Sign Language Recognition Using Convolutional

Neural Network

Shourya Jha Ayush Rai T. Muskan

School of ECE School of ECE School of ECE

Reva University Reva University Reva University

Bengaluru, Karnataka Bengaluru, Karnataka Bengaluru Karnataka

[Shouryajha480@gmail.com](mailto:Shouryajha480@gmail.com) [ayushrai420559@gmail.com](mailto:ayushrai420559@gmail.com) [muskanmusu812@gmail.com](mailto:muskanmusu812@gmail.com)

Varshitha L Madan H T

School of ECE Assistant Professor

Reva University School of ECE

Bengaluru, Karnataka Reva University, Bengaluru, Karnataka

[Varshithavarsh24@gmail.com](mailto:Varshithavarsh24@gmail.com) [madanht@reva.edu.in](mailto:madanht@reva.edu.in)

1. ABSTRACT

**Communication is how to express thoughts and emotions in life, it is also the key to sharing knowledge. Communication takes many forms, one of which is sign language. The Deaf community and the hearing majority have an undeniable communication problem. Automatic sign language recognition innovations are attempting to break down this communication barrier. Our proposed method aims to create a system that takes continuous gestures from a video capturing device and translates the hand gesture to recognize the movements. Our proposed method consists of 4 stages – creating an image capturing model and gathering images for the dataset, making the machine learning model, training the model, and finally testing the model and recognizing the gestures using a live camera feed. We could successfully detect 26 alphabets with an accuracy of 96.41% by training 1750 images per alphabet.**

1. KEYWORDS

Model, sign, language, accuracy, image, recognition, dataset, training, neural, machine, testing, training, gesture, network, convolution, learning.

1. INTRODUCTION

The whole world comprises different types of human beings, depending on different ways to communicate with each other where sign language is an exquisite language used by individuals with hearing loss. How exactly does this ensue?

Humans have five senses in their bodies, one of which is auditory perception (hearing). Hearing allows us to perceive sounds and noises and allows us to interact. A sound source generates sound waves that travel through the air and are caught by the outer ear. The sound waves then travel through a slender passage called the ear canal and reach the eardrum (Tympanic membrane); a flap of skin stretched tight like a drum. The sound waves from the source strike the eardrum, which causes vibrations in the bones of the middle ear, specifically the malleus, incus, and stapes. The anatomy of the human ear is depicted in Figure 1.

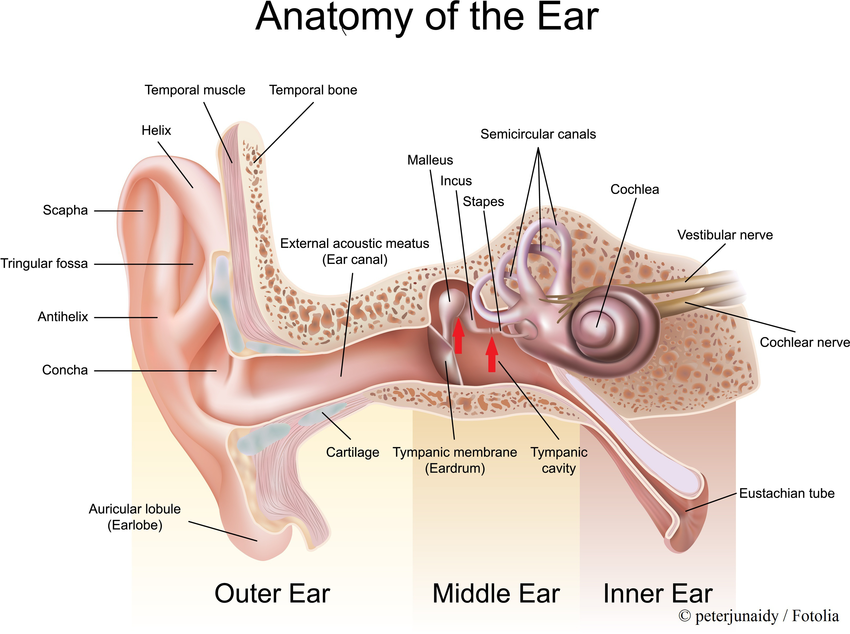


Fig 1: Anatomy of Human Ear

The vibrations are amplified and released into the cochlea by these bones. The cochlea is shaped like a snail and is made of fluid; vibration disturbs the fluid and causes ripples, which affects the hair-like structures called stereocilia that form in the group. Each hair cell on the cochlea's baseline generates an electrical signal in response to vibration effects from stereocilia, which is then carried by auditory nerves to the brain, where the brain predicts, recognises, and understands the sounds and rhymes. However, a single person can lose their ability to hear sound partially or completely during the transmission of sound waves from ear to brain, which is known as hearing impairment. Conductive hearing loss occurs when sound waves are blocked from reaching the inner ear. Sensorineural hearing loss is caused by an injury to the inner ear or the nerves that transmit sound to the brain and is more likely to be permanent. Hearing loss can occur at any age and can be caused by a variety of factors. People can lose their hearing power unexpectedly because of a virus or gradually lose their auditory perception due to disease, nerve damage, or injury caused by noise. Due to a variety of factors, both partial and total hearing loss can occur. Ear infections, fluid build-up in the back of the eardrum, holes in the eardrum, and problems with the middle ear bones can all cause conductive hearing loss. Tumours can also cause conductive hearing loss in rare cases, as they block sound from entering the ear.

Hearing loss is the third most common physical disability, following heart disease and arthritis. In this case, an Otoscope, which is a small handheld instrument with a light, is used to check for the presence of any blockage or ear infection, and the results are then sent to otologists. Even though we have the technology to detect hearing loss, sign language is the only way to communicate. In full communication, sign language is the movement of hands to convey meaning. Different countries have their sign language with their alphabets and word sets, such as America's American Sign Language (ASL) and Germany's German Sign Language (GSL). The similarity is that sign languages make use of hand movements. They can also communicate with their facial expressions, eyes, and heads. Because sign language is not universal, it is uncommon among healthy people and takes time to understand and interact with deaf or hard of hearing people. There is a need for hand gesture recognition, so our proposed system suggests a convolutional neural network that uses a live camera feed as input and could eventually achieve the goal of recognising sign language.

