Conversion of Sign Language to Speech with Human Gestures

*Ligitha Sakthymayuran, **Kajaluxshan Shanmugaratnam

* Temporary Information Technology Demonstrator, Trincomalee Campus, Eastern University

** Software Engineer, Virtusa (PVT) ltd, Colombo, Sri Lanka

Abstract

Sign language recognition is one of the most rapidly increasing disciplines of study today, and it is the most natural mode of communication for those who are deaf or hard of hearing. A gesture recognition system can allow persons with hearing impairments to interact with others without the need for a translator or an intermediary. With a tailored dataset, the system is set up for automatic recognition of American Sign Language. It can be taught to recognize any sign language in the world for which there is no well-established dataset. The suggested system allows any user to learn sign language using human sign language characters and a word database, and the training is provided offline. In the suggested approach a bigger sample is considered to identify the words that are taken using a camera and are not part of the regular sign language. In order to improve overall performance, picture pre-processing phases are employed to modify the image. For this first median and Gaussian filters are used to minimize sounds and morphological techniques are used as the pre-processing stage. Then Hue Saturation Value (HSV) color space is used to distinguish the palm from the arm. After that, for histogram equalization hands are set using the 5 x 10 boxes which can be kept for model training as well as recognition. TensorFlow object detection api used to build a model for a specific object identification which makes the entire methodology more approachable. Convolutional Neural Network (CNNs) can automate the process of feature building rather than constructing intricate handmade features. We have a good level of accuracy in recognizing 46 gestures.

***Keywords – Motion modeling, TensorFlow,* convolutional neural network, deep learning, gesture recognition, sign language recognition**

Introduction

In current times, spoken language has a range of communication channels via which everyone may connect with society. Unfortunately, not every human being is capable of using spoken languages owing to hearing disability or incapacity to speak.

According to the World Health Organization, around 466 million individuals (more than 5% of the world's population) suffer debilitating hearing loss, with 34 million of these being children. It is anticipated that by 2050, approximately 900 million individuals – or one out of every 10

people – would have debilitating hearing loss. -World Health Organization introduces the problem at http://www.who.int/news-room/factsheets/detail/deafness-and-hearing-loss

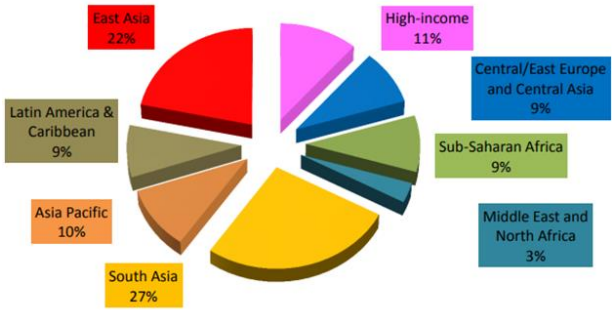

Figure 1: World health organization's pie chart of population with hearing impairment (adults)

Due to inadequate hygiene, excessive use of earphones, accidents, and the presence of 30 years of conflict in Sri Lanka, almost 1.7 million (8% of the population) Sri Lankans are deaf. Hundreds of sign languages may be found all over the world, including American Sign Language (ASL), Mexican Sign Language (MSL), British Sign Language (BSL), French Sign Language (FSL), and Italian Sign Language (ISL), to name a few. Almost every spoken language has a sign language based on the sign languages described above. With around 2000 sign-based terms, Sri Lankan sign language originated from British Sign Language.

People with hearing impairment confront a variety of challenges in their daily lives, particularly when it comes to public facilities such as banks, hospitals, and police stations, among others. According to the findings of the study, there is a need for a greater focus on deaf patients who utilize the health system in order for it to become genuinely universal and with equitable access for all populations and groups, especially minority ones. -Based on "Main difficulties and obstacles faced by the deaf community in health access: an integrative literature review" by Maria Fernanda Neves Silveira de Souza, Amanda Miranda Brito Araújo, Luiza Fernandes Fonseca Sandes, Daniel Antunes Freitas, Wellington Danilo Soares, Raquel Schwenck de Mello Vianna, Árlen Almeida Duarte de Sousa.

