Sign Language Detection and Translation: Text and Speech using CNN, NLP

***Abstract* —** People who struggle with speech impairment often rely on sign language for communication since they can't use hearing and speech effectively. However, conversing with those who don't understand sign language can be challenging for them. This underscores the necessity for sign language interpreters to bridge this communication gap, both in informal and formal settings. Recent advancements in deep learning have led to promising developments in gesture and motion recognition technology. A proposed solution aims to translate hand gestures into text in real-time, making communication more accessible for non-signers. Unlike previous research that primarily focused on translating individual letters or numbers, this system utilizes Convolutional Neural Networks (CNN) for hand gesture classification. By implementing such a system, the disparity between signers and non-signers can be reduced, easing communication for individuals with speech impairments. Research into Artificial Neural Networks provided the groundwork for this project, with previous models achieving an accuracy of around 86% in character detection. While Linear Discriminant Analysis (LDA) was considered, its limitation in handling complex data led to its exclusion. The project also delved into hardware implementation, recognizing the associated costs and maintenance requirements. Efforts were made to mitigate these factors in the developed system. Achieving a high accuracy rate of 96.5%, the model also offers word suggestions and sentence formation features, setting it apart from previous research efforts.

***Keywords —*** *CNN, ROI, OpenCV, NLP, ADA GRAD, N-Grams.*

1. INTRODUCTION

American Sign Language (ASL) stands out as the primary mode of communication for individuals with hearing and speech impairments. Because their disabilities mainly affect communication, those who are deaf and mute rely on sign language exclusively for conveying their thoughts and understanding others. Communication encompasses the exchange of ideas through various means, including speech, signals, behavior, and visual cues. Deaf and mute individuals, referred to as D&M people, utilize their hands to articulate different gestures, allowing them to communicate effectively with others. These gestures serve as nonverbal messages, comprehensible through visual perception. This form of nonverbal communication is known as sign language, a visual language comprising three major components.:

|  |  |  |
| --- | --- | --- |
| **WORD LEVEL VOCABULARY** | **FINGER SPELLING** | **NON-MANUAL FEATURES** |
| Used for most of the communication. | Used to spell words letter by letter | Facial expressions and body position and tongue, mouth. |

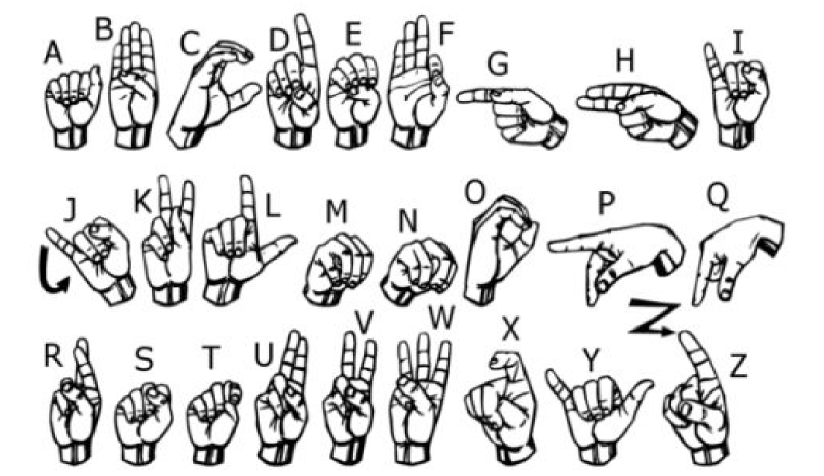
Our project primarily centers on developing a model capable of recognizing finger-spelling-based hand gestures and combining them to form complete words. The gestures we intend to train the model on are depicted in Figure 1.

Figure 1. Signs for all 26 English alphabets A-Z.

