








## A Deep Learning-Based Approach for Hand Sign Recognition Using CNN Architecture

Deepak Parashar<sup>1\*</sup>, Sudhanshu Thakur<sup>1</sup>, Kachapuram Basava Raju<sup>2</sup>, Garine Bindu Madhavi<sup>3</sup>,  
Kanhaiya Sharma<sup>1</sup>

<sup>1</sup> Symbiosis Institute of Technology Pune, Symbiosis International (Deemed) University, Pune 412115, India

<sup>2</sup> Department of Artificial Intelligence, Anurag University, Hyderabad 500088, India

<sup>3</sup> Geethanjali College of Engineering and Technology, Cheeryal, Keesara, Medchal, Hyderabad 501301, India

Corresponding Author Email: [parashar.deepak08@gmail.com](mailto:parashar.deepak08@gmail.com)

<https://doi.org/10.18280/ria.370414>

### ABSTRACT

**Received:** 13 May 2023

**Revised:** 25 July 2023

**Accepted:** 31 July 2023

**Available online:** 31 August 2023

#### **Keywords:**

*hand sign recognition; recognition, deep learning, Convolutional Neural Network, Kaggle dataset, max pooling*

The domain of hand sign recognition, an integral facet of computer vision, encompasses a wide array of practical applications, ranging from interpreting sign language and recognizing gestures to facilitating human-computer interaction. This research elucidates the introduction of a Convolutional Neural Network (CNN) model tailored to the identification of hand signs representing the English alphabet. For model training and validation, a dataset comprising 26,000 grayscale images of hand signs was employed. The model architecture embraced a profound CNN design, featuring numerous layers for convolution and pooling, followed by fully connected layers. Employing the Adam optimizer, the training procedure yielded an impressive accuracy of 96.7% when evaluated on the Kaggle dataset. These outcomes underscore the effectiveness of the proposed CNN model in precisely discerning hand signs corresponding to the English alphabet. The model's potential utility extends to the recognition of intricate manual gestures and real-time applications, including aiding individuals with motor impairments and enriching virtual reality experiences. Hence, this study accentuates the capacity of deep learning to propel the domain of hand sign recognition forward.

## 1. INTRODUCTION

Hand sign recognition constitutes a crucial realm of investigation within the domains of computer vision and machine learning, boasting an extensive array of utilities including but not limited to sign language translation, interfaces driven by gestures, and fostering human-computer interaction. A specific facet of hand sign recognition pertains to discerning sign language alphabets. Within this framework, the objective revolves around identifying manual signs representative of distinct letters in the alphabet. Convolutional Neural Networks (CNNs) have become a powerful tool for image identification efforts in recent years, demonstrating remarkable achievements across a spectrum of hand sign recognition. Within the scope of this endeavor, we present a CNN-based architecture tailored to the recognition of alphabet-associated hand signs. We harbor the belief that our proposed model has implications across domains encompassing education, communication, and enhancing accessibility for individuals with disabilities.

In the literature, Shahriar et al. [1] propose the Real-time American Sign-Language (ASL) identified with CNN -based ASL recognition system that can recognize ASL signs in real-time. The system is designed to recognize the signs from a live video stream and translate them into corresponding English text. The proposed system is trained on ASL dataset, which consists of over 100,000 images of 2000 ASL signs. The CNN architecture used in the system includes many convolutional and pooling layers followed by fully linked layers. For classification. The system also uses a technique called data

augmentation to increase the diversity of the training dataset, which improves the accuracy of the model. The trained model is then integrated into a real-time video recognition system that can recognize ASL signs from a live video stream. The proposed ASL recognition system achieves an accuracy of over 95% on the test dataset, which is comparable to the ASL recognition systems. The real-time video recognition system is also evaluated on a live video stream, and it achieves a recognition speed of 30 frames per second, making it suitable for real-time applications. The proposed system has significant potential applications in the field of assistive technology, where it can help people with hearing disabilities to communicate with others using ASL. The system can also be used in educational settings to teach ASL to students, as well as in the entertainment industry to create interactive ASL-based games and applications.

