American Sign Language (ASL) Detection Systemusing Machine Learning

Akshunya Banjarey[1], Bhavya Chauhan[2], Vibhor Sharma[3], Nitesh Kumar[4], Sachin Garg[5]
Department of Information Technology Maharaja Agrasen Institute of Technology, Delhi, India

Abstract: One of the main challenges of communicating with people who have hearing disabilities is to understand their sign language. This paper presents a dedicated research project to investigate the difficulties involved in recognizing characters in American Sign Language (ASL), which is the most widely used sign language in the world. Sign language is essential for communication among people with hearing or speech impairments, but it can be hard for those who are not familiar with it, as the signs made by people with disabilities may look complicated or messy. Effective communication requires a two-way exchange. This paper proposes a Sign Language detection system that uses American Sign Language, where users can take pictures of hand gestures through a web camera and get them analyzed. The system aims to predict and show the name that matches the picture. The proposed research proposes a method for detecting sign language that uses the collaborative features of the OpenCV and MediaPipe frameworks. The Convolutional Neural Network (CNN) is used to train the model and identify the pictures. The proposed methodshows a high detection rate and excellent accuracy.

Keywords: American Sign Language, OpenCV, MediaPipe, Convolutional Neural Network

1. Introduction

Effective interpersonal communication is paramount in today's interconnected global landscape, serving as a vital conduit for the seamless exchange of information, emotions, and ideas. However, the conventional modes of communication often present formidable challenges for individuals within the deaf and hard-of-hearing community. Consequently, sign language has evolved into a rich and dynamic linguistic system comprised of intricate movements and expressive gestures, offering a vibrant medium for conveying a spectrum of emotions and thoughts.

The elaborate interplay of hand gestures and facial expressions that constitutes sign language contributes to its inherent beauty, with each nuanced gesture bearing profound meaning. Nonetheless, relying solely on these gestures may not always capture the nuanced complexities inherent in human language. This is where finger spelling emerges as an invaluable adjunct. Finger spelling augments communication for the deaf and hard-of-hearing community by providing an additional layer that allows the expression of ideas and words lacking corresponding signs. Functioning as a metaphorical bridge, finger spelling facilitates a more comprehensive form of communication, empowering individuals to articulate the subtleties of language.

When seamlessly integrated with the user-friendly attributes of sign language, the amalgamation of finger spelling results in a more intricate and comprehensive mode of communication, enriching the ability to convey the nuances and intricacies of language in a manner that transcends the limitations of traditional communication methods.

A. Sign Language Recognition

A diverse array of methods and strategies is employed in the field of sign language recognition to decipher the intricate visual language conveyed through signs. Two predominant mechanisms, namely sensor-based and vision-based recognition, have been at the forefront of sign language recognition efforts [9]. Presently, sensor-based approaches prove cost-prohibitive for individuals with limited financial means. In contrast, vision-based methods offer users a natural mode of communication through hand gestures, closely mirroring real-life interactions. Consequently, approaches rooted in image and video analysis garner preference.

Convolutional Neural Networks (CNNs) stand out as a widely utilized tool in computer vision-based methodologies, leveraging video data analysis to identify and categorize signs based on their visual characteristics. Pose estimation techniques employ skeletal tracking to capture the gestures and body language of the signer, enabling the recognition of signs based on the locations and motions of key body parts. Deep learning models, such as transformers and recurrent neural networks (RNNs), facilitate sequence-to-sequence learning, empowering the recognition of complete sign language sentences and phrases.

Gloss-based recognition enhances the understanding of sign language by focusing on individual sign recognition and translation. Continuous sign language recognition eliminates the need for explicit segmentation, enabling real-time interpretation. Gesture and facial expression analysis play a pivotal role in deciphering the nuanced meanings and emotions conveyed in sign language. Data-driven approaches rely on extensive sign language datasets for model training, while multimodal systems integrate visual data with other sensory inputs to enhance accuracy and robustness.

These versatile methodologies are tailored to recognize specific sign languages, accommodate variations in signing styles, and support a spectrum of applications, ranging from accessibility technology to sign language translation. However, it's noteworthy that most of these techniques are contingent upon fundamental steps.

