Medicon Engineering Themes
Volume 6 Issue 1 January 2024
Article Type: Research Article
ISSN: 2834-7218

Study and Analysis of Deep Learning Models for the Recognition of Sign Language

Naima Azim*, Shamsia Afrin Jamema and Naznin Sultana

Daffodil International University, Dhaka, Bangladesh

***Corresponding Author:** Naima Azim, Daffodil International University, Dhaka, Bangladesh.

Received: December 13, 2023; **Published:** January 04, 2024

Abstract

A population of 430 million people and above, or over a population of 5% of the world's population, needs therapy to treat their "disabled" hearing and speaking condition. These people have the option to learn sign language to communicate with others. Hence, our project mainly targets the deaf and mute community. Around 5000 images of hand gestures have been used and divided into 10 categories for live detection. The categories are mainly American Sign Language (ASL) and are consisted of the first 10 numbers. Our model can detect these ten hand motions and categorize them correctly. We used the You Only Look Once Version 5 algorithm. The algorithm consists of a backbone namely CSPDarknet53, in which an SPP block is accustomed to accelerating the speed of the receptive field responsible to set apart prime traits and confirming that network operation speed is inclining in speed. The neck of the algorithm, PAN, is added to aggregate the parameters from different backbone levels. This model is very easy to use and understand and gives an accuracy above 98%. That is why we chose YoloV5 as our model for object detection due to its simplicity in usage. Therefore, an artificial sign language detection system has been suggested in this study which incorporates deep learning and image processing method. This study also gives a comparison between the two models to give a better understating of why we marked YoloV5 as a better algorithm even though both models gave an accuracy of above 98%. We believe that making a hand gesture detection system will encourage individuals to communicate with people who cannot hear or speak. That being the case, we aim to make the lives of the disabled better.

Keywords: Sign Language Detection; Deep Learning; YoloV5

Introduction

Communication is an important part of people's daily routines. It helps people to express many emotions and interact with others. Communication is also a vital way to educate people, learn from others, enlighten people with information, etc. This makes it very difficult for the deaf or mute to converse with others as they are unable to speak or listen to others [1]. Hence, sign language is learned by the disabled. Unfortunately, not everyone knows the sign language for which these disabled people need to hire a professional interpreter which can be very costly. Also, not all deaf or mute people know all the sign language as there are different sign languages for different countries for which we decided to make a sign language detecting system [1, 2]. Our system can be helpful for the deaf and mute to easily communicate face-to-face with people who do not know sign language. Our system is cost-free and suitable for everyone as it contains different language options for sign language. People who are interested to learn sign language can also use our application.

The YoloV5 algorithm from the You Only Look Once (YOLO) series has been used which is a sophisticated Convolutional Neural Network (CNN) in performing object detection in real-time object detection in real-time for identifying as well as predicting hand and body gesture language. It is a single-stage object detector that analyzes as well as forecasts photos as input by applying 3 key parts: The Backbone Model, The Neck Model, and The Head Model. It is an object detection technology that breaks down an image and uses a system in which every grid can recognize an item on its own.

1. We have also used the Convolutional Neural Network, also used for image processing and artificial intelligence (AI) instruments, to compare the results made of 3 layers: an input layer, an output layer, and a hidden layer that contains various layers.
2. The reason we chose YoloV5 is that it consists of simple codes which can give all the necessary results such as graphs, confusion matrix, etc. after the training process on its own. But in the case of CNN, it does not give graphs and another necessary results after the training process. We had to go through more complex codes to get those results.

Literature Review

This section discusses the various other papers we went through to help us understand more about sign language and the different algorithms used in making an application for sign language detection.

Risk factors observed repeatedly for loss of hearing included toxemia preterm, low birth weight, consanguinity, and birth asphyxia. According to the inspection, the major sources of loss of hearing are hyperbilirubinemia, pneumonia, meningitis, as well as ototoxicity. Furthermore, parents lack acknowledgment and guidance regarding the risk factors of deafness making it part of the reason why children suffered from this issue.