Yashas and Shivakumar [2] stated that hand gesture recognition has been a focus of research for some time due to the wide range of industries it has applications in, including robotics, sign language interpretation, and human-computer interaction. With the advancements in machine learning techniques, particularly deep learning algorithms, hand gesture recognition has been made more accurate and efficient. This survey paper provides a comprehensive review of various machine learning approaches used in hand gesture recognition. The paper starts by discussing the various types of hand gestures and the datasets used for training and testing the models. It then reviews different feature extraction techniques such as shape-based, appearance-based, and hybrid methods,

which are used to extract relevant information from the hand images.

The paper then discusses various ANN, SVM, and decision trees are examples of machine learning methods, which are commonly used for hand gesture recognition. It also covers the recent advancements in deep learning algorithms particularly CNNs and RNNs, and their applications in hand gesture recognition. The paper also provides a comparative analysis of different machine learning approaches used for hand gesture recognition based on factors such as accuracy, speed, and complexity. It also discusses the challenges and future directions in hand gesture recognition, such as robustness to changes in lighting conditions and hand occlusion. Overall, this survey the paper offers a thorough analysis of the most recent advancements in hand gesture recognition using machine-learning techniques. It highlights the potential of machine learning in this field and provides insights into the future direction of research in this area.

In recent study, Abdullahi and Chamnongthai [3] used the creation of a Convolutional Neural Network (CNN)-based real-time generation model for indigenous sign-language in Nigeria. The system was created to overcome the communication gap in Nigeria between the hearing and deaf communities. The system utilizes a CNN architecture for hand sign recognition and classification, specifically designed for Nigerian sign language. The dataset used for training the CNN model was collected from deaf individuals in Nigeria, and it consisted of images of hand signs for various words and phrases in Nigerian sign language. Two parts make up the created system: a part that recognises hand signs and a part that translates sign language. In order to identify and categorise hand signs, the CNN model is used by the hand sign recognition component. The sign language interpretation component then interprets the identified signs into spoken language. The system was evaluated using a dataset of recorded sign language conversations between deaf individuals and hearing individuals, and it achieved a high recognition accuracy of 95.7%. The system was also able to interpret the recognized signs into spoken language accurately. The proposed technology can assist close the communication gap between the hearing and deaf communities in Nigeria and enhance the quality of life for the deaf, according to the article's conclusion. The system can also be adapted for other indigenous sign languages in different parts of the world, providing a solution to the challenges faced by the deaf in communicating with the hearing community.

Côté-Allard et al. [4] propose the Hand Gesture Recognition Using Transfer Learning and CNN is a research paper that proposes a transfer learning-based approach for improving the performance of sEMG (surface electromyography) hand gesture recognition. The authors note that sEMG hand gesture recognition is a challenging task due to the high variability in signal patterns caused by individual differences, electrode placement, and muscle fatigue. The suggested technique makes use of an ImageNet-trained CNN. A large-scale image classification dataset, and fine-tunes it on a smaller sEMG hand gesture dataset. The authors show that the pre-trained CNN can learn robust feature representations from the ImageNet dataset, which can be transferred to the sEMG hand gesture recognition task. The more appropriate adjusts the pre-trained network's parameters to fit the sEMG dataset, improving the network's performance on this task. The authors conducted experiments on two publicly available sEMG hand gesture recognition datasets, and the results showed that the

proposed transfer learning approach outperforms several state-of-the-art methods for sEMG hand gesture recognition. The authors conclude that transfer learning is an effective approach for improving the performance of sEMG hand gesture recognition, especially when the available dataset is small. In summary, this research paper proposes a transfer learning-based approach for sEMG hand gesture recognition that utilizes a pre-trained CNN on a large-scale image classification dataset. The approach achieved superior performance in comparison to currently used state-of-the-art techniques and provides a promising direction for improving sEMG hand gesture recognition tasks.

