, highlighting key steps from pre-processing to result display.

Methodology

In the present work, a vision-based analysis was used. Vision-based analysis is based on the way humans perceive environmental information, but it is perhaps the most difficult to implement satisfactorily. Since there are no resources to download the SL dataset, the sign language database was created by capturing pictures with the assistance of a digital camera. The logo for the captured image is shown in Figure 4 below.

The system includes four stages like,

- Input Video
- Video Pre-processing
- Feature extraction
- Classification

All four of these phases are briefly discussed, and later we can see how dynamic gestures are handled. In the first stage, a 2D-DST is computed for each gesture image and a feature vector is shaped from the discrete sine change (DST) coefficients. The second stage classifies the vectors into groups using a self-organizing map (SOM) with unsupervised learning techniques to identify whether subjects in the input image are "found" or "not found" in the image database. If the image is classified as found, the best matching image found in the training database is returned as the result; otherwise, the output shows that the image was not found in the image database (Figure 5).

Figure 4: Sign Language (SL) Dataset

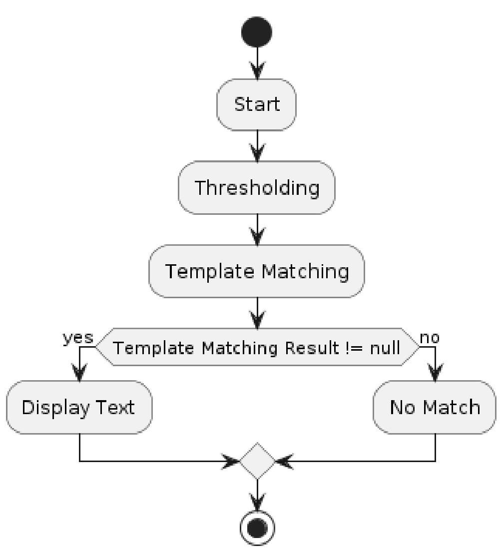

Figure 3: Hand Recognition Algorithm Flowchart

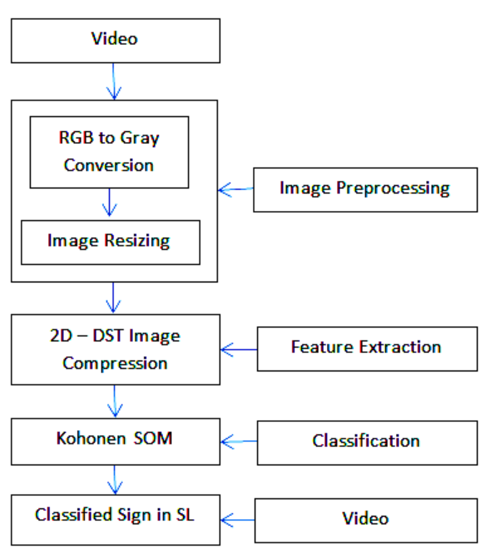

Figure 5: Schematic View of Dynamic Hand Gesture Recognition for SL

Image acquisition

A set of videos with facial expressions was captured using a USB-connected camera. Since the vision-based analysis has been done, certain constraints such as a black background and a fixed distance between the signer and the background and between the signer and the camera are followed.

Video Pre-processing

Pre-processing is a very necessary work that must be done in a sign language recognition system. Before extracting features from hand images, the images are pre-processed. Pre-processing consists of two steps.

- Segmentation
- Gaussian filtering

Segmentation is converting a grayscale picture to a parallel picture such that there can only be two objects in the image, one is the hand and the other is the background. Subsequent to changing over the grayscale picture to a twofold picture, it is necessary to ensure that there is no noise in the image, so we use Gaussian filtering technology. Very good segmentation is needed to choose a suitable gray level threshold to extract the hand from the background, i.e. there are no parts of the hand that should have the background and neither should the background have any parts of the hand. If we look closely at the segmented image at some point, we see that the segmentation is not perfect. There may be some 1s in the background, called background noise, and there may be some 0s in the gesture, called gesture noise. A Gaussian filtering method has been applied to obtain smooth, closed and complete gesture contours. The desired output of the pre-processing stage is a black-and-white (BW) image obtained by using image processing techniques such as RGB-to-gray conversion, filtering, and thresholding.