3. HSV model can be used for feature extraction of images which mainly relies on the pigmentation of the human skin. Segmentation has been done on the images and then edge detection has been used where the edges of high-contrast images to find the boundary of objects in the images. After normalizing the images, features are extracted from a black-and-white image [4]. In this study, a robustly estimated autoencoder (SAE) pattern instruction technique has been used and is a fundamental element examination to direct the identification of human gestures using RGB-D data. The results after testing on the ASL dataset show that related features of Active Learning significantly improve accuracy from 75% to 99.05% [5]. In this study, HOG has been used as a feature descriptor to extract features of images that were first segmented using YCbCr. The result showed an accuracy of 88% [6]. An average detection percentage of 92.4% has been observed by using the k-curvature algorithm that allocates the tips of fingers and dynamic time wrapping was used to recognize gestures [7]. This research proposes a new fusion of improved attributes for the categorization of sign language's static signs. It starts by describing how depth information can be used to distinguish the hand from the scenery and a combined edge detection approach is presented to obtain several pertinent features of an image [8]. An efficient deep attention network enabling concurrent identification and detection of hand gestures on static RGB-D pictures using a CNN frame-work that is based on a delicate attention mechanism in a holistic manner [9]. Videos have been used where hand motions in successive video sequences are represented by fused features. On various extracting features from the ISL dataset, the ANN Based classifier is evaluated against state-of-the-art classifiers including Adaboost, support vector machine (SVM), and other ANN approaches giving an accuracy of 92.79% [10]. Another approach using the ANN classifier shows the evaluation and comparison of two extraction methods namely hand contour-based ANN and complex moments-based ANN [11]. In another study, using NATOPS datasets, an authorized lexicon of aircraft flight control movements, they evaluated their approach in a simulation of real-world nonverbal communication. This gave an accuracy of 75.37% [12]. The ethical issues regarding sign language claim that computer scientists need to be aware of the history of sign language and must learn the language beforehand making a system for recognition [13].

The dataset was created using data-gathering processing which included the use of the webcam. Furthermore, we collected additional images from online sources to create a variety of hand shapes and sizes. The photos are cropped into the same size and converted to grayscale to get a more accurate output result. The images are then labeled in respective classes using LabelImg software. The model is then trained to recognize signs.

Methodology

This section has an explanation of our dataset, the methodology of our design, and the result we achieved.

Dataset Analysis

In this section, we have described how we prepared our dataset for the training process.

Below are the tables to show the structure of our dataset which has been labeled using LabelImg and made into two divisions for training and testing the YoloV5 model But for the CNN Model, we divided the dataset into each category of classes and used code to combine the images into one folder and separate them for training and testing.

The following table represents the total images for all classes with the format of the image we used as well as the size length and width of the images.

Class Name	*Image Numbers*	*Format of Image*	*Size of Image*
0	500	JPG	400 × 400
1	517	JPG	400 × 400
2	499	JPG	400 × 400
3	498	JPG	400 × 400
4	511	JPG	400 × 400
5	507	JPG	400 × 400
6	491	JPG	400 × 400
7	495	JPG	400 × 400
8	491	JPG	400 × 400
9	489	JPG	400 × 400

Table 3.1: Details of the dataset.

The dataset comprises 10 types of hand signs which has a total of 4998 images. The division of the dataset for training and testing is roughly 90% and 10% respectively.

The table below contains information on the size of the batch and epoch used for training. It also represents the percentage of training and testing images divided from the entire dataset. It further shows the number of classes, total training samples, total test samples, as well as the input shape of the images.

Size of Batch	32
Number of Epoch	300
Training	90%
Testing	10%
Output Label	10
Input Shape	416 × 416 × 3
Training Images	4206
Testing Images	792

Table 3.2: Training Specifications.

This table represents how many training and testing pictures are used for distinct classes. Total training and testing images are kept roughly the same for all the classes.

Class Name	*Number of Training Images*	*Number of Testing Images*
0	421	79
1	437	80
2	412	87
3	421	77
4	422	89
5	422	85
6	417	74
7	424	71
8	412	79
9	418	71

Table 3.3: Number of training and testing pictures of the individual label for YoloV5.

After we evaluate the two models, we concluded that both models give the same accuracy. We chose YoloV5 because it was easy to understand and with a few simple codes we can get all the necessary graphs and results as in the case of the CNN model it was the opposite. But both the models gave similar and high accuracy after testing which was suitable for usage.

Proposed Methodology

The design of our model that described our procedure has been described in this section.

The model given below shortly describes the procedure we followed for the training process and other steps we took for the entire project.