Zhan [5] developed the recognition of Hand Gesture Image Using Deep CNNs is a research paper that proposes a hand gesture recognition model using deep CNNs. The paper focuses on recognizing five hand gestures, including thumbs up, thumbs down, peace, rock, and OK, and evaluates the proposed model's performance against existing methods. Three convolutional layers make up the proposed model, which is then subsequently by max-pooling layers and two fully connected classification layers. The model is trained using the backpropagation algorithm with stochastic gradient descent optimization on a dataset of photos of hand gestures. The model's performance was enhanced by the authors using data augmentation techniques like rotation, flipping, and scaling. The outcomes demonstrate that the suggested model beats the current approaches in terms of recognition accuracy, obtaining accuracy equivalent to 98.62%. The authors also compared the model's performance against the human recognition rate and found that the model achieved comparable accuracy. The paper concludes that the proposed model using deep CNNs is an effective approach for hand gesture recognition and can be applied in various disciplines including sign language recognition, human-computer interface, and robotics. The authors suggest that future work could involve expanding the dataset to include more hand gestures and testing the model's performance on different hardware platforms.

We used a deep learning-based framework, specifically employing CNNs, which is justified for hand sign recognition due to their prowess in learning intricate visual patterns, accommodating spatial relationships, and ultimately contributing to enhanced accuracy and robustness in recognizing diverse hand signs. The main objective of this study is to develop an automated hand sign recognition system using Convolutional Neural Networks (CNNs) based on deep learning techniques. This endeavor seeks to enhance recognition accuracy through the utilization of a comprehensive deep-learning framework.

The study is structured as follows: Section I reviews the literature on issue formulation and several approaches that are currently in use, and Section II presents the suggested architecture. The results are then examined in Section III. The conclusion is described in Section IV.

## 2. PROPOSED METHOD

In this section, we described the dataset used in the study to measure the effectiveness of the proposed model. Besides, we describe the CNN-based architecture used for hand sign recognition. The signs illustrate A, B, Delete and Q are depicted in Figure 1.



**Figure 1.** Signs illustrate A, B, Delete and Q from top left to bottom right

## 2.1 Dataset

The deaf and hard-of-hearing community in the United States uses American Sign Language (ASL), a sophisticated and distinctive language. Body language, facial emotions, and hand gestures are all used to convey meaning. As the use of ASL continues to grow, so does the need for technologies that can recognize and translate ASL gestures into written or spoken language. One such technology is computer vision, which uses machine learning algorithms to analyze and interpret visual data. To develop and train these algorithms, large datasets of ASL gestures are required. Kaggle, a popular platform for data science competitions, offers an ASL Alphabet dataset consisting of images of each letter of the alphabet in ASL gestures. This dataset contains over 87,000 images and has been labeled for use in training and testing machine learning models [6].

The ASL Alphabet dataset from Kaggle provides a valuable resource for researchers and developers working on computer vision applications for ASL recognition. By analyzing and processing these images, algorithms can learn to identify and classify ASL gestures with a high degree of accuracy. Ultimately, this technology has the potential to greatly improve system and accessibility for the deaf and hard-of-hearing, opening up new opportunities for social and economic inclusion. There are 87,000 images of  $200 \times 200$  pixels. The 29 courses include 26 courses for the letters A through Z along with 3 classes having SPACE, DELETE, and NOTHING. To encourage the use of actual test photographs, just 29 images make up the test data set [6]. Besides, preprocessing is a crucial step in preparing an image dataset for machine learning applications. The ASL Alphabet dataset from Kaggle consists of images of the letters of the alphabet in American Sign Language gestures. Here are some preprocessing steps that can be applied to this dataset.

**Resizing:** The photographs in the collection could have various aspect ratios. The image processing can be made easier and the dataset more manageable by resizing the photos to a standard size. This can be done using a library like OpenCV.