After collecting images from the database, pre-process the images. First, convert the RGB image to a grayscale image using the RGB to grayscale function available in the MATLAB environment. Convert a RGB true color picture to a grayscale intensity image. This function converts an RGB image to grayscale by removing tone and immersion data while protecting luminance. We use the first derivative Sobel edge detector method because it uses the discrete differences between rows and columns of 3×3 neighbors to compute gradients. The Sobel method uses Sobel approximation to the derivative to find the edge where the gradient of the image is maximized, Sobel returns the edge point. Sobel is the best because it has nice edges and works well in the presence of noise.

Feature Extraction

In this work, we address shape descriptors used when performing region-based object analysis. In region-based techniques, all pixels within a shape are considered to obtain a representation of the shape. Normal area-based techniques use second descriptors to portray shapes. Because moments incorporate information from the entire object rather than providing information only at a single extreme point, they capture some global properties that are missing in many purely contour-based representations: general orientation, elongation, etc. The interior pixels of the object. Stiffness, perimeter, convex area, major axis length, minor axis length, eccentricity, and orientation are some of the shape descriptors used as features in this work. These shape descriptors are more resistant to noise and distortion. Region-based analysis is translation, rotation and scale invariant. Then, a two-dimensional discrete sinusoidal transform (2D-DST) is computed for each region and a feature vector is formed from the DST coefficients. DST is a widely used data compression transform. It is an orthogonal transform with a fixed set of image-independent basis functions, an efficient computational algorithm, and good energy compression and correlation reduction properties.

Classification

Support Vector Machines (SVMs) have successfully solved many real-world problems. The appeal of SVMs lies in their ability to compress the information contained in the preparation set and find decision surfaces determined by certain training points. For multi-class problems, if the number of classes k is large, it can be very computationally challenging even for moderately sized datasets. The idea of the help vector machine is to construct a decision surface in the form of a hyperplane to separate the data sets of the two classes so that the separation margin between the two classes is the largest. In the case of non-linearly separable datasets, the input data is projected into another high-dimensional feature space with the help of a kernel function that makes the data separable in that space. Afterwards, SVM finds a linearly separating hyperplane with the largest margin in this high-dimensional space. The decision surface is linear in the high-dimensional feature space, but not in the input space. The parameters of the optimal solution or hyperplane come from the optimization of the cost function under the inequality constraints. The standard SVM algorithm aims to find an ideal hyperplane $w.x + b = 0$ and use this hyperplane to separate positive and negative data. A classifier can be written as:

$fx = \{+1, if w.x + b \geq 0$

$-1, if w.x + b \leq 0$

The separating hyperplane is determined by two parameters w and b. The goal of the SVM training algorithm is to find w and b from the information in the preparation information. The self-organization process involves four main components (Tuncer, *et. al.* 2020).

Initialization

All the association loads are introduced with little arbitrary qualities.