To overcome this obstacle, we were inspired to design an application that would assist them in communicating, giving them a voice.

Objective

Convert human sign language to text using human gesture recognition and motion capture. To develop a simple application allowing non-deaf individuals to connect with deaf people, with improved user experience, flexibility, and portability.

To create a system that can be taught by anybody with hearing loss using tailored datasets based on their language choice, because only a few sign language datasets, such as ASL, BSL, CSL, and others, were well established and widely available. This program may also be used as a teaching module for deaf students who are learning sign language for the first time, as there are roughly 25 schools in Sri Lanka that teach sign language.

Contribution

Motion modeling, motion analysis, pattern recognition, and machine learning are all involved in gesture recognition. It is made up of procedures with both manual and non-manual parameters. The environment's structure, such as backdrop light and movement speed, influences predictive ability. Because of the variation in views, the gesture seems different in 2D space.

We met the aforementioned obstacles in this project and discovered solutions to overcome them with the help of convolutional neural networks (CNN) and OpenCV, as well as providing flexibility for users who are bound to different sign languages. To boost performance, we employed Gaussian blur, median filters for picture pre-processing, and CNN for

classification, which provides improved accuracy for predictions.

Literature review

SL is a distinct language that is distinct from spoken or written language. It has its own alphabet, numerals, words/sentences, and so on. The main distinction is that it has a smaller vocabulary than written or spoken language. It is also in its early stages in the majority of developing and underdeveloped countries. Sign language development in these nations will take years before it becomes an independent language. However, computer recognition for sign language has begun in these nations, and considerable progress has been described in the literature.

(Rajaganapathy, Aravind, Keerthana, & Sivagami, 2015) In this work, the system starts when the sensor, a Microsoft Kinect sensor, is turned on; when a human motion enters the frame, the skeletal data of the user is monitored with the 20 joints and their coordinates. The input stream is retrieved as individual skeleton frames, with each frame having a posture or a gesture. The obtained gestures are compared to the gesture input set. If the current skeleton frame fits the predetermined gesture pattern, the gesture's matching word is tossed as text to the Windows narrator. A sample of 100 spells for various indications was tested. Accuracy of up to 90% has been attained. The main disadvantage of this system is that it is not portable due to the high size of the sensor. Our suggested solution employs an Android camera as a sensor, allowing for portable functionality and device availability to all users.

(Shinde & Kagalkar, 2015) Edge detection technique is utilized in this article to recognize the input sign image by gray scaling and recognizing the edges of hand gesture. The system can handle various input sign pictures of alphabets, words, and phrases and translate them into text and vice versa. This method is intended to convert Marathi sign language into text. The dataset comprises a large number of hand motion photos captured from numerous users with varying hand sizes, which aids in recognizing the right output for every person utilizing the system. Concatenation method is utilized to construct words from alphabet motions, and words are formed. The technology is taught using a preconfigured database. The technique has been successfully deployed on forty-six Marathi sign gestures and 1000 hand gesture phrases created by concatenating alphabets from diverse hand gestures. For each sign picture supplied, the system produces correct results.

(Nanivadekar & Kulkarni, 2014) A sign language recognition system is suggested in this research. Indian Sign Language (ISL) database building is completed at the very first step. Hand tracking and segmentation are completed in the following phase. The system was successfully built, and the findings are provided in this publication. The findings show how motion tracking, edge detection, and skin color detection function separately as well as how they perform together. The gestures included in the database are alphabets A-Z and numbers 1-10.

(Saxena, Jain, & Singhal, 2014) This article contains an examination of the primary components, as well as a quick and effective approach for recognizing the identification of gestures. The suggested technique extracts three frames per second from the video stream. These photos are associated with the object database. This approach has been successfully tested and developed in a real-time context, with a success rate of roughly 90%. Ten sign motions derived from the Indian Index language and produced in the lab using a camera and an Android smartphone because the system database has a sign gesture of 60X80 pixels, it takes less memory and processing time. The recognition rate is between 70% and 80%, which is within acceptable limits. To work correctly, the system requires a dark backdrop.