1. RELATED WORK

The literature survey delves into recent advancements in sign language recognition, highlighting key findings from five distinct research papers. Barbhuiya et al. (2021) proposed a CNN+SVM approach for static sign language recognition, achieving an impressive accuracy of 99.82% in recognizing alphabets and numbers in American Sign Language (ASL). Their focus on isolated gestures underscores the significance of accurate classification for individual signs. Aly et al. (2020) employed a DeepLabv3+Bi-LSTM technique, achieving an accuracy of 89.59% in recognizing ASL alphabets. Their method, while static and isolated, showcases the efficacy of combining deep learning architectures for improved sign language understanding.

In a dynamic setting, Lee et al. (2020) presented a LSTM+KNN model capable of recognizing alphabets in Arabic Sign Language (SL). Their hybrid approach, accommodating both static and dynamic signs, reflects the complexity of sign language communication. Xiao et al. (2020) explored the use of GAN+LSTM+3DCNN for static sign language recognition, focusing on word-level classification in ASL with an accuracy of 99.44%. Their work emphasizes the importance of capturing temporal dependencies in sign language gestures to enhance recognition accuracy.

Lastly, Elakkiya et al. (2021) introduced a CNN + Bi-LSTM with attention mechanism for recognizing words and sentences in continuous sign languages like Chinese SL (CSL) and German SL (GSL). Their model achieved accuracies of 81.22% and 76.12% for CSL and GSL, respectively. By addressing the challenges of continuous sign language recognition, their research contributes to the development of more inclusive communication technologies. Collectively, these studies demonstrate the diverse range of techniques and approaches employed in sign language recognition, underscoring the ongoing efforts to improve accessibility and inclusivity for the deaf and hard of hearing community.

1. METHODOLOGY AND PROPOSED WORK

To In order to examine the effectiveness of several deep learning models in identifying multiple types of lung cancer using CT-Scan pictures, the following methodology was used:

1. Acquisition of Datasets: Initially, in our investigation, we looked for pre-existing datasets, but we were unable to locate any that met our requirements in the raw image format. Rather, we found datasets that were shown as RGB values. As such, we decided to generate our own dataset by doing the following:

* The OpenCV library was used to create our dataset. For training, we first took about 600 pictures of every ASL symbol, and for testing, we took about 150 pictures of each symbol.
* Our process begins by capturing each frame from the webcam of our machine. Within each frame, we delineate a region of interest (ROI) represented by a blue bounded square, as illustrated in Figure 2.

****

Figure 2. RGB image of alphabet “A”

* As shown in Figure 3, we extract our ROI—which is in RGB format—from this entire image and turn it into a grayscale image.

A close-up of a hand

Description automatically generated

Figure 3. Grayscale image of alphabet “A”

* Finally, to help with the feature extraction process, we apply a Gaussian blur filter to our image. After applying the Gaussian blur, the final image looks much like Figure 4.

**A close-up of a hand

Description automatically generated**

Figure 4. Image obtained of alphabet “A” after applying Gaussian filter.

1. Gesture Classification: We use two levels of algorithms to forecast the user's final sign.

* Algorithm Layer:
  + - 1. Apply Gaussian blur filter and threshold to OpenCV-captured frames. This approach pulls features from an image.
      2. The image is analyzed and supplied into the CNN model for prediction. If a letter is identified in more than 60 frames, it is printed and used to build the word. We also use the blank symbol to represent spaces between words.
* CNN Model:

1. In the first convolution layer, the input image with a resolution of 128x128 pixels is processed with 32 filter weights (3x3 pixels). This produces a 126x126 pixel image for each filter weight.
2. The first pooling layer uses max pooling with a 2x2 window to downsample images while maintaining the greatest value in each 2x2 square. As a result, the image is downsampled to 63x63 pixels.
3. Second Convolution Layer: The first pooling layer's output (63x63) is fed into the second convolution layer, which processes it with 32 filter weights (3x3 pixels each). This produces a 61x61 pixel picture.
4. 2nd pooling layer uses max pooling with a 2x2 window to reduce images to 30x30 pixels.
5. 1st Densely Connected Layer: Images are fed into a completely connected layer of 128 neurons.