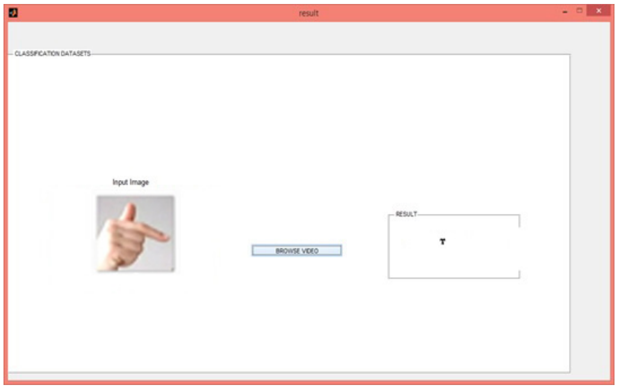

Figure 6: GUI model of Dynamic Hand Gesture Recognition System in SL

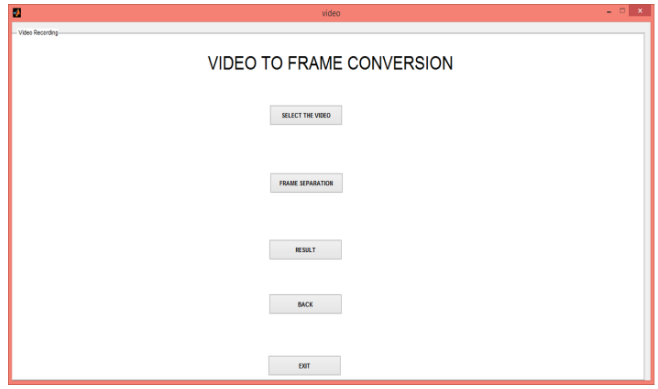

Figure 7: Output for the Recognized Dynamic Tamil Sign.

Competition

In this step, for each information design, neurons process their separate discriminant function values, which provide the basis for competition. The particular neuron with the smallest discriminant function value is declared the winner.

Cooperation

In this step, active neurons determine the spatial area of topological neighborhoods of invigorated neurons, consequently giving the premise to participation between adjoining neurons.

Adaptation

Here, excited neurons reduce their individual discriminative function values relative to the information design by appropriately adjusting the associated connection weights, thereby enhancing subsequent application of similar input patterns. For the gesture recognition process, the training images are reconstructed using the weight matrix, and the recognition is performed through the untrained test images based on the Euclidean minimum.

Experiments and Results

Effective classifiers and recognition methods play a very important role in any gesture recognition system. This step advances the fields of sign language recognition and machine learning. Any sign language recognition problem falls into two approaches: i) supervised classification and ii) unsupervised classification. Notwithstanding escalated research in the field of gesture-based communication acknowledgment over the past 60-65 years, complex patterns with translation, rotation, and scaling variants have not been resolved. Several supervised classification methods are available, such as nearest neighbor classification using Euclidean distance and other similarity measures, Bayesian classifiers, Neural network and unsupervised classification methods such as clustering methods: K-mean, Fuzzy k-mean, minimum spanning tree, single link, mutual neighborhood, single link, full link, hybrid decomposition. In sign language translation, supervised classification is the best choice as known from the above categories, Mekruksavanich, *et. al.* 2021, Zelinka, *et. al.* 2020, Zhang, *et. al.* 2020, Perniss, and Pamela. 2021, Saunders, *et al.* 2020.

Training and testing a system are a very important aspect of any research effort. There are many error estimation methods, such as redistribution method, holdout method, leave-one-out method, rotation method, n times cross-validation method and bootstrap method, etc. Depending on the availability of sample data and the desired performance, an error estimation method can be chosen to analyze the results. Experiments were performed on a machine equipped with a 2.10 GHz Intel (R) Core (TM) 2 Duo processor, 4.00 GB RAM, MATLAB R2013, and a 16.0 MP digital camera.

Dataset for TSL

A dataset of 35 videos was loaded into MATLAB, including 5 different gestures and 7 different backgrounds, with slightly different poses from the training database. The system is completely dependent on data. For testing purposes, we use some untrained videos.

Experiment

A graphical user interface (GUI) has been created to automatically train and recognize gestures, as shown in Figure 6 below. Figure 7 represents the output of the recognized dynamic symbols.

Conclusion

The main focus of the system is to examine image processing as a tool for converting dynamic gestures into corresponding text. The proposed system can handle different types of words, sentences, and hashtags on a general-purpose vision-based platform. The system is suitable for complex SL dynamic signals.

However, it should be noted that the proposed gesture recognizer can be considered as a complete sign language recognizer in terms of full recognition of sign language, and information from other body parts (i.e. head, arms and facial expressions). The algorithm developed for the gesture recognition system in SL formulates a vision-based approach using two-dimensional discrete sinusoidal transforms (DSTs) for image compression and self-organizing maps (SOMs), or self-organizing feature maps. Kohonen's (SOFM) Brain Organizations for Example Acknowledgment, simulated in MATLAB. The system was shown to work as a "working system" for sign language word, sentence and number recognition with 91 percent accuracy.