(Yang, 2010) A unique recognition approach of sign language spatio-temporal appearance modeling is introduced in this research for the vision-based multi-features classifier of Chinese

sign language identification. The obvious advantage of such a unique strategy is that we may omit certain skin-like items while following the moving recognized hand in the sign language video sequence more precisely. Experiments show that this novel modeling strategy is practicable and resilient. First, dynamic sign language appearance modeling is performed, and then a classification approach based on SVMs is employed for recognition. Experimenting with 30 sets of Chinese manual alphabet photos yields the best identification rate of 99.7% for the letter 'F' image group.

(Pigou, Dieleman, Kindermans, & Schrauwen, 2014) In this research, characteristics taken from frame sequences are used. This produces a representation made up of one or more feature vectors, often known as descriptors. This depiction will assist the computer in distinguishing between the various types of activities. The representations will then be used by a classifier to differentiate between the various activities (or signs). CNNs are used to automate feature extraction in this system. For classification, an artificial neural network (ANN) is utilized. The test set's accuracy is 95.68%, with a false positive rate of 4.13% due to noise movements.

DATA AND METHODOLOGY

The data set is a customized one that will consist of static photographs acquired by a web camera from numerous users.

Figure 2: Dataset

We chose 1200 photos as positive training examples, together with their left-right reflections (2400 images in all). For each move, we classified and tagged 2000 photos as train images and 400 as test images. The proportion of each gesture used in training and testing was 70% and 30%, respectively.

In the computer vision sector, there are numerous ways of tracking hands that are already in use. In addition, because many of these techniques are rule-based (e.g., removing background based on texture and border characteristics, discriminating between hands and backdrop using color histograms and HOG classifiers), they are not very robust. For example, if the background is odd, or if sharp changes in lighting conditions induce sharp changes in skin tone, or if the tracked item becomes occluded, these algorithms may become confused.

Furthermore, deep learning tools (such as the TensorFlow object detection API) that ease the process of building a model for specific object identification have made this entire area of study more approachable.

In this project, the first pixel values of the hand are set using the 5 x 10 boxes to compute the histogram, which can be kept and utilized in model training as well as recognition since we can easily replace the saved histogram as needed under varied illumination circumstances. This will provide for greater flexibility when testing in different situations and with different people. The picture under the green boxes will be utilized to construct the histogram, as illustrated in the Figure below.

Figure 3: Initialization histogram

Data Acquisition

The data obtained in vision-based gesture recognition is a frame of pictures. Images capturing equipment such as ordinary video cameras, webcams, stereo cameras, infrared cameras, or more modern active approaches such as Kinect and LMC are used to collect input for such systems. Stereo cameras, Kinect, and LMC are examples of 3D cameras that can capture depth data. In this study, we collect data using a webcam or an Android camera.

Image pre-processing

Picture pre-processing phases are used to change the image in order to increase overall performance. To reduce sounds, median and Gaussian filters are utilized in this research, followed by morphological procedures as the pre-processing stage. HSV color space is popular because the hue of the palm and arm vary substantially, allowing the palm to be easily divided from the arm.

Prior to the next stage, the acquired photos are shrunk to a lower resolution. This approach has demonstrated that decreasing the resolution of the input picture can enhance computing efficiency without impacting overall accuracy (Smith, Lobo, & Shah, 2007). Histogram equalization is used to improve the contrast of input photographs collected in varied environments in order to uniformly brighten and illuminate the images. The cropped picture size in this document is 50x50.

Feature extraction and classification using CNN

CNNs are deep learning feature extraction algorithms that have lately shown to be quite good in image identification. Currently, the models are used by industry heavyweights such as Google, Facebook, and Amazon. Recently, Google researchers used CNNs using video data. CNNs are inspired by the human brain's visual cortex. CNN's artificial neurons will link to a particular portion of the visual field known as a receptive field. This is performed by applying discrete convolutions to the picture and using filter values as trainable weights. For each channel, several filters are applied, and feature maps are formed by combining them with the activation functions of the neurons. This is followed by a pooling strategy in which just the important information from the feature maps is combined. Figure depicts the CNN architecture. (TensorFlow, n.d.) We built the model in Keras and TensorFlow using CNN, and we determined the accuracy of the prediction of these trained gestures using the sklearn.metrics.precision_ recall_fscore_support () function.

