American Sign Language Recognition

Sowmya Emani
IT department
IGDTUW
Delhi, India
sowmyaemani06@gmail.com

Anavi Srivastava
CSE department
IGDTUW
Delhi, India
anavisrivastava@gmail.com

Ipshita Tandon
CSE department
IGDTUW
Delhi, India
ipshitatandon@gmail.com

***Abstract*—American Sign Language (ASL) is a vital mode of communication for individuals with hearing impairments. Advances in technology have led to the development of systems that bridge the communication gap between sign language users and the wider community. This unique and expressive language relies on handshapes, facial expressions, and body movements to convey meaning. A standard approach involves collecting a diverse dataset of ASL gestures, preprocessing the data to extract relevant features, and training a machine learning model, such as a Convolutional Neural Network (CNN), to recognize these gestures. The trained model would be capable of converting hand and finger movements captured through video recordings into text, thereby enabling seamless communication between ASL users and those unfamiliar with sign language. This paper offers insights into the technical aspects of ASL detection using machine learning and underscores its potential to revolutionise communication inclusivity for diverse communities.**

I. Introduction

In computer science, artificial intelligence (AI) plays a pivotal role by aiming to replicate human intelligence for effective problem-solving. Within the AI domain, computer vision stands as a subset, primarily concerned with the extraction of meaningful information from images. Despite its inherent complexity, computer vision finds practical applications across various sectors, including robotics, automotive, medical, mathematics, and industry. On a parallel note, there is ongoing research aimed at facilitating communication for individuals with hearing impairments who use American Sign Language (ASL). This endeavor involves recognizing and translating ASL hand gestures into real-time text, a development with direct relevance to Human-Computer Interaction (HCI).

This research utilizes advanced Computer Vision and Pattern Recognition technology to create a real-time hand gesture detection application. It operates using live camera feeds, leveraging extensive American Sign Language (ASL) datasets and Convolutional Neural Networks (CNNs) for precise gesture classification. The primary goal is accurate recognition of individual alphabet letters, promptly translated into real-time text. Additionally, the study explores machine learning techniques for enhancing communication among ASL users. This involves constructing a diverse ASL gesture dataset, extracting key features, and training machine learning models like CNNs to translate hand movements from video recordings into text. The potential societal impact is substantial, offering improved communication support, enriched educational resources, and enhanced accessibility for individuals with hearing impairments.

II. Literature Review

When we were doing our background research for implementing the model, we came across a few research papers that solidified our conviction that developing a CNN model would be the correct approach for handling this issue. We observed that Convolutional Neural Networks are more accurate. The most significant advantage of neural networks is that they learn the most important classification features. However, they require considerably high memory and time to train.

In their work, W. Huang and colleagues [1], introduced a 3D Hopfield neural network designed for tasks such as hand tracking, feature extraction, and the recognition of gestures. Their model underwent testing using a dataset containing 15 distinct hand gestures, and the results indicated an average recognition accuracy of 91%.

Admasu and Raimond [2] successfully achieved an accurate classification of Ethiopian Sign Language with a remarkable accuracy rate of 98.5%. They accomplished this using a feedforward neural network. Their approach involved extensive image preprocessing, encompassing tasks like image size normalisation, background subtraction, contrast adjustment, and image segmentation.

Starner and Pentland [3] employed a Hidden Markov Model (HMM) in conjunction with a 3-D glove capable of tracking hand movements. Owing to the glove's ability to capture three-dimensional data from the hand irrespective of its spatial orientation, the researchers achieved an outstanding accuracy rate of 99.2% when tested on their evaluation dataset.

Over the years, researchers have explored various approaches to study sign language recognition, yielding intriguing findings with significant societal benefits. Consequently, this study will concentrate on developing a sign language recognition system using the American Sign Language (ASL) dataset, and employ Convolutional Neural Networks (CNN) along with Computer Vision and Pattern Recognition technology to devise a desktop application capable of real-time detection of hand movements by using a webcam or live camera feed