- **Sign Segmentation:** Segmenting signs entails the intricate process of recognizing and isolating individual signs within ongoing sequences of signing. This undertaking proves notably challenging given the fluid and interconnected characteristics inherent in sign language. To address this challenge, sophisticated computer vision techniques and machine learning models are utilized to identify the commencement and conclusion of each sign, thereby guaranteeing precise segmentation. Moreover, the process of sign segmentation plays a crucial role in discerning between actual signs and non-signing motions, thereby augmenting the overall precision of recognition.

- **Sign Recognition:** Sign recognition serves as the core component within a sign language recognition system, involving the identification of distinct signs, phrases, or gestures executed by the signer. This intricate process commonly relies on the utilization of machine learning models, including deep neural networks, trained on comprehensive datasets of sign language. These models scrutinize the segmented signs, conducting comparisons with a repository of recognized signs to ascertain their corresponding meanings. Given the considerable variability in signing styles, handshapes, and facial expressions, sign recognition represents a complex task that necessitates ongoing model refinement to ensure accuracy.

- **Translation to Text:** Once signs undergo accurate recognition, the subsequent stage involves their translation into text. Sign language recognition systems frequently incorporate natural language processing techniques to convert identified signs into written or spoken language. This translation facilitates seamless communication between users of sign language and individuals who lack proficiency in it. The output in text form can be displayed on a screen or converted into speech using text-to-speech synthesis, broadening accessibility to a more extensive audience. Collectively, these processes establish a connection between the intricate visual language of sign language and the textual or spoken modes of communication. Sign segmentation ensures precise identification of signs, sign recognition interprets their meanings, and translation to text renders these meanings accessible to individuals unfamiliar with sign language. These components collectively contribute to fostering a more inclusive community. The decision to utilize the ASL (American Sign Language) fingerspelling system was influenced by the greater availability of high-quality datasets on platforms such as kaggle.com [18], encompassing educational and other content. In the following figure, Fig. 1, various letters of the alphabet are depicted along with their corresponding fingerspelled equivalents.

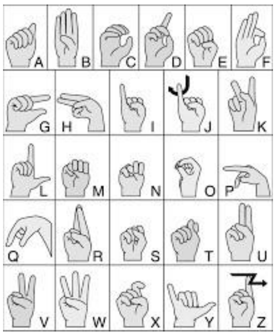


Figure 1: ASL finger-spelling alphabet [15]

2. Literature Review

- DongXu Li et al. [1] and Joshi's team [4] have made significant contributions to gesture recognition and sign language technology by providing large datasets and essential corpora, respectively.

- Pannattee et al. [7] and Bowen Shi et al. [8] have focused on identifying linguistic units within video data, addingvaluable insights to the field.

- Sharma and team [9] and Novopoltsev et al. [2] have advanced model training through exploration and definitionof optimal fine-tuning techniques.

- The convergence of these techniques has resulted in the development of the transformative MediaPipe Framework by Lugaresi et al. [5], significantly enhancing gesture recognition efforts.

- MediaPipe Framework, as highlighted by the work of Indira et al. [6], not only facilitates gesture recognition but also enables the application of transfer learning.

- Transfer learning, as demonstrated by Sharma et al. [9] and Veluri et al. [11], has empowered the developmentof superior models, ushering in new possibilities in sign language technology.

The combined impact of these methodologies has propelled the field forward, marking a new era of capabilities and possibilities in sign language technology.

3. Proposed Method

A. Preliminary

Researchers in the field of computer vision are motivated by three foundational concepts: sparse interactions, parameter sharing, and equivariant representations. In a standard neural network layer, interactions between input and output units are represented by a matrix of parameters, and computation involves multiplying numbers through matrix multiplication. This implies that all input and output units communicate with each other.

In contrast, Convolutional Neural Networks (CNN) exhibit sparse interactions achieved by using smaller kernels than the input. When processing an image through the kernel, relevant information can be identified, comprising tens or hundreds of pixels, even in the presence of images with millions or thousands of pixels. This results in fewer parameters that need to be stored, enhancing the statistical efficiency of the model while simultaneously reducing its memory footprint.