Architecture

As the input layer, a 50x50 picture is used. It will be mapped in the convolution layer, and then the useful information from the feature maps will be pooled together using the pooling strategy.

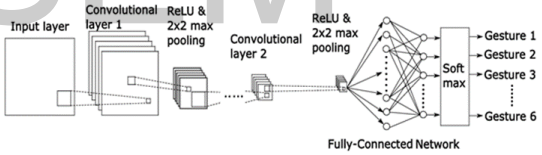

Figure 4: CNN Architecture fully connected layer

Gesture Recognition

When a gesture is caught, it is given a name gesture_name, which stores the value of the gesture, and a gesture_id, which is unique to each gesture. When a user gesture matches a specified gesture, the system returns the associated gesture_id. The system will then discover the relevant term as a text output.

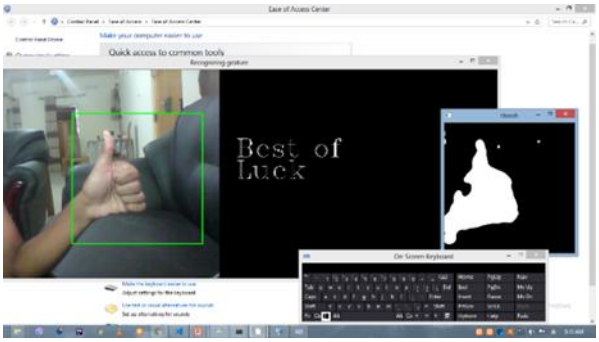

Figure 5: Gesture Recognition

Background

Convolutional Layer

(Zeiler & Fergus, 2014) Convolution is the initial layer in the process of extracting features from an input picture. By learning visual attributes with tiny squares of input data, convolution retains the link between pixels. It is a mathematical procedure with two inputs, such as an image matrix and a filter or kernel.

This phase is used to reduce the size of the image and make processing faster and easier. This stage removes some of the image's characteristics.

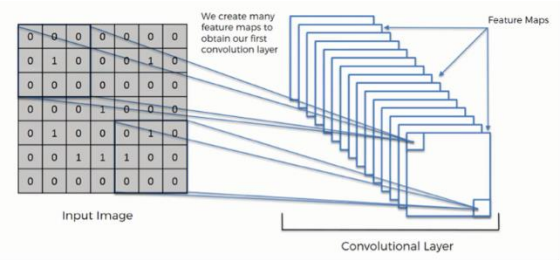

Figure 6: Convolution Layer

ReLU Layer

ReLU stands for Rectified Linear Unit for a non-linear operation, the output is;$f(x) = \max(0, x)$.The goal of ReLU is to inject non-linearity into the Convolutional Network. Since the real-world data that the Convolutional Network would want to learn would be non-negative linear numbers.

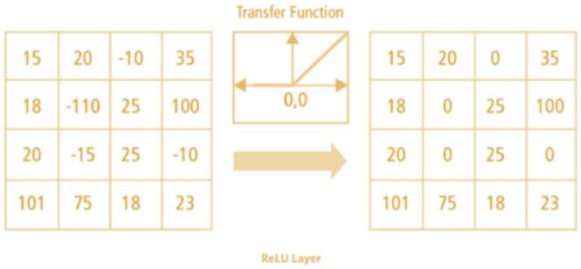

Figure 7: ReLU Layer

Other nonlinear functions, such as tanh or sigmoid, can be used in place of ReLU. However, in terms of performance, ReLU outperforms the other two.

Pooling

When the photos are too huge, the pooling layers' portion would lower the number of parameters. It also aids in the detection of characteristics in a variety of photos, regardless of differences in lighting and image angles.

