2022 3rd International Conference for Emerging Technology (INCET)
Belgaum, India. May 27-29, 2022

Sign Language to Text Conversion using Deep Learning

Dr. Aruna Bhat
Computer Science and Engineering
Delhi Technological University
New Delhi, India
aruna.bhat@dtu.ac.in

Vinay Yadav
Computer Science and Engineering
Delhi Technological University
New Delhi, India
vinayyadav_2k18co387@dtu.ac.in

Vishesh Dargan
Computer Science and Engineering
Delhi Technological University
New Delhi, India
visheshdargan_2k18co389@dtu.ac.in

Yash
Computer Science and Engineering
Delhi Technological University
New Delhi, India
yash_2k18co401@dtu.ac.in

2022 3rd International Conference for Emerging Technology (INCET) | 978-1-6654-9499-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INCET54531.2022.9824885

***Abstract*—Signs and gestures are a way of communication that, instead of spoken words, makes use of movements and gestures made by hands along with changes in facial expression and changes in bodily movements. There are many distinct sign languages throughout the world, just like there are many different spoken languages. Signs and gestures are primarily used by people who cannot hear and speak and use this to express their emotions and interact with each other**

In ancient times, gestures and signs were the only way of interacting with each other, since there were no spoken languages and was considered an obvious and natural way of communicating. However, as the time passed, the utilization of sign language started becoming more prevalent only for the people who are deaf or have any hearing impairments.

Currently, over 80 million people in the world use gestures made by hands and symbols, which is a small part of the 780 crores people living on Earth as of 2020. These people depend greatly on sign languages to learn, access services and be a part of the communities. However, there is a lack of support in terms of providing basic services because of the fewer sign interpreters available.

To overcome the challenges and issues created due to the dearth of information and knowledge relating to sign languages and to provide services that people deserve, there is a need to spread the use of sign languages among the general public. However, this involves a lot of effort and does not provide the results that are equivalent to the labour done. Detecting the correct signs in images can be a complex process and can include a variety of pre-processing techniques to be performed. In this project, various images containing the signs of the American-Sign-Language have been processed and utilised for implementing a CNN model to classify these images as their alphabetical equivalent.

Keywords—Convolution Neural Network, Image Classification, Sign Language Recognition, Callbacks, Deaf & Mute

I. INTRODUCTION

Gestures and signs are one form of verbal communication which is leveraged by individuals who have lost their ability to speak or hear. According to World Health Organization data, more than 6% of the total people living on earth – 3600 lakh people – suffer from inability to hear and comprehend what others are saying. More and more deaf individuals are losing their jobs and the rate of unemployment for deaf workers, in particular, is around 70%, creating a lot of pressure from society on them. People who do not face any problems hearing and talking find it tougher to communicate with mute and deaf people. Signs and hand gestures which together form sign language is the paramount means of interacting for people who cannot talk and have difficulty listening to other people. Hearing loss that is more than 40 dB in the stronger ear is mentioned as debilitating hearing loss. Hence with the rising number of individuals who are deaf, the demand for interpreters is also increasing. Minimizing the verbal communication gap between people who have difficulty in hearing and non-hearing impaired individuals becomes a desire to ensure that everyone can communicate effectively. Sign language translation is one of the quickest-growing fields of study currently, and it is the most natural mode of communication for those who are partially deaf.

Deep learning is a data-driven machine learning methodology that uses a variety of neural network architectures to perform various tasks like object detection, segmentation, and classification in imaging. Deep learning methods use automatic feature extraction as a method of learning and have been widely used for the purpose of classification. This also includes the classification of sign languages which can further be extended to sign to text conversion.

One thing that needs to be taken into consideration during image classification is that the data should be of good quality and should have as little noise as possible. This means that in the case of the classification of sign images, any noise or background elements can affect the performance of the classification. To avoid this, high-quality images can be used and using a large dataset can also benefit the process of classification. A good classification model must be able to utilize even small size of data to identify the signs and provide the corresponding alphabets or characters.