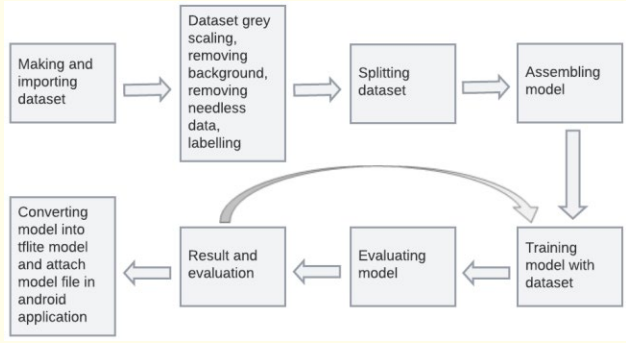

Figure 3.1: Proposed Methodology Design.

Input Image

The dataset consists of ten classes created using a webcam. The pictures are then taken while keeping a hand gesture for each gesture in multiple positions to increase the accuracy of real-time detection [14].

Pre-processing

It is a method applied to acquire pictures that must be anomalous in some respects. The main purpose of this step is to eliminate unwanted sections of the pictures or the backdrop to expand features. The entire dataset is produced in grayscale which means the images are in black and white to enhance accuracy [2]. The images were then made into two divisions for testing and training, after labeling them using LabelImg [1]. We trained YoloV5 and CNN models in Google Colaboratory using the same dataset except in the CNN model, labeled images were not used.

Conversion. We have converted the trained model file of YoloV5 into a TensorFlow Lite which is supported by the Android application. This is also done in Google Colaboratory. After converting the file, it is attached to the Android application using the right specifications to make the model in the application [15].

Here are some samples of the images used in the dataset.

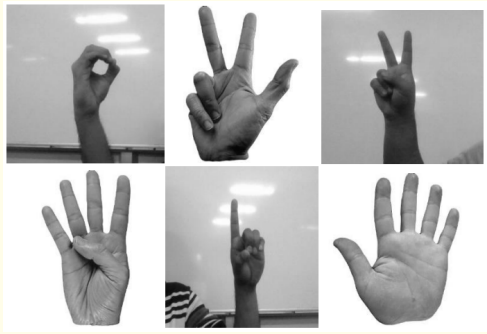

Figure 3.2: Sample images of our dataset.

Result & Analysis

This section contains the details of the accuracy we got from training both YoloV5 and CNN models using the same dataset and keeping all the necessary input the same for a fair test.

Below is a description of the training parameters used for both YoloV5 and CNN.

Size of Batch	32
Number of Epoch	300
Training	90%
Testing	10%
Output Label	10

Table 4.1: Training Specification for both YoloV5 and CNN.

These are the accuracy we received from training the two models. It shows that the accuracy between the two models has a difference of less than 1%.

Model Name	***Model Accuracy***
YoloV5m	98.60%
CNN	99.40%

Table 4.2: Accuracy Result.

Model Evaluation for YoloV5

This section shows the result we achieved from training the YoloV5 model.

In the following images, the table containing the class numbers shows which class each colored line in the graph represents.

This graph shows the F1 curve, Formula One, which expresses the top F1 value with a confidence threshold for each label as well as all the labels altogether.

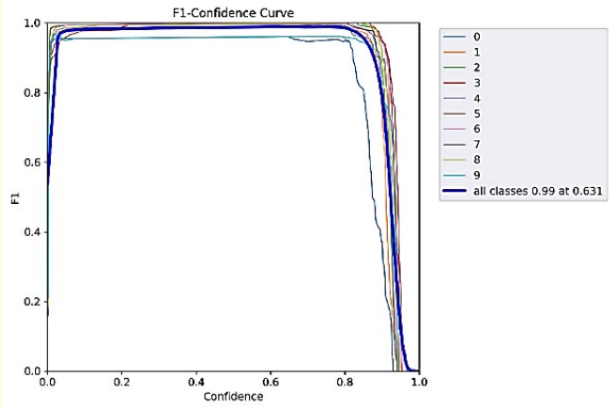

Figure 4.1: F1 Curve.

P Curve or Precision-Confidence Curve computes the possibility of a predicted bounding box having similarity to the actual ground truth box, called a positive predictor.

