IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 6, November 2024

# Sign Language to Text Conversion

**Annapoorna CM[1] and Mrs. Usha Sree R[2]**
Student MCA, IVth Semester[1]
Assistant Professor, Department of MCA[2]
Dayananda Sagar Academy of Technology and Management, Udaypura, Bangalore, Karnataka, India
annapoornacm115@gmail.com and ushashree-mca@dsatm.edu.in

**Abstract***: Sign language is crucial for Deaf and non-verbal individuals to communicate, yet it remains unfamiliarto many, and interpreters are often scarce. To address this challenge, we developed a real-time system that recognizes fingerspelling in ASL using neural networks. Our process begins by filtering hand images to enhance key features. These enhanced images are then classified by a CNN. This method achieves an impressive 98.00% accuracy in recognizing all 26 letters of the alphabet, making it a highly effective tool for improving communication.*

**Keywords:** Sign language recognition, text conversion, computer vision, machine learning, sensor-based systems,and human-computer interaction.

## I. INTRODUCTION

ASL is the most prevalent form of sign language in the U.S., used by around 2.5 million people in the deaf and non-verbal communities. For those who are deaf or hard of hearing, communication—typically involving spoken words, signals, behaviors, or visual cues—can be challenging because they depend on hand movements to express their ideas. Sign language functions as a visual means of communication and involves three main elements: the shape of the hands, the way the hands and fingers are oriented, and the movements made with them.

These facilitate this project's goal of developing an efficient and ideal model for recognizing fingerspelling-based hand movements in ASL. Fingerspelling is a way of gestural communication that spells out words by individual letters and is knitted to the ASL for providing names or technical terms or other words for which specific signs have not been established.

The central objective of this research therefore revolves around the identification of fingerspelling-based hand signs and the sequence linking for the formation of words. Not only does this model pose a real potential for changing the conditions of the D&M community's means of communication; it opens up a whole new realm of possibilities for the areas of communication and integration which include education, customer relation services, and health care. Based on the utilization of neural networks as the foundation of this project, the overall objective is to provide an opportunity for communication between those with and without hearing impairments to ensure that the society contains interconnectivity between the two ensuring social interaction is possible.

## II. LITERATURE SURVEY

Researchers have explored the application of CNNs for sign language recognition, focusing on their ability to learn spatial features from video frames. This approach has shown promising results in accurately identifying isolated sign gestures, demonstrating its potential for real-world applications where precise gesture recognition is essential. [1]

Vision-based recognition methods have employed techniques such as bacterial foraging optimization and Support Vector Machines (SVMs) to enhance the recognition of Indian Sign Language (ISL) alphabets and numerals. These approaches are designed to enhance the precision and dependability of identifying intricate sign gestures by employing effective feature selection and classification techniques. [2]

Systems utilizing sensory gloves and Artificial Neural Networks (ANNs) have been developed specifically for American Sign Language (ASL) word recognition. This research focuses on leveraging wearable technology to capture and interpret hand movements, aiming to create intuitive and accessible communication tools for the deaf community. [3]

DOI: 10.48175/IJARSCT-22513

ISSN 2581-9429 IJARSCT

Parallel Hidden Markov Models (HMMs) have been investigated for their ability to concurrently model different components of sign language gestures. This approach enhances the sturdiness and versatility of sign language recognition systems by effectively handling multiple aspects of gesture interpretation in real-time applications. [4]

Transition-movement models have been presented to advance large-vocabulary sign language recognition, addressing challenges in interpreting continuous signing. By refining the modelling of transitional movements between gestures, these models aim to expand the vocabulary and usability of sign language recognition systems. [5]

Depth silhouette-based systems have advanced real-time recognition of human activities, including sign language gestures. These systems utilize depth images to improve the precision and speed of gesture recognition, improving interaction and accessibility for individuals using sign language. [6]

Recent research has carried out in-depth reviews of deep learning methods used for sign language recognition. These surveys review various architectures and datasets, offering visions into the developments, challenges, and future directions of sign language recognition technologies. [7]

Continuous American Sign Language (ASL) recognition using Hidden Markov Models (HMMs) has been explored for its potential in real-time systems. This research focuses on capturing and interpreting gestures effectively, contributing to enhanced communication accessibility for the deaf and mute community. [8]

Gesture recognition for sign linguistics has been investigated using Dynamic Time Warping (DTW) and Neural Networks, aiming to improve accuracy in identifying sign gestures. This approach leverages dynamic temporal alignment techniques to enhance the robustness of sign language recognition systems. [9]