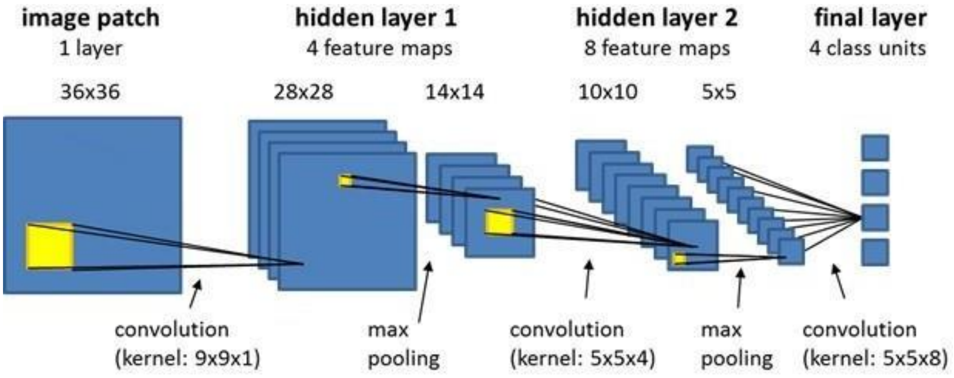


Figure 2: Convolutional Neural Networks (CNN) [20]

Some important frameworks/libraries used are:

a. **MediaPipe:** MediaPipe is an open-source framework developed by Google that focuses on building pipelines for real-time perception of hand, face, and body movement from images or video. It provides pre-trained models and a library of customizable solutions for computer vision tasks, making it a valuable resource for projects involving gesture recognition, sign language interpretation, and more. MediaPipe simplifies the process of integrating complex computer vision features into applications.

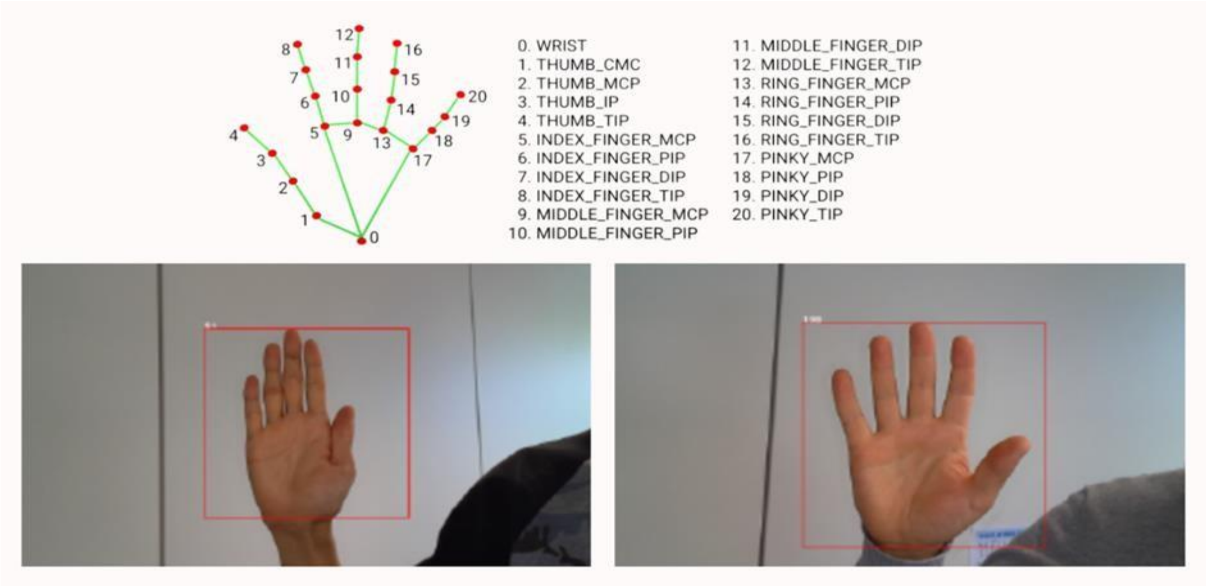


Figure 3: MediaPipe [18]

b. OpenCV: OpenCV, short for Open-Source Computer Vision Library, is a popular open-source computer vision and image processing library. It provides a wide range of tools and functions for image and video analysis, making it a fundamental resource for tasks such as object detection, feature extraction, image filtering, and more. OpenCV is widely used in various applications, including robotics, surveillance, and medical imaging.

c. Tensorflow/Keras: TensorFlow is an open-source machine learning framework developed by Google. It's known for its flexibility, scalability, and wide adoption in the machine learning community. Keras is an integral part of TensorFlow, providing an easy-to-use interface for building and training deep learning models. Together, TensorFlow and Keras are powerful tools for developing various machine learning and deep learning applications, including computer vision, natural language processing, and more.