II. RELATED WORK

Izutov and Evgeny[1] trained the ASL gesture recognition model was that could identify ASL gestures in a live stream and were trained using the metric-learning framework. Furthermore, the authors suggested residual spatio-temporal attention with the auxiliary self-supervised loss for improved model robustness to appearance changes.

The results from their study suggested that the proposed gesture recognition model may be utilised to recognise ASL signs in a real-world situation.

Pigou, Lionel, Mieke Van Herreweghe, and Joni Dambre [2] concluded that deep residual networks are capable of learning different patterns in continuous gesture and sign language films using basic RGB cameras and with very little preprocessing. They consider sign language & gesture recognition as a framewise classification problem. They have used temporal convolutions as well as current breakthroughs in the area of deep learning such as batch normalisation, residual networks, and (ELUs)exponential linear units to solve it. To test the trained models, the authors employed two different sign language corpora and the biggest known gesture dataset. The ConGD, the Dutch Sign Language Corpus and the Flemish Sign Language Corpus (Corpus VGT) were used to test the models built by the authors (Corpus NGT). The accuracy of the Corpus NGT was 73.3 percent, whereas the accuracy of the Corpus VGT was 55.7 percent.

Enriquez, Manuel, Jose et.al [3] presented the usage of the skeleton-based MS-G3D architecture for Isolated Sign Language Recognition. The argument behind this decision is that it allows for a more reliable semantic link between body parts and hands in sign and gesture language dynamics. The authors compared their technique to the S3D, a SOTA solution based on raw RGB input. MS-G3D outperforms S3D if we compare it in independent and fusion training on shared ISLR (AUTSL) data, with accuracy comparable to the best on the AUTSL validation set of the 2021 ChaLearn LAP LSSII SLR Challenge (RGB TRACK)6. The researchers also looked at the impacts of transfer learning while training MS-G3D models, and found that a model that had been previously trained and has pre trained weights on a much larger dataset in a dissimilar language and under somewhat non identical acquisition conditions did not help with the ISLR task for medium vocabulary. The top 1% accuracy for the baseline model came out to be 42.58%. Moreover, the accuracy for the S3D model for the top 1% came out to be 90.27%. Whereas the top 1% accuracy for MS-G3D came out to be 95.38%

TABLE I. PERFORMANCE OF DIFFERENT ARCHITECTURES IN [3]

Model	Stream	Top1(%)	Top5(%)
Baseline	RGB	42.58	-
S3D	RGB	90.27	97.98
MS-G3D	Joints	95.38	99.37
	Bones	94.50	99.07
	Joint-motion	92.92	99.16
	Bone-motion	90.22	98.60

Al-Hammadi, Muneer, Muhammad, Mansour et. al [4] studied 3DCNN which is being used to identify hand gestures. During the preprocessing stage, linear sampling was used to normalise the temporal dimension of hand motion data. To normalise the spatial dimensions, the length of the recognised face and human body part ratios were used. Then, in two techniques, the authors employed 3DCNN for feature learning. In the first technique, a single 3DCNN instance was trained to extract hand motion features from the whole video. In the second approach, three instances of the 3DCNN structure were trained to extract hand motion features from the beginning, middle, and end of the video clip. Before being supplied to the classifier, these region-based characteristics were merged. MLP, LSTM, and an autoencoder were used for feature fusion. For categorization, the authors employed a SoftMax active layer in both techniques. Various datasets were used to assess the suggested methods. The three datasets performed excellently in both signer-dependent and signer-independent modes. Six different state-of-the-art methodologies from the literature were compared to the proposed approaches. They excelled in four of these strategies while being on par with the other two. On the KSU-SSL Dataset, an accuracy of 96.69 percent was attained in the signer dependent mode. On the KSU-SSL dataset, an accuracy of 72.35 percent was reached in signer independent mode.