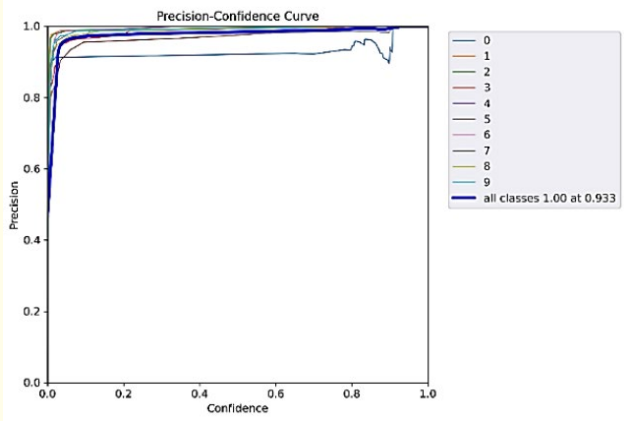

Figure 4.2: P Curve.

Below the R curve or Recall-Confidence Curve expresses a positive rate, also alluded to as sensitivity.

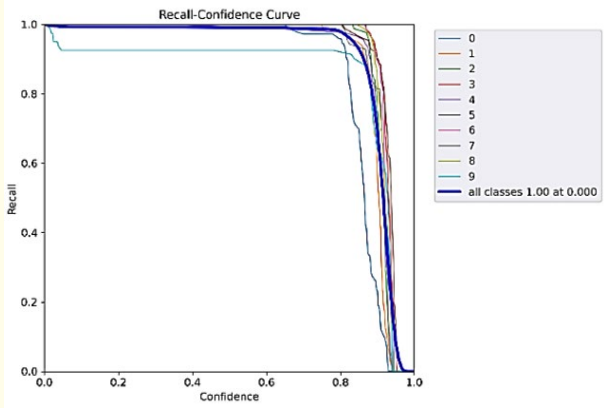

Figure 4.3: R Curve.

Here, the PR Curve or Precision-Recall Curve is a plot of precision and recall. This is used to evaluate the performance of object recognition models.

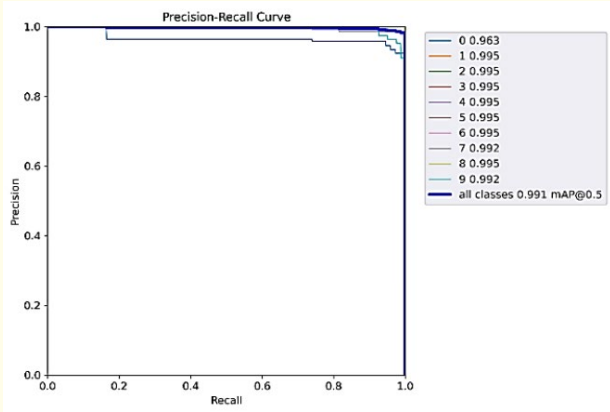

Figure 4.4: PR Curve.

This shows the Precision (P) score, Recall (R) score, mAP values from 0.5 to 0.95 over various IoU thresholds for the labels distinctively and altogether as well for the YoloV5 model.

Class	Images	Instances	P	R	mAP@.5	mAP@.5:.95: 100% 25/25
all	792	791	0.986	0.986	0.987	0.826
0	792	73	0.921	1	0.946	0.69
1	792	80	0.995	1	0.995	0.808
2	792	87	0.995	1	0.995	0.783
3	792	77	0.996	1	0.995	0.871
4	792	90	0.978	0.973	0.993	0.87
5	792	85	0.995	0.965	0.994	0.892
6	792	74	0.994	1	0.995	0.868
7	792	70	0.986	1	0.99	0.825
8	792	74	1	0.992	0.995	0.85
9	792	81	1	0.926	0.973	0.801

Figure 4.5: Performance Report for YoloV5 Model.

A confusion matrix is defined as a method to compute the execution of the Classifier model. The x-axis gives out the label for the images that the model detected and the y-axis gives out the predicted labels made by our trained model. The deeper the color goes following the color range on the right side of the confusion matrix, the better the accuracy.

