Sign to Speech

AND

Speech to Sign

By
Vemula Satya Pavan Kalyan (2019BCS-069)
Dasari Jayanth (2019BCS-016)
Jay Rajesh Shah (2019BCS-057)

TABLE OF CONTENTS

Abstract:

In this project, to close the gap in communication with impaired people and make life much easier for people with disabilities & for us to interact with them, we have trained various models on the sign language dataset. Dataset has been augmented to make it more generalized. Among various models (pre-trained and DL models) we trained, we got the best results for the MobileNet pre-trained model, 66.06% accuracy on the augmented data. This best model is being used for other tasks of the remaining project. For speech to text, we used Google's Deepspeech pre-trained model.

Introduction:

Sign Language is the most significant way of communication between impaired people. Sign language is a type of language that uses hand movements, facial expressions, and body language to communicate.

There isn't one universal sign language for all, every country has its own sign language, but there might be similarities. Most languages we have been exposed to have both a spoken and a written form, whereas sign language has neither, which raises the misconception that sign language is not actual language. Sign languages exhibit almost all features that spoken languages have, such as developing naturally rather than artificially and being a rule-governed communication system. Sign languages do not share the grammar of their spoken counterparts. ASL(American Sign Language) is the most widely used among various sign languages.

In ASL, there are distinct symbols for many characters, words, and phrases. The language is complex due to the use of so many symbols. Fingerspelling is expressing Sign Language by using alphabet characters and some other representational characters to spell out the text.

People with hearing and speaking difficulties face daily social isolation and miscommunication issues. This project is motivated to provide assistive technology that allows differently disabled people to communicate in their language.(ASL is used in this project).

Problem Statement:

Communication is essential in our daily lives. People with speaking and hearing disabilities use sign language for communication. As most people cannot understand sign language, it is challenging to communicate with them. So, To ease the communication, build different ML models to convert sign language to speech and speech to sign language.

Related Works:

Sign languages are specially defined hand gestures that are organized collections of gestures having specific meanings; it is used by people suffering from hearing and speech impairment to communicate in everyday life [3]. The use of movements of the hands, face, and body as communication mediums is employed by sign language. There are 300 different sign languages available worldwide [5].

Sign language recognition is a vast area for research where much work has been done, but still, various things need to be addressed. Researchers have developed efficient data acquisition and classification methods, which are divided into direct measurement techniques and vision-based methods [3][7]. The direct measurement techniques are based on using special devices to capture motion like motion data gloves, motion capturing systems, or sensors. The primary device employed as an input process in Sign language recognition systems is the camera [6][13]. The motion data that has been extracted provide a vivid and accurate tracking of different parts of hands and other body parts, which leads to more motion concise data to build a state of the art sign language recognition techniques research.

The computer vision-based sign language recognition approaches rely only on the extraction of discriminative spatial and temporal data from RGB images. Most computer vision techniques will initially try to track and extract the hand regions before their classification [3]. The current state of the art hand detection methods also use face detection and subtraction and subtraction of background to recognize only the moving parts in a scene [10][11].

Hands detection is achieved by semantic segmentation and skin color detection as the skin color is easily distinguishable [8][9]. Though the other body parts like the face and arms can be mistakenly recognized as hands, the current hand detection methods also use face detection and subtraction and background subtraction to recognize only the moving parts in a scene [10][11]. To attain accurate and robust hand tracking, particularly in obstructions, authors employed filtering techniques, such as Kalman and particle filters [10][12].

Leap Motion Controller is another system that is used for data acquisition[14][15]. It can operate around 200 frames per second and detect and track the hands, fingers, and objects that look like fingers. Most of the researchers collect their training dataset by recording it from their signer as finding a sign language dataset is a problem [2].