B. Proposed Model

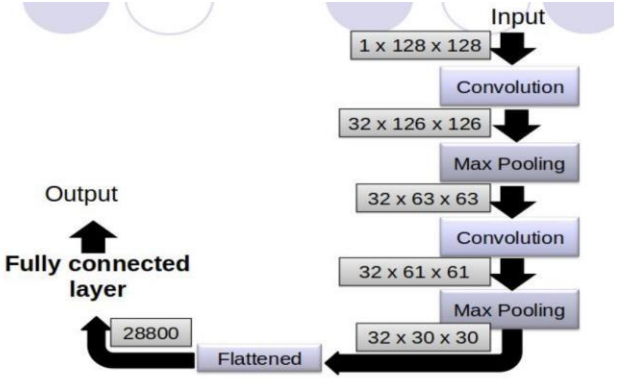


Figure 4: Proposed Model

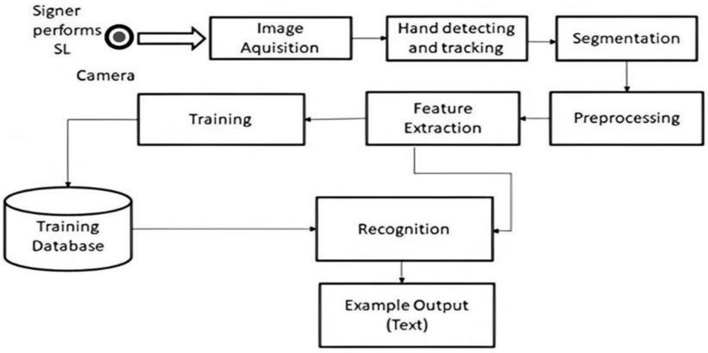


Figure 5: Proposed Methodology

In contrast to traditional Neural Networks, the layers of a Convolutional Neural Network (CNN) are arranged in three dimensions: width, height, and depth. The neurons in a layer are connected only to a small region of the preceding layer, referred to as the window size, rather than being fully connected.

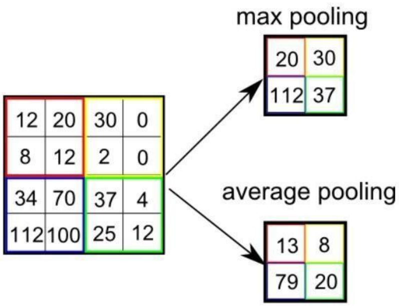
The final output layer is dimensioned according to the number of classes, as the CNN architecture reduces the entire image into a single vector of class scores by the end of the process.

1. Convolutional Layer: In the convolutional layer, a small window size, typically of dimensions 5*5, is used that extends to the depth of the input matrix. This layer consists of learnable filters of the same window size. During each iteration, the window is slid by a stride size, typically 1, and the dot product of the filter entries and input values at a given position is computed. This process results in a two-dimensional activation matrix that provides the response of that filter at every spatial position. In essence, the network learns filters that activate when they detect a specific type of visual feature, such as an edge of a certain orientation or a blotch of a particularcolor.

2. Pooling Layer: The pooling layer is used to decrease the size of the activation matrix, thereby reducing the number of learnable parameters. There are two types of pooling:

a. Max Pooling: In max pooling, a window size (for example, a window of size 2*2) is used, and only the maximum of the four values is taken. This window is slid across the activation matrix, resulting in a new activation matrix that is half the original size.

b. Average Pooling: In average pooling, the average of all values in a window is taken.