* The output of the second convolutional layer is reshaped into an array of 28800 values (30x30x32). To prevent overfitting, we use a dropout layer with a value of 0.5.
* The output of the first Densely linked Layer is sent into a completely linked layer of 96 neurons.

1. Final Layer: The 2nd Densely Connected Layer's output feeds into the final layer, which has neurons equal to the number of classes being classified (alphabets + blank symbol). Figure 5 depicts the integrated architecture of the CNN discussed earlier.

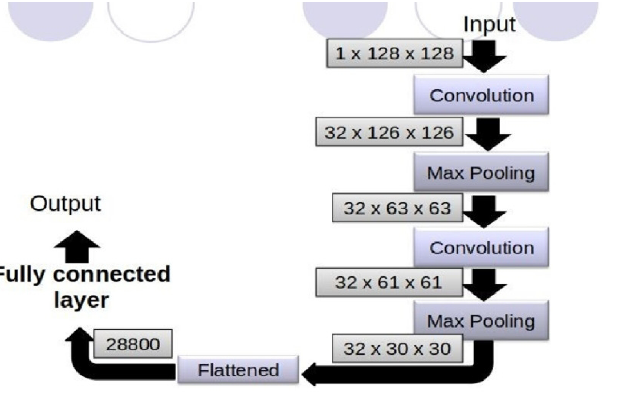


Figure 5. CNN Model

* Activation Function:

We've included ReLU (Rectified Linear Unit) in every layer, whether they're convolutional or fully connected. ReLU computes max (x, 0) for each input pixel, adding nonlinearity to the formula and improving the learning of complicated characteristics. It addresses the vanishing gradient problem and speeds up training by reducing calculation time.

* Pooling Layer:

We use Max pooling on the input image, with a pool size of (2, 2), in conjunction with the ReLU activation function. This minimizes the number of parameters, lowers computational costs, and reduces overfitting.

* Dropout Layers:

To avoid overfitting, in which the network's weights become overly specialized to the training instances, resulting in poor performance on fresh examples, we use dropout layers, as shown in Figure 6. These layers randomly deactivate a section of their activations by setting them to zero. Even having some activations removed, the network should still be able to provide appropriate classifications or outputs for specific samples.



Figure 6. Dropout Layers

* Optimizer:

We used the Adam optimizer to update the model based on the output of the loss function. Adam optimizer combines the advantages of two stochastic gradient descent algorithms: ADA GRAD and RMSProp.

1. Finger spelling sentence formation:

* Implementation:

1. If a detected letter exceeds a threshold value and no other letters are close, we print it and add it to the current string (set to 60 in our code).
2. If not, we clear the current dictionary with the number of detections of the symbol to avoid incorrect letter prediction.
3. If the number of blank (plain backdrop) identified exceeds a certain threshold and the buffer is empty, no spaces are recognized.
4. Otherwise, it prints a space to indicate the end of a word and appends the current buffer to the phrase..

* Autocorrect Feature:

We use the Python package Hunspell\_suggest to suggest appropriate alternatives for each invalid input word. Users are presented with a list of words that match the current term, from which they can select one to add to the current statement. This function helps reduce spelling errors and forecast complicated words.

1. Training and Testing:
2. We convert input images (RGB) into grayscale, apply Gaussian blur to remove unnecessary noise, and use adaptive thresholding to extract hands from the background. Images are resized to 128 x 128.
3. After preprocessing, we feed the input images to our model for training and testing, following the mentioned operations.
4. The prediction layer estimates the likelihood of the image falling under one of the classes. Outputs are normalized between 0 and 1, ensuring the sum of values in each class equals 1, achieved through the softmax function.
5. To improve prediction accuracy, we train the network using labeled data. Cross-entropy, a performance measurement in classification, is utilized. It is minimized to optimize network weights, accomplished through the Adam Optimizer, which is a variant of gradient descent. TensorFlow provides an inbuilt function for calculating cross-entropy.
6. RESULTS:

An application interface has been successfully developed, capable of interpreting American Sign Language (ASL) in real-time and converting it to text, aiding individuals who are deaf and mute in effective communication without the need for a translator. Figure 7 illustrates a character prediction by converting hand gestures using the Mediapipe library..