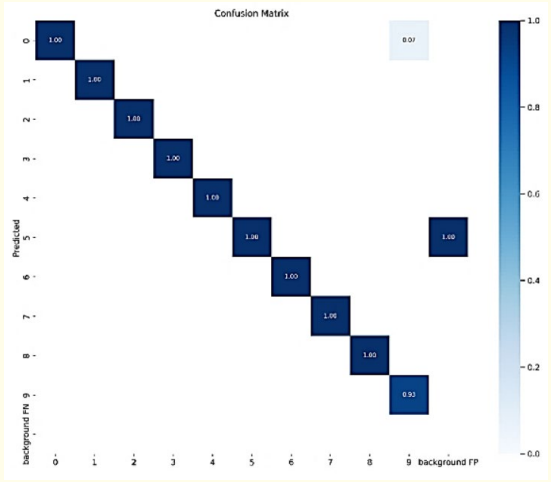

Figure 4.6: Confusion Matrix for YoloV5 Model.

Model Evaluation for CNN

This section shows the result we achieved from training the YoloV5 model.

Below shows accuracy graphs that measure the model's prediction performance for training in blue lines and validation in orange lines.

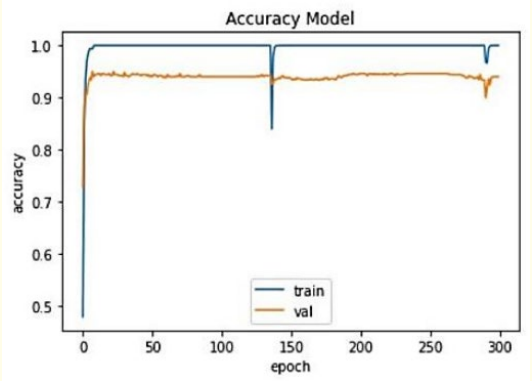

Figure 4.7: Accuracy Graph for CNN.

Next, the loss graphs measure the model errors it is making in training in blue lines and validation in orange lines. The fewer errors, the better the model is.

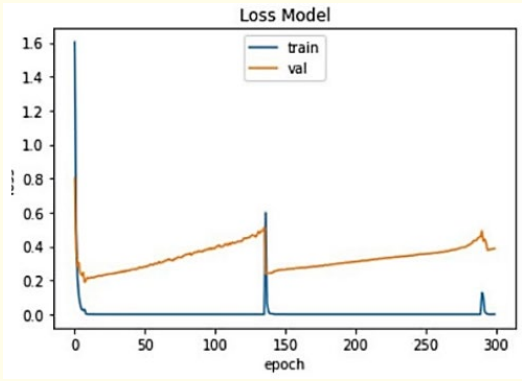

Figure 4.8: Loss Graph for CNN.

The below table gives the Precision, Recall, F1-Score, as well as Support values for all classes distinctively as well as altogether for the CNN model.

	precision	recall	f1-score	support
0	1.00	0.99	1.00	501
1	1.00	0.99	1.00	519
2	1.00	0.99	1.00	502
3	0.99	1.00	0.99	493
4	1.00	1.00	1.00	510
5	0.99	1.00	1.00	506
6	0.99	1.00	0.99	487
7	0.99	0.98	0.99	499
8	1.00	0.99	0.99	494
9	0.99	1.00	1.00	487
accuracy			0.99	4998
macro avg	0.99	0.99	0.99	4998
weighted avg	0.99	0.99	0.99	4998

Figure 4.9: Performance Report of CNN Model.

Below is the confusion matrix we achieved from training our CNN model. This matrix has the same structure as the matrix of YoloV5, but the only difference is that the higher the color chart goes following the color range on the right side of the confusion matrix, the better the accuracy.

Results of Working Camera

Below are given some results to show how our device can recognize and predict the gestures accurately. In the bounded box, we can see on the upper left side of the box that some values are showing. The first part of the value shows the sign, or the class of the gesture predicted, and the second part of the value shows the accuracy percentage.

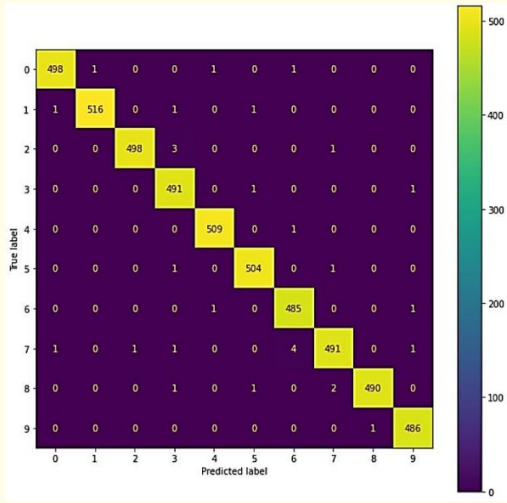

Figure 4.10: Confusion Matrix of CNN Model.