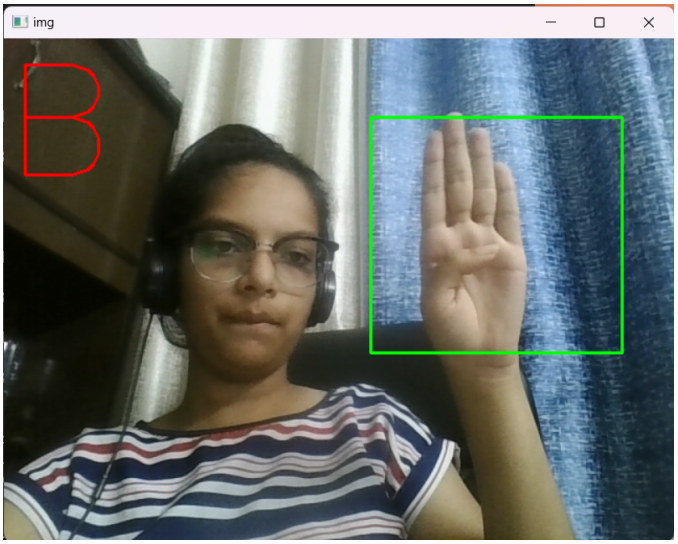

Fig. 1. Model testing using webcam

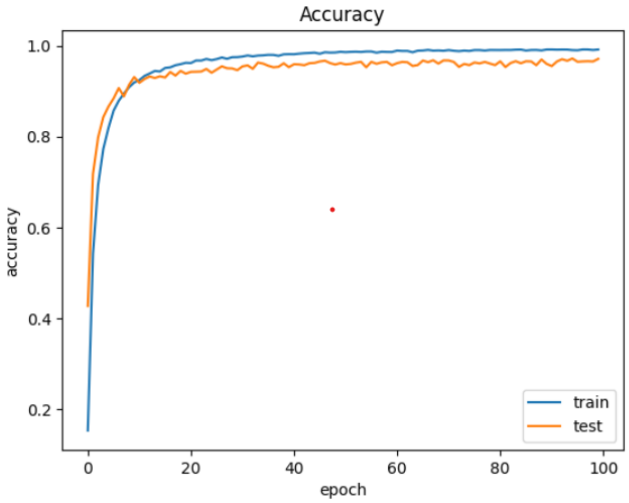

Fig. 2. Training and testing accuracy

III. Methodology

A. Data Acquisition

The first step of the proposed system is to collect data. We procured a dataset from the Kaggle website titled "Sign Language MNIST". Each training and test case represents a label (0-25) as a one-to-one map for each alphabet A-Z (it does not contain cases for 9=J or 25=Z due to gesture motions). The header row of labels, such as pixel1, pixel2 .... pixel784 depicts a 28x28 pixel image with grayscale values ranging from 0-255.

We divided the dataset using the 80:20 rule into the training and testing datasets respectively. The training dataset dimensions were 27455 rows x 785 columns whereas the testing dataset dimensions were 7172 rows x 785 columns.

We subsequently tested the hand gesture data through a webcam during the testing phase. This process involved capturing images of individuals performing American Sign Language (ASL) gestures using a camera. Many researchers had previously used sensors or cameras to capture the hand movements. For our system, we make use of the web camera to shoot the hand gestures as seen in 1. The choice of using a webcam allowed for real-time and convenient data verification.

B. Data Pre-processing

1) Data Inspection: We initiated the data analysis process by performing exploratory data analysis. This step was essential to gain insights into the dataset's structure and characteristics. Additionally, we aimed to identify any instances of missing values and duplicate samples within the dataset.

2) Data Cleaning: Following data inspection, we proceeded to cleanse the dataset. This involved two primary tasks:

- We removed missing values from the training dataset to ensure that our data was complete and ready for analysis.
- We also eliminated duplicate samples from the dataset to maintain data integrity and prevent redundancy.

3) Normalization: As a preprocessing step, we normalized the input images. This normalization was achieved by scaling the pixel values of the images to a standardized range of [0, 1]. Specifically, we divided each pixel value by 255.This scaling was performed to facilitate optimal model training and ensure consistency in the data.

4) Image Processing with OpenCV: Following data preparation and normalization, we utilized the OpenCV library for further image processing tasks. The objective was to ensure that all images were in the same format, eliminating variations between images of different gestures. This standardization process encompassed the following operations:

- Background Removal: We employed a color extraction algorithm called HSV (Hue, Saturation, Value) to detect and eliminate backgrounds from the images.
- Segmentation: Subsequently, we executed segmentation techniques to identify the regions corresponding to the skin tone within the images.
- Morphological Operations: To refine the image masks, we applied morphological operations. These operations involved the use of dilation and erosion with an elliptical kernel. The goal was to enhance the quality of the masks applied to the images.
- Image Reshaping: Finally, we amended the images to have the same size and shape. This resizing and reshaping step converted the images into a standardized 4D gray scale format suitable for deep learning purposes.