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

average pooling

13	8
79	20

Figure 6: Pooling [2]

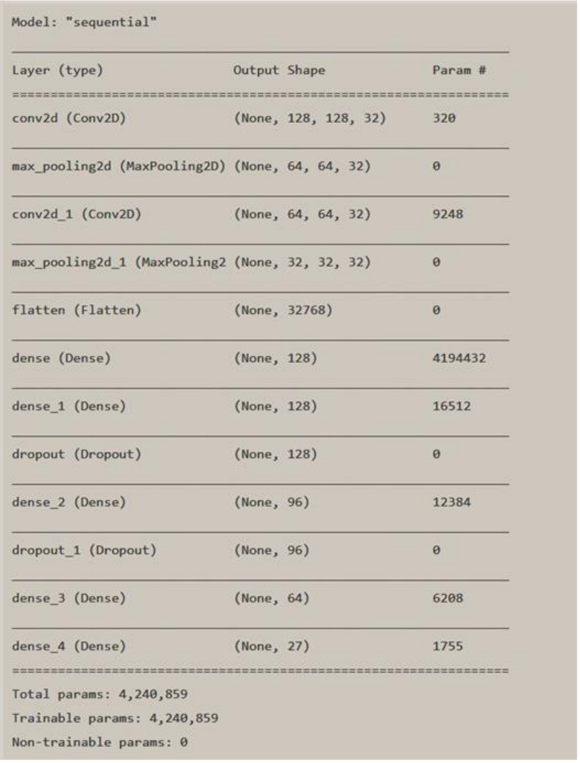
3. **Fully Connected Layer:** In the fully connected layer, all the inputs are connected to neurons, as opposed to the convolution layer where neurons are connected only to a local region.

4. **Final Output Layer:** After obtaining values from the fully connected layer, these are connected to the final layer of neurons, which is equal in count to the total number of classes. This layer predicts the probability of eachimage belonging to different classes.

C. Methodology

This section outlines the intricate process of data acquisition, pre-processing, model development, and evaluation. The methodology underscores the integration of advanced technologies to empower the ASL detection system, ensuring precision, adaptability, and real-time functionality. Each step in the methodology contributes to the holistic development of an efficient and user-centric solution for enhanced communication in the realm of sign language.

