Leveraging LLMs, computer vision, and ChatGPT to ease communication between deaf and hearing individuals

By: Amogh Khaparde

Abstract

Hearing and hard of hearing individuals might struggle to communicate without the help of a human interpreter. However, machine learning is a tool that can help enable ease of communication between both groups of people. This project involves an application that uses a convolutional neural network to solve this problem. This convolutional neural network model was custom-made from 46,000 augmented training images and tested with 8000 images. Its purpose in the application is to take camera frames of ASL fingerspelling as input and then translate it into text. This alphabet is appended to a sentence variable where it is stored. Once the user has completed a sentence, a ChatGPT API will correct any grammatical errors or misclassifications that the model made. With this ChatGPT edited sentence, a Play.HT API can turn it to speech in order to further smoothen the language barrier. A background cropping feature was added as well, which takes frames from a camera and uses the MediaPipe module to locate the hand, after which it will remove any part of the image except the hand in order to reduce noise for the model. The final model achieved a 97.5% accuracy and 0.1% loss with the testing dataset and also had a strong confusion matrix as well, with only some mistakes with similar looking characters such as M and N. The model also worked better in different rooms since the background noise was removed from the model entirely. Overall, this application is an innovative step for using AI and machine learning to slowly removing the language barrier between hard of hearing and hearing individuals.

Introduction

American Sign Language (ASL) allows many individuals around the world to communicate and be a part of society despite their physical challenges. Many of these hard of hearing individuals use human interpreters to communicate with hearing individuals, but in some cases, human interpreters may not be present. Even if an interpreter is present, access may be limited and costly. In addition, on online platforms, communication between a hearing impaired individual and a hearing individual may be difficult because these applications often lack the support of a human interpreter. Thus, to reduce reliance on human interpreters,, one possibility is to use artificial intelligence. In this study, I have developed an application using a deep convolutional neural network (CNN) to facilitate communication for the hearing impaired.

The landscape of American Sign Language (ASL) recognition has seen many new improvements with technologies such as Kinect systems (motion sensing input devices), data gloves (gloves that track hand motions in 3D), Leap Motion controllers (motion controller that allows users to control their computer with hand motions), and webcams (normally found on laptops and computers that take image input). While Kinect systems can track three-dimensional motion, their precision falls short of capturing subtle finger movements crucial for fingerspelling [1-4]. Although data gloves offer detailed finger motion tracking, they suffer from high cost, discomfort, and limited practicality in the real world [5-9]. Leap motion controllers are capable of fine-grained hand movement capture, but they have difficulty recognizing diverse

fingerspelling gestures accurately and are also exceptionally expensive [10-12]. In contrast, webcams, a cost-effective and non-intrusive solution, have the potential to properly take in input for ASL recognition algorithms. Webcam-based recognition models rely only on prior training data as well as camera input, making them more cost-effective and comfortable than other alternatives. In addition, their ubiquity and ease of integration make them an attractive choice [13-17]. While some challenges exist in developing algorithms for the accurate interpretation of fingerspelling gestures, these challenges can usually be overcome by using different testing data and frame cropping, which enable the machine learning model to focus on the relevant parts of an image.

Despite the promise of webcam-based recognition systems, there is a clear need within the landscape of American Sign Language (ASL) fingerspelling recognition for a model that not only achieves high accuracy but also offers user-friendly features. Regarding accuracy, the limited success of these types of models in decoding gestures remains a hurdle, often resulting in misinterpretations and miscommunications [18-20]. In addition, existing models often lack the crucial capability of allowing users to delete or modify recognized signs, hindering the user experience.

To meet these challenges, I have developed a model that takes video input of a user signing the spelling of a sequence of words and that decodes the video images in realtime to output each letter of the word or phrase. In addition, there is a feature in the UI that allows the user to indicate when signing is complete. Once this occurs, the model passes the decoded output to ChatGPT, which - in the event of potential misclassifications - revises the output to ensure correct spelling, interpretability, and reasonableness.