C. Dataset Preparation

1) Dataset Size: The dataset used in this project consisted of a substantial collection of American sign gesture images. This sizable dataset provided an extensive range of examples for the model to learn from, increasing its ability to recognize a wide variety of ASL gestures.

2) Data Split: To evaluate the model's performance effectively, the dataset was divided into two distinct subsets:

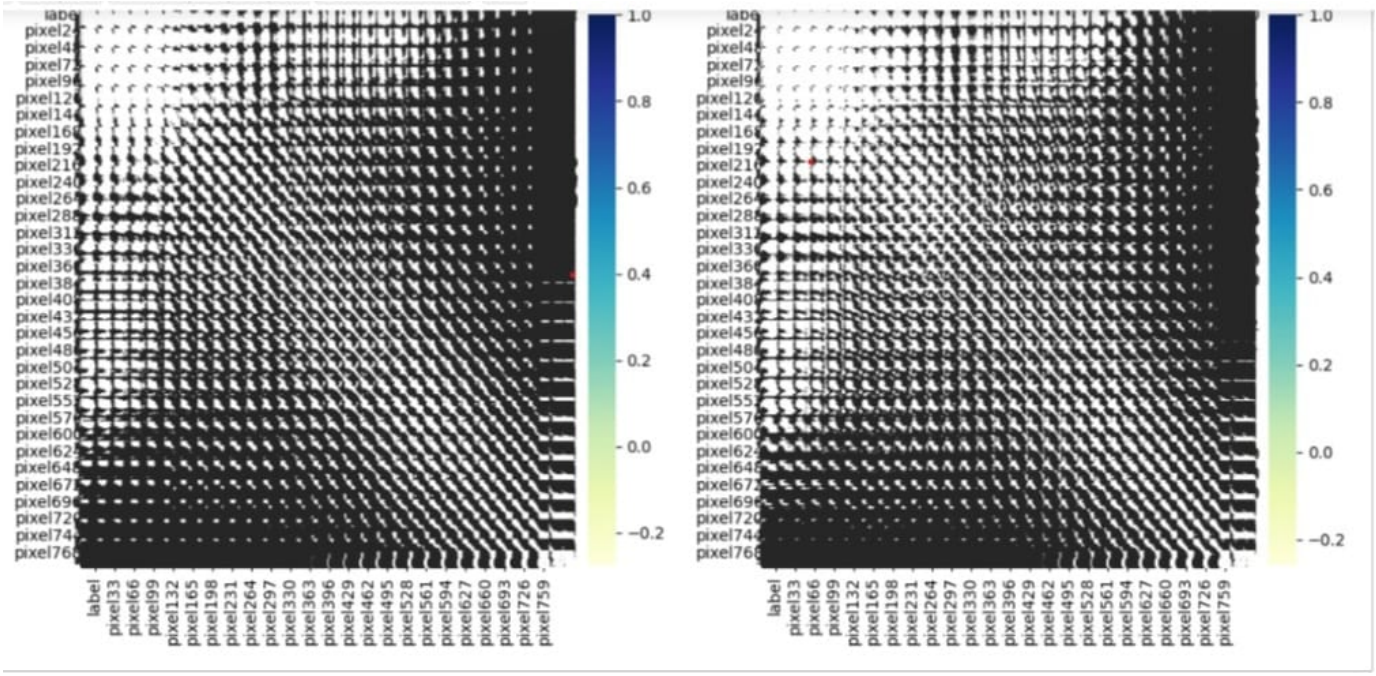

Fig. 3. Heat maps

- Training Subset (80%): This portion of the dataset was used to train the deep learning model. During the training phase, the model learned to recognize patterns and features within the ASL gesture images.
- Testing Subset (20%): The testing subset was reserved for rigorous testing and evaluation. It served as an independent dataset that the model had never seen during training. Testing on this subset provided insights into the model's generalization and predictive accuracy as seen in figure 2.

D. Data Transformation

1) Binary Pixel Values: A critical step in the project involved the extraction of binary pixel values from the preprocessed images. This binary representation converted each pixel in the images into a binary value (0 or 1) based on certain criteria. This transformation allowed the model to work with a simplified representation of the images, which is essential for deep learning.

2) Label Encoding: To represent ASL gestures in a format suitable for machine learning, a label encoding technique called Label Binarization was applied. This encoding method transformed each ASL gesture label into a binary vector. Each element of the binary vector corresponded to a specific ASL gesture class. This encoding prepared the gesture labels for further analysis and model training.

3) Normalisation: Normalisation was performed on the binary pixel values. This step involved scaling the pixel values to ensure uniform pixel intensity across all images. Normalisation is a crucial preprocessing step, as it helps the model converge faster during training and prevents certain features from dominating others.

