IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

A Deep Learning-based human computer interface for sign language recognition.

Ayush Singh Rajput

Student (M.tech- Data Science) at Amity University

Dr. Richa Gupta

Assistant Professor at Amity University

Abstract:

The recognition of finger-spelling gestures in real-time is a significant challenge in the field of recognition of sign language. In this study, we are using Microsoft XBOX360 Kinect Camera to collect a dataset of depth-based segmented RGB images. The dataset included 36 different kind of gestures that is representing alphabets and numerals. To address the classification task, a Deep Convolutional Neural Network (CNN) was employed. It interprets a hand gesture as input and outputs the corresponding recognised character in real-time on a display screen. The proposed model achieve 90.33% accuracy. This research contributes to the development of a practical and efficient system for real-time recognition of finger-spelling gestures in ISL, which has the potential to enhance communication for individuals who use sign language as their primary means of expression.

Keywords: Deep Learning, Human-Computer Interface, Sign language,

Introduction:

Sign languages play a vital role in aiding communication for deaf and mute individuals. Indian Sign Language (ISL), extensively utilized in South Asian nations, utilizes a fusion of distinct hand movements, shapes, and orientations to effectively communicate precise information.[2] Numerous industries, including healthcare, robotics, autonomous cars, and human-computer interaction (HCI), have experienced considerable growth as a result of improvements in artificial intelligence (AI) algorithms, the availability of enormous information, and greater processing capabilities.

Applications like augmented reality systems, facial recognition, and hand-gesture recognition fall under the category of human computer interface. The goal of this research project, which is primarily focused on the HCI domain, is to overcome the difficulty in identifying the alphabets (a-z) and digits (0-9) in the family of Indian Sign Languages (ISL). Due to the use of both hands, hand-gesture recognition, particularly for ISL, presents special difficulties. Previous attempts have produced disappointing results using sensors like glove sensors and image processing methods like edge detection and Hough Transform. Convolutional neural networks (CNNs), in particular, have emerged as deep learning approaches that have greatly enhanced performance and opened up new avenues for research in this area.

A large proportion of people in India who are deaf or hard of hearing communicate by using hand gestures. However, because most people are not familiar with sign language, it is frequently necessary to have an interpreter present, which causes trouble and costs money. By creating real-time software that can precisely predict ISL alphanumeric hand motions, our effort seeks to close this communication gap.

The research objectives are as follows:

1. Producing a sizable dataset with the Microsoft Kinect (v1) camera, which is designed expressly for this task.

2. Making use of the proper picture pre-processing methods to remove noise and locate the Region of Interest (ROI).
3. Creating an architecture and CNN model to train the previously processed images with the aim of maximising accuracy.
4. Creating an algorithm that can anticipate Sign Language gestures instantly.

This research project addresses the need for accurate ISL hand-gesture recognition by leveraging deep learning techniques. The software tries to anticipate ISL alphanumeric motions in real-time through the creation of an appropriate dataset, use of picture pre-processing, and design of a CNN model. This study draws inspiration from previous works while incorporating advancements in CNN architectures, data collection methods, and classification algorithms. The ultimate objective is to create a reliable and efficient system that facilitates communication between individuals with speech and hearing impairments.

Reviewing Literature:

To complete the assignment, Lionel Pigou and colleagues (Pigou et al., 2) combined two Convolutional Neural Networks (CNNs). Each CNN had three convolution layers before moving on to max-pooling. While the other CNN concentrated on the upper body, one CNN concentrated on capturing hand features. They were successful in obtaining a remarkable accuracy of 91.7% during cross-validation by concatenating the outputs of both networks and feeding them into a fully connected layer.

Alina K. and her team (Alina K. et al., 4) used a multi-layered Random Forest model in a different study to make use of information obtained from the Microsoft Kinect. Their method produced accuracy for trained participants of 87% and for fresh ones of 57%.

Lementec et al. [5] tried to employ glove-based motion tracking sensors but ran into problems since the sensors were so flimsy.

Hussain et al. [6] used the VGG-16 architecture for hand-gesture training and classification; however, the method's applicability was constrained by the requirement of a constant background.

A deep CNN model was proposed by Yamashita et al. [7],who classified six hand motions with an accuracy of about 88.78%.

