IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 9, September 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 9, September 2024| DOI: 10.15680/IJIRCCE.2024.1209033

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Human-Computer Interaction in AR/VR with Gesture Recognition: A Deep Learning Perspective

Chandrika E, Rajshekar M, Varun R, Bharathi Ramesh

PG Student, Department of M.C.A, Surana College (Autonomous), Kengeri, Bangalore, India

Assistant Professor, Department of M.C.A, Surana College (Autonomous), Kengeri, Bangalore, India

ABSTRACT: The rapidly growing field of human-computer interaction in AR and VR environments integrates the physical and digital worlds to create immersive experiences. These technologies have the potential to transform various sectors like education, healthcare, entertainment, and design with user-friendly and interactive interfaces. HCI in AR/VR focuses on designing user experiences using natural interactions like gesture recognition, voice commands, and haptic feedback. Gesture recognition plays a key role in AR and VR environments by enabling users to interact with virtual objects more naturally and intuitively. This proposed study employs deep learning algorithms to accurately detect and interpret hand or body gestures, empowering users to interact with virtual objects and navigate virtual spaces for an enhanced AR/VR experience.

KEYWORDS: Augmented Reality, Virtual Reality, Deep Learning, Deep Neural Network, Convolutional Neural Network, Recurrent Neural Network, Gesture Recognition

I. INTRODUCTION

Deep learning, a subset of machine learning, empowers computers to learn from data inputs using artificial neural networks. This approach facilitates self-directed learning of features [1], significantly advancing technology in fields like image recognition, speech recognition, and natural language processing. Deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are adept at identifying complex patterns in data. This technique represents the cutting edge in addressing various large-scale data challenges, with applications spanning computer vision, robotics, agriculture, and healthcare [2].

Gesture recognition allows computers to interpret and understand human movements, enabling more natural and intuitive interaction with virtual objects [3]. In AR/VR environments, gesture recognition can enhance the user experience by enabling more direct and intuitive interactions. Typically, gesture recognition involves hand detection and image segmentation as its primary steps. This process includes capturing visual data from the hand as seen by the camera. Factors such as color, shape, and motion can contribute to inaccuracies and reduced performance in image segmentation. After detecting the hand, a tracking mechanism monitors the segmented hand regions frame by frame to analyze hand movements. This in turn ensures that every action is tracked, increasing the efficiency of data analysis and interpretation. The last stage is recognition when the system deciphers the meaning of hand position, posture, or gesture. The extracted data will be organized to identify patterns that the trained algorithm uses to compare and determine the gesture being performed.

Despite advancements, gesture recognition techniques continue to encounter challenges, particularly in the complex task of breaking down continuous signals into distinct components. Recent improvements in this technology offer boundless possibilities for interacting with virtual reality, overcoming the limitations imposed by physical devices. I utilized machine learning methods, including deep neural networks (DNN) and convolutional neural networks (CNN), for gesture recognition.

A. Challenges of Gesture Recognition

The recent research that shows promise for applications in HCI interaction, healthcare, and various other fields. The review emphasizes key challenges and opportunities, providing insights into future research directions in gesture recognition [4].

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- Hand gestures are different: Everyone makes gestures a little differently, so computers have trouble recognizing them.
- Inadequate training data: Insufficient labeled data (examples of gestures with accurate annotations) poses a challenge for training computers to recognize gestures effectively.
- Real-time computing: Gesture recognition needs to be performed quickly; however, deep learning representations can be slow and require substantial computing resources.
- Lighting conditions and Background disturbances: Changes in lighting conditions or background may mislead computer gesture recognition.
- Similar gestures may be confusing: Some gestures are similar, and hence computers get confused and fail to distinguish between them.

B. Deep Neural Network

A deep neural network (DNN) algorithm is a machine learning model inspired by the structure and function [5] of the human brain. It consists of a multi-layered network, usually with more than three layers, where interconnected units process and convert input data into output.

Deep neural networks (DNNs) are utilized across a wide range of domains, such as speech and gesture recognition, machine translation, sentiment analysis, and healthcare. They also play a key role in computer vision tasks like image classification, object recognition, and image separation. Additionally, DNNs are employed in natural language processing (NLP) tasks, including language modeling and text classification.

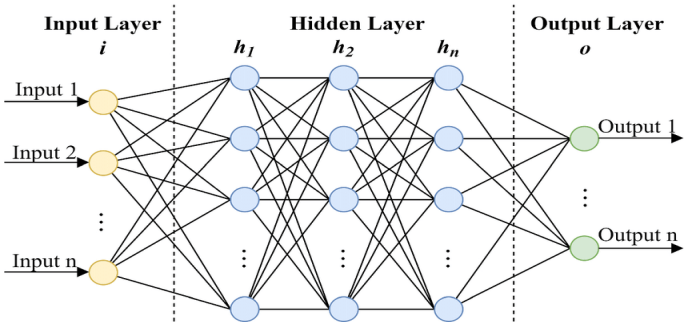


Figure 1: Architecture of Deep Neural Network

Figure (1) above illustrates the overall architecture of DNN, where several layers of artificial neurons, known as perceptron, are stacked in processing complex data-dependent problems.

