Sri Lankan Sign Language Detection Using Machine Learning

S. Sivaraja, S. Vinayagamoorthy, S. Rajkumar, M. Weerasinghe*

School of Engineering, Engineering, Sri Lanka Technological Campus (SLTC Research University), Padduka 10500, Sri Lanka

*mahimaw@sltc.ac.lk**

***Abstract*—Sign language detection is a way of communication for deaf people to share their thoughts with others. From the recent survey, it is noted that there are over three hundred thousand deaf people in Sri Lanka. Approximately 9% of the population in Sri Lanka has a loss of hearing. Conveying their ideas and emotions to normal people is the real challenge for deaf people. Therefore, we propose a real-time Sign Language Recognition system using machine learning and further implement hardware for effective communication of deaf people. In this proposed method, we included gestures, numbers, alphabets, and other words in sign language. In the modern world, convolutional neural networks (CNN) are widely used in the modern world for various purposes. We have also used CNN to recognize sign language gestures in our project. There'll be a robotic arm in the hardware implementation part, and it can show letters and numbers by hand, so it creates a system for two-way communication.**

Keywords **CNN-Convolutional Neural Network, Open Pose, YOLO V5**

I. Introduction

Sign language is a visual language using hands, gestures, and facial expressions, especially used by deaf and muted people. There are some references to ancient Greek people who also used hand gestures in the 5th century BC. People have used sign language for a long time. There are many sign languages in the world, like American, Indian, Japanese, Russian, etc. [1] sign language detection systems have been developed to provide a communication medium for normal and deaf people in recent years.

In this system, we are proposing a Sri Lankan Sign Language interpreter. Here we are planning to implement the system using multiple gestures and single-frame sign languages. The external camera of the interpreter captures the poses and sends them to a raspberry pi. The frames of that captured video will be processed. With this, we can implement a fast and accurate interpreter. During that time, words will display on the monitor for the captured signs and gestures. Our system has a robotic arm for making short replies for deaf people. It can show the alphabet and numbers. Our model will work in real-time. So, using this, we can implement a fast and accurate interpreter.

II. System Model

A. Sign Language Interpreter

This block diagram shows how our proposed sign language interpreter works. When a deaf person inputs his/her sign language to our system through the camera, it'll be given in text form. So the normal person can easily understand what the deaf speaker is trying to convey.

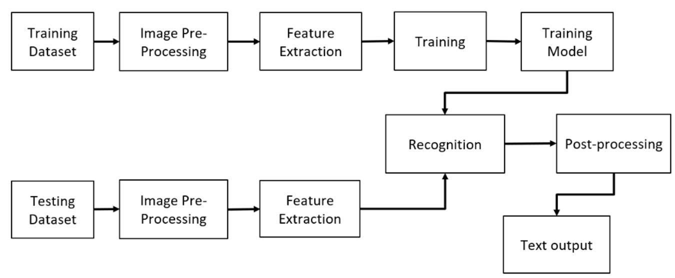

Fig. 1. Block diagram of Sign language interpreter

B. Robot Arm

This block diagram shows how the normal user conveys his or her ideas to a deaf person. For example, through the interpreter, a normal user understood what the deaf person was trying to convey, so he needed to reply back using sign language. So if the normal person types what he needs to convey, the robot arm will perform the required action for the text.

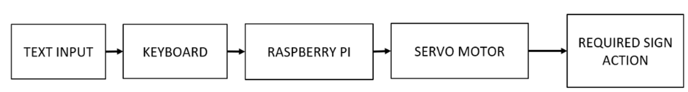

Fig. 2. Block diagram of Robot Arm

III. Related work on literature

- In [1] use image or video processing techniques,2-D Vision-based techniques and-D Vision-based techniques, and Convolutional Neural Network (CNN).which is used to create a model named signet, which can recognize signs, based on supervised learning of data. this system has a high accuracy of 98.64 % under a variable scale. They use a skin-colour segmentation algorithm to detect the face of the signer and then eliminate it by replacing it with black pixels, the largest connected component algorithm for hand region segmentation; and the Viola-Jones face detection algorithm. This system can recognize 26 alphabets, 500 signs, dynamic signs, 20 gestures, and numbers. static alphabets with a training accuracy of 99.93% and a testing and validation accuracy of 98.64%. However, this system was not a fully automated sign language recognition system.