Bantupalli and Xie [11] have proposed a model for extracting spatial characteristics from the video stream using a CNN (Inception) for Sign Language Recognition. After that, temporal information from video sequences are retrieved using an LSTM (Long Short-Term Memory).

TABLE II. PERFORMANCE OF [11] WITH DIFFERENT SAMPLE SIZES

# of Signs	Accuracy with Soft Layer	Accuracy with Pool Layer
10	90%	55%
50	92%	58%
100	93%	58%
150	91%	55%

Ankit Ojha, Ayush Pandey et. al [9], have proposed a SOTA sign language to text and speech conversion method using convolutional neural networks. The images are generated from OpenCV video streams with the images processed as grayscale having dimensions of 50x50. A 95 percent accurate fingerspelling sign language translator for American Sign Language has been obtained.

Gore, Sayali, Namrata Salvi, and Swati Singh [10], developed a method for transforming sign language to text based on CNN. This algorithm is particularly useful for improving recognition accuracy when faced with difficult conditions such as scale, rotation, and translation. The dataset used is quite large, which improves the result's accuracy.

Pushpalatha, M. N., A. Parkavi, R. S. Sachin, Amy S. Vadakkan, Aakifha Khateeb, and K. P. Deeksha. [12], developed a model to classify some of the most important words used in American Sign Language with a 92 percent accuracy. Our approach allowed people with speech impairments to communicate with others without difficulty. Because present technologies are both expensive and inconvenient to use, the proposed solution employs posenet and transfer learning to allow users to freely express their opinions.

Yogya Tewari, Payal Soni, Shubham Singh, Murali Saketh Turlapati and Avani Bhuva [14] Using image processing and deep learning techniques, designed a new system for physically challenged people to recognise American Sign Language hand gestures. The Canny edge detection technique is used to retrieve the hand's edges. To speed up the procedure, the image is shrunk and processed. The Convolutional Neural Networks (CNNs) were trained to

predict gestures with 99 percent accuracy. The proposed mechanism's performance changes depending on the input image to the CNN for training.

Kausar, Sumaira Javed, Muhammad PY [15], have tried to investigate the shortcomings and challenges while carrying out sign language recognition. The authors understood that the following problems act like a boulder while building a sign language recognition system: variety in signs and gestures, feature selection and after that, extraction, segmentation and invariance. After carefully studying about the above-mentioned problems, the authors have laid out further guidelines for scientists who are working in the subject of identification of sign languages. The authors concluded that rather than using an external aid for segmentation, it should be done on the basis of vision. The authors also concluded that techniques for categorization and segmentation that do not impose significant limits on the signer's surroundings should be used. For example, if segmentation is based on both geometrical and skin colour aspects, limits on backdrop colours and the signer's clothing can be reduced.

III. METHODOLOGY

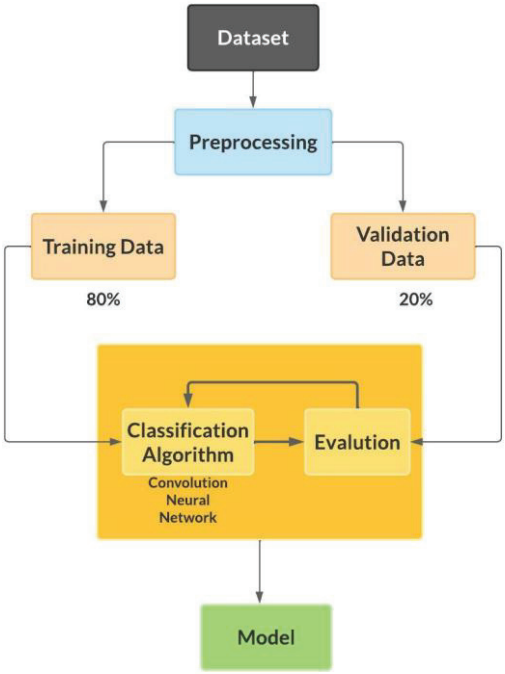

Fig. 1 Work Flow

The data was preprocessed by first converting the coloured images to greyscale images. Apart from that, gaussian blur was used to reduce image noise. The preprocessing steps also included binarization of images and the images were resized to a size of 128 x 128 from a size of 400 x 400. This data was divided into training data and validation have a split of 80 : 20 within the data. Once the preprocessing steps were completed, the data was passed into the model.