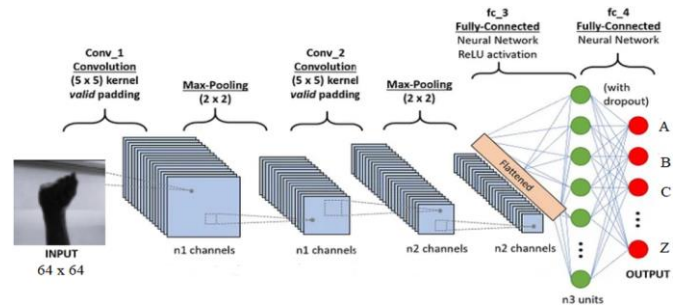
**Grayscale conversion:** Converting the images to grayscale can simplify the data and reduce the amount of computational resources required for processing. This can also help in reducing overfitting and improving the accuracy of the model.

**Normalization:** The pixel values to a same range, such as  $[0, 1]$ , can help in reducing the impact of lighting and contrast variations across images. This can be done by dividing each pixel value by the maximum pixel value of the image.

**Label encoding:** The ASL Alphabet dataset from Kaggle has images labeled with the corresponding alphabet letter. However, machine learning algorithms work better with numeric labels. Therefore, label encoding can be applied to convert the alphabet labels to numeric labels.

## 2.2 Convolutional Neural Networks

Figure 2 shows the CNN-based deep learning architecture for recognizing hand signs. Convolutional Neural Networks (CNNs), a type of deep learning neural network, are frequently employed for image identification and classification applications. CNNs are designed to automatically detect features in input images through a process called convolution, which extracts specific features such as edges, corners, and textures [7]. In sum, CNN architecture is chosen for Hand Sign Recognition due to its innate ability to capture spatial features and patterns within images, which aligns perfectly with the distinctive visual characteristics of hand signs, leading to superior recognition accuracy.



**Figure 2.** Proposed architecture

**Convolution:** A third function is created via convolution, a mathematical procedure that mixes two functions. In image analysis and machine vision, convolution is a widely used technique for filtering and extracting features from images. In convolution, a low matrix called a filter is applied to every pixel of a picture to produce a new pixel value. The kernel is typically a small square matrix, and the size of the kernel determines the size of the output image [8]. The process involves sliding the kernel over each pixel of the input image and performing a mathematical operation between the kernel and the input image. The result is a new pixel value in the output image. Convolution can be used for a variety of tasks, such as edge detection, image sharpening, blurring, and feature extraction. It is also an essential component of deep learning neural networks, where it is used to extract features from images or other data types [9-15].

**Pooling:** Convolutional Neural Networks (CNNs) frequently employ the pooling approach to scale back the spatial dimension of feature maps. Pooling is used to downscale the feature maps while keeping the most crucial details. Max pooling is the most commonly used type of pooling in CNNs, where a kernel of a fixed size slides over the feature map and selects the maximum value within the kernel. Max pooling is a strategic inclusion in CNN architecture for Hand Sign Recognition as it serves two key purposes. First, it aids in reducing the spatial dimensions of feature maps, enabling the network to focus on the most essential information while discarding redundant details. Second, max pooling enhances the network's robustness against slight variations in hand sign positioning, scale, or orientation, ensuring accurate recognition by capturing the most prominent features regardless of their precise location within the image. By reducing the spatial descriptors of the feature maps, pooling helps to capture the most significant features of the input data while discarding irrelevant details. This makes the model more effective to variations in the input feature and

helps to prevent overfitting [16].

**Activation function:** ReLU is a typical activation function in neural networks, including Convolutional Neural Networks (CNNs). It is a simple function that returns the input value if it is positive, and zero otherwise. The primary advantage of using ReLU over other activation functions is that it is computationally efficient, and it allows for faster training of deep neural networks. Additionally, ReLU has been shown to improve the accuracy of the network, compared to other activation functions. Another advantage of ReLU is that it helps to address the problem of vanishing gradients, which can occur when using other activation functions such as sigmoid or tanh. This is because ReLU does not saturate for positive input values, which means that it does not cause the gradients to become small, and thus, it does not hinder the learning process [17, 18]. Besides, ReLU is an optimal choice for CNN architecture in Hand Sign Recognition due to its non-linearity and computational efficiency. Its ability to facilitate gradient propagation addresses vanishing gradient problems, promoting faster convergence during training. This, in turn, empowers the network to effectively learn intricate features from hand sign images, resulting in improved recognition performance.