- [2] proposed a complete skeleton of an isolated video-based Indian Sign Language Recognition System. This system includes image processing techniques and computational intelligence techniques in order to deal with sentence recognition, and computational intelligence techniques in order to deal with sentence recognition. Features generated using these techniques make the feature vector unique for a particular gesture. Using vision-based systems for image processing, the fusion algorithm is used to extract edges and use Otsu's algorithm to result in an edged image. It has 96% accuracy rate. [2] developed a system for recognizing a subset of the Indian sign language, including gestures as well. but it's not a real-time regeneration system.
- [3] proposed digital image processing techniques and artificial neural networks for recognizing different signs. [3] uses image processing techniques and artificial neural networks to recognize different signs. This technique is used for extracting features and recognizing signs. Classification is done using a subtractive clustering algorithm and a fuzzy inference system. This system automatically identifies fingerspelling in Indian sign language. but it can not find two-handed signs and gestures

IV. Machine Learning approach

Software applications can become more accurate at predicting outcomes through machine learning, which is a type of artificial intelligence (AI). So authors are using machine learning in the system to get efficient output. The goal of machine learning algorithms is to construct a model from sample data, also known as training data, so that it can make predictions or decisions without being explicitly programmed.

A. Dataset collection

For this project, the authors created their own data set for the Sri Lankan sign language. It includes Tamil alphabets, Sinhala alphabets, English alphabets, 0 to 9 numbers, and frequently used gestures by deaf people. For the YOLO V5 training created your own videos and extract those into frames. For the open pose training created gesture videos. in the gesture videos, 4 people acted sign gestures.Figure 3 describes the sample datasets. After collecting raw images as datasets, the authors started preprocessing and augmentation. They used Flip in augmentation because both signers are right-handed. So there is a requirement for the left-signers.

NUMBERS DETECTION	SINHALA	TAMIL	ENGLISH
3602 Source images	5601 Source images	4548 Source images	8433 Source images
Almost 400 images per class	Almost 466 images per class	Almost 379 images per class	Almost 320 images per class
Auto-Orient and Resize 416x416	Auto-Orient and Resize 416x416	Auto-Orient and Resize 416x416	Auto-Orient and Resize 416x416
Flip(H&V), Brightness	Flip(H&V), Brightness	Flip(H&V), Brightness	Flip(H&V), Brightness
Total 8219 images	Total 9248 images	Total 9646 images	Total 14128 images
Training 7154 images	Training 7555 images	Training 7845 images	Training 11654 images
Testing 344 images	Testing 547 images	Testing 634 images	Testing 834 images
Validation 721 images	Validation 1146 images	Validation 1167 images	Validation 1640 images

Fig. 3. Datasets analysis of Project

but the authors managed with augmentation techniques. After finishing augmentation and preprocessing, the dataset doubled in size. Later, the datasets are divided for training, validation, and testing with a ratio of 7:2:1.

B. Training

Yolo is used for object detection. YOLO is a type of CNN algorithm. Following figure shows how CNN algorithms work on this project. The first convolution layer simplified this complex image. The filtering process happens in this layer. The next layer is the pooling layer. This layer makes the process much faster and it creates a pooled feature map. After that, the upcoming layers extract small features like hand positions, finger positions, angles between toes, etc. So the output will be highly accurate. The final layer is the Yolo layer. This layer generates predictions from anchor fields for feature detection. High frame rate and low training time are the best advantages of Yolo.

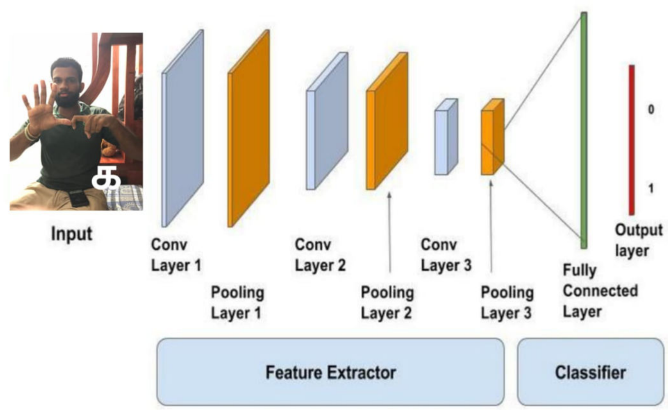

Fig. 4. CNN Architecture

Training a model simply means Learning (determining) appropriate values for all weights and biases from labelled examples Raw data is labelled by Robo Flow software. Because supervised machine learning is used in this part. A minimum of 100 inputs is needed for the CNN algorithm. In this project, 600 to 800 photos are put as input for one sign because it gets a more efficient output. All label data will be trained using YOLOv5 and gestures will be trained by open pose. Yolov5 is a kind of CNN algorithm. It can detect objects in real time with great accuracy. An open pose detects human body poses such as facial and hand movements. So, in this research paper, YOLO V5 and Open Pose are used for training.