Moreover, the incorporation of ChatGPT within my application sets it apart by not only recognizing fingerspelling, but also providing users with a valuable tool to fix sentence structures or clarify their expressions in real time. This dual functionality addresses a common issue in sign language recognition systems, where linguistic nuances and context can be challenging to capture accurately. In contrast, my model boasts an exceptionally high accuracy of 97% by leveraging advanced machine learning techniques and a robust training dataset to enhance precision. This aspect is crucial for the effective and reliable communication between individuals using ASL, as inaccuracies can lead to misunderstandings and hinder the overall efficacy of sign language recognition systems. By significantly improving the accuracy, my application aims to contribute to a more seamless and reliable communication experience for users. Furthermore, the incorporation of a realistic Text-to-Speech (TTS) feature within my application is a pioneering approach to facilitating communication between deaf and hearing individuals. This TTS functionality goes beyond the conventional scope of ASL recognition models, acknowledging the diverse communication needs of the deaf community. It serves as an inclusive tool, allowing deaf users to communicate effortlessly with those who may not be familiar with ASL, thereby bridging the gap between the deaf and hearing worlds. This innovation aligns with the broader goal of creating accessible technologies that enhance communication and promote inclusivity.

As a solution to this issue, I created my own model that is highly accurate with over 70% accuracy as defined by percent of testing images classified correctly within the testing dataset. It also includes ChatGPT integration in order to correct any misclassifications, and a high quality text-to-speech (TTS) program. This model is a Resnet-18 model, using Pytorch's default weights to train. Then, this model was initialized and trained with images of my hand. I first took jpeg images with a python file that I made to make it easier to capture training data. In addition, to further improve the accuracy, this model utilizes a background cropping tool that reduces background noise and focuses on the hand, which in turn allows the model to notice smaller changes within the hand structure and learn how to better classify the input images.

Method

Dataset creation: For the model, I used an ASL Fingerspelling dataset that I built, consisting of approximately 46000 images of individual fingerspelling gestures. The dataset covers all 26 letters of the English alphabet, and is then augmented via synthetically modified images consisting of around 1700-1900 images per class. All images were resized to a uniform 224x224 pixel size. **Augmentation:** To generate more training samples, each image was transformed using a brightness variation with a brightness variation of 0.5 to 1.4 and a contrast variation of 0.5 to 1.4, along with a random rotation of 20 degrees. Before training the model with this data, By doing this, the images vary in contrast, brightness, and rotation, which allows the model to learn and function better under different backgrounds. **Training:** The model uses pre-built training weights called the RESNET18.IMAGENET weights. The simulation was trained using Adam optimizer with a default learning rate of 0.001, which allowed the model to learn and perform well. I implemented a batch size of 64 images and early stopping criteria based on validation loss. The training procedure was run for 20 epochs. The model made predictions using a softmax layer. **Performance:** In this experiment/project, the loss is defined as comparing the the "distance" between the model's predictions and the actual labels. In addition, the accuracy is defined as the number of correct predictions divided by the total predictions. The simulation achieved an accuracy of 98% on the test set, along with an approximate 0.08% loss, which indicates a very trivial difference between a model's predicted output and the actual output. **Background crop:** A background cropping feature was used on the model's data to isolate the hand, enhancing the model's focus on crucial signing details during image capture. This innovative technique aims to improve accuracy and reduce the impact of background noise in the training dataset.

As for the features of this application, it has a user-friendly interface with an input field for hearing users. This includes a webcam-based input field for hard of hearing users, a visual output pop-up for the new GIF for hard of hearing users, and a vocal output for hearing users. **ChatGPT Conversion:** As an example, if a user signs "OK TTYL," the application instantly transforms it into "OK, talk to you later!" in the form of a voice for the hearing user to hear. **Realistic TTS:** This real-time feature aids users in quickly translating their thoughts into normal sentences. **Manual edit features:** The application includes a spacebar feature where a hearing impaired user can move their hand away from the webcam to mark a space in their sentence, which allows for a typing-free, smooth user experience. In addition, if one is not as experienced in fingerspelling, they may use a backspace feature to manually edit their sentences as well.

Results/Discussion

The final model (Version 10) was tested for accuracy and loss (Figure 1). An older version (Version 3) was included in the comparison to assess the evolution of the model. The final model achieved considerably higher performance than the older version. However, in both models, the loss is low (1% and lower) and the accuracy is high (90%+) for both the test and the training dataset, which exceeded the benchmark goals of >70% accuracy, and <1% loss. Therefore, these results indicate both models have robust reliability in predictions. This illustrates the raw CNN model's ability to make accurate predictions, even with new data, which is promising for real world usage. The more recent model had augmented images, as the previous models .