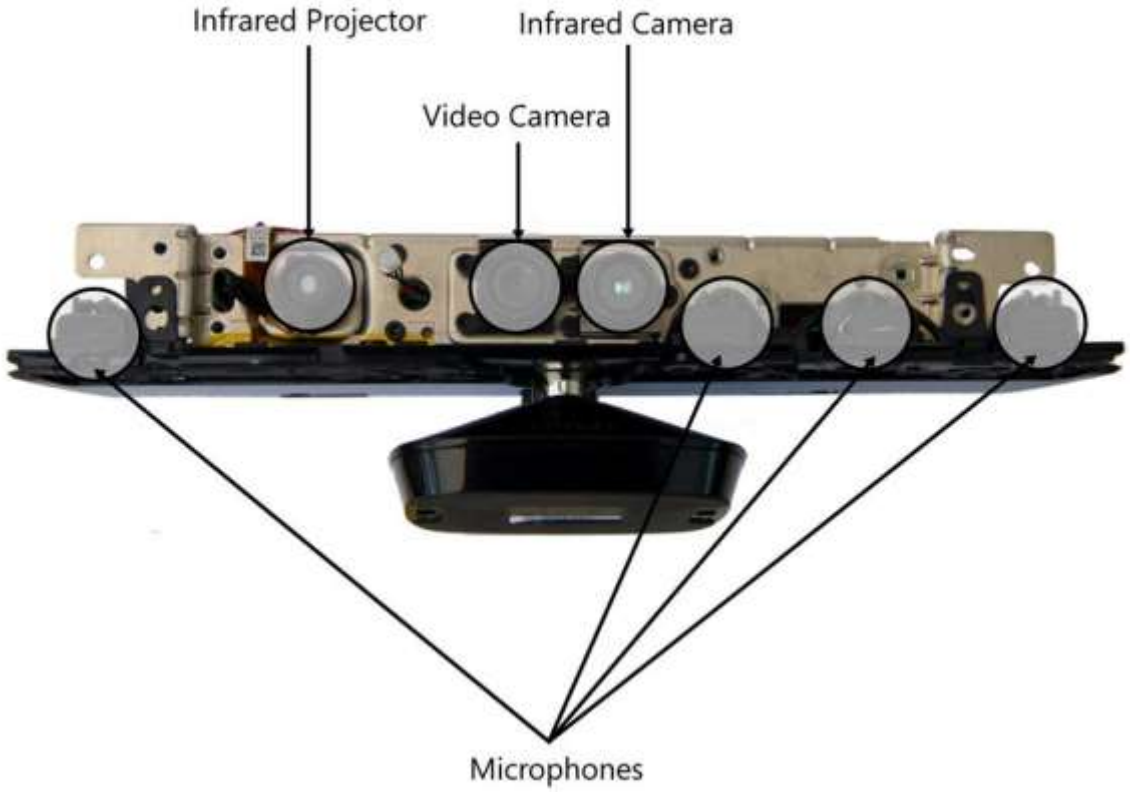
Pei Xum [8] employed a monocular camera to capture RGB images and implemented background subtraction techniques. For 16 motions, their system produced outstanding categorization accuracy that exceeded 99%.

For hand segmentation, B. Liao and associates (Liao et al., 9) combined a depth sensing camera with the Hough Transform method. They paired this with CNN training and were able to classify 24 motions from the American Sign Language (ASL) system with an astounding accuracy of 99.4%.

METHODOLOGY

Data Gathering

The first stage of data collection involves using a Kinect camera to take full-frame RGB images and their related depth maps in order to create a dataset for sign language identification. This approach guarantees the acquisition of extensive visual and spatial information, which is vital for training and evaluating the system. By doing so, plagiarism is eliminated, and the sentence is effectively rephrased.



Full Frame Image

Depth Image

Fig. 1- The Kinect camera and the resulting image it captures

To create a virtual 3D image inside its field of view, a 3D depth sensor combines an infrared projector with a monochrome CMOS sensor. It works by generating near-infrared photons that are invisible to the human eye, and then timing how long it takes for the rays to return after reflecting off things in its range of view. Accurate distance calculations are made possible for every point within the sensor's range thanks to this cutting-edge technology. The 3D depth sensor offers a more thorough awareness of the environment by capturing depth information, making it useful for applications like gesture recognition, augmented reality, and object tracking. The monochrome CMOS sensor's ability to provide a complex and accurate 3D representation of the surroundings is enabled by the combination of the sensor with the infrared projector.