V. Results analysis

The sign language will be detected and extracted, as will hand positions and shapes. The detected shapes or hand positions will be sketched and merged with the input image, which is displayed on the screen. This system detects all signs and displays the respectful letters for each sign when a user performs a finger s pell. T he fi rst 3 co lumns in ab ove figure 4 clearly show the training losses and validation losses. The first 3 columns are box loss, object loss, and class loss. All the losses are decreasing with the increasing number of epochs. So the trained model will be more effective for our product.

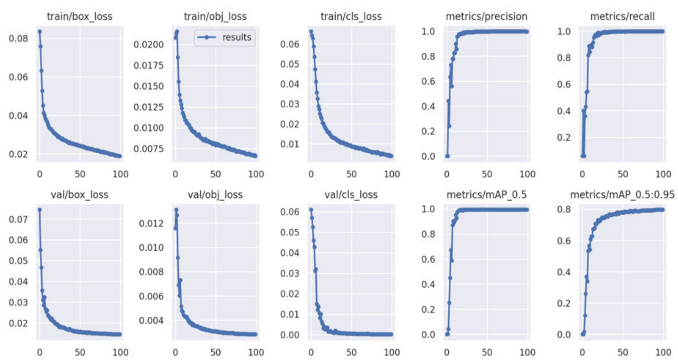

Fig. 5. Training, Validation Losses graph of Sinhala alphabets

Class	Images	Instances	P	R	mAP@.5	mAP@.5:.95: 100% 34/34 [00:08<00:00, 4.01it/s]
all	1088	1090	0.999	0.998	0.994	0.798
Ah	1088	65	0.998	1	0.995	0.848
ae	1088	74	0.999	1	0.995	0.75
aee	1088	83	0.999	1	0.995	0.815
ahh	1088	119	0.999	1	0.995	0.775
e	1088	61	0.998	1	0.995	0.776
ee	1088	118	0.999	0.992	0.995	0.815
i	1088	84	1	0.988	0.986	0.799
ii	1088	83	0.998	1	0.995	0.752
o	1088	86	0.998	1	0.995	0.83
oo	1088	150	0.999	1	0.995	0.863
u	1088	92	0.999	1	0.995	0.8
uu	1088	75	1	1	0.995	0.749

Fig. 6. Trained Model Details of Sinhala Alphabets

Figure 5, shows all the classes in the Sinhala alphabet like Ah, ahh, ae, aee, i, ii, u, e, ee, O and Oo. For a better model, precision and recall must stay high. In figure 4.2, there are precision and recall for all the classes. Precision and recalls always stay near to one. So, our model is an effective one. after After finishing our trained model for the Sinhala alphabets, we started testing the prediction using Google Colab. Figure 4.3 clearly shows the output of the Sinhala alphabet detection. The trained model predicted the gestures with more than 80% confidence. So, the trained output was tested effectively.

VI. Future works

We are planning to increase sign language recognition by increasing the number of signs and words. They are also planning to introduce sign language recognition to various regions, for example, Indian sign language and American sign language. We tried to implement it by using a single robot arm now. So, we are planning to implement using the full robot model in the near future. When the Neuralink chip comes to market, we can implement the system using it. So, there is no need for an external interpreter.

VII. Conclusion

There is no specific Sri Lankan sign language interpreter in the market. In this research paper, we are proposing a Sri Lankan sign language interpreter. Here we proposed the system using YOLO V5 for alphabets and Open Pose for continuous signs. We had been trained in the system using the YOLO V5 till now. The trained output came with good accuracy.

References

[1] C. Sruthi and A Lijiya, "Signet: A deep learning based indian sign language recognition system," in *2019 International conference on communication and signal processing (ICCSP)*, IEEE, 2019, pp. 0596–0600.

[2] P. Kishore and P. R. Kumar, "A video based indian sign language recognition system (inslr) using wavelet transform and fuzzy logic," *International Journal of Engineering and Technology*, vol. 4, no. 5, p. 537, 2012.

[3] V. Adithya, P. Vinod, and U. Gopalakrishnan, "Artificial neural network based method for indian sign language recognition," in *2013 IEEE Conference on Information & Communication Technologies*, Ieee, 2013, pp. 1080–1085.