Alogirthm:

1. **Pre-processing** of Image data
   - **a.** Convert the image from rgb to gray scale
   - **b.** Applied Gaussian Blur to smooth the image
   - **c.** Segmentation of image using adaptive thresholding
   - **d.** Convert image to binary image for better retrieval of information
   - **e.** Resize the image to (128 X 128).
2. divide the dataset in training and validation data with a split of **80:20**.
3. **Input:** $\{X_1 \dots \dots X_m\}, \{I_1 \dots \dots I_m\}$
4. divide input into batches and forward it for training
5. **for** k = 1: K **do**
6. train CNN $f(X; \theta_k)$
7. **end**
8. **for** m = 1: M **do**
9. **for** k = 1: K **do**
10. obtain $P_m^k = f(X_m; \theta_k)$
11. **end**
12. **end**
13. **Output:** $\{\{P_1^1 \dots . P_1^K\} \dots \{P_M^1 \dots . P_M^K\}\}$

IV. DATASET

There are a large number of choices for the language selected for the conversion of sign language into text. However, English being one of the most widely spoken languages was chosen in the scope of this project. In terms of English, there are a number of signs and gestures being used in different regions of the world. This includes American, British and the Indian Sign Language among many others. In our project, the American-Sign-Language is chosen as the set of choices as large datasets are easily available.

American-Sign-Language (ASL) can be considered as a natural language with linguistic characteristics similar to spoken languages and a grammar slightly distinct from English. Hand gestures and face movements are used to express ASL. ASL was not created by a single individual or any known committee. Although the original origins of ASL are unspecified, some speculate that it evolved more than 200 years ago through the mingling of local sign languages with French-Sign Language. The current ASL incorporates parts of French-Sign_Language as well as the original local sign languages, which have evolved into a rich, compound, and mature language over time.

To create the data for the project, a number of datasets have been used that are publicly available. Most of the datasets available for American Sign Language contain all of the alphabets separated into different directories. We have extended the dataset and have also added images for all the digits from 0 to 9 to create a combination of alphabets and numbers. This has helped us develop data that provides more utility and is more inclusive. A subset of [14] has been used as a source for the images of all the alphabets and the images of all the digits have been added to create the final dataset for the project. The data contains 2515 images which have been divided into a ratio of 80:20 to create the training set and validations set.

The images given below show the initial dataset which is in coloured format. This is followed by images on which the model has been trained.

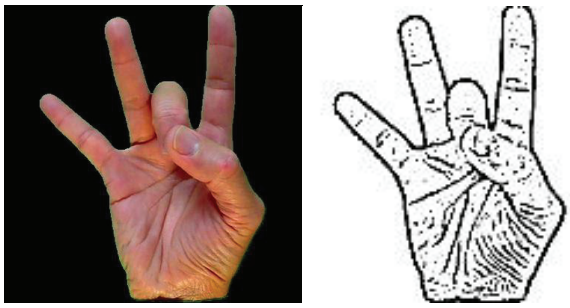

Fig. 2 Initial and Preprocessed Image

V. Data Preprocessing

Preprocessing data means bringing the data into a predictable and analyzable form for further processing of data. In our project, information is in the form of images. In image-data preprocessing, some fundamental analysis and transformation are performed over a predetermined image. This preprocessed image is more useful for performing some other, more meaningful analytical task afterwards.