1. LITERATURE SURVEY

Izzah [1] proposed a technique where 380 images were converted from RGB to YCbCr to reserve luminance. Further to pull out the features Generic Fourier Descriptor is applied and filtering using a mask is done. Ultimately for testing purposes, the pre-processed dataset is fed to Support Vector Machine, Principal component analysis with a performance percentage of 81.39 and 61.36. The work presented in [2] uses 2D wavelet transform to pull out the features and then uses multiclass SVM for classification. Their methodology involves image acquisition, processing of the image, feature extraction and then classification. The image was recorded and altered to black and white images and rescaled. Wavelet transformation was applied to these pre-processed images. The feature vector obtained was given as input to SVM. They achieved an accuracy of 94% after training and testing over 350 samples. The work presented in [3] proposes a model where initially pre-processing is done with 720 gesture images by resizing and global thresholding method and maintain the structural properties by canny edge detection, further usage of the histogram was seen to extract the features and eventually the results were observed using support vector machine with an accuracy of 93.75%.In the work presented in [4], the photographs are initially captured, and image pre-processing is done to eliminate the unwanted noise and regulate the brightness. Image segmentation is carried out to spot the hand objects and edges of images. Then apply image analysis and convexity algorithm to bring out the outline position of the hand. This method can only be used to detect numbers and hence becomes a drawback. Sign in the form of gestures is given as an input to the system proposed in [5] segmentation phase is performed based on the skin colour to identify the shape of the sign. The region identified is then modified into a binary image on which the Euclidean distance transformation was applied at a later stage. On the distance transformed image, Row and column projection is applied. For extracting the features central moments together with HU’s moments are used. For classification, neural networks and SVM are used. The average accuracy for 13 feature sets obtained is 92.12%.

In the work presented in [6], Identification of Sign language was done using a gyroscope, accelerometer, and a hardware module to control the sensors. An improved K- mean algorithm has achieved an average accuracy of 96.55 %. The static gestures were recognized effectively as compared to continuous gestures.

In the work presented in [7], recognition is done by converting the hand moves into an oral language. Flex sensors are placed inside the gloves on every finger which helps to get data employing various methods, the K-Nearest Neighbours (KNN) is used as a classifier. They obtained an average accuracy of 87.8% in drop3. The work presented in [8] uses high-level CNN to identify the static sign images and processes them. RGB dataset is used. It classifies ten digits, 23 alphabets of English and 67 commonly used words. Different optimizers are used compared. 99.17% of training accuracy and 98.80 % of validation accuracy were achieved. Shreyashi Narayan Sawant [9] suggests a technique using MATLAB. Otsu algorithm has been used as a part of prepossessing of the image where the resolution of the image taken is 380 x 420 pixels. Principle component analysis has been implemented to find out more descriptive data from the image by using Eigenvectors and ultimately recognizing the hand gestures. The work presented in [10] describes the usage of simple CNN architecture to recognize the sign language. The datasets used were SIBI (Indonesian Language Signal System) and ASL (American Sign Language) to test and train the model. Using the above method, they attained an accuracy of 96.82% and this was compared with AlexNet. Alina Kuznetsova [11] has proposed a method using a publicly available dataset on which an ensemble of shape function (ESF) descriptor is applied and applied to real-time data. And eventually using the multi-layered random forest to differentiate between clusters using features decided by the algorithm. The work presented in [12] focuses on vision-based techniques to identify using principal component analysis. At first, the hand region is divided by using the skin colour model in the YCbCr colour space. Next, we apply thresholding to split the background and at last template-based matching is developed using principal component analysis. They achieved an accuracy of 91.25% after testing with 20 images for each gesture. In the work presented in [13], the objective is from sign language videos to generate spoken language translations, considering grammar and the order of words different word orders and grammar. To evaluate the approach, the first continuous sign language translation dataset, PHOENIX14T has been collected. An identifying method using the K Nearest Neighbour classifier was proposed in [14]. According to the study they made, there was a change in the accuracy when the pattern is represented by full dimensions compared to when it was represented by PCA reduced dimension feature. From their study, we were able to conclude that the KNN classifier is more preferred with full dimension in place of reduced dimension. In the work presented in [15], SubUNets a unique approach to unravelling simultaneous alignment and recognising problems named “Sequence-to-sequence” learning are being employed. A framework for modelling the subunits of learning. The matter has been decomposed into a specialized unit called subunits. The work presented in [16] collected the dataset using Microsoft Kinect. The depth from the dataset along with the segmented static model gave an accuracy of 98.81 % and with the dynamic model, 99.08 % accuracy was recorded. The model could classify the data into 36 different classes.

1. MOTIVATION

The Convolution Neural Network method can achieve high accuracy for sign language recognition. The motivation is to achieve comparable results for the machine learning model and to ease the communication gap with the deaf community.

1. DOMAIN

*Machine Learning* is the study of computer algorithms that can improve themselves automatically based on experience and data. It is regarded as a component of artificial intelligence.

*Deep Learning* is part of a broader family of machine learning methods based on artificial neural networks with representation learning.

A *Convolutional Neural Network* (CNN) is a type of artificial neural network (ANN) that is commonly used in deep learning to analyse visual imagery.

1. PROBLEM DEFINITION

Disabled people use sign language for communication; hence others have difficulty in communicating with them which creates segregation for them in society. Hence, there is a need for a system which recognizes the different signs and conveys the information to the rest.

1. PROBLEM STATEMENT

The Deaf community have an undeniable communication problem. Automatic sign language recognition innovations are attempting to break down this communication barrier using machine learning.

1. INNOVATIVE CONTENT

Our model can recognise the 26 alphabets, and we achieved an accuracy of 96 to 98 per cent while training the model and 94 to 96 per cent while using the validation set. As a result, our model can recognise gestures with high accuracy and is suitable for hearing-impaired people.

1. REPRESENTATION OR DESIGN

The proposed system has four major steps that guide us to achieve the objective. The first step is the generation of the dataset using a capturing device followed by building the model and training our machine learning model. Finally testing the accuracy of the trained machine learning model.

A window of 64 x 64 pixels is drawn while capturing which contains the gesture image. We have used an HSV colour model for the images. After the images are captured, they are saved locally in the gesture name folder respectively.

After all of the images have been captured, they are labelled one by one with the LabelImg package. The images and their XML files are then separated into training and validation data. Each alphabet has 1750 images captured for the training dataset and 250 images captured for the validation dataset.

Keras has a useful API which makes us easier to define the layers of our neural network. Here the input shape is 64,64 which is our image size and 3 represents colour channel RGB. If it is a grayscale image, we should specify it as 1.

Conv2D(): Neural networks apply a filter to an input image to create a feature map that summarizes the presence of detected features in the input. In our case, there are 32,64,128 and 128 filters or kernels in respective layers and the size of the filters are 3X3 with activation functions as relu.