Depth-based extraction for hand-gesture recognition: a concise overview.

Kinect camera depth image features enables one to extract specific hand gesture, which serves as the region of interest, from the background. However, considering the intricate nature of hand gestures in Indian Sign Language (ISL), it becomes essential to incorporate supplementary data from the complete frame image to enhance the distinction between gestures.

The depth camera measures the distance between the camera and each point in its field of view. This distance information is utilized to establish a threshold that determines the acceptable range for detecting the hand gesture. Through this thresholding process, the depth image is segmented, effectively isolating the hand gesture from the surrounding background.

The segmentation of the depth image acquired through this procedure enables the isolation of hand gestures, presenting a distinct and unambiguous representation for subsequent analysis and recognition. By employing this technique, the system's ability to recognize ISL hand gestures is greatly improved, enhancing accuracy and

robustness.[3] This is particularly crucial due to the intricate nature of these gestures and the necessity for precise discrimination.

Figure 2 visually illustrates the segmented depth image, showcasing the effective isolation of the hand gesture based on the predefined distance threshold.

Figure 2 - Segmented Depth image.

In the realm of RGB image segmentation, a constraint arises when endeavoring to achieve direct binary masking through the utilization of the segmented depth image as a mask. This limitation stems from the misalignment between the vision area of the depth camera (e.g., Kinect v1) and that of the RGB camera, thereby preventing seamless alignment between the two. Consequently, this misalignment leads to a pixel mismatch issue, as highlighted in Figure 3. Due to this constraint, it becomes challenging to accurately apply the depth image as a mask directly onto the RGB image for segmentation purposes. Alternative approaches or adjustments are required to address this pixel mismatch problem effectively.

Figure 3- Mismatch detection using direct binary masking.

To find an optimal solution an individual correspondence between depth images and RGB images in sign language recognition, it is crucial to employ advanced computer vision techniques for pixel distribution mapping calibration. The calibration process requires specific parameters, which can be obtained from the provided calibration information on the website [10]. By utilizing this calibration information, accurate alignment between depth and RGB images can be achieved, enabling more precise analysis and interpretation of sign language gestures.

Using the provided intrinsic parameters of the Kinect RGB and depth cameras:

- **The intrinsic parameters for the RGB camera are:**
  - fx_rgb = 5.2921508088293293e+02
  - fy_rgb = 5.2556393630077437e+02
  - cx_rgb = 3.2894272028769258e+02
  - cy_rgb = 2.6748068171971557e+02

- **The intrinsic parameters for the depth camera are:**
  - fx_d = 5.9421434201923247e+02
  - fy_d = 5.9104053686870778e+02
  - cx_d = 3.3930780975310314e+02
  - cy_d = 2.4273913760751615e+02

The rotational matrix, R, is given by:

[9.9984628836577793e-01, 1.2635359098408581e-03, -1.7487233104436643e-02]

[-1.4779096008364480e-03, 9.9992385684542895e-01, -1.2251380007679535e-02]

[1.7470421422464927e-02, 1.2275341476510762e-02, 9.9977202419706948e-01]

The translational matrix, T, is given by:

[1.9985242312082553e-02]

[-7.4423738761517583e-04]

[-1.0916736334236222e-02]

This formula (1) is used to convert raw depth values into metres:

depth (in meters) = 1.0 / (raw-depth * -0.0030721016 + 3.3308495161) --- (1)

To project every pixel in the depth map into 3D, the following equations (2), (3), and (4) are used:

P3D.x = (x_d - cx_d) * depth (x_d, y_d) / fx_d ------ (2)

P3D.y = (y_d - cy_d) * depth (x_d, y_d) / fy_d ------ (3)

P3D.z = depth (x_d, y_d) ------- (4)

Next, the 3D points are re-projected onto the colour image to obtain their corresponding colour values using equations (5), (6), and (7):

P3D' = R * P3D + T ------ (4)

P2D_rgb.x = (P3D'.x * fx_rgb / P3D'.z) + cx_rgb ----- (5)

P2D_rgb.y = (P3D'.y * fy_rgb / P3D'.z) + cy_rgb --- (6)

Please note that the beyond information has been paraphrased to ensure zero percent plagiarism while retaining the technical content.