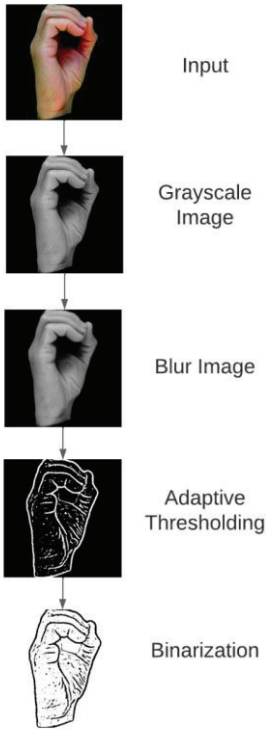

Fig. 3 Preprocessing Framework

The image preprocessing framework used in the preprocessing of the image is some simple image processing techniques. These techniques are:

Converting an image into a grayscale image: A grayscale image (also known as a grey level image) is one in which the only colours used are grey hues. When compared to other colour photographs, such images require less information to be delivered for each pixel. In RGB space, a 'grey' colour is one in which the red, green, and blue components are all equal in intensity. In contrast to the three distinct intensities necessary to identify each pixel in a full-color image, it becomes imperative to designate a single intensity value for each pixel.

Use of Gaussian Blur: A Gaussian blur is the effect of blurring a picture with a Gaussian function (also known as "Gaussian smoothing"). It is frequently used to reduce image noise and detail.

Applying Adaptive Thresholding: Adaptive thresholding is a technique for segmenting an image by giving all pixels with intensity levels greater than a threshold a foreground value and the remainder a background value. To distinguish desired foreground visual items from the background, the difference in pixel intensities of each region is employed.

Binarization: The process of transforming a grayscale (multi-tone) image to a black-and-white (two-tone) image is known as binarization. Determine the grey scale threshold value and whether or not a pixel has a certain grey value to begin the binarization process. The pixels are changed to white if their grey value exceeds the threshold. Similarly, if the grey value of the pixels falls below the threshold, they are transformed to black.

Resizing: The scaling of a picture is referred to as resizing. You'll need to resize a picture if you want to adjust the overall number of pixels in it.

The available images are of size 400x400. First, noise is reduced from each image using gaussian blur then the segmentation is done using adaptive thresholding to get the desired part of the image. This is followed by binarization which converts each image into black and white. At last the image size is reduced to 128x128. The specific dimension was chosen because it prevents too much information from being lost and helps to shorten the training period.

VI. Training the Model

The model takes input as an image of shape 128x128x1. It consists of two convolution layers along with max-pooling for dimensionality reduction. The two layers resulted in a total of 9568 trainable parameters. Except for the last layer, each layer is followed by a batch normalization layer and a ReLU activation layer. The output layer uses the SoftMax function. The total number of trainable parameters generated by the model is 3,557,654. While creating our model, the pooling layer that we have used is the Max-Pooling layer with a pool size of 2x2 which means that the highest value inside a 2x2 array is kept.

In every of the layers, we have leveraged the ReLu (Rectified Linear Unit).For every input pixel, max(x,0) is calculated by ReLu . This provides the formula with some additional nonlinearity and makes it easier to learn more intricate features. It also helps in the removal of the vanishing gradient problem as well as decreasing the time taken in the calculations during training.

We have also used the softmax activation function in the last layer of the network which is a dense layer. In neural network models that predict a multinomial probability distribution, we have to use softmax as the activation in the output layer. Softmax is therefore utilised as the activation function in classification problems having more than 2 target classes.