- Input Layer**:** These layers are responsible for receiving input data from external sources and directly transmitting it to the next layer without performing any computations.
- Hidden Layers: These layers carry out intermediate computations and generate outputs. By linking several hidden layers, a network can reveal various underlying features. For instance, in image processing, the initial hidden layers might detect edges, whereas the later layers assist in identifying whole objects.
- Output Layer: These layers take input from the previously hidden layers and produce the final prediction, drawing on the model's inferences, to deliver to the user.

C. Convolutional Neural Network

Convolutional neural networks (CNNs) are a form of deep learning classifier tailored for image recognition and processing [6]. Their distinctive capability to effectively handle grid-like data, such as images, makes them especially well-suited for tasks involving image analysis.

Convolutional neural networks (CNNs) are applied in numerous fields. They are extensively used in object detection, image classification, medical image analysis, and natural language processing. Additionally, CNNs are employed in image segmentation, face recognition, speech recognition, and video analysis. Furthermore, CNNs play a significant role in promising technologies such as self-driving cars, robotics, and surveillance systems.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

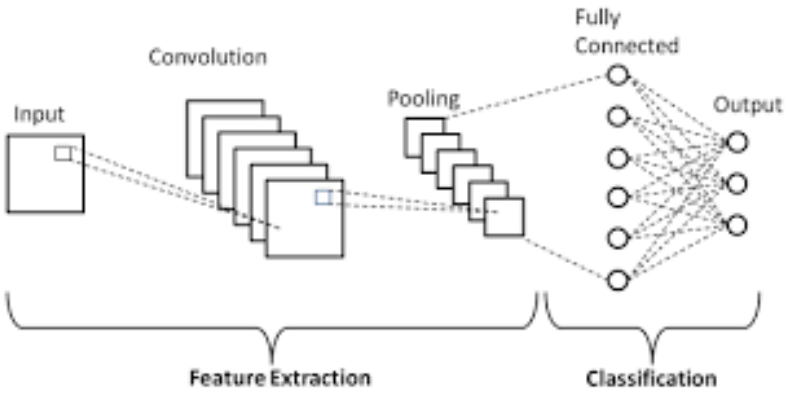


Figure 2: Architecture of Convolutional Neural Network

Figure (2) above illustrates the overall architecture of a CNN, where multiple layers hierarchically process the input data.

- Convolutional Layers: These layers apply filters to small regions of input data, scanning horizontally and vertically to detect local patterns, edges, or textures.
- Activation Function: The output from the convolutional layers is processed by an activation function, commonly ReLU (Rectified Linear Unit), which introduces non-linearity to the features, allowing for a more accurate representation.
- Pooling Layers: These are down-sampling layers to reduce the spatial dimensions of the feature maps while preserving important information, thereby minimizing overfitting by reducing the parameters and computations.
- Flatten Layer: This layer reshapes the output from the convolutional and pooling layers into a 1-D array, preparing it for input into fully connected layers.
- Fully Connected Layers: Also called dense layers, they are accountable for regression or classification tasks taking the flattened output to generate the final prediction.

II. LITERATURE REVIEW

Human-computer interaction (HCI) involves the interactive relationship between people and computers or machines [7], focusing on how human actions achieve specific objectives. When designing an HCI system, two crucial factors are functionality and usability [8]. Functionality encompasses the range of services and features the system provides, while usability assesses how effectively the system understands and executes the user's intentions. A system that achieves a balance between these elements is considered highly effective and efficient. Additionally, gestures are essential for enabling communication between humans and machines, as well as for facilitating sign language [9] communication among individuals.

A significant study [10] on hand gesture recognition employed a multivariate Gaussian distribution to detect gestures applying non-geometric features. This method involved segmenting the input hand image through two complementary techniques: skin color-based segmentation utilizing the HSV color model, and thresholding methods [11] based on clustering. The hand's shape was then captured through a sequence of operations that extracted pertinent hand features, with a modified directional analysis algorithm applied to examine the variance and covariance of a statistical parameter, enabling calculations of slope and direction (trend).

This study [12] introduces a neural network-based system for recognizing American Sign Language (ASL) static poses. The system begins by converting the input image into the HSV color model and resizing it to 80x64 pixels. Preprocessing operations are then applied to isolate the hand from a uniform background. Feature extraction utilizes histogram and Hough algorithm techniques. A three-layer feed-forward neural network is employed for gesture classification, with 5 samples used for training and 3 for testing each of the 26 ASL signs. The system achieves a recognition accuracy of 92.78% in MATLAB. Additionally, the real-time vision-based gesture recognition system

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[13] is capable of accurately identifying 35 different hand gestures from both Indian Sign Language (ISL) and American Sign Language (ASL). To minimize false detections, the RGB to GRAY segmentation technique is implemented, and an enhanced Scale Invariant Feature Transform method is proposed for feature extraction. The system is designed with a GUI model in MATLAB to provide an efficient and user-friendly hand gesture recognition experience.

This research [14] reviews various hand gesture recognition systems and sign language detection methods proposed by different researchers. Sign language is crucial for communication among deaf or hard-of-hearing individuals who depend on manual gestures and body language. Typically, sign language detection systems [15] follow a three-step process: preprocessing, classification, and character extraction. Classification techniques employed in these systems include neural networks, support vector machines, hidden Markov models, scale-invariant transform functions, and other methods. Users of gesture detection systems demand quick and precise responses, as any delays or inaccuracies can significantly impact user satisfaction and performance [16]. To ensure a positive user experience (UX), it is crucial for augmented reality (AR) systems to be both accurate and robust, especially in settings beyond casual use, such as medical, educational, or professional environments, where smooth and normal interaction is critical despite any technological limitations.