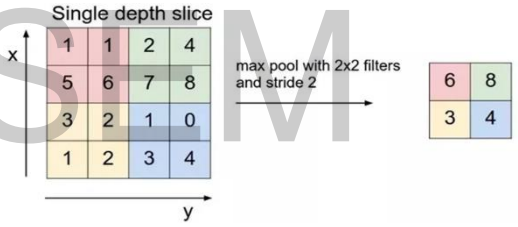

Figure 8 : Pooling

Spatial pooling, also known as subsampling or down sampling, decreases the dimensionality of each map while retaining the essential information. There are several forms of spatial pooling: maximum pooling, average pooling, sum pooling, and so forth.

Flattening

Flattening entails converting the whole pooled feature map into a single vector that is then fed into a fully linked layer, such as a neural network.

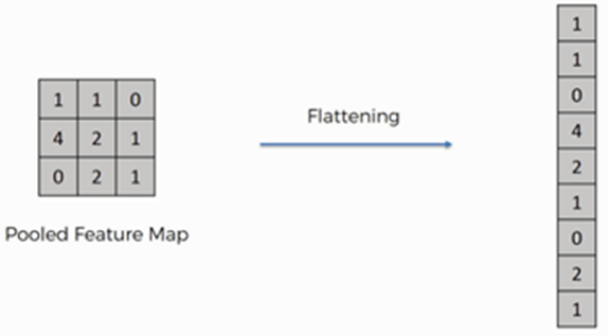

Figure 9: Flattening

Fully connected layer

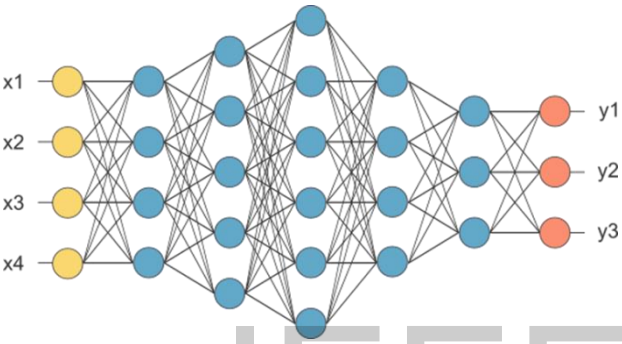

Figure 10: Fully connected layer

The input layer, the fully linked layer, and the output layer comprise this stage. The classes are predicted by the output layer. The data is sent to the network, and the prediction error is determined. The mistake is then transmitted back through the system in order to improve the forecast. Attaching a fully connected layer to the network's end results in an N-dimensional vector, where N is the number of classes from which the model chooses the required class using the probability from the soft-max activation function.

Soft-Max

$$Loss = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{e^{f_{i,y_i}}}{\sum_{j=1}^{C} e^{f_{i,j}}}\right) \quad \text{——— (1)}$$

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{C} e^{z_k}} \quad \text{——— (2)}$$

N = total number of training examples
C = total number of classes

Figure 11: Soft-Max

The soft-max function is represented by equation (2). It takes a feature vector z for a particular training sample and compresses its values to a vector of [0, 1]-valued real numbers that add to 1. Equation (1) calculates the total soft-max loss by taking the mean loss for each training case. Using a soft-max-based classification head, we may produce values that are similar to probabilities for each alphabet. This is distinct from another common option: the SVM loss. Using an SVM classification head would yield scores for each letter that would not transfer directly to probabilities. The probabilities provided by the soft-max loss help us to comprehend our results more intuitively and are valuable when running our classifications through a language model.