MaxPool2D(): Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after the max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

Flatten(): This method converts the multi-dimensional image data array to a 1D array.

We have used Convolutional Neural Network for the prediction of the Sign Language. The architecture of the Convolutional Neural Network is shown in figure 2.

Diagram

Description automatically generated

Fig 2: Architecture of CNN

A convolution layer is added that takes a 64 x 64 image as input. This layer's output is routed to a MaxPooling Layer. This Convolution and MaxPooling layer pair are repeated. A Flatten layer is added after three pairs of Convolution and MaxPooling layers. The Flattening layer reduces the two-dimensional feature matrix to a one-dimensional feature vector. To overcome the model's overfitting, a Dropout layer is added. Finally, a dense layer is added to form the Convolutional Neural Network. As an optimizer, standard gradient descent is used to improve the performance of the Convolutional Neural Network.

We can train our model by invoking the fit generator() function, which accepts both our training and validation images as input for training and validation. We should also specify epochs and steps per epoch. The most common method for determining steps per epoch is the number of train images/batch size. In our case, it is approximately 800. The trained model is saved as a file in the format of an h5 file.

For all values, the HSV values for the camera feed are set using a track bar. The model predicts the Sign for the indicated gesture in the camera after the values are fine-tuned.

1. METHODOLOGY

The proposed system has four major steps that guide us to achieve the objective. The first step is the generation of the dataset using a capturing device followed by building the model and training our machine learning model. Finally testing the accuracy of the trained machine learning model completes the process. Figure 3 explains the workflow followed to reach the objective. The dataset is generated by using an image capturing device. The dataset is then divided into validation and training sets. Then the model that needs to be trained is fed with a training dataset. We have used a Convolutional Neural Network to make the predictions for the sign language. This model after training is then tested against the dataset which is created by performing a split on the overall dataset. At last, the model is given images from the live camera feed and the predicted alphabets are displayed as output.

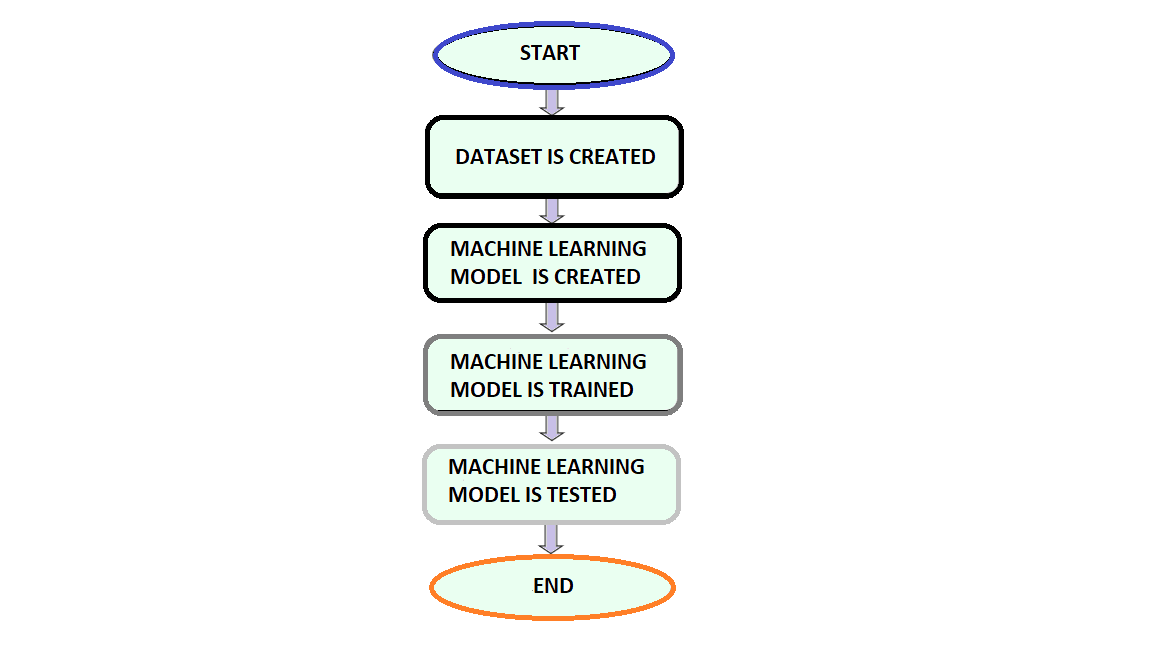


Fig 3: Flowchart for sign language recognition

1. Generation Of Dataset

The images are captured using an image capturing device. A window of 64 x 64 pixels is drawn while capturing which contains the gesture image. We have used an HSV colour model for the images. After the images are captured, they are saved locally in the gesture name folder respectively. The images are already in the size of 64 x 64 pixels as it was normalized while capturing. In this way, the quality of the images is also ensured. Figure 3 displays each gesture from our dataset.  Dependencies such as cv2, os, and time have been imported for data generation. The dependency os is used to assist with working with file paths and it provides functions for interacting with operating systems. Time in Python can be represented in code in a variety of ways, including objects, numbers, and strings, thanks to the time module. It can be used to measure code efficiency or wait during code execution in addition to representing time. It is used here to insert breaks between image captures to allow for hand movements.

After all of the images have been captured, they are labelled one by one with the LabelImg package. When you save a labelled image, an XML file is created. The XML files for all of the images are now available after they have been labelled. This is used to generate TF (TensorFlow) records. The images and their XML files are then separated into training and validation data. Each alphabet has 1750 images captured for the training dataset and 250 images captured for the validation dataset.