III. METHODOLOGY

The proposed method combines the strengths of deep neural networks (DNN) and convolutional neural networks (CNN) to provide a highly accurate and reliable system for identifying and categorizing hand gestures. CNNs are particularly effective in processing image data, as they can capture spatial hierarchies and patterns through convolutional layers, pooling, and non-linear activation functions. This enables the system to identify subtle details and variations in hand movements, such as finger positions or orientations, which are crucial for distinguishing between different gestures. By integrating DNN layers, the model further enhances its ability to extract meaningful features from the gesture data. DNNs are excellent at identifying complex relationships and high-level abstractions in data, allowing the model to better generalize and classify gestures into distinct categories. The combination of CNNs for spatial feature extraction and DNNs for decision-making leads to a robust system that excels in both accuracy and efficiency.

The methodology is organized into a clear, step-by-step process that begins with data preprocessing, where raw gesture images or sequences are normalized and augmented to improve the model's training robustness. Following preprocessing, CNN layers are applied to extract spatial features from the gesture data. These features are then passed through the DNN for classification, where the system determines the most likely gesture category. Finally, the model adapts to complex gesture patterns through iterative learning, enhancing its ability to recognize gestures across various lighting conditions, hand sizes, and orientations. Overall, the use of DNN and CNN in tandem ensures that the proposed method can handle the complexities of hand gesture recognition, providing a scalable and adaptable solution for applications such as human-computer interaction, virtual reality, and sign language interpretation.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

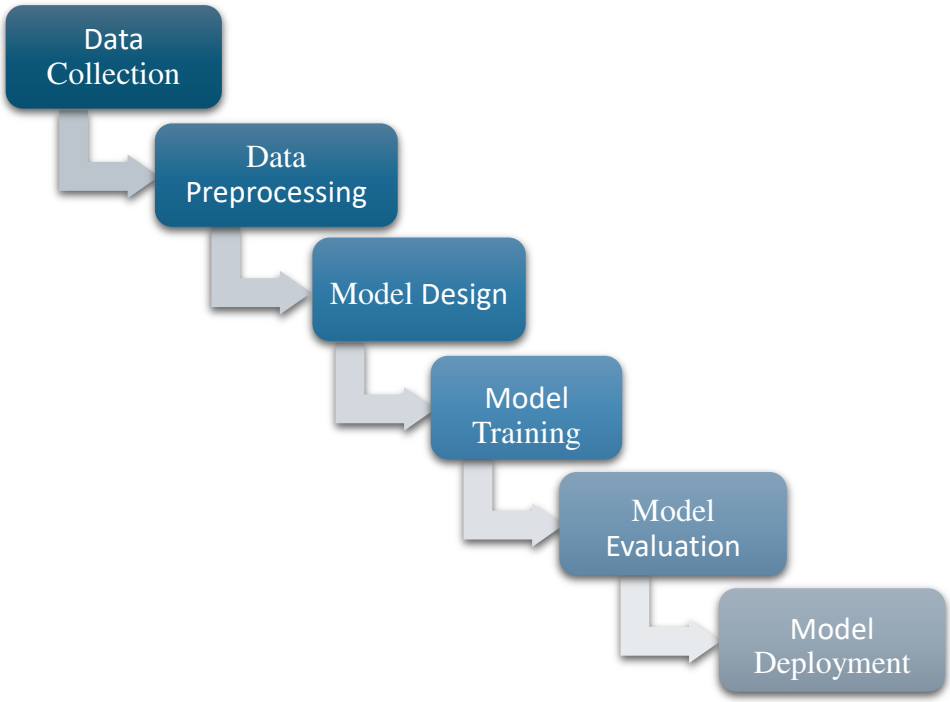


Figure 3: Proposed Methodology

- **Data Collection**: Initially, the data collection phase focuses on gathering and preparing an ASL dataset, which forms the foundation for developing an operational model.
- **Data Preprocessing**: In this phase, the collected data is prepared for model training through processes such as resizing images, normalizing pixel values, and applying data augmentation techniques.
- **Model Design**: This phase involves designing convolutional neural network (CNN) and deep neural network (DNN) models for gesture detection, with the CNN model concentrating on extracting features and the DNN model focusing on classification.
- **Model Training**: This phase trains the proposed CNN and DNN models using pre-processed data and optimizes weights and hyper-parameters to increase performance.
- **Model Evaluation**: This phase assesses the performance of the trained model by utilizing metrics such as accuracy, precision, and recall.
- **Model Deployment**: This phase includes incorporating the trained models into appropriate environments, such as computer vision systems or mobile applications, for practical gesture recognition.