Objective:

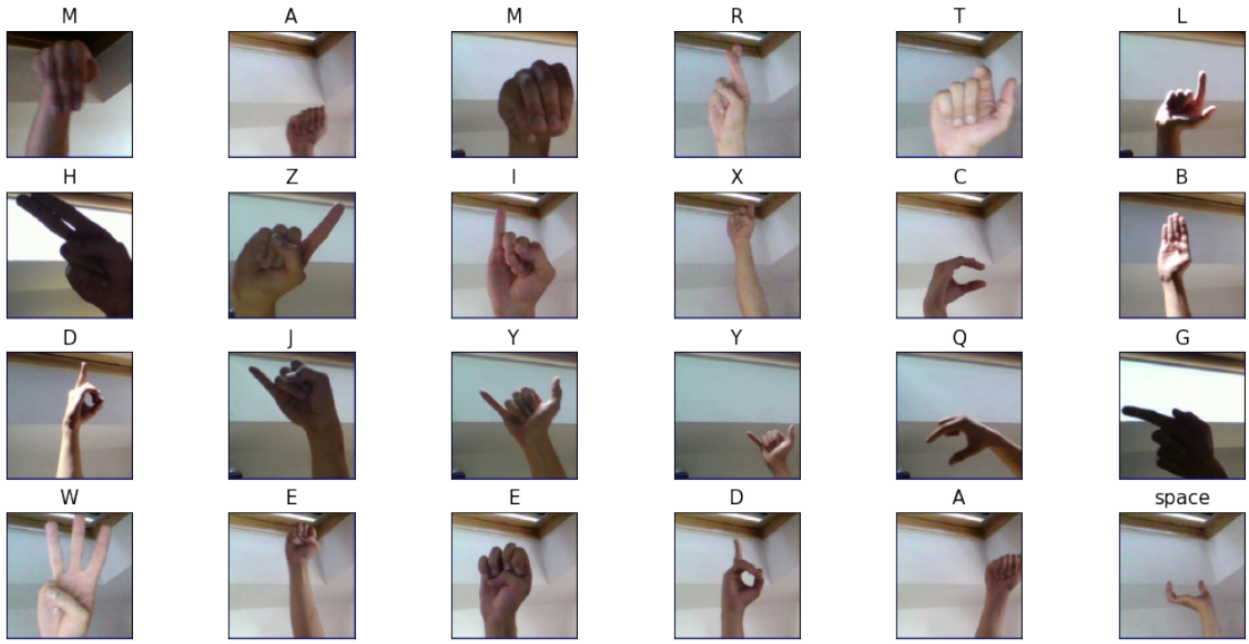
This project aims to make communication easier with impaired people for both them and others. To achieve our goal, we have to understand Sign Language. So, to make it easier, we want to build an ML model which translates the American Sign Language into speech to the English language for us(sign to text and text to speech). Another ML model to translate the speech into sign language video (speech to text and text to sign and merge those glosses into the video) for those with deaf disabilities.

Dataset:

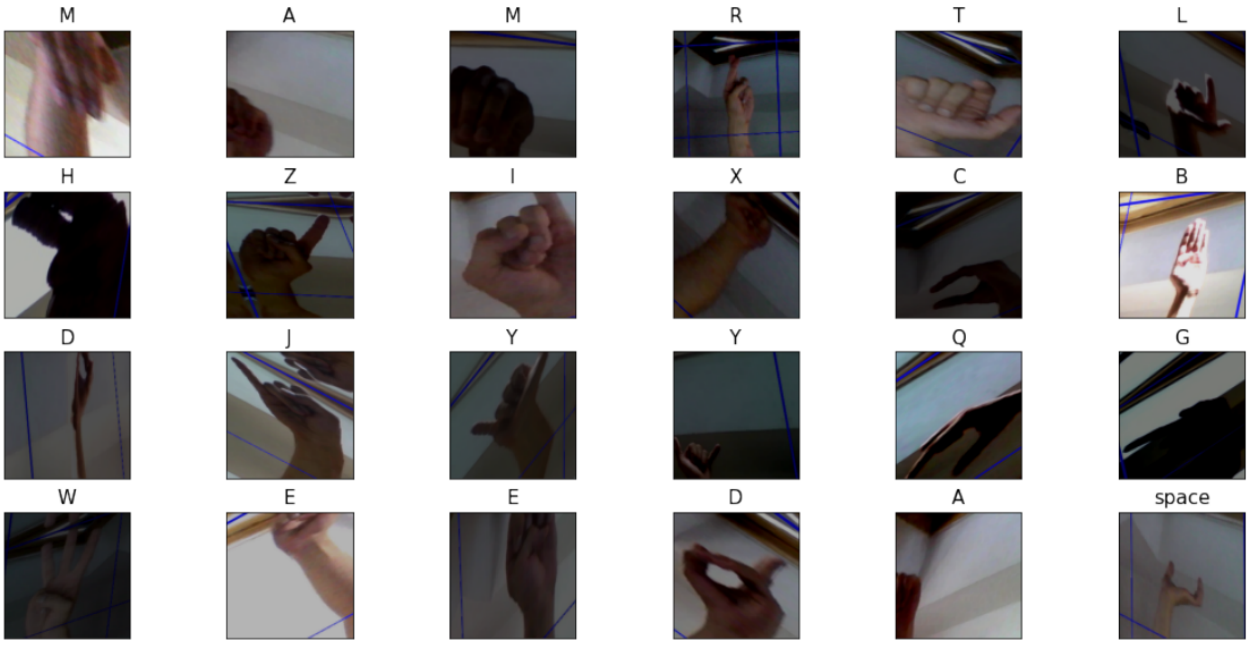
The ASL Dataset was taken from kaggle[1]. This Dataset Contains 84,000 images of 28 classes[A-Z, space, nothing]. In Transfer Learning, All the pre-trained models require their input image size to be 224x224 pixels. So, training data has been scaled to that size.
The data is clean. So, to make it more generalized without more data, to improve performance for new users following data augmentation has been performed on the previous data.