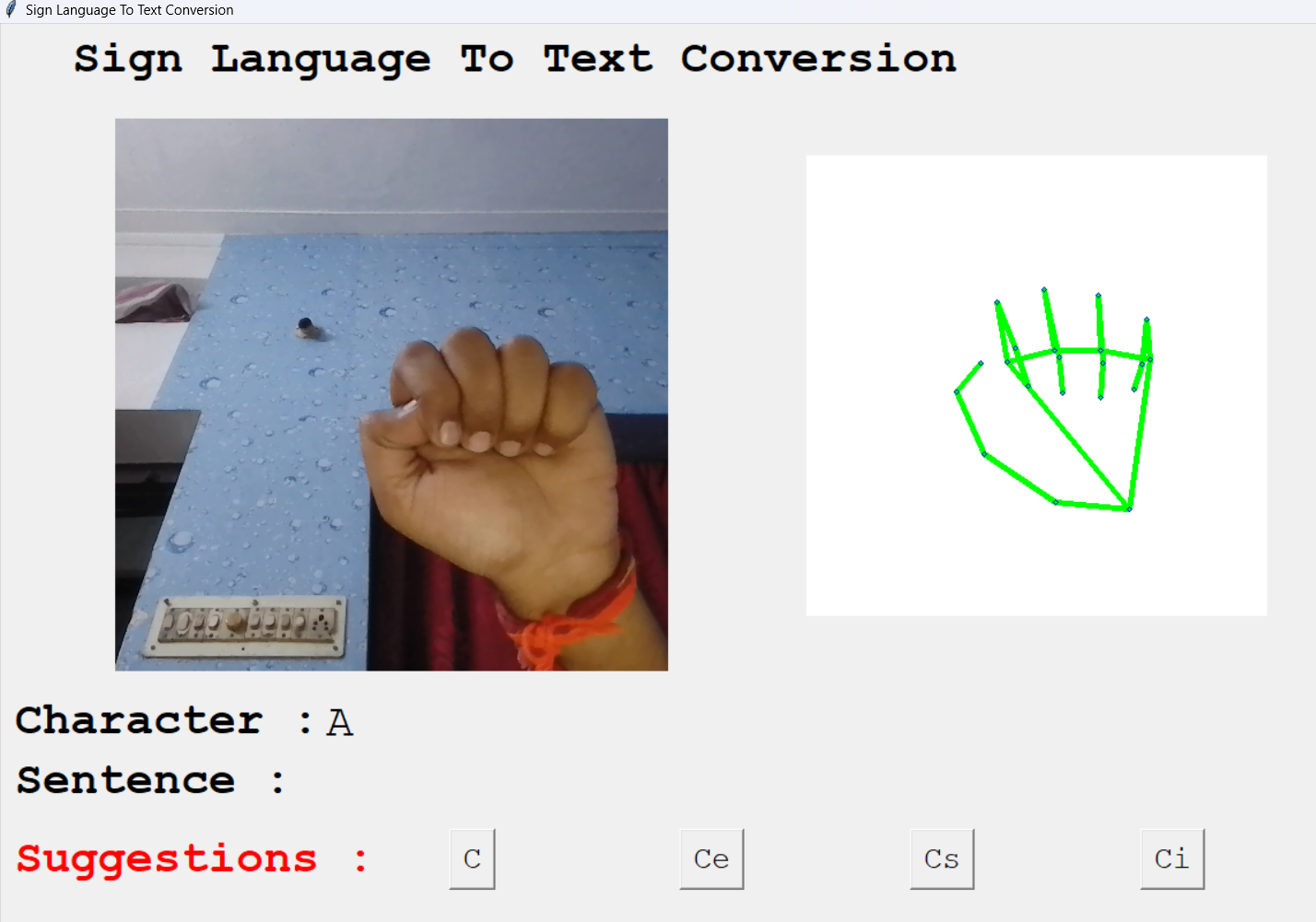


Figure 7. Result obtained for a single character “A”.

The devised model achieved an impressive accuracy of 96.5%, surpassing many of the models researched. Figure 8 displays the loss and accuracies of 20 epochs, demonstrating the model's performance.

* To ensure accurate detection, the sign had to be held stable for at least 60 frames in front of the camera.
* Various frame capturing values were tested for prediction, including 40 and 80 frames.
* However, a frame value of 60 proved to be the optimal choice, balancing accuracy and speed of detection..

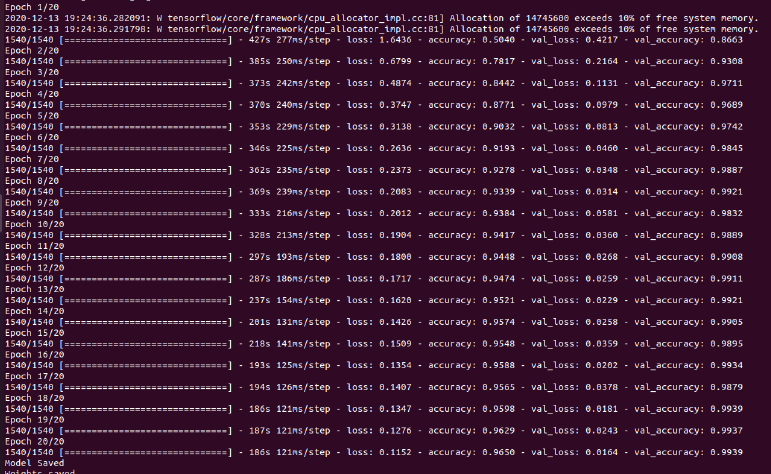


Figure 8. Epochs for CNN model trained.

This project is Implemented using a deep learning model in Python, the project requires only a laptop webcam and a computer system for deployment, making it cost-effective and efficient compared to hardware-based systems using flex sensors. Maintenance costs are minimal, with only a functional webcam needed for sign detection. The model offers suggestions based on the current word being translated, leveraging a US dictionary.

For instance, if a speech or hearing-impaired person wishes to sign "ELECTRON" to a normal person, the system provides suggestions matching the initial alphabets written so far, such as "ELECTOR," "ELECTRON," "ELECTRA," or "ELECTROCUTE." The suggestions provided by the Hunspell library help detect and correct spelling mistakes that may arise from incorrect alphabet detection or limited knowledge of word spellings. Multiple characters can be combined to form words, enabling the formulation of complete sentences—an innovative feature not found in other researched models.

1. CONCLUSIONS AND FUTURE WORK:

Convolutional Neural Networks (CNNs) are integrated into sign language recognition represents a significant advancement in artificial intelligence, particularly in the domain of computer vision. In our study, we have proposed a novel method for recognizing fingerspelling in American Sign Language (ASL) leveraging CNNs, alongside pattern recognition techniques. Through extensive training on a dataset comprising 27 symbols, which includes 26 English alphabets and also a 'blank' symbol for spacing, our CNN classifier achieved an impressive accuracy rate of 96.5%. This high level of accuracy not only showcases the efficacy of our approach but also underscores its potential to reduce reliance on human interpreters, thereby enhancing accessibility for individuals with hearing impairments.