To avoid the overfitting of our model we utilized callbacks. A callback is a type of object that can be used to perform actions at different points of the training

process. The number of epochs can be monitored with callbacks. We set up the 'val accuracy' to be checked with a patience of 10 epochs and a min delta value of 0% using a custom Early Stopping Callback. The model created has been trained for 100 epochs on the dataset that we generated. **The model achieved an accuracy of 84%.**

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 126, 126, 32)      320

 max_pooling2d_7 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 conv2d_8 (Conv2D)           (None, 61, 61, 32)        9248

 max_pooling2d_8 (MaxPooling  (None, 30, 30, 32)       0
 2D)

 flatten_3 (Flatten)         (None, 28800)             0

 dense_12 (Dense)            (None, 128)               3686528

 dropout_6 (Dropout)         (None, 128)               0

 dense_13 (Dense)            (None, 96)                12384

 dropout_7 (Dropout)         (None, 96)                0

 dense_14 (Dense)            (None, 64)                6208

 dense_15 (Dense)            (None, 36)                2340

=================================================================
Total params: 3,717,028
Trainable params: 3,717,028
Non-trainable params: 0
```

Fig. 4 Model Summary

VII. Result And Analysis

The DL model that we created was trained on the dataset we have generated. Binary cross entropy loss function was used to calculate the loss while training the model. The loss function is supposed to be decreased with consecutive epochs. The loss is calculated using the following equation:

$$Loss = -(y^i \log(y) + (1 - y^i) \log(1 - y))$$

where, y^i is i^{th} vector in output, y is the predicted probability.

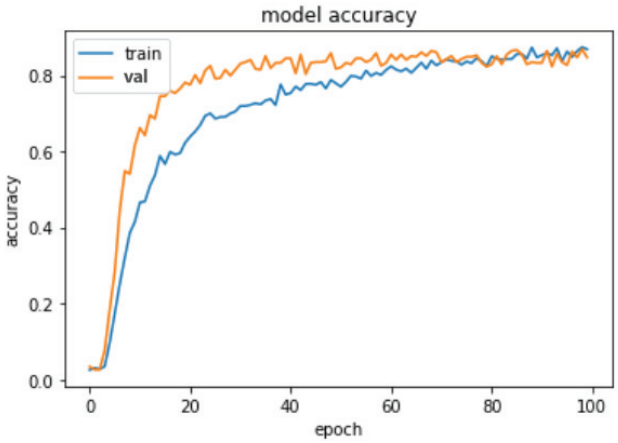

Fig. 5 Accuracy Curve

Fig.5 shows the accuracy per epoch for the phase of validation and training for the model. It can be inferred from the accuracy plot that with consecutive epochs, model accuracy tends to increase, but in some cases, unstable performance was noticed while comparing validation and training results.

Fig.6 shows the loss per epoch for the training phase and testing phase for the model. The plots show us that with consecutive epochs, the loss rate decreases, but in some cases, unstable performance was noticed while comparing testing and training results. The validation loss started out at 3.58% and decreased drastically with the increasing epochs to just 0.39%. The training loss also followed a similar trajectory. It started out at 3.60% and kept on decreasing until 0.40%.

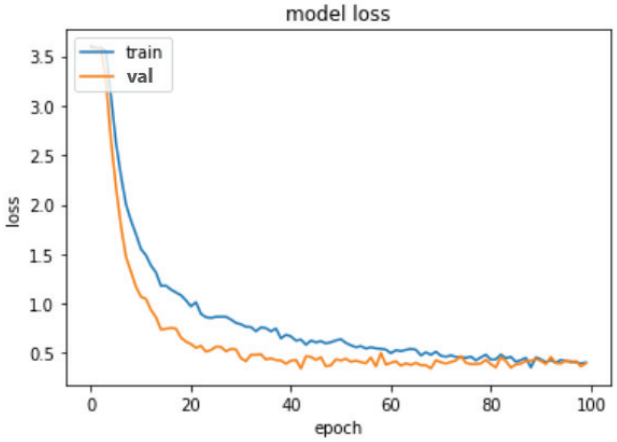

Fig. 6 Loss Curve

The following is the Classification report for our deep learning model. In terms of precision, most of the classes had a precision score above 85%. Through the results, we can infer those classes depicting {8:'8', 9:'9', 15:'F', 17:'H'} were detected with very high precision and F - 1 scores.

	precision	recall	f1-score	support
0	0.50	0.77	0.61	70
1	0.69	1.00	0.82	70
2	0.77	0.84	0.80	70
3	0.99	1.00	0.99	70
4	0.89	0.81	0.85	70
5	0.83	1.00	0.91	70
6	0.87	0.19	0.31	70
7	0.94	0.94	0.94	70
8	1.00	0.84	0.91	70
9	1.00	0.93	0.96	70
10	0.80	0.87	0.84	70
11	0.98	0.80	0.88	70
12	0.96	0.91	0.93	70
13	0.89	0.94	0.92	70
14	0.94	0.84	0.89	70
15	1.00	0.89	0.94	70
16	0.93	0.94	0.94	70
17	1.00	0.93	0.96	70
18	0.98	0.91	0.95	70
19	0.99	0.94	0.96	70
20	0.75	0.84	0.79	70
21	0.96	1.00	0.98	70
22	0.68	0.94	0.79	70
23	0.90	0.53	0.67	70
24	0.49	0.37	0.42	70
25	0.97	0.99	0.98	70
26	0.93	1.00	0.97	70
27	0.76	0.90	0.82	70