E. Model Training

1) Feature Extraction: The heart of the process lies in training a deep learning model with 100 epochs. This model was constructed as a Convolutional Neural Network (CNN) comprising three layers. The CNN was designed to extract meaningful features from the ASL gesture images. Convolutional layers scanned the input image using small filters to detect patterns, edges, and features at different spatial scales.

2) Regularisation: Max-pooling layers were introduced to reduce spatial dimensions and abstract key features, while dropout layers served as a regularisation mechanism to prevent overfitting. These layers randomly deactivated a fraction of neurons during training, forcing the model to learn more robust and general features. This regularisation technique improved the model's ability to classify ASL gestures accurately on new, unseen data.

3) Flattening and Dense Layers: After feature extraction and regularisation, the model flattened the output from the convolutional layers into a one-dimensional vector. This step transformed the spatial information learned by the CNN into a format suitable for traditional neural network layers. Subsequently, dense layers were employed for image classification. These layers connected every neuron to every other neuron in adjacent layers, allowing the model to learn complex relationships between the extracted features. The combination of flattening and dense layers served as the bridge between the feature extraction and classification stages.

4) Softmax Activation: In the final dense layer, the softmax activation function was used. Softmax is a powerful activation function for multi-class classification tasks. It took the output of the previous layer and normalised them into a probability