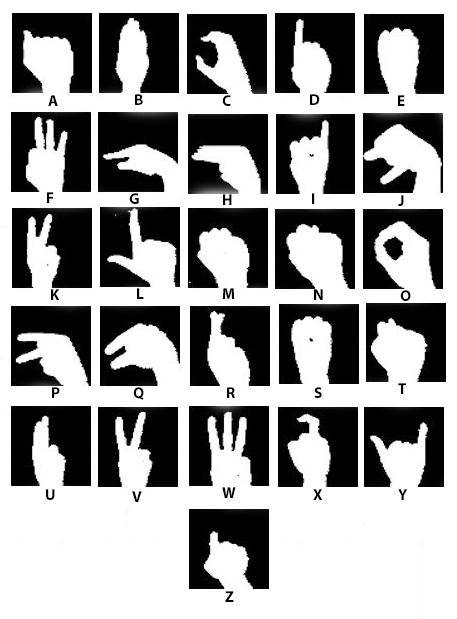


Fig 4: Gestures for each alphabet

1. Building Our Machine Learning Model

We have used Convolutional Neural Network for the prediction of the Sign Language. The architecture of the Convolutional Neural Network is shown in figure 4. A sequential layer is initialized using the Keras library and we have added some layers for increasing the performance of the Convolutional Neural Network. A convolution layer that takes an image of 64 x 64 as input is added. The output from this layer is passed on to a MaxPooling Layer. This pair of Convolution and MaxPooling layers is repeated. After three pairs of Convolution and MaxPooling layers, a Flatten layer is added. The Flattening layer converts the 2 D feature matrix into a one-dimensional feature vector. A Dropout layer is added to overcome the overfitting of the model. And finally, a dense layer is added which constitutes the Convolutional Neural Network. Standard gradient descent is used as an optimizer for increasing the performance of the Convolutional Neural Network.

1. Training Our Machine Learning Model

The training dataset is used to perform the training of our model. The proposed model is trained for 40 epochs with a step size of 800 per epoch. The validation step is set to 6500. The trained model is saved as a file in the format of an h5 file. This is done so that the model can be used on different systems. The accuracy of the training set was found to be 96.41% with an error value of 0.14. These values are justified in figure 5.

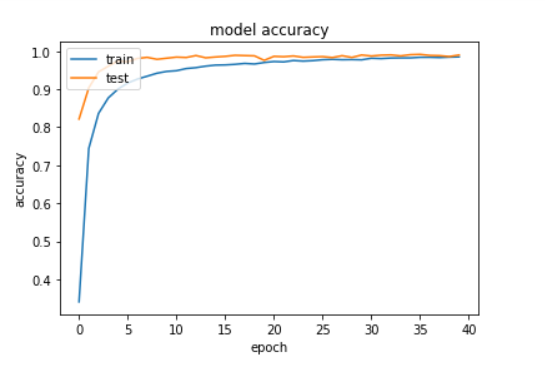


Fig 5: Accuracy plot of our model

1. Testing our Machine Learning Model

The test data is fed into the model to see its performance of the model. The model performs well on the test data as well with an accuracy of 94.21%.

Finally, the live camera feed is connected to the model for real-time detection of the Sign Language. The HSV values for the camera feed are set using a track bar for all the values. After the values are fine-tuned, the model predicts the Sign for the indicated gesture in the camera.

1. Results and ANALYSIS

We propose a model that recognizes the Sign language using a convolutional neural network. The input is given as a live image feed. The proposed model is trained on a labelled dataset that contains images of all 26 alphabets. Our model can identify all the 26 alphabets with considerable accuracy.

Table I

From our results we can conclude that our trained model can recognize the alphabets with the following accuracy values:

Accuracy = 96.41 % (during the training of the model)

Accuracy = 94.21 % (while using the validation Set)

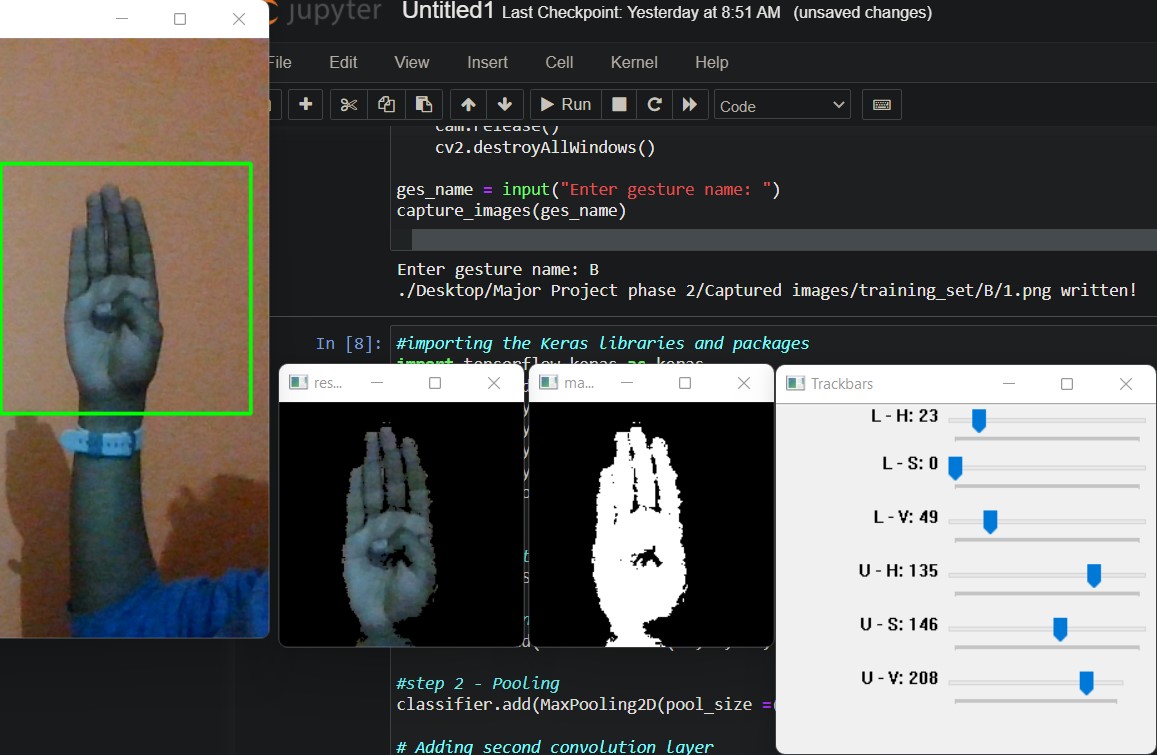


Fig 6: Capturing of Dataset

The images are captured by pressing the trigger key(C). The image is normalized and masked in the process as shown in Figure 6.