IV. DATASET DESCRIPTION

The American Sign Language (ASL) dataset [17] comprises images of alphabetic characters, organized into 29 folders, each representing a different class. This dataset is used to recognize the gestures, focusing on identifying ASL characters from hand gestures. The images of various letters are employed to train convolutional neural networks (CNNs) and deep neural networks (DNNs) to accurately recognize the characters depicted in every image. The goal is to develop a system capable of reliably identifying ASL characters, thereby supporting the creation of computer vision applications that enhance communication for deaf and dumb populations.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

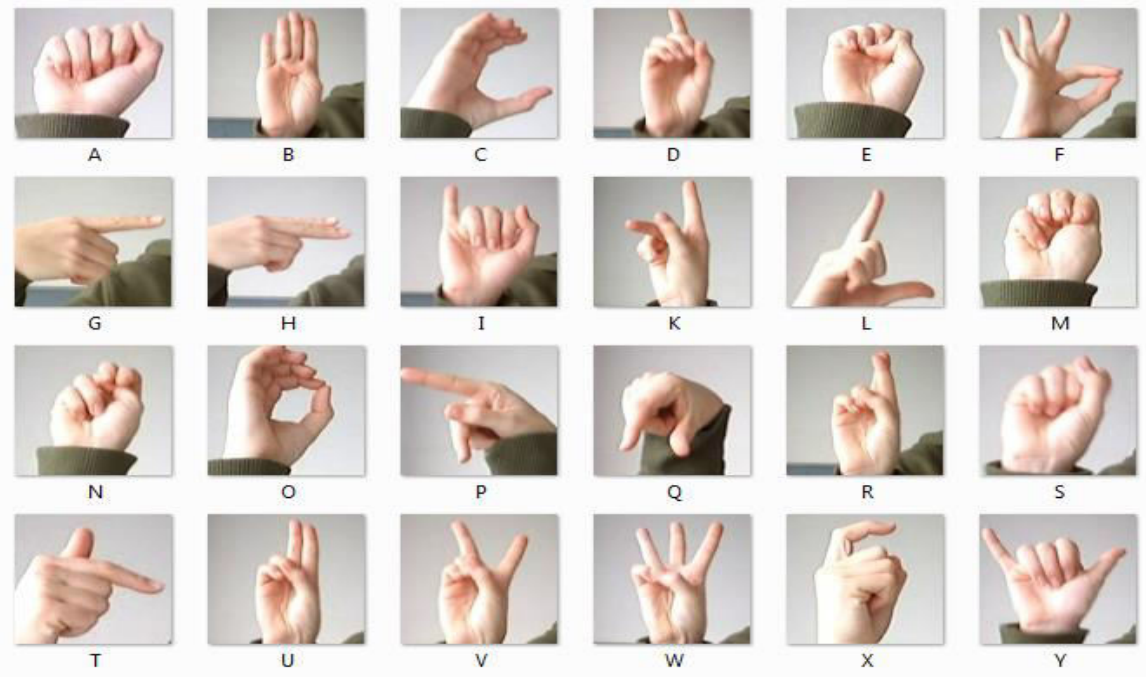


Figure 4: The American Sign Language Dataset

V. ARCHITECTURE SUMMARY

Deep learning models have transformed the field of artificial intelligence, with convolutional neural networks (CNNs) and deep neural networks (DNNs) emerging as the most prominent and widely used architectures. CNNs are designed to handle grid-like data structures such as images and videos, utilizing multiple layers for convolution, pooling, and feature extraction. On the other hand, DNNs consist of numerous interconnected layers of neurons enabling them to learn the complex patterns in the data, by making them highly effective across various applications.

A. Deep Neural Network Model Description

1. Dataset loading and preprocessing**:** The ASL dataset is from Kaggle, on which this network was trained. It contains 87,000 images of 200 x 200 pixels, each divided into 29 classes containing 26 English alphabets and 3 additional characters SPACE, DELETE, and NOTHING.
2. Data augmentation: For better real-world training, we have augmented the data with the brightness shift in conditions equal to 20% darker and a zoom shift of up to 120%.
3. Building the DNN model: The model comprises of,

- Six convolutional layers: The primary function of these layers is to extract features from the input data. The core size was consistently set at 4x4, where the units varied, including 64, 128, and 256, and the step lengths were also adjusted accordingly.
- Convolution operation:

$$Z_{i,j,k} = \sum_{m=1}^{H_f} \sum_{n=1}^{W_f} \sum_{c=1}^{C_{in}} X_{i+m-1,j+n-1,c} \cdot W_{m,n,c,k} + b_k$$

.. (1)

- $Z_{i,j,k}$ the output feature map at position (i j) corresponding to the kth filter.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- $X_{i,j,c}$ Input at position (i, j) in the i-th row, j-th column for the c-th channel.
- $W_{m,n,c,k}$ the weight of the filter at position (m, n) for the k-th filter and c-th channel.
- b_k the bias term for the k-th filter.

- ReLu Activation:

$$A_{i,j,k} = \max(0, Z_{i,j,k}) \quad .. (2)$$

- $A_{i,j,k}$ is the activated output after applying the ReLU function.

5.1 Three dropout layers: The layers have a dropout rate of 0.5, which randomly deactivates 50% of neurons in the training process, this technique will help to avoid overfitting.

$$A_{dropped} = A.M \quad .. (3)$$

- A is an input activation.
- M is the mask matrix that randomly sets each element to 0 through a probability of p (the dropout rate) or to 1 by the probability of 1− p.