a. Model Summary:



```
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 7: Model Summary

b. Implementation

- The initial phase of this project involves the creation of directories for storing the training and testing data. Inthis project, a unique dataset is constructed.

- Following the creation of these folders, the next step is to generate the training and testing datasets. This is accomplished by capturing each frame displayed by the machine's webcam. Within each frame, a Region of Interest (ROI) is defined, which is represented by a blue bounded square. After capturing the image from the ROI, a Gaussian blur filter is applied to the image, aiding in the extraction of various image features. The image after applying gaussian blur looks like below.

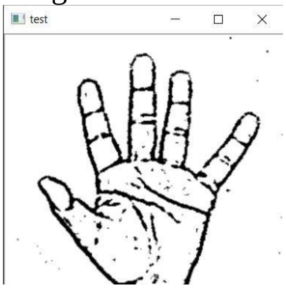


Figure 8: ROI

- Upon the completion of the training and testing data, the next phase involves the creation of a model fortraining. In this case, a Convolutional Neural Network (CNN) is utilized for model construction.

▪ **Training:**

The process begins with the conversion of input images (originally in RGB format) into grayscale. This is followed by the application of a Gaussian blur to eliminate unnecessary noise. An adaptive threshold is then applied to extract the hand from the background. The images are resized to a uniform size of 128 x 128 pixels for consistency.

These pre-processed images are fed into the model for both training and testing. The prediction layer of the model estimates the likelihood of the image belonging to one of the classes. The output is normalized between 0 and 1, ensuring that the sum of the values for each class equals 1. This normalization is achieved using the SoftMax function.

Initially, the output of the prediction layer may deviate from the actual value. To improve this, the networks are trained using labelled data. Cross-entropy, a performance measurement used in classification, is employed here. It is a continuous function that is positive at values that differ from the labelled value and is zero when it matches the labelled value.

The goal is to optimize the cross-entropy by minimizing it as close to zero as possible. This is done by adjusting the weights of the neural network within the network layer. TensorFlow provides an inbuilt function to calculate the cross-entropy.

Once the cross-entropy function is set up, it is optimized using Gradient Descent. The most effective gradient descent optimizer, known as the Adam Optimizer, is used in this case.

▪ **Testing:**

During the testing phase of the application, it was observed that some of the symbol predictions were incorrect. To address this, a two-layer algorithm was implemented to verify and predict symbols that are more similar to each other, with the aim of improving the accuracy of symbol detection.

In the tests, the following symbols were not correctly identified and were instead misclassified as other symbols:

- Symbol 'D' was misclassified as 'R' and 'U'
- Symbol 'U' was misclassified as 'D' and 'R'
- Symbol 'I' was misclassified as 'T', 'D', 'K', and 'I'
- Symbol 'S' was misclassified as 'M' and 'N'

To handle these cases, three different classifiers were created to classify the following sets of symbols:

- {D, R, U}
- {T, K, D, I}
- {S, M, N}

These classifiers aim to improve the accuracy of symbol prediction by focusing on sets of symbols that were found to be commonly misclassified as each other. This approach allows for more targeted error correction and improved overall performance of the sign language detection system.

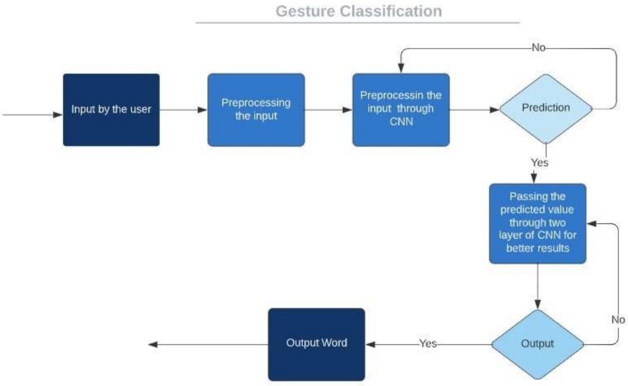

Gesture Classification
Input by the user
Preprocessing the input
Preprocessin the input through CNN
Prediction
No
Yes
Passing the predicted value through two layer of CNN for better results
Output
No
Yes
Output Word

Figure 9: Gesture Classification

- The final stage, following the training of the model, involves the creation of a Graphical User Interface (GUI). This GUI will be used to convert signs into text and form sentences, facilitating communication with individuals who are deaf and mute.

4. Result