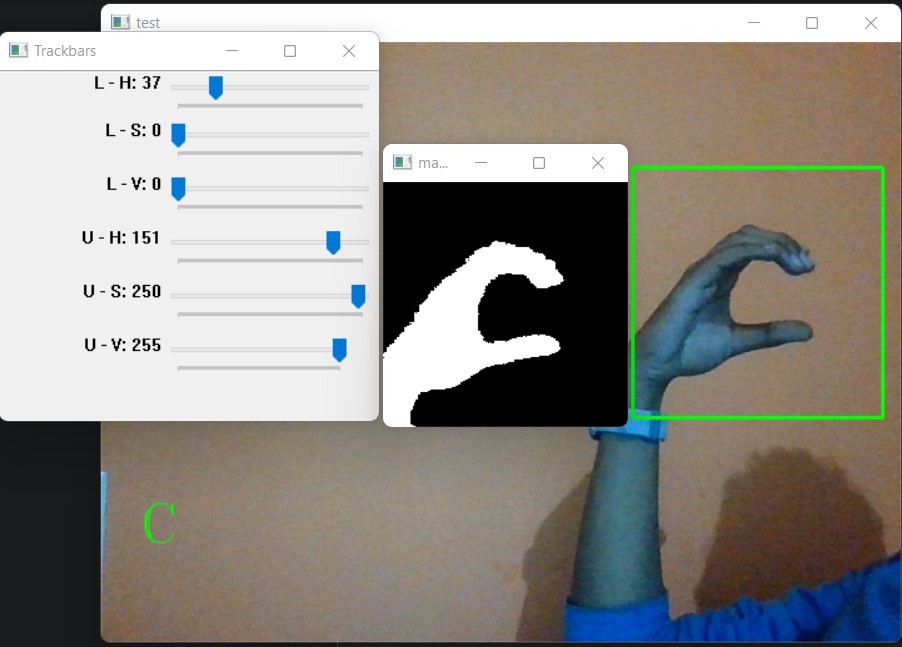


Fig 7: Testing the Model(C)

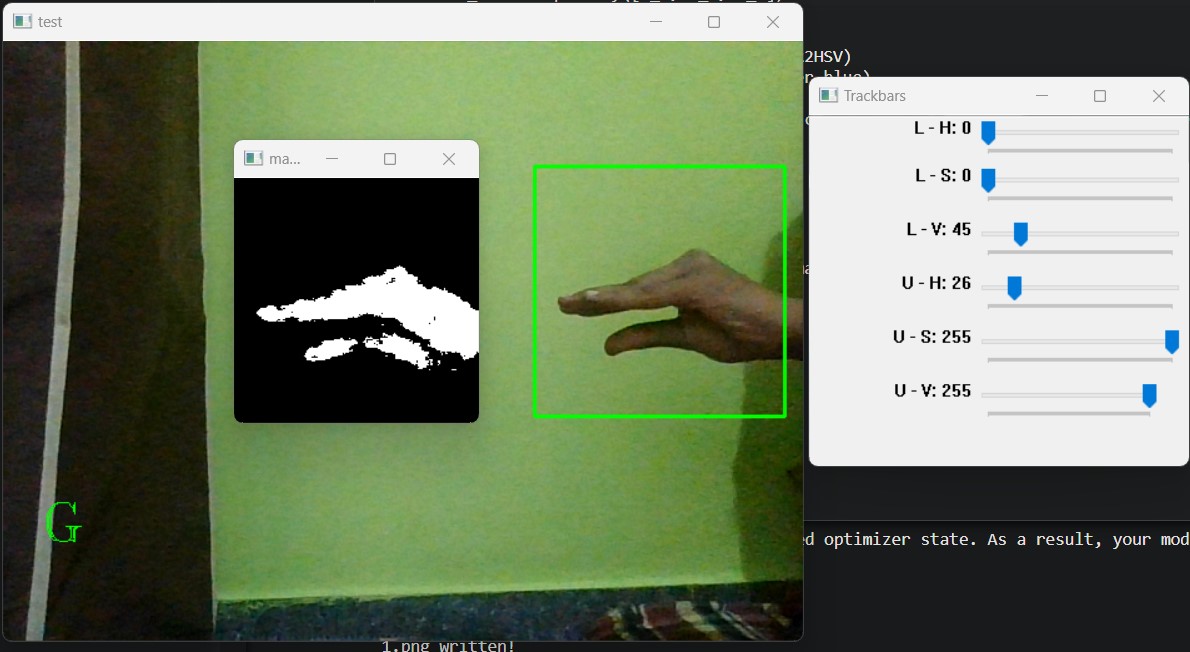


Fig 8: Testing the Model(G)

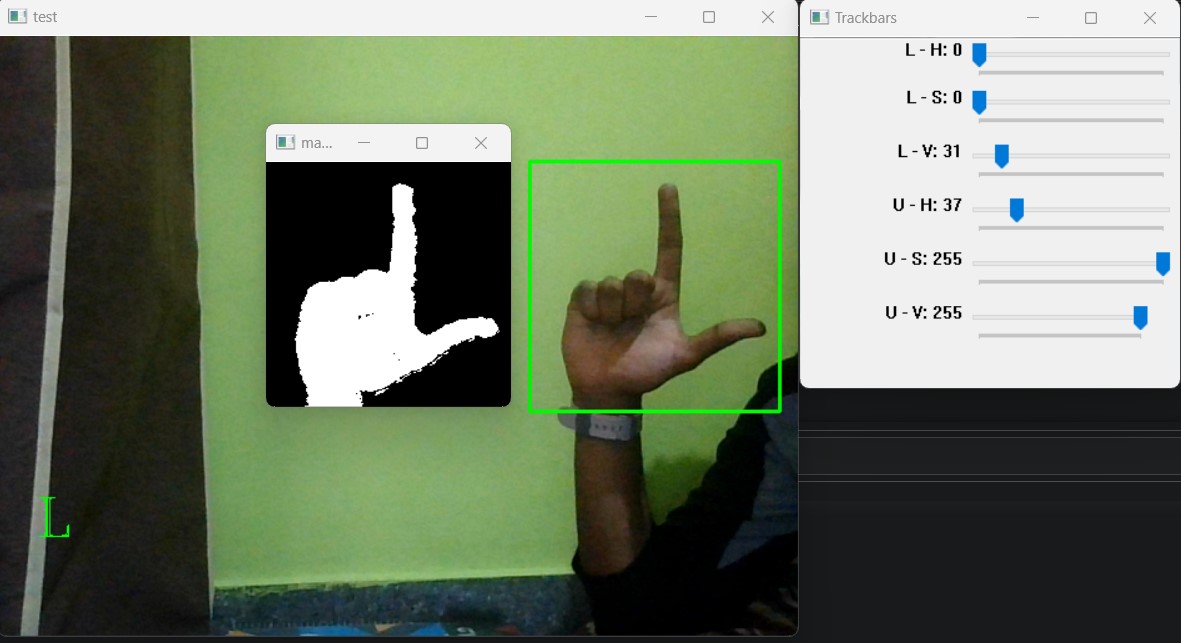


Fig 9: Testing the Model(L)