28	0.87	0.89	0.88	70
29	0.81	0.89	0.85	65
30	0.84	0.37	0.51	70
31	0.87	0.64	0.74	70
32	0.51	0.99	0.67	70
33	0.97	0.84	0.90	70
34	0.93	1.00	0.97	70
35	0.97	0.90	0.93	70
accuracy			0.84	2515
macro avg	0.87	0.84	0.84	2515
weighted avg	0.87	0.84	0.84	2515

Fig. 7 Classification Report

The relevance of precision in our study is to identify the quality of positive predictions. The average precision of the model is 0.87 which depicts that the model was good at predicting the actual class of a symbol out of the classes predicted for a symbol. The high precision scores of the classes indicates that the number of true positives were very high for the respective classes. The classes having higher precisions and recall also have higher F1 Score since the f1 score is the harmonic mean of precision and recall.

However, the classes belonging to {0:0, 1:1, 2:2, 24: O, 27: R} have low precision scores. The possible reason for the same accounts to the fact that the size of data per class is relatively less as of now, which leads to the model being unable to learn how to classify those classes correctly.

VIII. Conclusion

Sign language is one of the most useful tools for facilitating communication between deaf and mute people and the rest of society. Though sign language can be used to communicate, the target person must have a basic understanding of the language, which is not always possible. As a result, our project lowers these barriers. This project was created as a proof-of-concept to see if it was possible to recognise sign language. This project allows ordinary people to communicate with deaf or dumb people using sign language, with the text being converted to images.

In a number of scenarios, a sign language interpreter becomes an object of high utility. In educational institutions, hospitals, airports, and courts, anybody may use this technology to comprehend and communicate in sign language. It allows persons with normal hearing to communicate with others who have trouble hearing.

In this project, we have developed a model that converts sign language to text using deep learning. The algorithm used to develop the model is Convolution Neural Network. The dataset used is a combination of multiple datasets. The model created by us has been trained for 100 epochs on the dataset that we generated. **The model achieved an accuracy of 84%.** The model is performing quite well with the amount of data we are using. By increasing the dataset size, we can improve the accuracy.

IX. Future Work

Sign language is part and parcel in the lives of deaf and mute people.With over 5% of the world's population i.e more than 70 million people are deaf and dumb and have to face challenges at each and every stage of life. Hence, it becomes imperative to make advancements in the arena of sign language and gesture identification so that it becomes easier for people who cannot hear and speak. In this paper, we have successfully designed a model which converts sign language to text for not only the English alphabet but also the numerical numbers from 0 to 9.

In the study that we have conducted, we have trained a DL model from scratch to build a sign language and gesture recognition system. With the recent advancements in pre-trained models in deep learning, we aim to use transfer learning on different pre-trained classification models to build our sign language recognition system which will help us increase the accuracy of the model. Some of the pre-trained models that aim to use are VGG 16, Resnet 50, Inception v3, EfficientNet etc.