When we are training the model, accuracy and loss in model for validation data could be variating with variouscases. Normally with every increasing epoch, loss should be going lower and exactness should be going higher.

But with validation loss (keras validation loss) and validation accuracy, numerous cases can be conceivable likeunderneath:

1) Validation loss starts increasing, validation accuracy starts diminishing. This implies model is cramming valuesnot learning.

2) Validation loss starts increasing, validation accuracy likewise will increase. This could be instance of overfitting or diverse probability values in situations where soft max is being used in output layer.

3) Validation loss starts decreasing, validation accuracy starts increasing. This is additionally fine as that impliesmodel fabricated is learning and dealing fine.

After testing our model, we have acquired the following results, we have plotted the graph of accuracy and losswith respect to epochs.

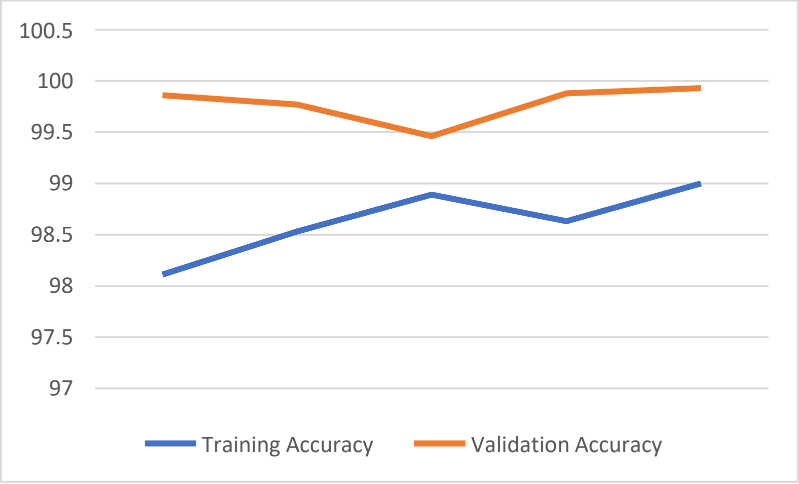


Figure 10: Training and Validation Accuracy

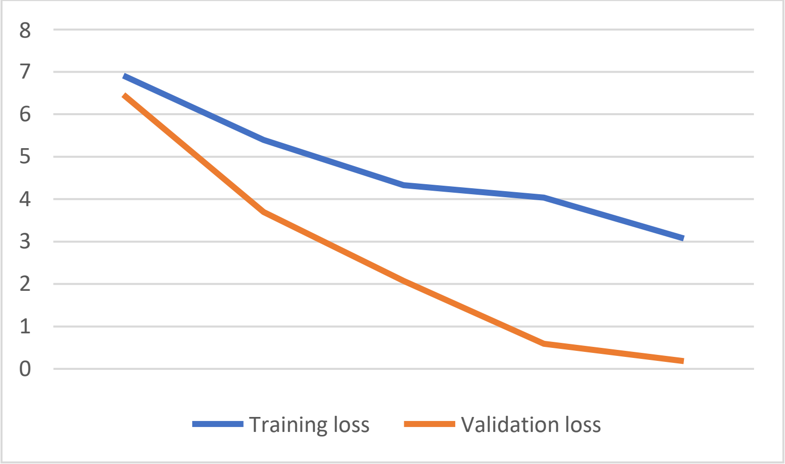


Figure 11: Training and Validation loss

In the above graphs, it has been seen that validation loss is decreasing and validation accuracy is increasing noticeably. We attempted to build a model using a Convolutional Neural Network in this project. This results ina validation accuracy of about 95.8% that was achieved in the model using only the first layer of the algorithm. By combining the first and second layers of the algorithm, the accuracy increased to 98.0%.

5. Future Scope

- Multilingual Support: Expand the system's capabilities to recognize and interpret sign languages beyond American Sign Language (ASL). Incorporate datasets and training methodologies to enable the model tounderstand a broader range of sign languages, promoting inclusivity on a global scale.
- Real-time Interpretation: Optimize the system for real-time sign language interpretation, enabling instantaneous recognition and translation of signs. This could facilitate dynamic communication between individuals using sign language and those unfamiliar with it in various settings, such as live events, conferences, or educational environments.

- Gesture Refinement: Further refine the system's ability to interpret subtle nuances in hand gestures and facial expressions within sign language. This could involve leveraging advanced computer vision techniques and machine learning algorithms to enhance the accuracy and granularity of gesture recognition.
- User-Friendly Applications: Develop user-friendly applications that leverage the sign language detection system for everyday use. This could include mobile applications, web platforms, or smart devices that empower individuals with hearing impairments to communicate effortlessly in diverse contexts.

6. Conclusion

The research delineated in this paper underscores the effective utilization of a Convolutional Neural Network (CNN) in the intricate task of detecting and classifying characters in American Sign Language (ASL), which is acknowledged as the most widely used sign language across the globe. The proposed method, ingeniously integrating the capabilities of the OpenCV and MediaPipe frameworks, exhibits a remarkable detection rate coupled with exceptional accuracy, thereby demonstrating its potential in real-world applications.

The crux of this research lies in facilitating communication with individuals who have hearing disabilities, a challenge that often leads to their exclusion from mainstream communication channels. By developing a system that can accurately interpret ASL, this research contributes significantly towards fostering inclusivity and promoting better understanding and interaction with individuals with hearing impairments.

The model, trained using a CNN, achieved a validation accuracy of 95.8% with just the first layer of the algorithm. This accuracy saw a substantial increase to 98.0% when both layers of the algorithm were employed, indicating the model's effective learning from the training data and its ability to generalize well to unseen data in the validation set.