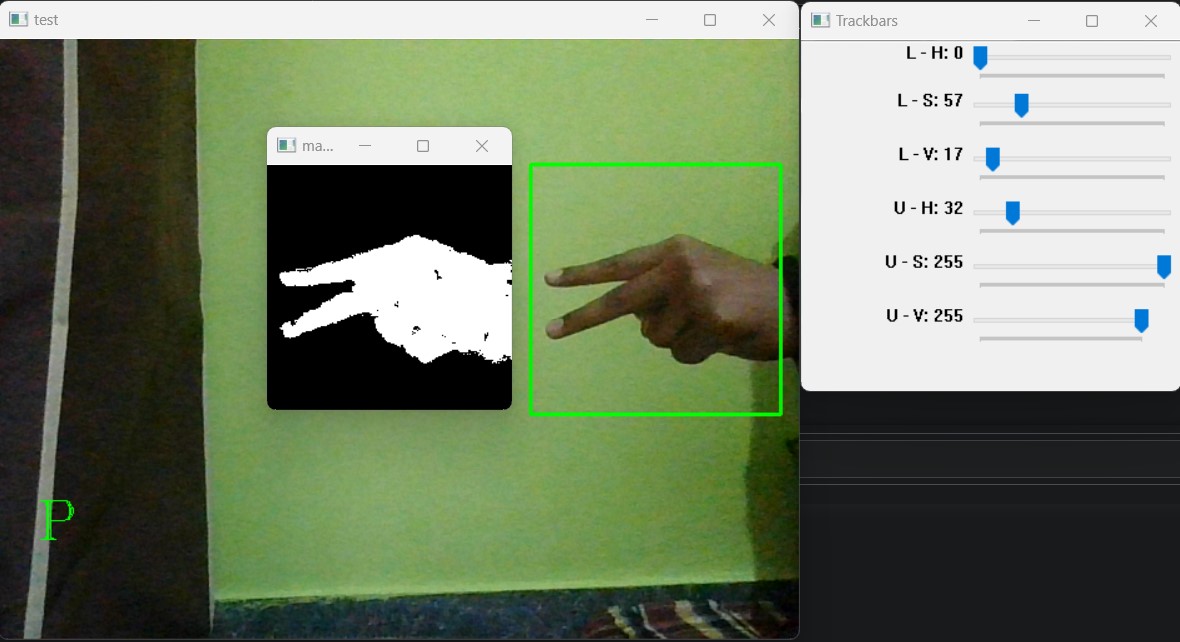


Fig 10: Testing the Model(P)

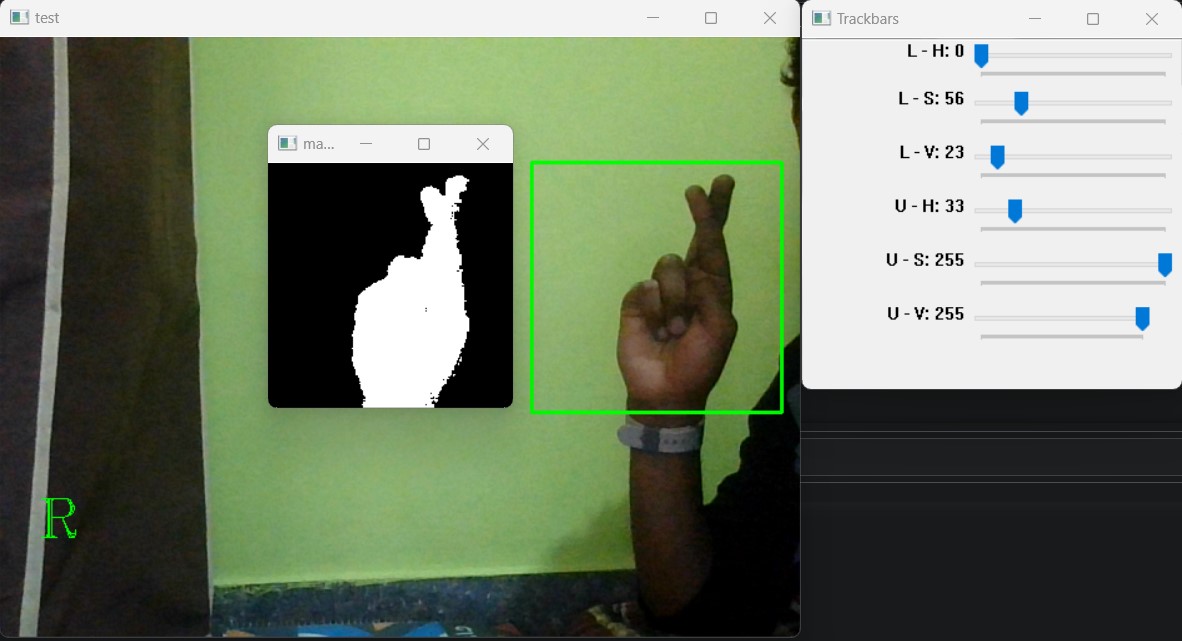


Fig 11: Testing the Model(R)