5.2 One flattened layer: The flattened layer plays a vital role in preparing the output from the convolutional layers for further processing. It achieves this by transforming the multidimensional output into a 1D vector, and efficiently collapses the 2D feature maps into a 1D vector through a straightforward transforming operation:

Flattened_output = Reshape (A, [1, total_number_of_elements]) .. (4)

- Two dense layers: Fully connected layers handle the final classification task, with the first layer comprising 512 units and using ReLU activation, and the second layer featuring 29 units with SoftMax activation, serving as output layer for multi-class classification problems.
- Fully connected layer:

$$Z^{(l+1)} = W^{(l)} . A^{(l)} . b^{(l)} \quad .. (5)$$

- W(l) denotes weight matrix for layer l.
- A(l) is an input activation from the preceding layer.
- b(l) is a bias vector.

➢ ReLu Activation (for the first dense layer):

➢

$$A^{(l+1)} = \max(0, Z^{(l+1)}) \quad .. (6)$$

➢ SoftMax Activation (for the final dense layer):

➢

$$\sigma(Z_i) = \frac{e^{Zi}}{\sum_j^K e^{Zj}} \quad .. (7)$$

➢ Z_i is input for the softmax function for class i.

➢ K is the sum of classes which in this problem is 29.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Amongst the sequence of transformations, each layer processes a input data by applying a combination of mathematical operations, ultimately producing a final output for the probability distribution across all 29 classes.

Table 1: DNN Model Summary

Layer (type)	Output Shape	Parameters #
Conv2D (Conv2D)	(None, IMAGE_SIZE-3, IMAGE_SIZE-3, 64)	3136
Conv2D (Conv2D)	(None, (IMAGE_SIZE-6)//2, (IMAGE_SIZE-6)//2, 64)	65600
Dropout (Dropout)	(None, (IMAGE_SIZE-6)//2, (IMAGE_SIZE-6)//2, 64)	0
Conv2D (Conv2D)	(None, ((IMAGE_SIZE-9)//2)-3, ((IMAGE_SIZE-9)//2)-3, 128)	131200
Conv2D (Conv2D)	(None, (((IMAGE_SIZE-12)//2)-6)//2, (((IMAGE_SIZE-12)//2)-6)//2, 128)	262400
Dropout (Dropout)	(None, (((IMAGE_SIZE-12)//2)-6)//2, (((IMAGE_SIZE-12)//2)-6)//2, 128)	0
Conv2D (Conv2D)	(None, ((((IMAGE_SIZE-15)//2)-9)//2)-3, ((((IMAGE_SIZE-15)//2)-9)//2)-3, 256)	524544
Conv2D (Conv2D)	(None, (((((IMAGE_SIZE-18)//2)-12)//2)-6)//2, (((((IMAGE_SIZE-18)//2)-12)//2)-6)//2, 256)	1048832
Flatten (Flatten)	(None, ((((IMAGE_SIZE-21)//2)-15)//2) *((((IMAGE_SIZE-21)//2)-15)//2) *256)	0
Dropout (Dropout)	(None, ((((IMAGE_SIZE-21)//2)-15)//2) *((((IMAGE_SIZE-21)//2)-15)//2) *256)	0
Dense (Dense)	(None, 512)	2097216
Dense (Dense)	(None, 29)	14877
Total Parameters		**4,088,805**
Trainable Parameters		**4,088,805**
Non-Trainable Parameters		**0**

This model features a robust architecture, beginning with multiple convolutional layers designed to capture complicated patterns in the data. Dropout layers are included afterward as regularization methods to prevent overfitting. The final dense layers enable a model to interpret complex representations and classify them into 29 categories. Overall, the model has a total of 4,088,805 trainable parameters.

B. Convolutional Neural Network Model Description

1. Dataset loading and preprocessing: The ASL Alphabet dataset is from Kaggle, on which this network was trained. It contains 87,000 images of 200 x 200 pixels, each divided into 29 classes containing the 26 English alphabets and 3 additional characters SPACE, DELETE, and NOTHING.
2. Data augmentation: Keras' ImageDataGenerator offers a preprocessing tool that introduces various distortions into an image dataset. This is accomplished by randomly applying transformations, for example rotation, zoom, and scaling to images on a pixel-by-pixel basis. Additionally, the scaling parameter normalizes the image dataset by dividing each pixel value by 255, which ensures the resulting pixel values fall within the range of 0 to 1.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3. Building the CNN model: The model consists of,

- Three convolutional layers: A proposed CNN architecture includes three convolutional layers with progressively fewer filters: 128, 64, and 32. The kernel sizes for these layers are 5x5, 3x3, and 2x2, respectively. Each layer employs the ReLU activation function and uses a stride of 1. Notably, the input shape is set at 28x28x1, indicating the images are processed in grayscale. These layers are primarily responsible for extracting features from the input images.
- Convolution Operation:

$$Z_{i,j,k} = \sum_{m=1}^{H_f} \sum_{n=1}^{W_f} \sum_{c=1}^{C_{in}} X_{i+m-1,j+n-1,c} \cdot W_{m,n,c,k} + b_k$$

..(8)

- $Z_{i,j,k}$ prints feature map at position (i, j) for the k-th filter.
- $X_{i,j,c}$ the input at position (i, j) for the c-th channel.
- $W_{m,n,c,k}$filter weight at position (m, n) for the k-th filter and c-th channel.
- $b_k$the bias term for the k-th filter.
- This is followed by the convolution operation and activation function, usually ReLU.

Padding**:** When padding = 'same' is used, zero-padding is added to the input so that output retains the same spatial dimensions as input.