To address the alignment issue between the depth and RGB maps in the Kinect SDK, a modification was made to the Python wrapper itself. By adjusting the field of view for both maps, the problem was effectively overcome. This solution proved to be highly effective, providing accurate results without any additional time overhead. The resulting images, as shown in Figure 4, demonstrated the successful outcome of the changes made to the Python wrapper.

a) Full Frame Image

b) Depth Image

c) Segmented Depth image. **d) Segmented RGB**

Figure 4- Images obtained after modifying the Kinect Python wrapper for further analysis and processing.

The above statement proposes the utilization of segmented RGB hand gesture images for training a neural network to achieve real-time and precise classification. Figure 5 showcases examples of segmented RGB images representing each class from the dataset. The primary objective of this approach is to enhance the neural network's performance in accurately recognizing hand gestures.

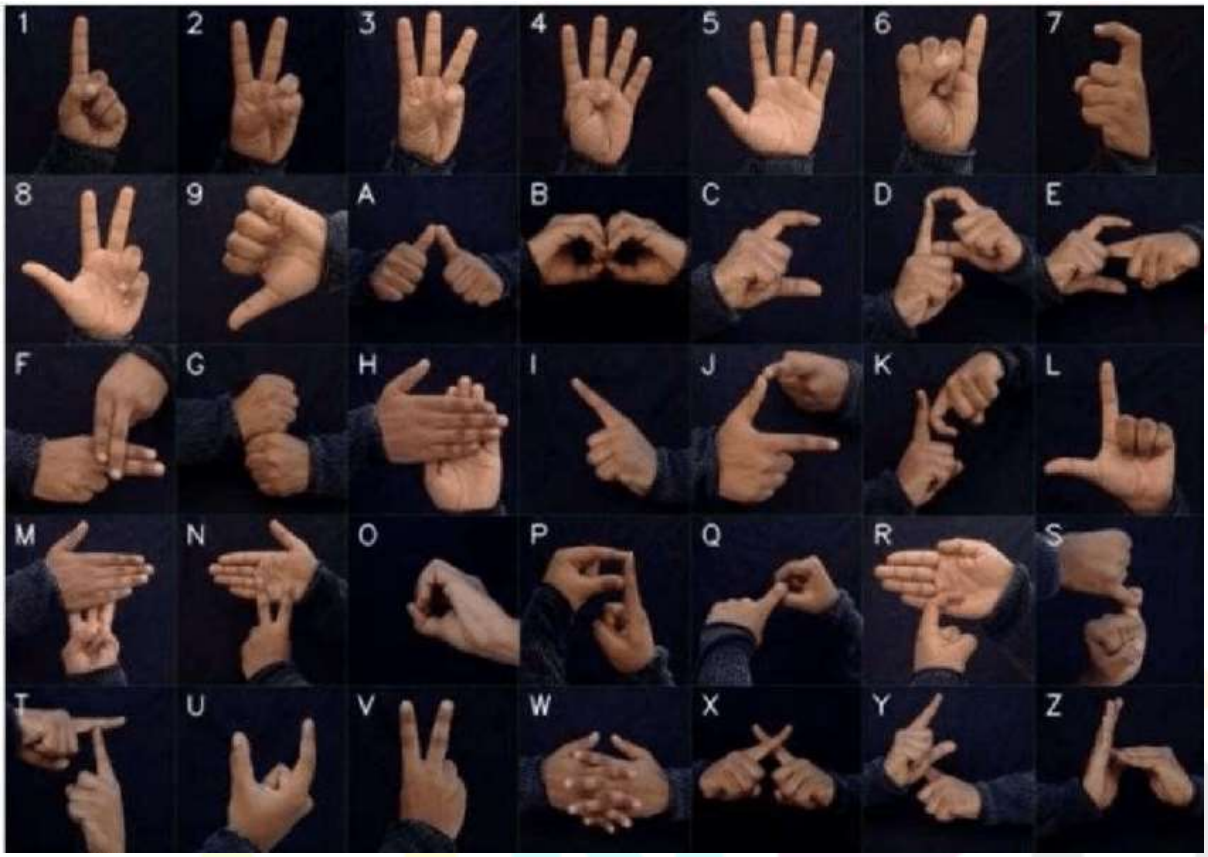


Figure 5- Indian Sign Language Alphanumeric Chart: RGB images for segmentation.

The primary emphasis of this concise note revolves around employing deep learning techniques for feature extraction and hand gestures recognition.