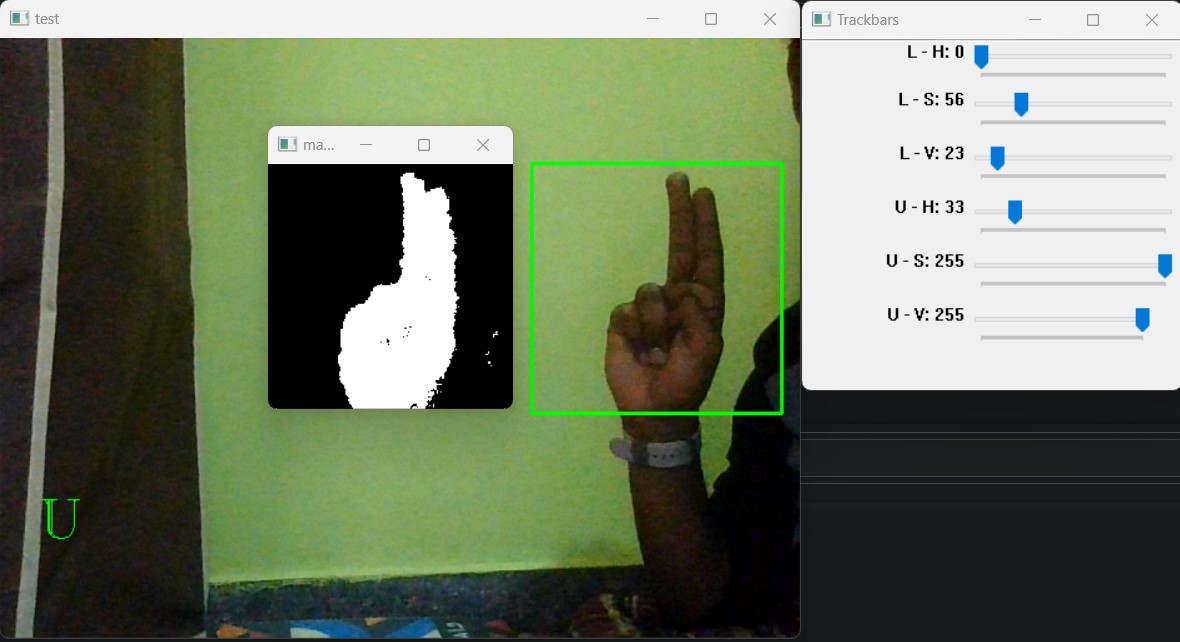


Fig 12: Testing the Model(U)

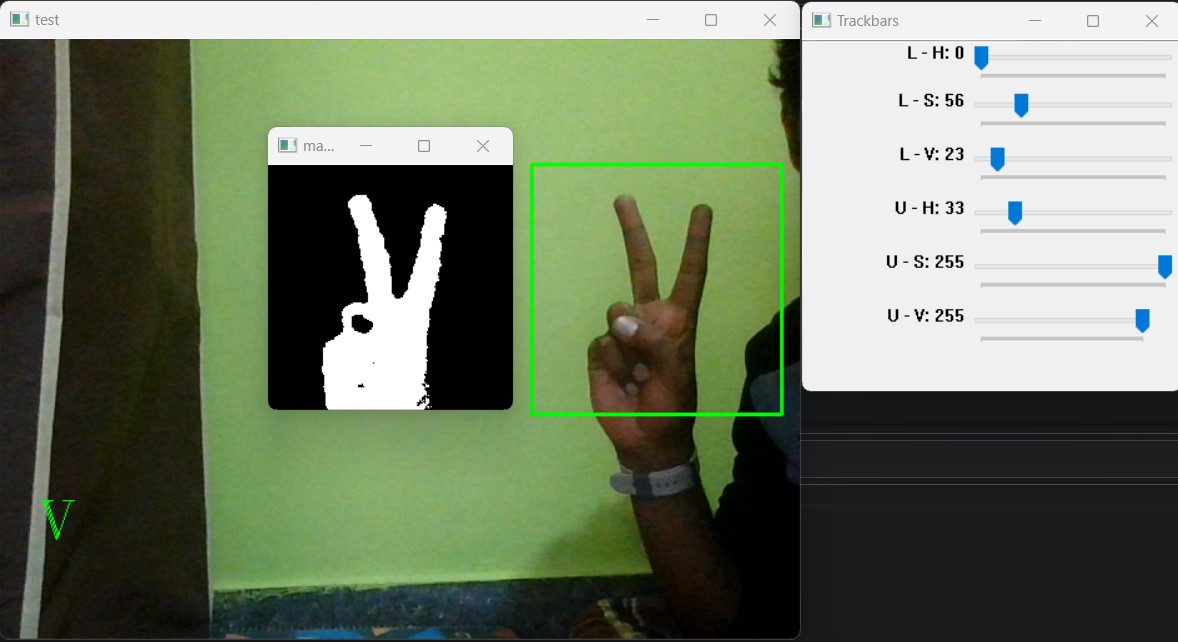


Fig 13: Testing the Model(V)

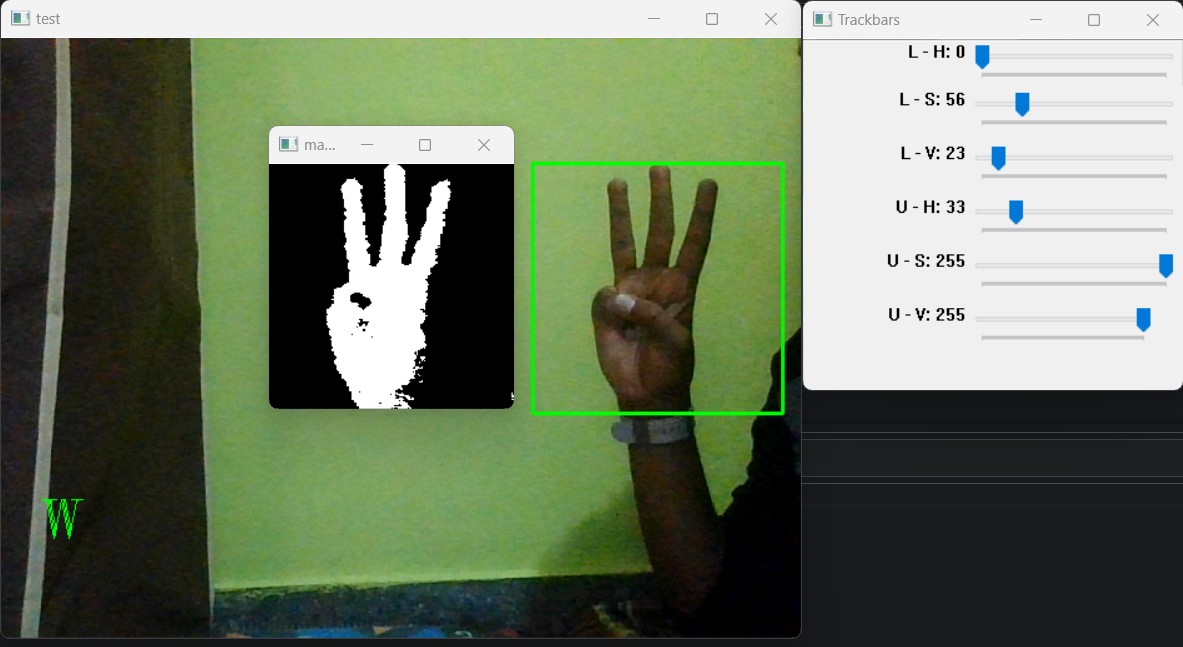
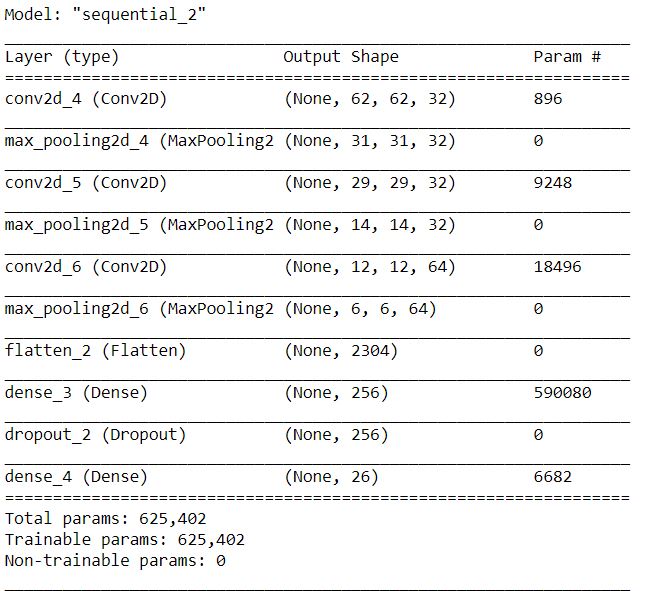