Hybrid classification models have been proposed to enhance sign language recognition by integrating multiple classification techniques. This research explores the synergies between different models to achieve higher accuracy and reliability in recognizing diverse sign gestures. [10]

## III. PROPOSED SYSTEM

Our proposed system aims to create a real-time American Sign Language (ASL) recognition tool using advanced neural network frameworks. The process starts with gathering and annotating a detailed dataset of ASL gestures, ensuring a wide range of handshapes and movements are included. Preprocessing steps, such as image normalization and noise reduction, are applied to improve gesture clarity and minimize variability. The core of our model will utilize CNNs to extract spatial features from ASL gesture images, ensuring reliable recognition despite varying lighting conditions and backgrounds. To capture the temporal aspects of continuous sign language sequences, Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks will be used, enabling smooth and accurate gesture interpretation. During training, the model will undergo supervised learning on the annotated dataset to refine performance metrics such as accuracy, precision, and recall. The evaluation will emphasize the system's recognition accuracy and real-time processing capabilities through thorough testing and validation. Our study aims to provide valuable insights into addressing challenges related to dataset creation, model training, and practical deployment, ultimately improving communication accessibility for individuals using sign language.

## IV. METHODOLOGY

### 1. Data Folder Creation

First, we set up folders to organize our data. Each letter of the ASL alphabet gets its folder for both Instructional and examination data

### 2. Data Collection and Pre-processing

We gathered images of hand gestures representing the 26 letters of the ASL alphabet from different public datasets. These images were then expanded and adjusted to create a diverse training set.

Each webcam frame captures a hand gesture, and we define the area of interest (ROI) with a blue square around it.

The image within the ROI is then filtered using a Gaussian blur to enhance features and reduce noise.

DOI: 10.48175/IJARSCT-22513
ISSN 2581-9429 IJARSCT

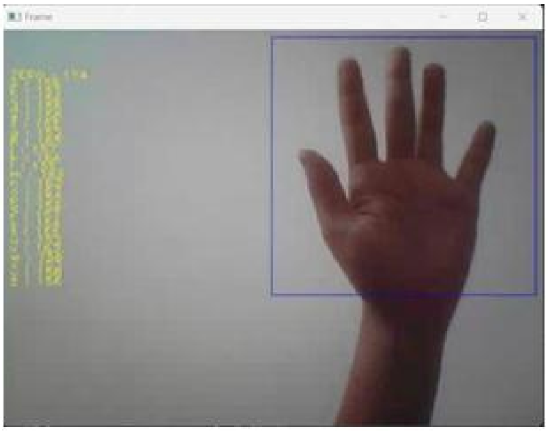


Figure 1: Training Data Collection Image

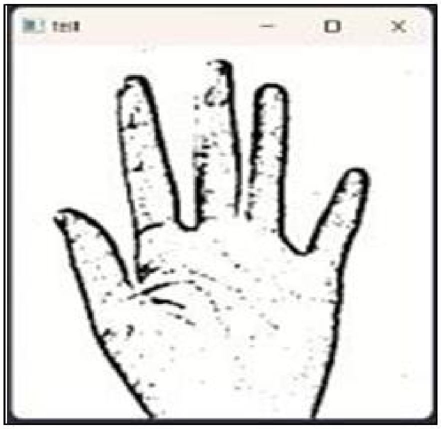


Figure 2: ROI (Region of Interest)

**a) Pre-processing Steps**

- Hand Detection: We extract the region of the image that shows the hand.
- Filtering: Techniques like Gaussian blur are used to make hand features clearer and reduce unwanted details (noise).
- Normalization: All images are reshaped and adjusted to maintain uniform dimensions and brightness levels for training.

These steps are essential for preparing our dataset to train a model capable of accurately recognizing ASL gestures.

### 3. Model Creation

**a) Convolutional Neural Network (CNN)**

CNN is specialized for understanding images by arranging their neurons in a way that replicates the image's shape: width, height, and depth. Unlike regular neural networks, CNNs don't connect every neuron to all others; instead, each layer focuses on small parts of the image at a time, using a sliding window approach to capture details. As data passes through these layers, they extract different features from simple shapes like edges to complex patterns like textures and objects.