**CNN architecture:** (1) The model consists of several layers that are designed to process 2D images with a size of 64x64 pixels and 3 color channels (RGB). (2) Convolutional layer is the initial layer that applies 32 filters of size 5x5 to the input image. This layer is followed by an activation function that applies a rectified linear unit (ReLU) to the output of the convolutional layer. The next layer is a max-pooling layer that reduces the size of the feature maps by a factor of 2. (3) Similar to the first layer, the second and third levels have 64 filters of size 3x3. A ReLU activation function and a max pooling layer come after each convolutional layer. (4) The 2D feature maps are transformed into a 1D feature vector and then passed on to a dense layer using the flatten layer. The dense layer uses a ReLU activation function and has 128 neurons. (5) Finally, a thick layer of 29 neurons, one for each class in the dataset, makes up the output layer. The output layer generates a probability distribution over the classes using a softmax activation function. The model summary shows the architecture of the model, including the number of parameters in each layer. This information can be used to optimize the model for training and inference [19-25].

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 60, 60, 32)	2432
activation (Activation)	(None, 60, 60, 32)	0
max_pooling2d (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_1 (Conv2D)	(None, 28, 28, 64)	18496
activation_1 (Activation)	(None, 28, 28, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	36928
activation_2 (Activation)	(None, 12, 12, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 128)	295040
dense_1 (Dense)	(None, 29)	3741

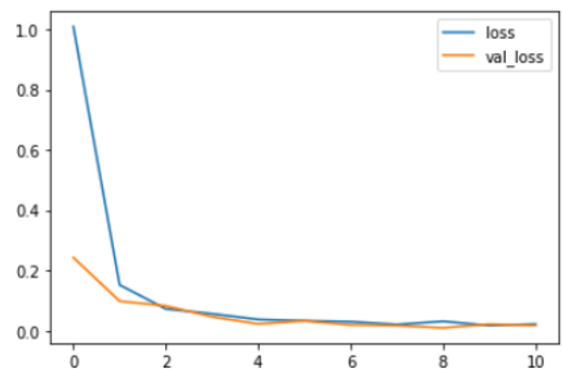
Total params: 356,637  
 Trainable params: 356,637  
 Non-trainable params: 0

**Figure 3.** Show that the number of parameters in each layer

**Early Stopping:** Early stopping is used to guarantee that the model fitting ends at the accuracy point where it is best optimized. The model could begin overfitting as it reaches the early stopping point. This step can be omitted while testing and the entire training process can proceed instead. CNN employ the method of early stopping to avoid overfitting and enhance model generalization. The idea behind early stopping involves keeping an eye on the model's validation loss while it is being trained and terminating the process when the loss stops reducing or starts to increase. In order to assess the model's performance on unobserved data, the validation loss is computed on a different dataset from the training data. The validation loss is continuously checked throughout training, and the training process is terminated. By stopping the training process early, the model can prevent overfitting and improve its generalization performance. This is because overfitting occurs when the capacity of the model to generalize to new data is lost when it begins to memorize the training data. Early stopping is a simple and effective technique for preventing overfitting in CNNs, and it is widely used in practice [26, 27]. It is typically implemented using the Keras library in Python, which provides a built-in callback function for early stopping. The callback function can be configured to monitor the validation loss with no improvement, and restore the weights of the best-performing model shown in Figure 3.

### 3. RESULT AND DISCUSSION

In this section, we described the obtained results with comparative analysis of the study. We also present the detailed description of CNN-based architecture for Hand Sign Recognition. Loss is a performance indicators used in machine learning to quantify the discrepancy between a model's anticipated output and the actual output. It is a gauge of how closely the model may get to the intended purpose. Learning a mapping from input features to output labels is the aim of supervised learning. The input features and the accompanying output labels are presented in a labelled dataset on which the model is trained. The loss function is used to quantify the discrepancy between the model's projected output and the output label that actually occurred. The problem being solved and the nature of the predicted output from the model influence the choice of loss function.