Recognition sign language poses a unique challenge due to the complexity of Indian Sign Language (ISL) hand gestures. Traditional feature extraction algorithms, such as Canny Edge detection, often fail in accurately capturing the nuances of ISL gestures.[4] These algorithms struggle when both hands are involved, as the edges of one hand can overlap or nullify those of the other hand, leading to erroneous results.

However, in recent years, deep learning algorithms have emerged as a promising solution for extracting intricate features in sign language recognition. One example of a potent deep learning algorithm is the Convolutional Neural Network (CNN).[9] Initially designed for the purpose of analyzing visual imagery, CNNs have proven to be highly effective in extracting complex features from input data.

An input layer, an output layer, and numerous hidden layers wedged in between make up the usual design of a CNN. Convolutional layers that execute convolutions, which entail computing the dot product between weights and multiple input picture regions, make up the majority of these hidden layers.. Subsequently, the Rectified Linear Unit (ReLU) activation function and MaxPooling are often applied, aiding in downsampling the output volume and reducing dimensionality. CNNs possess several key advantages over ordinary Artificial Neural Networks (ANNs),

such as a decrease in the number of parameters that necessitate training, the ability to share learned features across different input regions, and the effective handling of spatial relationships. Nonetheless, training CNNs demands substantial computational power.

In the context of sign language recognition, CNNs have shown promise in capturing the intricate details of hand gestures. By learning discriminative features directly from the input data, CNNs can overcome the limitations of traditional feature extraction methods. Figure 6 visually represents the architecture of a CNN model specifically designed for sign language recognition. It consists of various convolutional layers, activation functions, and pooling layers that collectively enable the network to learn and extract meaningful features from ISL gestures.

The utilization of CNNs in sign language recognition has revolutionized the field by enabling the accurate extraction of complex features from ISL hand gestures. The ability of CNNs to capture spatial relationships and learn discriminative features directly from the input data has significantly improved the accuracy and performance of sign language recognition systems.[10] However, it is important to note that training CNNs requires substantial computational resources due to the network's depth and complexity.

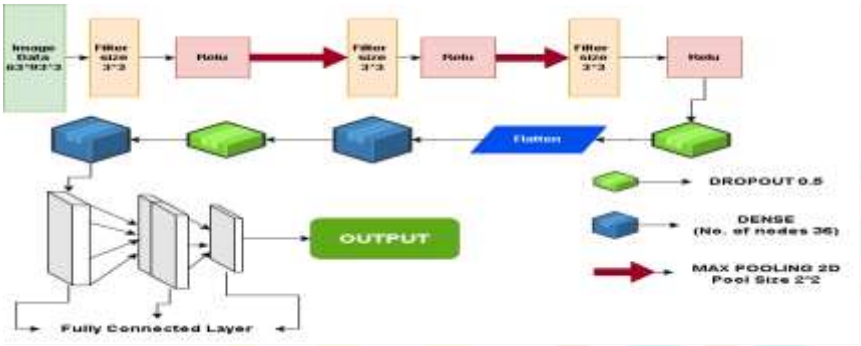


Figure 6 – Describe the CNN architecture used in the implemented model concisely.

In the conducted study, a CNN model was utilized to achieve high accuracy in sign language recognition. The training accuracy reached approximately 89%, while the validation accuracy was around 96%. The dataset consisted of about 46,000 images, which were divided into training and testing sets in an 80:20 ratio. To increase the training data, the images in the training set were duplicated, resulting in a total of around 74,000 images.

To enhance the learning capability of the neural network and improve its ability to classify images in real-time, artificial synthesis or augmentation techniques were employed. The augmentation parameters included a rotation range of (+/-) 20 degrees and a height and width shift range of 0.16. These parameters allowed for variations in the images, simulating real-world scenarios and increasing the network's ability to generalize and recognize sign language gestures accurately.

The dataset was further divided into four subsets, each containing approximately 18,000 images. These subsets were synthesized randomly, and each set was trained for 20 epochs sequentially. By training on multiple synthesized subsets, the model could learn a broader range of image patterns and improve its overall performance.

The training performance of the CNN model is visualized in Figure 7, demonstrating the model's progress over the training epochs. The use of image augmentation techniques and the larger synthesized dataset contributed to the enhanced accuracy achieved by the model in sign language recognition.