Augmentation Type	Value
Rotation	20%
Width Shift	10%
Height Shift	10%
Brightness	20%-100%
Shear	45%
Zoom	50%-150%
Channel Shift	100 px

Table:1 Augmentations Applied



pic:1 Images before Data Augmentation



pic:2 Images after Data Augmentation

Methodology:

ADAM Optimiser:

The previous optimization algorithms were shown not to really generalize that well to a wide range of neural networks. So, to overcome this Adam optimizer is introduced. Adam stands for “Adaptive moment estimation”. Adam Optimization algorithm combines the effect of gradient descent with momentum together with gradient descent with RMSprop. For a better understanding of how Adam works refer[5]

Categorical Cross Entropy:

Also called Softmax Loss. It is a Softmax activation plus a Cross-Entropy loss. If we use this loss, we will train a CNN to output a probability over the classes for each image. It is used for multi-class classification. In the specific (and usual) case of Multi-Class classification the labels are one-hot, so only the positive class which means discarding the elements of the summation which are zero due to target labels, we can write

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

1. Deeper CNN Model

The First Model we designed for this problem is a Deeper CNN. The hyper parameters were tuned through rigorous experimentation and analysis.

Layers : Convolutional, Dense
Activation Function : ReLU
Pooling : Max Pooling
Output Layer : Softmax Activation, units 28

```
Epoch 1/5
2175/2175 [==============================] - 202s 93ms/step - loss: 0.5102 - accuracy:
0.8423 - val_loss: 1.1267 - val_accuracy: 0.7298
Epoch 2/5
2175/2175 [==============================] - 192s 88ms/step - loss: 0.1394 - accuracy:
0.9595 - val_loss: 2.0690 - val_accuracy: 0.7263
Epoch 3/5
2175/2175 [==============================] - 198s 91ms/step - loss: 0.0932 - accuracy:
0.9753 - val_loss: 1.6161 - val_accuracy: 0.7867
Epoch 4/5
2175/2175 [==============================] - 195s 89ms/step - loss: 0.0784 - accuracy:
0.9794 - val_loss: 2.5941 - val_accuracy: 0.7496
Epoch 5/5
2175/2175 [==============================] - 198s 91ms/step - loss: 0.0628 - accuracy:
0.9847 - val_loss: 2.0475 - val_accuracy: 0.7675
```

Training Accuracy⇒98.47%, Validation Accuracy ⇒ 76.75%

Layer (type)	Output Shape	Param #
conv2d_29 (Conv2D)	(None, 224, 224, 16)	448
conv2d_30 (Conv2D)	(None, 224, 224, 32)	4640
max_pooling2d_15 (MaxPooling	(None, 74, 74, 32)	0
conv2d_31 (Conv2D)	(None, 74, 74, 32)	9248
conv2d_32 (Conv2D)	(None, 74, 74, 64)	18496
max_pooling2d_16 (MaxPooling	(None, 24, 24, 64)	0
conv2d_33 (Conv2D)	(None, 24, 24, 128)	73856
conv2d_34 (Conv2D)	(None, 24, 24, 256)	295168
max_pooling2d_17 (MaxPooling	(None, 8, 8, 256)	0
batch_normalization_7 (Batch	(None, 8, 8, 256)	1024
flatten_5 (Flatten)	(None, 16384)	0
dropout_6 (Dropout)	(None, 16384)	0
dense_7 (Dense)	(None, 512)	8389120
dense_8 (Dense)	(None, 29)	14877

pic3: CNN Model Layers

2. Transfer Learning

We have experimented with various ImageNet pre-trained models to find suitable models for the task of sign language recognition.