RESULTS AND DISCUSSION

Classification Report

	precision	recall	f1-score	support
0	0.98	1.00	0.99	394
1	1.00	1.00	1.00	434
2	0.95	1.00	0.97	386
3	0.99	1.00	1.00	396
4	0.97	1.00	0.98	410
5	0.95	1.00	0.98	383
6	0.99	1.00	0.99	391
7	0.98	1.00	0.99	390
8	1.00	1.00	1.00	401
9	0.99	1.00	1.00	413
10	0.99	1.00	0.99	397
11	0.99	1.00	0.99	405
12	1.00	1.00	1.00	399
13	1.00	1.00	1.00	381
14	0.94	1.00	0.97	429
15	0.98	1.00	0.99	399
16	1.00	1.00	1.00	396
17	0.98	1.00	0.99	393
18	0.93	1.00	0.96	397
19	0.99	1.00	0.99	375
20	0.98	1.00	0.99	406
21	1.00	1.00	1.00	411
22	1.00	1.00	1.00	428
23	0.98	1.00	0.99	401
24	0.96	1.00	0.98	355
25	1.00	1.00	1.00	423
26	1.00	1.00	1.00	410
27	1.00	1.00	1.00	375
28	1.00	1.00	1.00	419
29	0.96	1.00	0.98	396
30	0.99	1.00	0.99	420
31	0.84	1.00	0.91	403
32	1.00	1.00	1.00	404
33	0.99	1.00	1.00	415
34	0.99	1.00	1.00	426
35	0.99	1.00	0.99	396
36	0.99	1.00	0.99	374
37	0.94	1.00	0.97	395
38	0.99	1.00	1.00	407
39	0.99	1.00	0.99	398
40	0.94	1.00	0.97	372
41	0.97	1.00	0.98	375
42	1.00	1.00	1.00	405
43	0.99	1.00	0.99	416
avg / total	0.96	1.00	0.97	18000

Figure 12: Confusion Matrix

Precision is defined as tp / (tp + fp), where tp is the number of true positives and fp is the number of false positives. Precision is intuitively defined as the classifier's ability to avoid labeling a negative sample as positive.

The recall is defined as the ratio tp / (tp + fn), where tp represents the number of true positives and fn represents the number of false negatives. The recall is intuitively the classifier's capacity to locate all positive samples.

The number of instances of each class is the support. The graph depicts the total accuracy of the model we trained.

Conclusion and Future Work

We have demonstrated that we can achieve great accuracy without the need of extra devices such as Kinetic Cameras, which are both expensive and inconvenient. Simultaneously, we created the recommended system for persons with hearing difficulties, so that they may use it to communicate with normal people much more effectively in emergency circumstances where writing the message would take too long. This suggested approach may potentially be utilized as a learning module for students with hearing impairments.

Techniques for improving hand detection performance that will be part of our future study include: Because everyone nowadays carries a smartphone, designing the above-mentioned system in Android would provide mobility; Clutter model: Because the signer's face contributes the most to background variance in hand chips, (Smith, Lobo, & Shah, 2007) present a useful way to modeling facial clutter. A forearm detector can be utilized to further refine the ROI set for input to the hand detector.

REFERENCES

[1] Nanivadekar, P., & Kulkarni, V. (2014). Indian Sign Language Recognition: Database creation, Hand tracking and Segmentation. International Conference on Circuits, Systems, Communication and Information Technology Applications, (pp. 358-363).

[2] Pigou, L., Dieleman, S., Kindermans, P.-J., & Schrauwen, B. (2014). Sign Language Recognition using Convolutional Neural Networks. European Conference on Computer Vision 2014 Workshops, (pp. 572-578).

[3] Rajaganapathy, S., Aravind, B., Keerthana, B., & Sivagami, M. (2015). Conversation of Sign Language to Speech with Human Gestures. Procedia Computer Science, 10-15.

[4] Saxena, A., Jain, D. K., & Singhal, A. (2014). Sign Language Recognition Using Principal Component Analysis. Fourth International Conference on Communication Systems and Network Technologies, (pp. 810-813).

[5] Shinde, A., & Kagalkar, R. (2015). Sign Language to Text and Vice Versa Recognition using Computer Vision in Marathi. International Journal of Computer Applications, 23-28.

[6] Smith, P., Lobo, N. d., & Shah, M. (2007). Resolving hand over face occlusion. Image and Vision Computing, 1432-1448.

[7] TensorFlow. (n.d.). TensorFlow Core - Tutorials - Create an Estimator from a Keras model. Retrieved from TenserFlow Learn: tensorflow.org/tutorials/estimator/keras_model_to_estimator

[8] Yang, Q. (2010). Chinese sign language recognition based on video sequence appearance modeling. 5th IEEE Conference on Industrial Electronics and Applications, (pp. 1537-1542).

[9] Zeiler, M., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. European Conference on Computer Vision, (pp. 818-833).