These promising results pave the way for future research in this direction. Potential future work could explore the application of this model to other sign languages, thereby broadening its scope and impact. Additionally, integrating this system into real-time applications could revolutionize the way we communicate with individuals who use sign language, making communication seamless and more inclusive. This research, thus, marks a significant stride in leveraging technology to bridge communication gaps and foster a more inclusive society.

7. References

[1] Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, & Hongdong Li. (2020). Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison.

[2] Novopoltsev, M., Verkhovtsev, L., Murtazin, R., Milevich, D., & Zemtsova, I. (2023). Fine-tuning of sign language recognition models: a technical report. ArXiv, abs/2302.07693.

[3] Rupesh Kumar, Ashutosh Bajpai, & Ayush Sinha. (2023). MediaPipe and CNNs for Real-Time ASL

GestureRecognition.

[4] Joshi, A., Bhat, A., S, P.R., Gole, P., Gupta, S., Agarwal, S., & Modi, A. (2022). CISLR: Corpus for Indian Sign Language Recognition. Conference on Empirical Methods in Natural Language Processing.

[5] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.

[6] Indriani, Moh. Harris, & Ali Suryaperdana Agoes (2021). Applying Hand Gesture Recognition for User Guide Application Using MediaPipe. In Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021) (pp. 101-108). Atlantis Press.

[7] Shi, B., Rio, A.M., Keane, J., Michaux, J., Brentari, D., Shakhnarovich, G., & Livescu, K. (2018). American Sign Language Fingerspelling Recognition in the Wild. 2018 IEEE Spoken Language Technology Workshop (SLT), 145-152.

[8] Bowen Shi, Diane Brentari, Greg Shakhnarovich, Karen Livescu; Proceedings of the IEEE/CVF Conferenceon Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4166-4175

[9] Sharma, Chandra Mani and Tomar, Kapil and Mishra, Ram Krishn and Chariar, Vijayaraghavan M, Indian Sign Language Recognition Using Fine-tuned Deep Transfer Learning Model (September 20, 2021). Procc. of INTERNATIONAL CONFERENCE ON INNOVATIONS IN COMPUTER AND INFORMATION SCIENCE (ICICIS), pp 62-67, 2021, Jiangxi China, Available at SSRN: https://ssrn.com/abstract=3932929 or http://dx.doi.org/10.2139/ssrn.3932929

[10] Shin J, Matsuoka A, Hasan MAM, Srizon AY. American Sign Language Alphabet Recognition by Extracting Feature from Hand Pose Estimation. Sensors. 2021; 21(17):5856. https://doi.org/10.3390/s21175856

[11] Veluri, R.K., Sree, S.R., Vanathi, A., Aparna, G., Vaidya, S.P. (2022). Hand Gesture Mapping Using MediaPipe Algorithm. In: Bindhu, V., Tavares, J.M.R.S., Du, KL. (eds) Proceedings of Third International Conference on Communication, Computing and Electronics Systems. Lecture Notes in Electrical Engineering, vol 844. Springer, Singapore. https://doi.org/10.1007/978-981-16-8862-1_39

[12] Goel, P., Sharma, A., Goel, V., & Jain, V. (2022, November). Real-Time Sign Language to Text and Speech Translation and Hand Gesture Recognition using the LSTM Model. In 2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) (pp. 1-6). IEEE.

[13] Liu, Z., Pan, C., & Wang, H. (2022). Continuous Gesture Sequences Recognition Based on Few-Shot Learning. International Journal of Aerospace Engineering, 2022.

[14] Sharma, A., & Kumar, Y. (2021). Hand Gesture Recognition using Machine Learning and Computer Vision.

[15] Pedersoli, Fabrizio & Benini, Sergio & Adami, Nicola & Leonardi, Riccardo. (2014). XKin: an Open Source Framework for Hand Pose and Gesture Recognition Using Kinect. The Visual Computer: International Journal ofComputer Graphics. 10.1007/s00371-014-0921-x.

[16] https://www.kaggle.com/datasets/grassknoted/asl-alphabet

[17] Ameen, S. & Vadera, S. A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. Expert Syst. 34(3), e12197. https://doi.org/10.1111/exsy.12197 (2017).

[18] https://omdena.com/wp-content/uploads/2021/12/MediaPipe-Python-Hand-Tracking.png

[19] https://github.com/luvk1412/Sign-Language-to-Text/blob/master/presentation.pdf

[20] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53