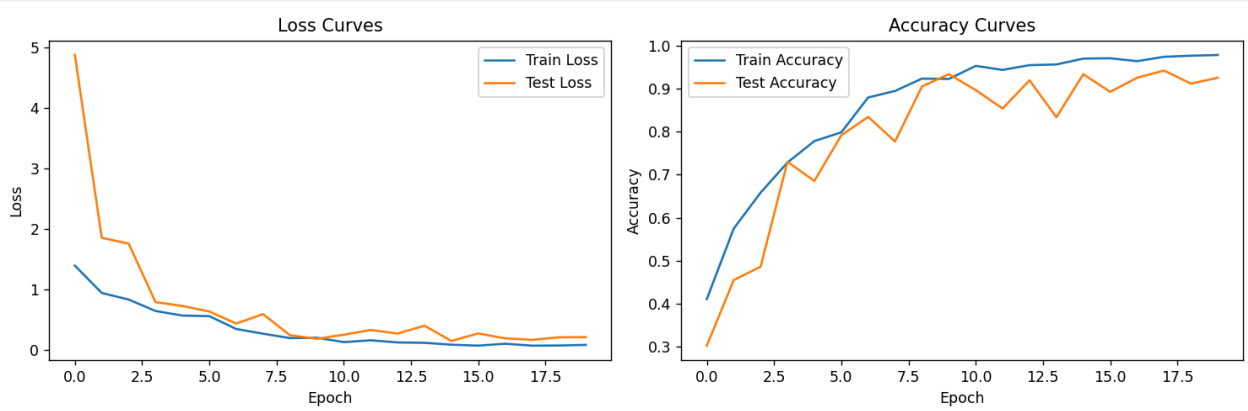

Figure 1. (A) Loss (the graph on the left) and (b) accuracy (the graph on the right) of the testing and training datasets for version 3

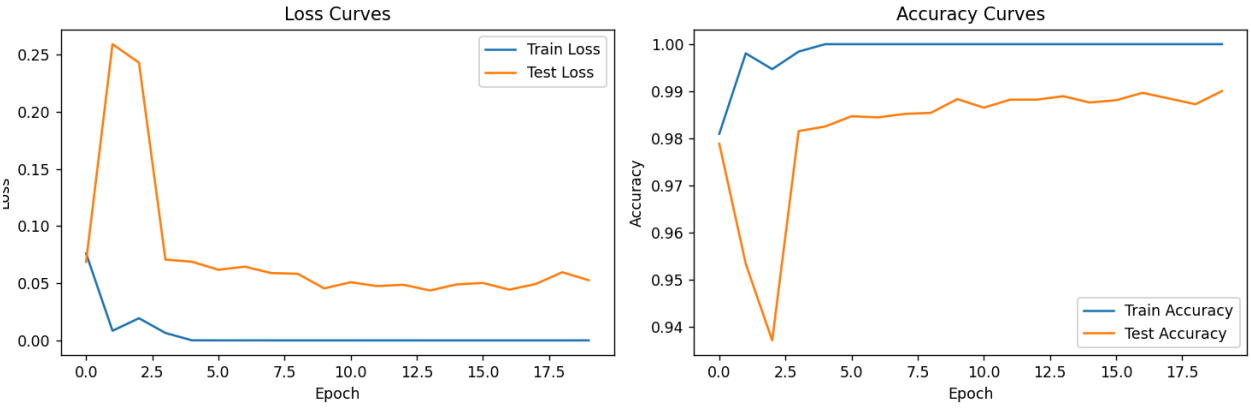

Figure 2. (A) Loss (the graph on the left) and (b) accuracy (the graph on the right) for the testing and training datasets for the final model (version 10)