**Figure 4.** A plot of Loss over time (running epochs)

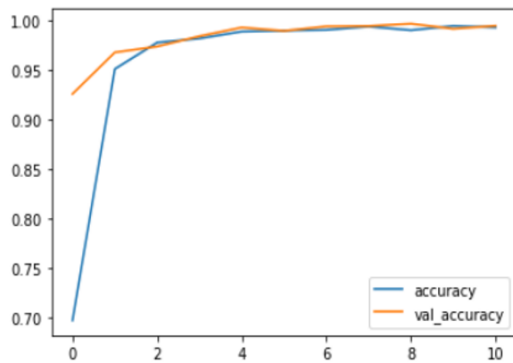
During training, the loss function is improved using an optimization approach such stochastic gradient descent (SGD). In order to assess the model's performance during training and testing, the loss function's value is used. Better performance is



indicated by a lower value of the loss function since it suggests that is able to better approximate the target function. A plot of Loss over time (running epochs) are provided in Figure 4. Table 1 provides information about the set's loss in training and validation. Epochs accuracy is a ML performance parameter that assesses how well a model can categories or forecast the outcome of a given input. This the percentage of correctly predicted outputs over the total number of inputs. In supervised learning, accuracy is a widely utilized metric in classification tasks, which involve predicting a categorical output label. For instance, when dealing with binary classification problems. A positive or negative label is predicted by the model, and accuracy is the proportion of properly categorized occurrences relative to the total number of examples. Accuracy is a simple and intuitive performance metric that is easy to understand and interpret. However, it has some limitations. For example, it may not be a suitable metric when the classes are imbalanced, as a high accuracy may be achieved by simply predicting the majority class. A plot of accuracy with running epochs is shown in Figure 5.

**Table 1.** Loss in training and validation set with epochs

Epochs	Loss	val Loss
1	0.008968	0.097741
2	0.151430	0.082014
3	0.072046	0.045856
4	0.055787	0.022153
5	0.036911	0.032015
6	0.032577	0.018841
7	0.029511	0.016817
8	0.020253	0.008985
9	0.030768	0.021358
10	0.021693	0.017304



**Figure 5.** A plot of accuracy with running epochs

**Table 2.** Obatined accuracy in training and validation set with epochs

Epochs	Accuracy	val Accuracy
1	0.697570	0.926284
2	0.951445	0.968238
3	0.982282	0.974176
4	0.989261	0.984598
5	0.990263	0.993372
6	0.990952	0.989962
7	0.994335	0.994521
8	0.990624	0.994866
9	0.994975	0.997165
10	0.993432	<b>0.995096</b>

The obtained accuracy in training and validation set with Epochs is provided in Table 2. Confusion Matrix Heat Map: A

table commonly utilized to assess the effectiveness of a classification model is known as a confusion matrix. A visual representation of data in which colors are used to indicate the individual values within a matrix is known as a heat map. Combining the two, a confusion matrix heat map is a visual representation of a confusion matrix where each cell in the table is color-coded according to the value it contains. This makes it easy to identify the performance of a classification model at a glance. One possible way to visually represent the results of a binary classification problem is through a confusion matrix heat map. The amount of False positives and negatives as well as real positives and true negatives may all be depicted in this sort of diagram using various colours, with the former two commonly portrayed in green and the latter two in red. This would allow you to quickly see which classes your model is performing well on and where it might need improvement.

In this work, the Hand Sign Recognition Model for Alphabets Using Convolution Neural Network has been successfully developed and trained on a dataset of 87000 images. The model achieved an impressive accuracy of 99%, which indicates that it is highly reliable for recognizing hand sign gestures for alphabets. The early stopping significantly influences the final accuracy of the model by preventing overfitting. As training progresses, there is a risk of the model becoming too specialized to the training data, leading to poor generalization on unseen examples. Early stopping monitors, the validation performance and halts training when it starts to deteriorate, preventing the model from memorizing noise in the data. This ensures that the model captures meaningful patterns, resulting in a better-balanced accuracy on new data and a more robust outcome. The model's great accuracy suggests that it can be used for a variety of practical purposes, such as sign language interpretation, hearing aids for the deaf, and other interactive applications. Additionally, the successful development of this model highlights the effectiveness of convolution neural networks for image recognition tasks. The model tends to find crucial features in hand sign recognition that capture the distinctive shapes, contours, and spatial arrangements which are indicative of the underlying alphabetic gestures.