5.3 Three maxpooling layers**:** This architecture uses three maximum pooling layers that gradually reduce the size of window, 3 x 3, 2 x 2, and 2 x 2, using step 2. The "same" padding will help preserve all spatial dimensions for each layer. In this network, max-pooling layers resample feature maps by reducing spatial dimensions while preserving important features.

$$M_{i,j,k} = \max_{(m,n)\in window} A_{i+m,j+n,k}$$

..(9)

- $M_{i,j,k}$is output of max pooling layer.
- The max function chooses the maximum value within the pool_size×pool_size pooling window.

5.4 One dropout layer: Layer has a speed of 0.5; in other words, during each run or training session, 50% of neurons, randomly selected from the preceding layer, would be discarded, thus avoiding overcrowding.

5.5 One flattened layer: This layer converts the multidimensional output from the convolutional layers into 1D vector. Two dense layers: These fully connected layers handle the final classification task. The first dense layer contains 512 units with ReLU activation, and the second dense layer includes 24 units, also using ReLU activation.

A convolutional neural network is designed to process 28x28 single-channel grayscale images and output of 24-class probability distribution.

Table 2: CNN Model Summary

Layer (type)	Output Shape	Parameters #
Conv2D (Conv2D)	(28, 28, 128)	3,328
MaxPooling2D(Maxpool2D)	(14, 14, 128)	0

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Conv2D (Conv2D)	(14, 14, 64)	32,832
MaxPooling2D(Maxpool2D)	(7, 7, 64)	0
Conv2D (Conv2D)	(7, 7, 32)	8,224
MaxPooling2D(Maxpool2D)	(4, 4, 32)	0
Flatten (Flatten)	512	0
Dense (Dense)	512	524,800
Dropout (Dropout)	512	0
Dense (Dense)	24	12,312
Total Parameters		**319,352**
Trainable Parameters		**319,352**
Non-Trainable Parameters		**0**

This model is designed for image classification tasks involving a dataset of 24 potential classes, such as characters or digits. It effectively extracts characteristics from images through three convolutional layers, followed by max pooling to reduce dimensionality and computational load. The extracted features are then processed by dense layers to classify input images into one of the 24 classes. This approach helps avoid overfitting. With a total of 319,352 parameters, the model complexity is managed to effectively learn the relevant patterns in input data.

VI. RESULTS AND INTERPRETATION

A. Gesture Recognition with DNNs: Accuracy and Efficiency Results

Gesture recognition, a crucial element of human-computer interaction, has advanced considerably with the rise of deep learning, allowing machines to accurately interpret human gestures. Developing a DNN-based gesture recognition model involves multiple stages, including data collection, preprocessing, model training, and testing. This section analyzes the results and implications of a DNN-based algorithm for gesture recognition, emphasizing the efficiency of DNNs in this domain and their potential to transform industries such as healthcare and entertainment, as well as revolutionize human-computer interaction.

Tabular Result

Table 3: DNN Model Result

Metric	Value
Training Accuracy	0.9887 (~98.87%)
Training Loss	0.1100
Validation Accuracy	0.9575 (~95.75%)
Validation Loss	0.1926
Test Accuracy	96.43%

Table (3) above illustrates that the deep neural network (DNN) model, trained for 24 epochs, demonstrates strong performance in gesture recognition. Both training plus validation accuracies is impressive, at 98.87% and 95.75% respectively, while the test accuracy reaches 96.43%. These results indicate, DNN model is highly effective at recognizing gestures.

Graphical Results

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

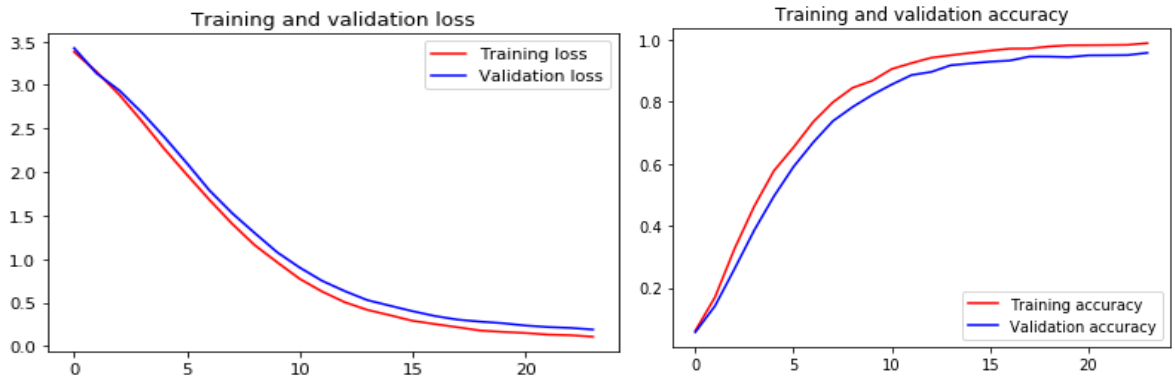


Figure 5: Graphical Result of DNN Model

The above figure (5) shows the graphical results where the DNN model was trained for 24 epochs and gave a final training accuracy of 98.87%, validation accuracy of 95.75%, and test accuracy of 96.43%. The training loss falls from 3.5 to 0.1 and the validation loss from 3 to 0.2. This result says that the DNN algorithm can recognize gestures effectively.