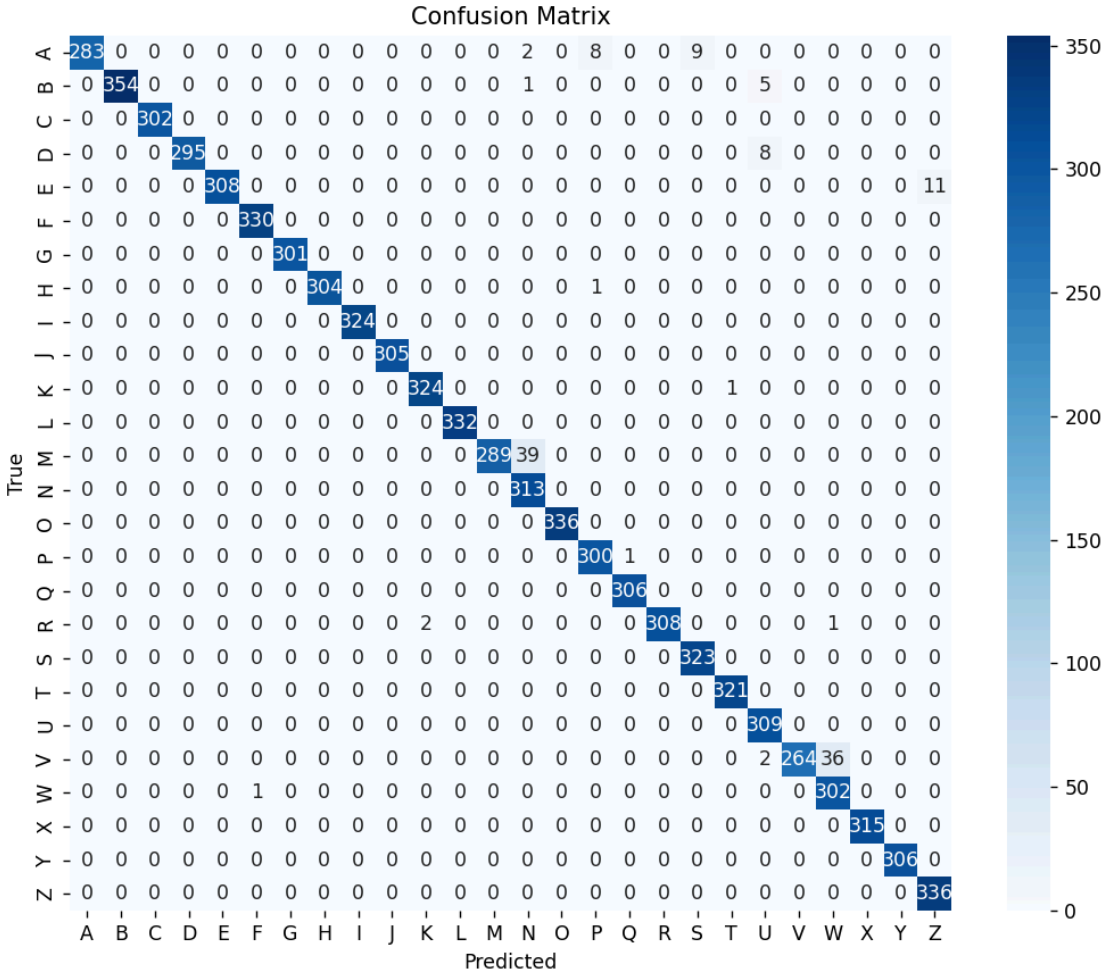

Figure 3. A confusion matrix showing false positives and true positives for each alphabet letter. The dark blue line through the center of the heatmap signifies the model was able to predict the correct letter for almost all of the testing images.

Figure 3 displays a confusion matrix of the multivariate classification results for the final model without the cropped backgrounds. The strong true positive to false positive ratio – indicated by the dark diagonal blue line through the center of the heat map – provides evidence of the eliability of the raw CNN model .

Background crop model:

Figure 4. (A) Loss (the graph on the left) and (b) accuracy (the graph on the right) of the testing and training datasets for the background cropping model