#### 4. CONCLUSION

In summary, this hand sign recognition model has highlighted its capability to offer an effective and precise solution for detecting hand sign gestures corresponding to alphabets. Potential enhancements can involve broadening the dataset and investigating alternative methods for image augmentation, thereby bolstering the model's proficiency in identifying intricate hand gestures. In future work, we intend to present class-specific performance analysis and different model comparisons by exploring model deep learning architectures like Recurrent Neural Networks (RNNs) or Transformer-based models, along with transfer learning and multi-modal data integration, which could further enhance hand sign recognition beyond the proposed CNN-based approach.

#### REFERENCES

[1] Shahriar, S., Siddiquee, A., Islam, T., et al. (2018). Real-time american sign language recognition using skin

- segmentation and image category classification with convolutional neural network and deep learning. In TENCON 2018-2018 IEEE Region 10 Conference, Jeju, Korea (South), pp. 1168-1171. <https://doi.org/10.1109/TENCON.2018.8650524>
- [2] Yashas, J., Shivakumar, G. (2019). Hand gesture recognition: A survey. In 2019 International Conference on Applied Machine Learning (ICAML), Bhubaneswar, India, pp. 3-8. <https://doi.org/10.1109/ICAML48257.2019.00009>
- [3] Abdullahi, S.B., Chamnongthai, K. (2022). American sign language words recognition using spatio-temporal prosodic and angle features: A sequential learning approach. *IEEE Access*, 10: 15911-15923. <https://doi.org/10.1109/ACCESS.2022.3148132>
- [4] Côté-Allard, U., Fall, C.L., Campeau-Lecours, A., Gosselin, C., Laviolette, F., Gosselin, B. (2017). Transfer learning for sEMG hand gestures recognition using convolutional neural networks. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, pp. 1663-1668. *IEEE*. <https://doi.org/10.1109/SMC.2017.8122854>
- [5] Zhan, F. (2019). Hand gesture recognition with convolution neural networks. In 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), Los Angeles, CA, USA, pp. 295-298. <https://doi.org/10.1109/IRI.2019.00054>
- [6] Aly, S., Aly, W. (2020). DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. *IEEE Access*, 8: 83199-83212. <https://doi.org/10.1109/ACCESS.2020.2990699>
- [7] 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): 662-700.
- [8] Khetavath, S., Sendhilkumar, N.C., Mukunthan, P., et al. (2023). An intelligent heuristic manta-ray foraging optimization and adaptive extreme learning machine for hand gesture image recognition. *Big Data Mining and Analytics*, 6(3), 321-335. <https://doi.org/10.26599/BDMA.2022.9020036>
- [9] Wang, Z., Wang, D., Yao, T., An, J., Lv, D. (2021). A ring-shaped hand motion sensor based on triboelectricity. *IEEE Sensors Journal*, 21(20): 22472-22479. <https://doi.org/10.1109/JSEN.2021.3106008>
- [10] Natarajan, B., Rajalakshmi, E., Elakkiya, R., Kotecha, K., Abraham, A., Gabralla, L.A., Subramaniaswamy, V. (2022). Development of an end-to-end deep learning framework for sign language recognition, translation, and video generation. *IEEE Access*, 10: 104358-104374. <https://doi.org/10.1109/ACCESS.2022.3210543>
- [11] Dong, Y., Liu, J., Yan, W. (2021). Dynamic hand gesture recognition based on signals from specialized data glove and deep learning algorithms. *IEEE Transactions on Instrumentation and Measurement*, 70: 2509014. <https://doi.org/10.1109/TIM.2021.3077967>
- [12] Dwijayanti, S., Sahirah, I.T., Hikmarika, H., Suprpto, B.Y. (2021). Indonesia sign language recognition using convolutional neural network. *International Journal of Advanced Computer Science and Applications*, 12(10): 415-422.