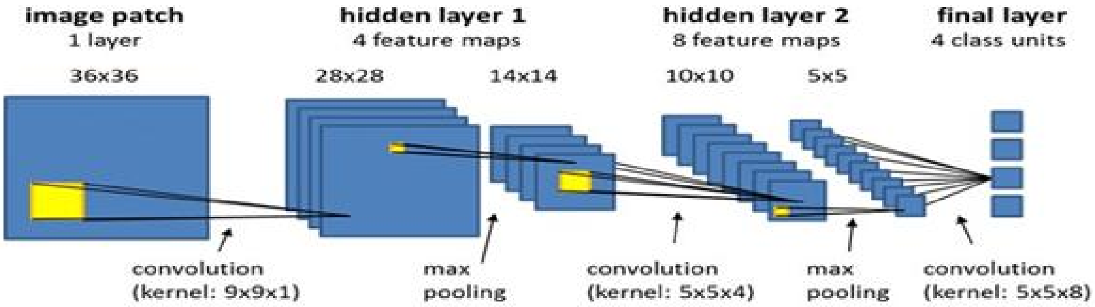


Figure 3: Model Creation

**b) Convolutional Layer:**

In a convolutional layer, we use small filters, usually 5x5 in size, that span across the entire dimensionality of the input matrix These filters are trainable, meaning the network adjusts them during training to detect specific features. As we slide these filters across the input data, typically moving them one pixel at a time (stride of 1), we compute the dot product between the filter weights and the corresponding input values at each position. This process generates a 2-dimensional activation map that shows how strongly each filter responds to different spatial locations in the input.

DOI: 10.48175/IJARSCT-22513
ISSN 2581-9429 IJARSCT

Essentially, these filters act like detectors, learning to activate strongly when they detect certain visual patterns like edges of specific orientations or patches of particular colours. Through repeated iterations and adjustments during training, the convolutional layer helps the network progressively recognize and extract meaningful features from the input data, which is essential for activities such as image recognition and classification.

**c) Pooling Layer**

Pooling layers in CNN act like filters that simplify and condense the information learned in earlier layers. They come in two main types: max pooling and average pooling.

**I. Max Pooling:**

Max pooling operates by examining small segments of the data and picking the highest value within each segment. This process reduces the data's size while retaining the most significant features. For example, in a 2x2 grid of numbers, max pooling would select the largest number among those four.