B. Gesture Recognition with CNNs: Efficiency and Accuracy Results

The performance of convolutional neural network (CNN) algorithm is evaluated based on the developed gesture recognition system. The model was tested with a separate test dataset, and the accuracy results reflect the system's effectiveness in recognizing various gestures.

Tabular Results

Table 4: CNN Model Result

Metrix	Value
Training Accuracy	0.9896(~98.96%)
Training Loss	0.0314
Validation Accuracy	1.0000(100%)
Validation Loss	0.0467
Test accuracy	99.40%

Table (4) above presents the training results of CNN model, which was trained for 35 epochs. The model achieved remarkable accuracy in gesture recognition, with a training accuracy of 98.96% and a training loss of 0.0314. It also performed nearly perfectly on the validation set, attaining 100% accuracy and a validation loss of 0.0467, indicating minimal overfitting. The high-test accuracy of 99.40% suggests that a model generalizes well to new, unseen data, making it a strong candidate for practical applications in gesture recognition.

Graphical Results

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

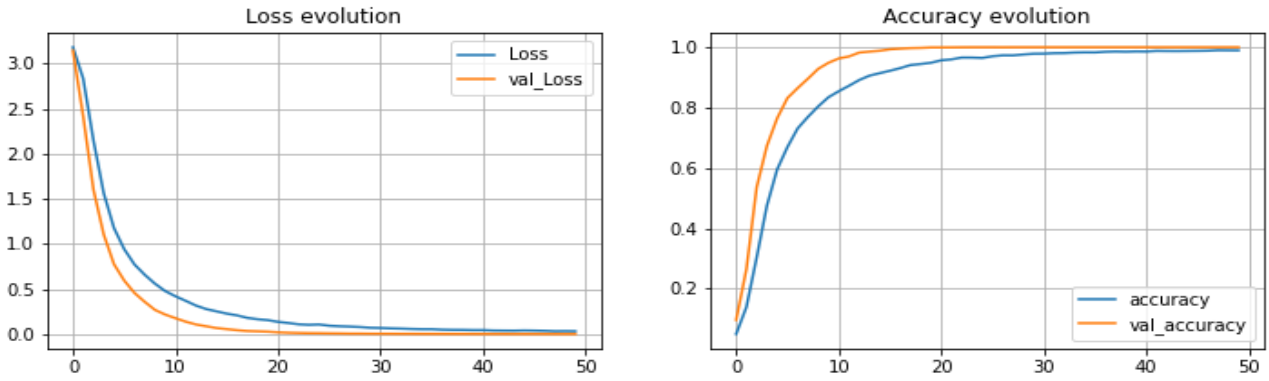
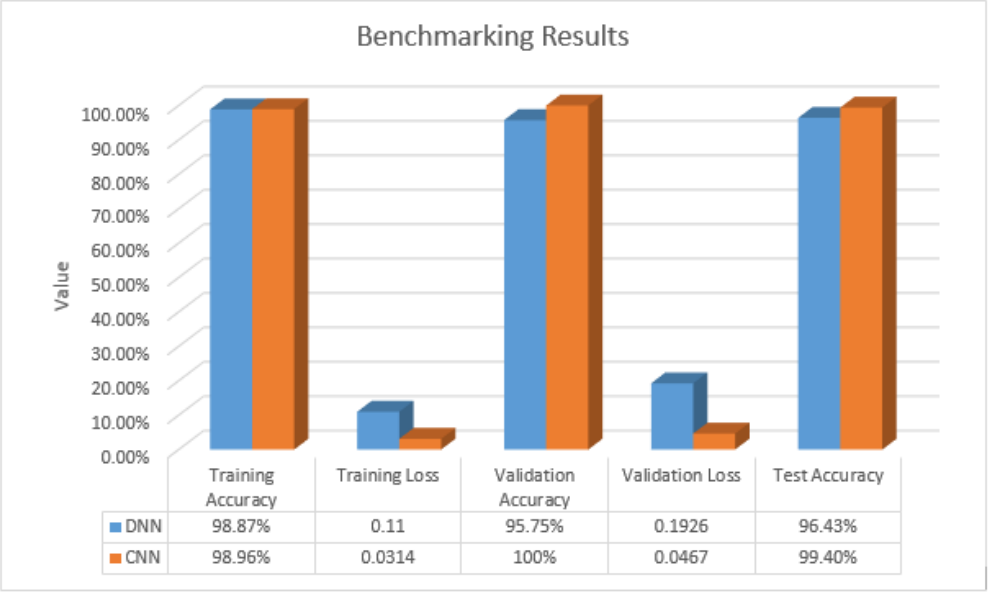


Figure 6: Graphical Result of CNN Model

Figure (6) above displays the results of CNN model for gesture recognition, which achieved training accuracy of 98.96%, validation accuracy of 100%, a validation loss of 0.0467, and a test accuracy of 99.40%. The model demonstrates strong robustness in identifying and classifying gestures with high accuracy and minimal loss, even for unseen data.

C. Comparative analysis of Deep Neural Network and Convolutional Neural Network

Popular machine learning algorithms include DNNs and CNNs. DNNs are effective for tasks involving text, speech, plus time series data, as they can learn complex patterns, though they may sometimes be slow to train and adapt [18]. In contrast, CNNs are specifically designed for image and video data, offering quick training and strong pattern recognition in images. However, CNNs are limited to image and video tasks, while DNNs are more versatile across various types of data [19] [20].