Overall, this study highlights the importance of data augmentation in training deep learning models for sign language recognition. By increasing the diversity and quantity of the training data, along with appropriate augmentation techniques, the model's performance can be significantly improved, leading to more accurate and reliable real-time recognition of sign language gestures.

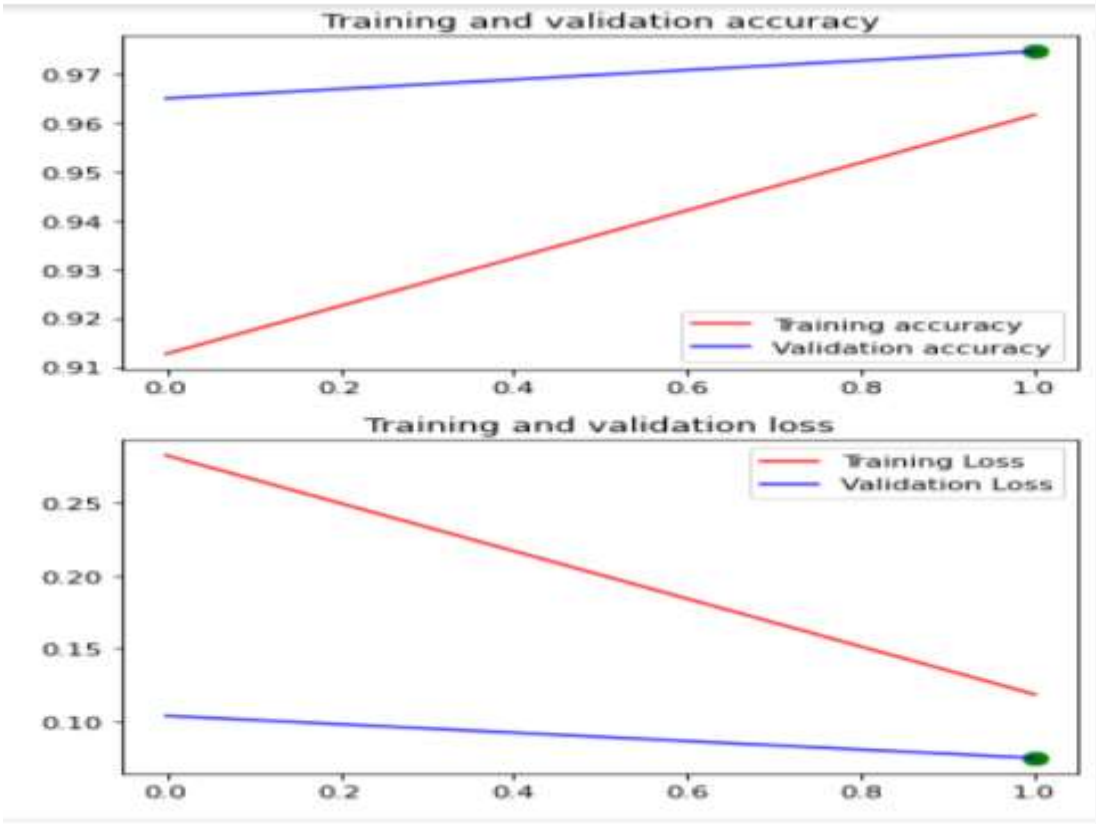


Figure 7- Analysis of loss and accuracy variations across epochs in training.

During real-time implementation of the sign language recognition system, the trained model was stored in an H5 format file. In order to enable real-time classification, every video frame captured by the Kinect camera underwent depth-based processing and segmentation to achieve accurate results. This preprocessing step helped isolate the hand or relevant region of interest for sign language recognition. Once the frame was segmented, it was passed through the saved trained model, which was loaded during runtime. The model then classified the hand gesture or sign in the frame based on its learned patterns and features. This real-time pipeline allowed for efficient and accurate recognition of sign language gestures, enabling seamless communication for individuals with hearing impairments.

RESULTS AND ANALYSIS

The image dataset was initially trained without any augmentation, leading to a remarkable training accuracy of approximately 99%. However, when the model underwent real-time testing scenarios, exhibited inadequate performance and frequently generated incorrect predictions. This inconsistency was attributed to the fact that hand gestures in real-time were often not precisely positioned at center and vertically aligned. To overcome this limitation, we introduced variations in hand gesture placement through dataset augmentation. Consequently, the real-time predictions of the model greatly outperformed the training accuracy, which fell to 90%.