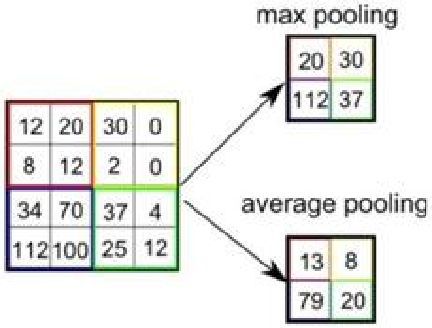


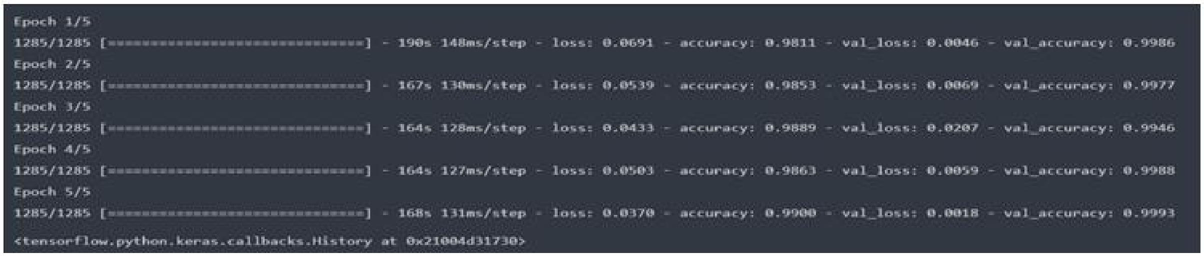
Figure 4: Pooling Layer

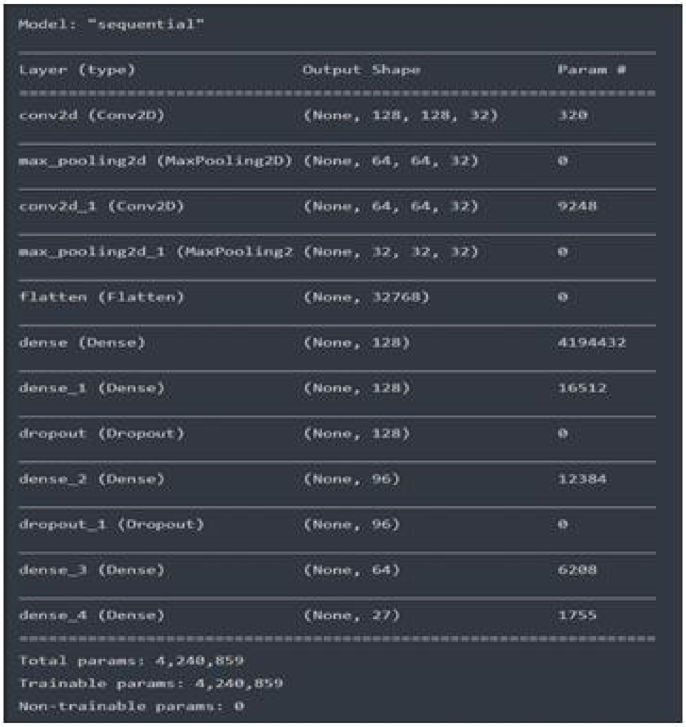
**II. Average Pooling:**

Conversely, average pooling computes the average (mean) of the values within each segment. This technique also decreases the data size, by averaging the values instead of picking the highest. Both pooling methods are employed in CNNs to streamline the data, enhancing their efficiency in identifying patterns in images and other types of data.

**d) Final Output Layer:**

Once the values are processed by the fully connected layer, they are fed into the final layer of neurons, with the number of neurons matching the total number of classes. This layer will generate predictions for the probability of each image belonging to the various classes.


```
Epoch 1/5
1285/1285 [==============================] - 190s 148ms/step - loss: 0.0691 - accuracy: 0.9811 - val_loss: 0.0046 - val_accuracy: 0.9986
Epoch 2/5
1285/1285 [==============================] - 167s 130ms/step - loss: 0.0539 - accuracy: 0.9853 - val_loss: 0.0069 - val_accuracy: 0.9977
Epoch 3/5
1285/1285 [==============================] - 164s 128ms/step - loss: 0.0433 - accuracy: 0.9889 - val_loss: 0.0207 - val_accuracy: 0.9946
Epoch 4/5
1285/1285 [==============================] - 164s 127ms/step - loss: 0.0503 - accuracy: 0.9863 - val_loss: 0.0059 - val_accuracy: 0.9988
Epoch 5/5
1285/1285 [==============================] - 168s 131ms/step - loss: 0.0370 - accuracy: 0.9900 - val_loss: 0.0018 - val_accuracy: 0.9993
<tensorflow.python.keras.callbacks.History at 0x21004d31730>
```