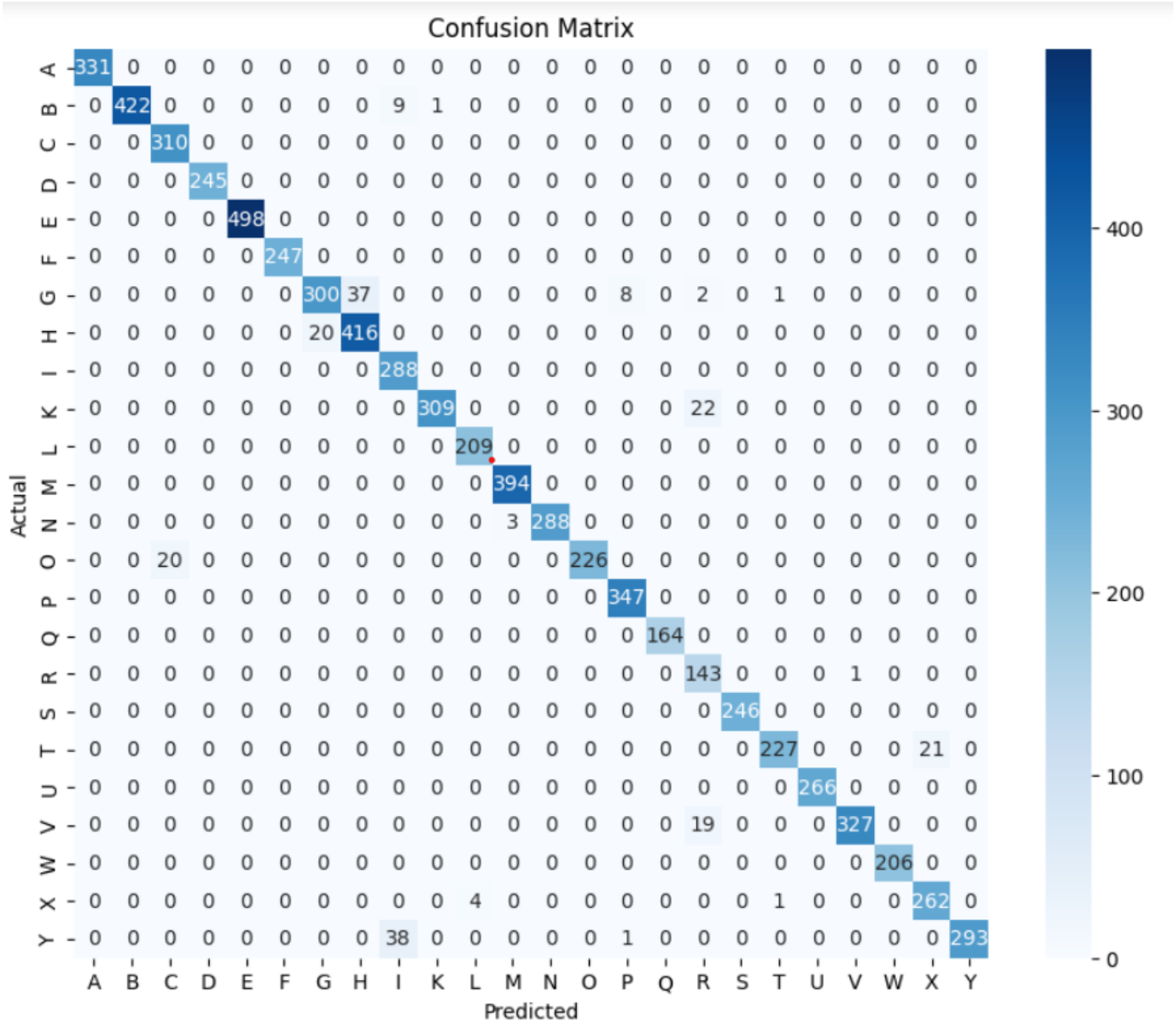

Fig. 4. Confusion matrix

distribution over multiple classes. Each class received a probability score, and the class with the highest probability was predicted as the output. This was particularly useful for ASL gesture classification because it provided a clear and intuitive way to determine which sign or letter the model predicted based on the loaded model's output probabilities.

IV. Classification and Prediction

When classifying new ASL gesture images using our trained model, we passed the image through a custom function. This function preprocessed the image by resizing it to a consistent 28x28 pixel size, normalising pixel values, and converting it into a NumPy array. Additionally, a batch dimension was added to the image to match the input format expected by the model. The model then predicted the class index for the input image using its learned weights and the softmax activation in the final layer.

After we updated the model, incorporated new datasets, and completed the training, we decided to evaluate its performance using a live webcam application. In this application, we used various libraries, including TensorFlow, Keras, NumPy, and OpenCV, to assist in the image classification process. When the camera captured a sign language gesture, the application processed the image by sending it through our trained model. Finally, the corresponding letter was retrieved from an alphabet list based on the predicted class index, providing an intuitive means to classify ASL gestures.We create a confusion matrix to summarise the performance of our model more graphically.

The accuracy resulting from the confusion matrix is 98%. We have provided an image of the confusion matrix for our results in figure 4. We have also provided a visualisation of the confusion matrix in the form of heat maps as shown in figure 3

V. Results and Conclusion

Several researches have been conducted on ASL gesture recognition using different techniques like Artificial neural networks (ANNs), Long short-term memory(LSTM) and 3D Convolutional Neural Network (CNN). However, most of them require extra computing power. On the other hand, our research paper requires low computing power and gives a remarkable accuracy. In our research, we first took a dataset from Kaggle and normalized and rescaled our images to 28x28 pixels in order to extract binary pixels and make the system more robust. We then succesfully created and trained a new model using the two layers of Convolutional Neural Network (CNN) to classify the alphabetical sign gestures and successfully achieved an accuracy of 98% demonstrating its potential in bridging the communication gap for the deaf and impaired hearing communities. Thus, our approach contributes to the field of assistive technology and inclusive communication.

VI. Future Scope

We can enhance the quality of our research by broadening our dataset to incorporate a larger quantity of distorted images, even covering variations in lighting and other factors. We can also include hand gesture movements for comprehensive word and sentence detection in ASL to empower our model further. This ambitious effort reflects our commitment to advancing ASL recognition for greater inclusivity and accessibility. There is also scope of merging speech recognition to create a comprehensive communication solution for the deaf and blind communities.

References

[1] C.-L. Huang and W.-Y. Huang, "Sign language recognition using model based tracking and a 3d hopfield neural network," Machine vision and applications, vol. 10, no. 5-6, pp. 292–307, 1998.

[2] Y.F. Admasu, and K. Raimond, Ethiopian sign language recognition using artificial neural network. 10th International Conference on Intelligent Systems Design and Applications, 2010. 995-1000.

[3] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," IEEE Transactions on pattern analysis and machine intelligence, vol. 20, no. 12, pp. 1371–1375, 1998.

[4] Obi, Yulius & Claudio, Kent & Budiman, Vetri & Achmad, Said. (2023). "Sign language recognition system for communicating to people with disabilities", Procedia Computer Science. 216. 13-20.

[5] Ahmed KASAPBAŞI, Ahmed Eltayeb AHMED ELBUSHRA, Omar AL-HARDANEE, Arif YILMAZ, "DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals", Computer Methods and Programs in Biomedicine Update, Volume 2, 2022, 100048, ISSN 2666-9900.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105

[7] B. Garcia and S. A. Viesca, "Real-time american sign language recognition with convolutional neural networks," Convolutional Neural Networks for Visual Recognition, vol. 2, 2016.