- [13] Breland, D.S., Skriubakken, S.B., Dayal, A., Jha, A., Yalavarthy, P.K., Cenkeramaddi, L.R. (2021). Deep learning-based sign language digits recognition from thermal images with edge computing system. *IEEE Sensors Journal*, 21(9): 10445-10453. <https://doi.org/10.1109/JSEN.2021.3061608>
- [14] Zhang, J., Wang, Q., Wang, Q., Zheng, Z. (2023). Multimodal fusion framework based on statistical attention and contrastive attention for sign language recognition. *IEEE Transactions on Mobile Computing*. <https://doi.org/10.1109/TMC.2023.3235935>
- [15] Singha, J., Das, K. (2013). Indian sign language recognition using eigen value weighted euclidean distance based classification technique. *International Journal of Advanced Computer Science and Applications*, 4(2): 188-195.
- [16] Gupta, S. D., Kundu, S., Pandey, R., Ghosh, R., Bag, R., Mallik, A. (2012). Hand Gesture recognition and classification by discriminant and principal component analysis using machine learning techniques. *International Journal of Advanced Research in Artificial Intelligence*, 1(9): 46-51.
- [17] AlKhuraym, B. Y., Ismail, M.M.B., Bchir, O. (2022). Arabic sign language recognition using lightweight CNN-based architecture. *International Journal of Advanced Computer Science and Applications*, 13(4): 319-328.
- [18] Liu, Z., Pang, L., Qi, X. (2022). MEN: Mutual enhancement networks for sign language recognition and education. *IEEE Transactions on Neural Networks and Learning Systems*. <https://doi.org/10.1109/TNNLS.2022.3174031>
- [19] Lin, F., Wang, Z., Zhao, H., Qiu, S., Liu, S., Shi, X., Wang, C., Yin, W. (2023). Hand movement recognition and salient tremor feature extraction with wearable devices in Parkinson's patients. *IEEE Transactions on Cognitive and Developmental Systems*. <https://doi.org/10.1109/TCDS.2023.3266812>
- [20] Saqib, S., Kazmi, S.A.R., Masood, K., Alrashed, S. (2018). Automatic sign language recognition: Performance comparison of word based approach with spelling based approach. *International Journal of Advanced Computer Science and Applications*, 9(5): 401-405.
- [21] Yosinski, J., Clune, J., Bengio, Y., Lipson, H. (2014). How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*.
- [22] Anwar, S., Hwang, K., Sung, W. (2015). Fixed point optimization of deep convolutional neural networks for object recognition. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, pp. 1131-1135. <https://doi.org/10.1109/ICASSP.2015.7178146>
- [23] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1): 1929-1958.
- [24] Al-Timemy, A.H., Khushaba, R.N., Bugmann, G., Escudero, J. (2015). Improving the performance against force variation of EMG controlled multifunctional upper-limb prostheses for transradial amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(6): 650-661. <https://doi.org/10.1109/TNSRE.2015.2445634>

- [25] Seddiqi, M., Kivrak, H., Kose, H. (2020). Recognition of turkish sign language (TID) using sEMG sensor. In 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, pp. 1-6. <https://doi.org/10.1109/ASYU50717.2020.9259859>
- [26] Banerjee, K., Harsha, K.G., Kumar, P., Vats, I., Vinooth, P., Dasila, P., Akhtar, N., Kumar, A., Gautam, P. (2022). A review on artificial intelligence based sign language recognition techniques. In 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, pp. 2195-2201. <https://doi.org/10.1109/IC3I56241.2022.10073186>
- [27] Wachs, J.P., Kölsch, M., Stern, H., Edan, Y. (2011). Vision-based hand-gesture applications. *Communications of the ACM*, 54(2): 60-71. <http://dx.doi.org/10.1145/1897816.1897838>