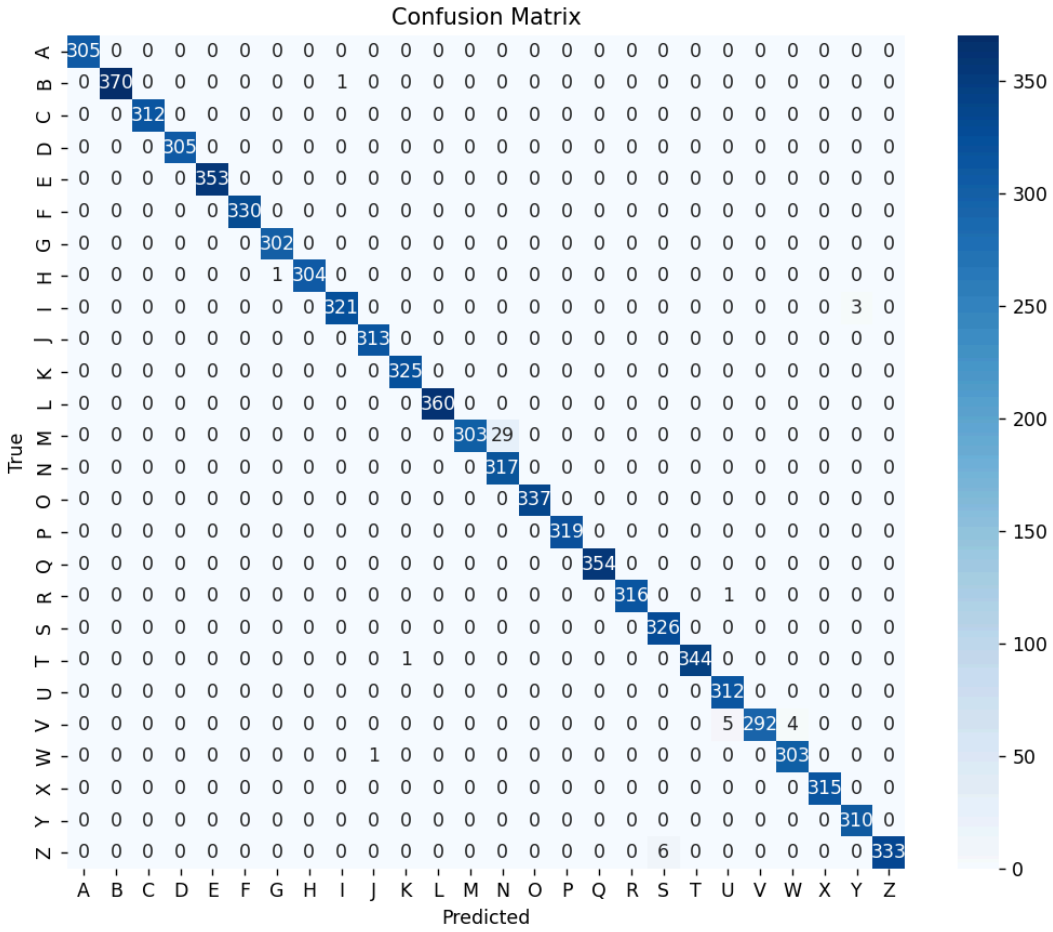

Figure 5. A confusion matrix showing false positives and true positives for each alphabet letter. The dark blue line through the center of the heatmap signifies the model was able to predict the correct letter for almost all of the

testing images. In addition, the background crop model's confusion matrix did slightly better compared to that in Figure 3.

Surprisingly, the background cropping feature, designed to refine the model's predictions, yielded minimal improvements in accuracy. Since the decision-making process involves weighing the trade-offs between model complexity and performance. the simplicity and efficiency of the CNN model is arguably more effective and faster. However, the background cropping feature may be more useful for more users whom the model is not already acquainted with. To offer a visual representation of the models' capabilities, demonstration videos have been created, providing a tangible illustration of the fingerspelling recognition process. The videos are linked below along with the code:

Video 1:
https://drive.google.com/file/d/1_0uPYHjiTpY2Yje-AanQxdND-IkrWIX_/view?usp=sharing
Video 2:
https://drive.google.com/file/d/1rs_es06TVGMGOb4HIny81h_IPLh0AaYC/view?usp=sharing
Video 3:
https://drive.google.com/file/d/143N607riEWj2SXL9TrKCIQ1tiM-qJo7n/view?usp=sharing
Code:
https://github.com/AmoghKhaparde/ASL_Detection

Conclusion

In summary, this project aimed to tackle the intricacies of ASL fingerspelling recognition through the development of a convolutional neural network integrated with an innovative background cropping feature. Despite initial expectations, the model demonstrated surprising effectiveness without the background cropping feature, as shown by a strong confusion matrix, high accuracy (90%+) and very low loss (0.1%). However,, the model with the background cropping feature did perform slightly better on the confusion matrix, and will likely work better as part of a real world application since it is able to work around any background noise that may come from the input. In order to bridge communication gaps between sign language users and those unfamiliar with ASL, the application includes the integration of a realistic Text-to-Speech (TTS) feature and a Chat-GPT feature, which corrects user's sentences for them and increases the usability of this application.

This work lays the groundwork for subsequent investigations involving improving the model and assessing its performance. For instance, one future direction is to explore the effects of variables such as diverse signing styles on the model's performance. By diversifying the training and testing data, it will enable us to determine the extent to which the CNN is generalizable. Therefore, the next steps for his project involve refining and improving the model based on these insights, creating more dynamic gestures, such as a backspace gesture, and other common gestures such as "Hi", "How are you" in ASL. Lastly, this application can be improved further with user feedback in order to optimize the model's practical use in the real world.