	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
DNN	98.87%	0.11	95.75%	0.1926	96.43%
CNN	98.96%	0.0314	100%	0.0467	99.40%

Figure 7: Comparative results of DNN and CNN Model

The above figure 7, a bar chart, visually compares the effectiveness of the DNN and CNN machine learning algorithms across various metrics.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The findings displayed that CNN outperforms DNN in all aspects, achieving higher training accuracy, lower validation loss, lower training loss, higher validation accuracy, lower validation loss, and significantly higher test accuracy.

These findings suggest that CNN is the superior model for this task, as it is better equipped to learn from the training set and makes accurate predictions on new data, ultimately leading to enhanced performance.

VII. CONCLUSION

The emergence of CNN and DNN algorithms in machine learning has significantly transformed human-computer interaction in AR/VR environments, particularly in gesture recognition. These algorithms have enabled the development of more efficient and accurate hand gesture recognition systems, enhancing the ability to simulate real-world interactions. The integration of CNN and DNN algorithms allows for the analysis of complex data from various sensors, leading to a better understanding of user behavior and preferences. This advancement has resulted in the creation of more user-friendly interfaces, greatly enhancing the user experience in AR/VR settings. As research progresses, the application of CNN and DNN algorithms is expected to drive further innovation and improvements in AR/VR interfaces.

REFERENCES

[1] Guo, Yanming, et al. "Deep learning for visual understanding: A review." Neurocomputing 187 (2016): 27-48.
[2] Kamilaris, Andreas, and Francesc X. Prenafeta-Boldú. "Deep learning in agriculture: A survey." Computers and electronics in agriculture 147 (2018): 70-90.
[3] Mitra, Sushmita, and Tinku Acharya. "Gesture recognition: A survey." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37.3 (2007): 311-324.
[4] Yasen, Mais, and Shaidah Jusoh. "A systematic review on hand gesture recognition techniques, challenges and applications." PeerJ Computer Science 5 (2019): e218
[5] Samek, Wojciech, et al. "Explaining deep neural networks and beyond A review of methods and applications." Proceedings of the IEEE 109.3 (2021): 247-278.
[6] Abdelhafiz, Dina, et al. "Deep convolutional neural networks for mammography: advances, challenges and applications." BMC Bioinformatics 20 (2019): 1-20.
[7] Krig, Scott, and Scott Krig. "Feature learning and deep learning architecture survey." Computer Vision Metrics: Textbook Edition (2016): 375-514.
[8] Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh, Mo Nours Arab, (2008) "Human-Computer Interaction: Overview on State of the Art", International Journal on Smart Sensing and Intelligent Systems, Vol. 1(1).
[9] Mokhtar M. Hasan, Pramoud K. Misra, (2011). "Brightness Factor Matching for Gesture Recognition System Using Scaled Normalization", International Journal of Computer Science &Information Technology (IJCSIT), Vol. 3(2).
[10] Mokhar M. Hasan, Pramod K. Mishra, (2012) "Features Fitting using Multivariate Gaussian Distribution for Hand Gesture Recognition", International Journal of Computer Science & Emerging Technologies IJCSET, Vol. 3(2).
[11] Mokhar M. Hasan, Pramod K. Mishra, (2012). "Robust Gesture Recognition Using Gaussian Distribution for Features Fitting', International Journal of Machine Learning and Computing, Vol.2 (3).
[12] V. S. Kulkarni, S.D. Lokhande, (2010) "Appearance Based Recognition of American Sign Language Using Gesture Segmentation", International Journal on Computer Science and Engineering (IJCSE), Vol. 2(3), pp. 560-565.
[13] Nakul Nagpal, Dr. Arun Mitra., Dr. Pankaj Agrawal, "Design Issue and Proposed Implementation of Communication Aid for Deaf & Dumb People", International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 3 Issue: 5, pp- 147 – 149.
[14] Chandandeep Kaur, Nivit Gill, "An Automated System for Indian Sign Language Recognition", International Journal of Advanced Research in Computer Science and Software Engineering.
[15] Neelam K. Gilorkar, Manisha M. Ingle, "Real Time Detection and Recognition of Indian and American Sign Language Using Sift", International Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 5, Issue 5, pp. 11-18, May 2014
[16] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. 2013. User-Defined Gestures for Augmented Reality. In Human-Computer Interaction – INTERACT 2013 (Berlin, Heidelberg), Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer Berlin Heidelberg, 282–299.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[17] Bantupalli, Kshitij, and Ying Xie. "American sign language recognition using deep learning and computer vision." 2018 IEEE International Conference on big data (big data). IEEE, 2018.
[18] Bouwmans, Thierry, et al. "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation." Neural Networks 117 (2019): 8-66.
[19] Li, Zewen, et al. "A survey of convolutional neural networks: analysis, applications, and prospects." IEEE transactions on neural networks and learning systems 33.12 (2021): 6999-7019.
[20] J. Shen et al., "Comparative Analysis of Performance of 1D-CNN and DNN Model in Line Loss Rate Prediction," 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 2021, pp. 450-453, doi: 10.1109/IAECST54258.2021.9695863.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details