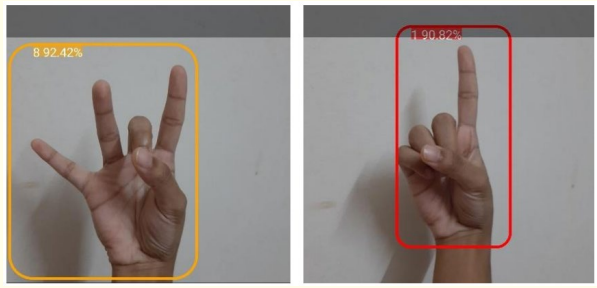

Figure 4.11: Camera Results.

Conclusion

Lastly, this section concludes with the background of sign language and how research on sign language has been established.

We ended up with a result showing that both YoloV5 and CNN give the same accuracy and were in high percentages as well. This is a satisfactory value for usage. More-over, YoloV5 was easy to use as we got all the results right after the model was done training on the dataset which made it easy for us and was time-saving as well.

In this generation, many new technologies are being developed to bring more comfort to life and help individuals struggling through multiple issues in their everyday life. In this fast-developing generation, more and more ways are being made for the deaf and mute to communicate with ease, cutting down boundaries between normal people and the disabled. Before, it was difficult to get an idea of the sign languages of certain countries due to a lack of databases or research. But now databases are being made to include a large range of vocabularies for different sign languages to help others trying to learn new sign languages easily. With the new technologies and studies, we wish to improve our system for faster and better detection. Moreover, we are inclined to make a variety of devices and software adding a huge dataset of more sign languages that are yet to be known by the majority to make the lives of the deaf community a lot easier and simpler.

References

1. Sahla Muhammed Ali. "Comparative Analysis of YoloV3, YoloV4, and YoloV5 for Sign Language Detection". Department of Information Technology, Rajagiri School of Engineering and Technology, Kerala, India (2021).
2. Abul Abbas Barbhuiya, Ram Kumar Kash and Rahul Jain. "CNN-based feature extraction and classification for sign language". Multimedia Tools and Applications 80 (2021): 3051-3069.
3. Sharddha Jain and Sirjan Singh. "Factors associated with deaf mutism in children attending special schools of rural central". J Family Med Prim Care 9.7 (2020): 3256-3263.
4. Hasan MM and Misra PK. "HSV brightness factor matching for gesture recognition system". IJIP 4.5 (2011): 456-467.
5. Nagarajan S and Subashini TS. "Static hand gesture recognition for sign language alphabets using edge-oriented histogram and multi-class SVM". International Journal of Computer Applications 82.4 (2013): 28-35.
6. Omkar Vedak., et al. "Sign Language Interpreter using Image Processing and Machine Learning". Department of Computer Engineering, Datta Meghe College of Engineering, Mumbai University, Airoli, India (2019).
7. Plouffe G and Cretu AM. "Static and dynamic hand gesture recognition in-depth data using dynamic time warping". IEEE Transactions on Instrumentation and Measurement 65.2 (2015): 305-316.
8. Jadooki S., et al. "Fused features mining for depth-based hand gesture recognition to classify blind human communication". Neural Computer & Application 28.11 (2017): 3285-3294.
9. Li Y., et al. "Deep attention network for joint hand gesture localization and recognition using static RGB-D images". Information Sciences 441 (2018): 66-78.
10. Sign language recognition with multi-feature fusion and ANN class.
11. Badi H. "Recent methods in vision-based hand gesture recognition". International Journal of Data Science and Analytics 1.2 (2016): 77-87.
12. Ghotkar AS and Kharate GK. "Hand segmentation techniques to hand gesture recognition for natural human-computer interaction". ACM Transactions on Interactive Intelligent Systems 3 (2012): 15.
13. Annelies Braffort. Research on Computer Science and Sign Language Ethical Aspects (2002): 2298.
14. Kanchan Dabre and Surekha Dholay. "Machine Learning Model for Sign Language Interpretation using Webcam Images". Department of Computer Engineering Sardar Patel Institute of Technology Student of M.E.(Computer) Mumbai, India (2014).
15. Suharjitoa., et al. "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output". Computer Science 116 (2017): 441-448.

Volume 6 Issue 1 January 2024