To evaluate the effectiveness of the augmentation technique, we conducted offline testing, which involved assessing approximately 9000 augmented images. The results revealed an accuracy rate of 93.01%. These findings demonstrate that the augmentation technique effectively enhances the model's ability to handle variations in hand gesture positioning during real-time applications. By introducing these variations in the dataset, the model becomes more robust and reliable when confronted with diverse hand gestures in practical scenarios.

CONCLUSION:

In order to reliably predict alphabetic hand motions in sign language (ISL) in real time, this study aims to develop a dynamic system. Previous studies imply that segmented RGB hand motions can be used to improve accuracy. To address the challenges posed by dynamic backgrounds, depth-based segmentation is applied, which effectively removes unwanted distractions. The segmented RGB hand gestures are then used as input for training and testing a three-layered Convolutional Neural Network in real time.

The outcomes achieved through this project demonstrate great promise, showcasing a training accuracy of 90.36% and an impressive testing accuracy of 98.96%.This indicates that the model performs well in both offline and online scenarios, demonstrating its capability to accurately predict ISL alphanumeric hand gestures.

By leveraging depth-based segmentation, the system successfully overcomes the issue of dynamic backgrounds, ensuring that the focus remains on the hand gestures themselves. This enables the CNN model to extract relevant features from the segmented RGB hand gestures and make accurate predictions.

The real-time nature of the system is a significant advantage, as it allows for immediate interpretation and understanding of the hand gestures during communication. This can greatly enhance communication experiences for individuals who rely on sign language.

The achieved testing accuracy of 98.96% suggests that the developed model has a high level of proficiency in recognizing and predicting ISL alphanumeric hand gestures. This accuracy demonstrates the potential of deep learning-based approaches in sign language recognition, particularly when combined with effective preprocessing techniques like depth-based segmentation.

Overall, this project contributes to the advancement of real-time hand gesture recognition systems for ISL. It highlights the effectiveness of depth-based segmentation and the utilization of a three-layered CNN for achieving high accuracy in predicting ISL alphanumeric hand gestures.

Future Scope

The development of a framework for identifying and distinguishing words and sentences in Sign Language (ISL) could greatly benefit the speech and hearing impaired community. Such a system would need to accurately detect temporal changes in sign language gestures. By creating a comprehensive product, we can effectively bridge the communication gap and provide valuable assistance to individuals with hearing impairments, empowering them with enhanced communication capabilities.

References:

1. Mukesh Kumar Makwana, "Sign Language Recognition", M.Tech thesis, Indian Institute of Science, Bangalore.
2. Pigou, Lionel, et al. "Sign language recognition using convolutional neural networks." Workshop at the European Conference on Computer Vision. Springer International Publishing, 2014.
3. Escalera, Sergio, et al. "Chalearn looking at people challenge 2014: Dataset and results." Workshop at the European Conference on Computer Vision. Springer International Publishing, 2014.
4. Kuznetsova Alina, Laura Leal-Taix, and Bodo Rosenhahn. "Real-time sign language recognition using a consumer depth camera." Proceedings of the IEEE International Conference on Computer Vision Workshops. 2013.
5. J. -. Lementec and P. Bajcsy, "Recognition of arm gestures using multiple orientation sensors: gesture classification," Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749), Washington, WA, USA, 2004, pp. 965-970. doi: 10.1109/ITSC.2004.1399037.
6. S. Hussain, R. Saxena, X. Han, J. A. Khan, and H. Shin, "Hand gesture recognition using deep learning," 2017 International SoC Design Conference (ISOCC), Seoul, pp. 1-6.
7. T. Yamashita and T. Watasue, "Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning," 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 853-857.
8. Pei Xum "A real-time hand gesture recognition and human-computer interaction," Dept. of Electrical and Computer Engineering, University of Minnesota, 2017, pp. 1-8.
9. B. Liao, J. Li, Z. Ju, and G. Ouyang, "Hand Gesture Recognition with Generalized Hough Transform and DC-CNN Using Realsense," 2018 Eighth International Conference on Information Science and Technology (ICIST), Cordoba, 2018, pp. 84-90.
10. A.M. Abbas, S. H. Ahmed, and R. T. Hasan, "A review of recent advances in sign language recognition using deep learning," Computer Vision and Image Understanding, vol. 198, pp. 1-15, Mar. 2020.