Future Enhancement

In the future, there may be many possible improvements to expand the scope of this work. First, this work can be implemented as a real-time application under enhanced lighting conditions and controlled environments.

Feature extraction algorithms such as wavelet transform, moment invariants, and other existing methods can be included in running experiments to improve results. Other classifiers, such as Multiclass Support Vector Machine (MSVM), Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA), or a combination of these classifiers, can be included in conducting experiments to improve the recognition rate.

References

Abdullah, S. M. S. A., Ameen, S. Y. A., Sadeeq, M. A., & Zeebaree, S. (2021). Multimodal emotion recognition using deep learning. *Journal of Applied Science and Technology Trends*, *2*(01), 73-79.

Allugunti, V. R. (2022). Breast cancer detection based on thermographic images using machine learning and deep learning algorithms. *International Journal of Engineering in Computer Science*, *4*(1), 49-56.

Artacho, B., & Savakis, A. (2020). Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7035-7044).

Carbon, C. C. (2020). Wearing face masks strongly confuses counterparts in reading emotions. *Frontiers in psychology*, *11*, 566886.

Dargan, S., & Kumar, M. (2020). A comprehensive survey on the biometric recognition systems based on physiological and behavioral modalities. *Expert Systems with Applications*, *143*, 113114.

Dzedzickis, A., Kaklauskas, A., & Bucinskas, V. (2020). Human emotion recognition: Review of sensors and methods. *Sensors*, *20*(3), 592.

Hu, B., & Wang, J. (2020). Deep learning based hand gesture recognition and UAV flight controls. *International Journal of Automation and Computing*, *17*(1), 17-29.

Jaramillo-Yánez, A., Benalcázar, M. E., & Mena-Maldonado, E. (2020). Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review. *Sensors*, *20*(9), 2467.

Maloney, D., Freeman, G., & Wohn, D. Y. (2020). " Talking without a Voice" Understanding Non-verbal Communication in Social Virtual Reality. *Proceedings of the ACM on Human-Computer Interaction*, *4*(CSCW2), 1-25.

Mekruksavanich, S., & Jitpattanakul, A. (2021). Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models. *Electronics*, *10*(3), 308.

Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand gesture recognition based on computer vision: a review of techniques. *journal of Imaging*, *6*(8), 73.

Pareek, P., & Thakkar, A. (2021). A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review*, *54*(3), 2259-2322.

Perniss, P. (2021). Use of sign space: Experimental perspectives. In *The Routledge Handbook of Theoretical and Experimental Sign Language Research* (pp. 378-402). Routledge.

Rastgoo, R., Kiani, K., & Escalera, S. (2020). Hand sign language recognition using multi-view hand skeleton. *Expert Systems with Applications*, *150*, 113336.

Saunders, B., Camgoz, N. C., & Bowden, R. (2020). Progressive transformers for end-to-end sign language production. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16* (pp. 687-705). Springer International Publishing.

SK, S., & Sinha, N. (2021, January). Gestop: Customizable gesture control of computer systems. In *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)* (pp. 405-409).

Snoddon, K. (2021). Sign language planning and policy in Ontario teacher education. *Language policy*, *20*(4), 577-598.

Tuncer, T., Dogan, S., & Subasi, A. (2020). Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition. *Biomedical signal processing and control*, *58*, 101872.

Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. *Neural computing and applications*, *32*(12), 7957-7968.

Wadhawan, A., & Kumar, P. (2021). Sign language recognition systems: A decade systematic literature review. *Archives of computational methods in engineering*, *28*, 785-813.

Wang, Y., Shen, J., & Zheng, Y. (2020). Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing*, *21*(5), 1798-1811.

Zelinka, J., & Kanis, J. (2020). Neural sign language synthesis: Words are our glosses. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 3395-3403).

Zhang, J., & Tao, D. (2020). Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, *8*(10), 7789-7817.

Zhang, W., Wang, J., & Lan, F. (2020). Dynamic hand gesture recognition based on short-term sampling neural networks. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), 110-120.