References

1. "American Sign Language alphabet recognition using Microsoft Kinect." ScholarsMine, scholarsmine.mst.edu/cgi/viewcontent.cgi?article=8391&context=masters_theses. Accessed 20 Nov. 2023.
2. Bowden, Richard. "Spelling it out: Real-time ASL fingerspelling recognition." ieeexplore.ieee.org, ieeexplore.ieee.org/document/6130290. Accessed 20 Nov. 2023.
3. Starner, Thad. "American sign language recognition with the kinect." *dl.acm.org*, dl.acm.org/doi/10.1145/2070481.2070532. Accessed 20 Nov. 2023.
4. Yang, Hee Deok, editor. "Sign Language Recognition with the Kinect Sensor Based on Conditional Random Fields." *ncbi.nlm.nih.gov*, www.ncbi.nlm.nih.gov/pmc/articles/PMC4327011/. Accessed 16 Nov. 2023.
5. "Wearable-tech glove translates sign language into speech in real time." *UCLA*, newsroom.ucla.edu/releases/glove-translates-sign-language-to-speech. Accessed 20 Nov. 2023.
6. "Sign Language Glove." *Cornell*, people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2014/rdv28_mjl256/webpage/. Accessed 20 Nov. 2023.
7. "Recognition of Finger Spelling of American Sign Language with Artificial Neural Network Using Position/Orientation Sensors and Data Glove." *SpingerLink*, link.springer.com/chapter/10.1007/11427445_25. Accessed 20 Nov. 2023.
8. "Dataglove for Sign Language Recognition of People with Hearing and Speech Impairment via Wearable Inertial Sensors." *MDPI*, www.mdpi.com/1424-8220/23/15/6693. Accessed 20 Nov. 2023.
9. R. Fatmi, S. Rashad and R. Integlia, "Comparing ANN, SVM, and HMM based Machine Learning Methods for American Sign Language Recognition using Wearable Motion Sensors," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2019, pp. 0290-0297, doi: 10.1109/CCWC.2019.8666491.
10. Chong, Taek Wei. "American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach." *National Library of Medicine*, www.ncbi.nlm.nih.gov/pmc/articles/PMC6210690/. Accessed 20 Nov. 2023.
11. "The Leap Motion controller: A view on sign language." *Griffith University*, research-repository.griffith.edu.au/bitstream/handle/10072/59247/89839_1.pdf. Accessed 20 Nov. 2023.
12. "Sign language recognition through Leap Motion controller and input prediction algorithm." *Journal of Physics*, iopscience.iop.org/article/10.1088/1742-6596/1715/1/012008/pdf#:~:text=Leap%20motion%20controller%20(LMC)%20is,language%20letters%20and%20digits%20recognition. Accessed 20 Nov. 2023.
13. "Sign Language Recognition with Advanced Computer Vision." *Towardsdatascience*, towardsdatascience.com/sign-language-recognition-with-advanced-computer-vision-7b74f20f3442. Accessed 20 Nov. 2023.
14. *Science Direct*. 16 Nov. 2021, www.sciencedirect.com/science/article/pii/S2667305321000454. Accessed 12 Oct. 2023.

15. S. Mhatre, S. Joshi and H. B. Kulkarni, "Sign Language Detection using LSTM," 2022 IEEE International Conference on Current Development in Engineering and Technology (CCET), Bhopal, India, 2022, pp. 1-6, doi: 10.1109/CCET56606.2022.10080705.
16. Parades, Brian. "American Sign Language Interpret using web camera and deep learning." *Ieomsociety.org*, ieomsociety.org/proceedings/2022rome/89.pdf?CFID=20565289-bddb-402a-93a8-93cf81c3e18e&CFTOKEN=0. Accessed 20 Nov. 2023.
17. "DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals." *Science Direct*, www.sciencedirect.com/science/article/pii/S2666990021000471#:~:text=Jain%20et%20al.,for%20a%20two%2Dlayer%20CNN. Accessed 12 Oct. 2023.
18. "ASL Detection - 99% Accuracy." *Kaggle*, www.kaggle.com/code/namanmanchanda/asl-detection-99-accuracy. Accessed 17 Nov. 2023.
19. "MiCT-RANet-ASL-FingerSpelling." *Github*, github.com/fmahoudeau/MiCT-RANet-ASL-FingerSpelling. Accessed 17 Nov. 2023.
20. Shi, Bowen. "Fingerspelling Detection in American Sign Language." openaccess.com, openaccess.thecvf.com/content/CVPR2021/papers/Shi_Fingerspelling_Detection_in_American_Sign_Language_CVPR_2021_paper.pdf. Accessed 21 Nov. 2023.