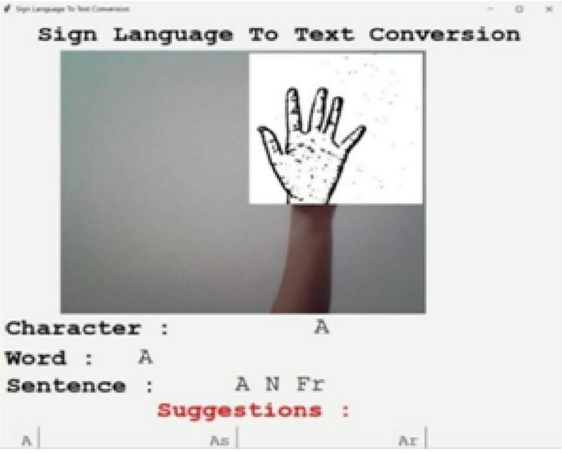


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 128, 128, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 32768)             0
_________________________________________________________________
dense (Dense)                (None, 128)               4194432
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 96)                12384
_________________________________________________________________
dropout_1 (Dropout)          (None, 96)                0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                6208
_________________________________________________________________
dense_4 (Dense)              (None, 27)                1755
=================================================================
Total params: 4,240,859
Trainable params: 4,240,859
Non-trainable params: 0
```

Figure 5: Model Summary

### 4. GUI Development

After training the model, a GUI is developed for converting signs into text, facilitating communication with D&M individuals.



Sign Language To Text Conversion

Character : A

Word : A

Sentence : A N Fr

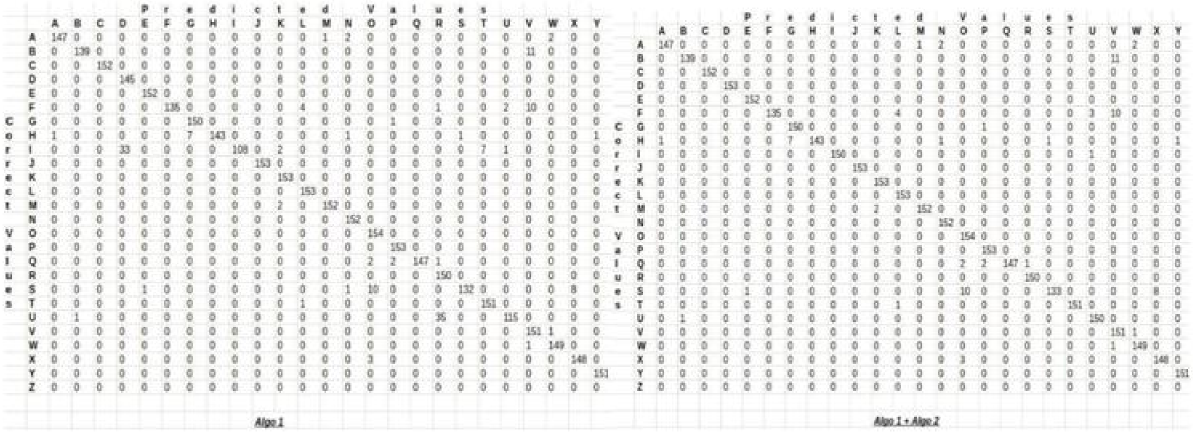
Suggestions :

Figure 6: GUI Application

## V. RESULTS AND DISCUSSIONS

### Results

The model achieved an accurateness of 95.8% with the initial algorithm and 98.0% using an improved two-layer algorithm. Below are the confusion matrices for the results:



Predicted Values

Correct Values

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 147 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| B | 0 | 139 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 | 0 |
| C | 0 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 145 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 135 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 10 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| H | 1 | 0 | 0 | 0 | 0 | 0 | 7 | 143 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| I | 0 | 0 | 0 | 33 | 0 | 0 | 0 | 0 | 108 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 1 | 0 | 0 | 0 | 0 |
| J | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| L | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| M | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| N | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| O | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 154 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Q | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 147 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 10 | 0 | 0 | 0 | 132 | 0 | 0 | 0 | 0 | 8 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 151 | 0 | 0 | 0 | 0 | 0 |
| U | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 35 | 0 | 0 | 115 | 0 | 0 | 0 | 0 |
| V | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 151 | 1 | 0 | 0 |
| W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 149 | 0 | 0 |
| X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 148 | 0 |
| Y | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 151 |
| Z | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Algo 1

Predicted Values

Correct Values

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 147 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| B | 0 | 139 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 | 0 |
| C | 0 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 135 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 10 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| H | 1 | 0 | 0 | 0 | 0 | 0 | 7 | 143 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| J | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| L | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| M | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| N | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 152 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| O | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 154 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Q | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 147 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 133 | 0 | 0 | 0 | 0 | 8 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 151 | 0 | 0 | 0 | 0 | 0 |
| U | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 150 | 0 | 0 | 0 | 0 |
| V | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 151 | 1 | 0 | 0 |
| W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 149 | 0 | 0 |
| X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 148 | 0 |
| Y | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 151 |
| Z | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Algo 1 + Algo 2

Figure 7: (A) Algo 1 (B) Algo 1+ Algo 2

**Limitations**

While the model performs well under controlled conditions, its accuracy may decrease in real-world scenarios with varying lighting conditions, backgrounds, and hand orientations. Future work will focus on addressing these challenges by incorporating more diverse drill data and advanced pre-processing techniques.

## VI. CONCLUSION

To wrap up, this paper has examined the basics of CNNs, focusing on their design and how they are used in image recognition. We covered how CNNs utilize convolutional layers to identify features like edges and textures, pooling layers to simplify and reduce the data size, and fully connected layers for classifying images. The research shows how CNNs are effective in tasks like image categorization and target detection because they can learn detailed features directly from the raw pixel data.

We also looked at how CNNs impact different areas, such as healthcare and self-driving cars, showing their wide range of applications and effectiveness. However, challenges like overfitting and high computational needs still need to be addressed in future research. Overall, this paper highlights how CNNs are changing the field of computer vision and sets the foundation for further progress in artificial intelligence and machine learning.

## REFERENCES

[1] Koller, O., et al. (2015). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding, 141, 108- 125.

[2] Ong, S. C. W., & Ranganath, S. (2005). Automatic sign language analysis: A survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 873- 891.

[3] Starner, T., & Pentland, A. (1997). Real-time American sign language recognition from video using hidden Markov models. Proceedings of International Symposium on Computer Vision.

[4] Kaur, S., & Vig, R. (2012). A comparative study of various image pre-processing techniques for hand gesture recognition. International Journal of Computer Science & Engineering Survey (IJCSES), 3(2), 67-74.

[5] Zhang, Z. (2012). Microsoft Kinect sensor and its effect. IEEE Multimedia, 19(2), 4-10.

[6] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[7] Szegedy, C., et al. (2015). Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[8] He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

DOI: 10.48175/IJARSCT-22513

ISSN 2581-9429 IJARSCT
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

[9] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[10] Chollet, F. (2017). Xception: Deep learning with depth-wise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

DOI: 10.48175/IJARSCT-22513

ISSN 2581-9429 IJARSCT