Furthermore, to facilitate testing and real-world application of our classifier, we developed a user-friendly graphical user interface (GUI) application. This application gives ease for users to form characters, words, and sentences in ASL according to their communication needs. Additionally, the application predicts various suggestions for the corresponding word formed currently, contributing to smoother and more accurate communication. By successfully eliminating the need for human interpreters through our CNN-based approach and providing a practical tool for ASL communication, our study marks a significant step forward in advancing accessibility and inclusivity for the deaf and hard of hearing community.

The creation of an American Sign Language (ASL) recognition system based on real-time vision that is adapted for the deaf and mute (D&M) community is described in this study. Our method uses two layers of algorithms to improve prediction accuracy, especially for related symbols, and achieves a final accuracy of 95% on our dataset. With appropriate hand movements, low background noise, and sufficient lighting, our technology can accurately identify ASL symbols and display the associated text on the screen. We appreciate the significance of facial emotions in sign language, even though our main focus was on gesture identification; these features might be included in further system updates.

Our technology has a wide range of possible uses. For example, it can be used to access government websites that do not have sign language films or to fill out online forms without the need for an interpreter. With an estimated 70 million people with speech and hearing impairments globally, our project strives to close the communication gap between the D&M community and the general public. Future improvements might include running the system on inexpensive hardware such as Raspberry Pi and improving image processing skills to allow for two-way communication, meaning that spoken and sign language can be translated.

Our future objectives include the recognition of motion-based signals and the conversion of gesture sequences into words and sentences, with the possibility of adding speech synthesis for aural output. Even with such high accuracy, there is still space for improvement and extension because our dataset lacks word-level sign language data. Furthermore, real-time efficiency is just as important for practical implementation as algorithmic accuracy.

1. REFERENCES:
2. M.M.Gharasuie, H.Seyedarabi, “Real-time Dynamic Hand Gesture Recognition using Hidden Markov Models”, 8th Iranian Conference on Machine Vision and Image Processing (MVIP), 2013.
3. Pradumn Kumar, Upasana Dugal, “Tensorflow Based Image Classification using Advanced Convolutional Neural Network”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-6, March 2020, March, 2020.
4. Xin Jia, “Image Recognition Method Based on Deep Learning”, CCDC, DOI: 978-1-5090-4657-7/17, 2017.
5. Jiudong Yang, Jianping Li, “Application Of Deep Convolution Neural Network”, IEEE, DOI: 978-1-5386-1010-7/17, 2017.
6. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, Journal of Machine Learning Research 15 (2014) 1929-1958, 2014.
7. Ankit Ojha, Ayush Pandey, Shubham Maurya, Abhishek Thakur, Dr. Dayananda P, “Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, NCAIT - 2020 Conference Proceedings.
8. Nobuhiko MUKAI, Naoto HARADA, Youngha CHANG, “Japanese Fingerspelling Recognition based on Classification Tree and Machine Learning”, NICOGRAPH International, 2017.
9. Qi Wang, Zequn Qin, Feiping Nie, Yuan Yuan, “Convolutional 2D LDA for Nonlinear Dimensionality Reduction”, Proceedings of the TwentySixth International Joint Conference on Artificial Intelligence (IJCAI-17).
10. Aarthi M, Vijayalakshmi P, “Sign Language To Speech Conversion”, Fifth International Conference On Recent Trends In Information Technology, 2016.
11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ‘‘Going deeper with convolutions,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.
12. Tang Z., Chen K., Pan M., Wang M. and Song Z. 2019 An augmentation strategy for medical image processing based on Statistical Shape Model and 3D Thin Plate Spline for deep learning IEEE Access 7 1-1.
13. "Rethinking the Inception Architecture for Computer Vision" by Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna.
14. Y. Song, Y. Zhang, Y. Chen, and Y. Wu, "Lung nodule detection using 3D deep convolutional neural networks with multiple GPUs," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 10-18.
15. K. He, X. Zhang, and S. Ren, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 770–778.