Fig 14: Testing the Model(W)

The model is given images from the live camera feed and the predicted alphabets are displayed as output in the bottom-left section of the test window. The model is tested with the alphabet C, G, L, P, R, U, V and W as shown in the figures above. It detects sign language with high accuracy as proposed.

1. DATA MODEL



1. COMPARISON OF RESULTS

|  |  |  |
| --- | --- | --- |
| Paper | Method | Accuracy  (in %) |
| Nagarajan [3] | Support Vector Machine | 93.75 |
| Agarwal Rajat [2] | Multiclass Support Vector Machine | 94 |
| Rokade [5] | Support Vector Machine | 92.12 |
| Izzah [1] | Support Vector Machine | 81.39 |
| Proposed method | Convolutional neural network | 96.41 (training)  94.21 (validation) |

1. JUSTIFICATION OF RESULT

Our model can identify all 26 alphabets with considerable accuracy. The accuracy of the training set was found to be 96.41% with an error value of 0.14. The model performs well on the test data as well with an accuracy of 94.21%. By using the convolution neural network of deep learning, we achieved higher accuracy and minimal loss.

1. Conclusion

Sign language recognition using a convolutional neural network was introduced in our work. The machine learning model is created using the Keras API with TensorFlow as the backend. The model was trained using a labelled dataset with 1750 images. Our model can identify the 26 alphabets and we have achieved an accuracy of 96.41% during the training of the model and an accuracy of 94.21% while using the validation set. So, our model can recognize the gestures with considerable accuracy and is feasible for hearing impaired people.

1. future scope

The proposed method for the identification of sign language using a Convolutional neural network identifies the 26 alphabets of the English language. In future, we can improvise our model to recognize numeric characters. Further can be improved by recognizing some of the words which are commonly used. For more convenient use of our model, we can develop a mobile application.

1. Acknowledgement

We, students at REVA UNIVERSITY, would like to express our gratitude towards our Hon’ble Chancellor, Dr P. Shyama Raju, Hon’ble Vice-Chancellor, Dr S Y Kulkarni and our respected Director, Dr Rajshekhar C. Biradar for providing us with the opportunity to enhance our knowledge in an environment that promotes growth and innovation. We would also like to thank our Project Guide, Prof. Madan H T, for his constant support and guidance and for mentoring us throughout the project. We are also thankful to all faculty members for their kind contributions.

1. REFERENCES

[1] Izzah, Abidatul, and Nanik Suciati. "Translation of sign language using generic Fourier descriptor and nearest neighbour." *International Journal on Cybernetics and Informatics* 3, no. 1 (2014): 31-41.

[2] Agarwal, Rajat, Balasubramanian Raman, and Ankush Mittal. "Hand gesture recognition using discrete wavelet transform and support vector machine." In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 489-493. IEEE, 2015.

[3] Nagarajan, Sathish, and T. S. Subashini. "Static hand gesture recognition for sign language alphabets using edge-oriented histogram and multi-class SVM." *International Journal of Computer Applications* 82, no. 4 (2013).

[4] Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image-based hand gesture recognition techniques." In *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, pp. 1-5. IEEE, 2016.

[5] Rokade, Yogeshwar I., and Prashant M. Jadav. "Indian sign language recognition system." *International Journal of Engineering and Technology* 9, no. 3 (2017): 189-196.

[6] Ma, Daniel Nareshkumar, and Vijayalakshmi Sb. "Recognition and Analysis of Indian Sign Language Using Improved K-means Algorithm." *European Journal of Molecular & Clinical Medicine* 7, no. 11: 2020.

[7] Rosero-Montalvo, Paul D., Pamela Godoy-Trujillo, Edison Flores-Bosmediano, Jorge Carrascal-García, Santiago Otero-Potosi, Henry Benitez-Pereira, and Diego H. Peluffo-Ordóñez. "Sign language recognition based on intelligent glove using machine learning techniques." In *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, pp. 1-5. IEEE, 2018.

[8] Wadhawan, Ankita, and Parteek Kumar. "Deep learning-based sign language recognition system for static signs." *Neural Computing and Applications* 32, no. 12 (2020): 7957-7968.

[9] Sawant, Shreyashi Narayan, and M. S. Kumbhar. "Real-time sign language recognition using PCA." In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 1412-1415. IEEE, 2014.

[10] Pratama, Yohanssen, Ester Marbun, Yonatan Parapat, and Anastasya Manullang. "Deep convolutional neural network for hand sign language recognition using model E." *Bulletin of Electrical Engineering and Informatics* 9, no. 5 (2020): 1873-1881.

[11] Kuznetsova, Alina, Laura Leal-Taixé, and Bodo Rosenhahn. "Real-time sign language recognition using a consumer depth camera." In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 83-90. 2013.

[12] Ahuja, Mandeep Kaur, and Amardeep Singh. "Static vision-based Hand Gesture recognition using principal component analysis." In *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, pp. 402-406. IEEE, 2015.

[13] Campos, Necati Cihan, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. "Neural sign language translation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7784-7793. 2018.

[14] Aryanie, Dewinta, and Yaya Heryadi. "American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier." In *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pp. 533-536. IEEE, 2015.

[15] Campos, Necati Cihan, Simon Hadfield, Oscar Koller, and Richard Bowden. "Subunits: End-to-end hand shape and continuous sign language recognition." In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3075-3084. IEEE, 2017.

[16] Bhagat, Neel Kamal, Y. Vishnusai, and G. N. Rathna. "Indian Sign Language Gesture Recognition using Image Processing and Deep Learning." In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1-8. IEEE, 2019.