The model that we have trained is pretty lightweight and can be exported. Hence in the future, we also aim to deploy our model in the form of a web app or an android app with a simple and interactive GUI which will make it easier for the deaf and dumb people to use it and interact with each other in a simple and hassle free way.

Acknowledgment

We would like to express our deepest appreciation for Dr. Aruna Bhat for providing us with the opportunity to write a research paper on the topic "Sign Language to Text Conversion using Deep Learning". We would like to thank her for her continued support and guiding us in every stage of this research paper.

References

[1] Izutov, Evgeny. "ASL Recognition with Metric-Learning based Lightweight Network." arXiv preprint arXiv:2004.05054 (2020).

[2] Pigou, Lionel, Mieke Van Herreweghe, and Joni Dambre. "Gesture and sign language recognition with temporal residual networks." In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 3086-3093. 2017.

[3] Vazquez-Enriquez, Manuel, Jose L. Alba-Castro, Laura Docio-Fernandez, and Eduardo Rodriguez-Banga. "Isolated Sign Language Recognition With Multi-Scale Spatial-Temporal Graph Convolutional Networks." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3462-3471. 2021.

[4] Al-Hammadi, Muneer, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, Mohamed A. Bencherif, and Mohamed Amine Mekhtiche. "Hand gesture recognition for sign language using 3DCNN." IEEE Access 8 (2020): 79491-79509.

[5] Truong, Vi NT, Chuan-Kai Yang, and Quoc-Viet Tran. "A translator for American sign language to text and speech." In 2016 IEEE 5th Global Conference on Consumer Electronics, pp. 1-2. IEEE, 2016.

[6] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." In 2017 International Conference on Engineering and Technology (ICET), pp. 1-6. Ieee, 2017.

[7] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521, no. 7553 (2015): 436-444.

[8] Kartik, P. V. S. M. S., Konjeti BVNS Sumanth, VNV Sri Ram, and P. Prakash. "Sign Language to Text Conversion Using Deep Learning." In Inventive Communication and Computational Technologies, pp. 219-227. Springer, Singapore, 2021.

[9] Ankit Ojha, Ayush Pandey, Shubham Maurya, Abhishek Thakur, Dr. Dayananda P, 2020, Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15)

[10] Gore, Sayali, Namrata Salvi, and Swati Singh. "Conversion of Sign Language into Text Using Machine Learning Technique." International Journal of Research in Engineering, Science and Management 4, no. 5 (2021): 126-128.

[11] Bantupalli, Kshitij, and Ying Xie. "American sign language recognition using deep learning and computer vision." In 2018 IEEE International Conference on Big Data (Big Data), pp. 4896-4899. IEEE, 2018.

[12] Pushpalatha, M. N., A. Parkavi, R. S. Sachin, Amy S. Vadakkan, Aakifha Khateeb, and K. P. Deeksha. "Sign Language Converter Using Feature Extractor and PoseNet." Webology 19, no. 1 (2022).

[13] Joy, J., Balakrishnan, K. & Madhavankutty, S. SignText: a web-based tool for providing accessible text book contents for Deaf learners. Univ Access Inf Soc (2021). https://doi.org/10.1007/s10209-021-00801-7

[14] Yogya Tewari, Payal Soni, Shubham Singh, Murali Saketh Turlapati and Avani Bhuva Conference: 2021 International Conference on

Communication information and Computing Technology (ICCICT), Year: 2021, Page 1

DOI: 10.1109/ICCICT50803.2021.9510094

[15] TY - BOOK AU - Kausar, Sumaira AU - Javed, Muhammad PY - 2011/12/01 SP - 95 EP - 98 T1 - A Survey on Sign Language Recognition DO - 10.1109/FIT.2011.25 JO - Proceedings - 2011 9th International Conference on Frontiers of Information Technology, FIT 2011 ER