Imported the pre-trained models from the tensorflow library along with their respective image preprocessing methods and performed transfer learning to our problem by removing top layers of the pretrained model and adding 2 extra dense layers with 128 neurons and one layer for softmax classification. We got the following results for different pre-trained models.

	Model	Validation Accuracy	Training Time (sec.)
0	MobileNet	0.660600	1166
1	EfficientNetB7	0.628512	2016
2	ResNet50	0.617900	1315
3	DenseNet201	0.613690	1430
4	ResNet101V2	0.610000	1396
5	ResNet50V2	0.578200	1252
6	VGG19	0.538300	1499
7	VGG16	0.526600	1489
8	MobileNetV2	0.504702	1221
9	Xception	0.501800	1485
10	InceptionV3	0.451500	1241

Pic4: Comparison of pre-trained models

Speech recognition:

We have used Mozilla Foundation pretrained model DeepSpeech. It has been pretrained on the dataset Common Voice Project at its core it is a RNN model with 5 layers of hidden units specially designed for training on multiple GPUs it takes speech spectrograph, it's wave files as an input. First three layers are non recurrent layers using clipped ReLU activation function. The fourth layer is Bi-Directional recurrent neural network and the final layer is a non recurrent-layer which takes inputs sum of both forward and backward units of previous layer.

System	Clean (94)	Noisy (82)	Combined (176)
Apple Dictation	14.24	43.76	26.73
Bing Speech	11.73	36.12	22.05
Google API	6.64	30.47	16.72
wit.ai	7.94	35.06	19.41
Deep Speech	**6.56**	**19.06**	**11.85**

Pic:4 Comparison of Deep Speech model with other SOTA Models

WorkFlow:

Completed Tasks

- Sign Language to Text.
    1. Data Augmentation and Preparing Data.
    2. Model Training.
        - ❖ Trained various models(from scratch) and pre-trained models, tried them out for different parameters.
    3. Converting Images of Sign Language to Text with best accuracy model from above – for live implementation.
- Speech to Text.
    1. Trained different pretrained models and saved the best model (in terms of accuracy metrics).

Future Work, To be Done

- Using a wide range of diverse dataset and Improving dataset by Image Recognition Using Object Detection methods like yolo.
- Text to Speech and Text to Sign Language.
- Merging all the above implementations.
- Sign to Speech for Live implementation(for video).
    1. Data Extraction and Augmentation of sign language glosses from videos.
    2. Predicting the language text.
    3. Output ⇒ speech of the text from the trained classifier.
- Speech to Sign for Live implementation(for video).

References:

1. American Sign Language Alphabet | Kaggle Dataset .
2. (PDF) American Sign Language Fingerspelling Recognition Using Wide Residual Networks (researchgate.net)
3. Konstantinidis, D., Dimitropoulos, K., Daras, P.: Sign language recognition based on hand and body skeletal data. 3DTV-Conference. 2018-June, (2018).
4. Better Sign Language Translation with STMC-Transformer
5. Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., Morris, M.R.: Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. 21st Int. ACM SIGACCESS Conf. Comput. Access. (2019).
6. Rosero-Montalvo, P.D., Godoy-Trujillo, P., Flores-Bosmediano, E., Carrascal-Garcia, J., 12 Otero-Potosi, S., Benitez-Pereira, H., Peluffo-Ordonez, D.H.: Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques. 2018 IEEE 3rd Ecuador Tech. Chapters Meet. ETCM 2018. (2018).
7. Zheng, L., Liang, B., Jiang, A.: Recent Advances of Deep Learning for Sign Language Recognition. DICTA 2017 - 2017 Int. Conf. Digit. Image Comput. Tech. Appl. 2017-Decem, 1–7 (2017).
8. Rautaray, S.S.: A Real Time Hand Tracking System for Interactive Applications. Int. J. Comput. Appl. 18, 975–8887 (2011).
9. Zhang, Z., Huang, F.: Hand tracking algorithm based on super-pixels feature. Proc. - 2013 Int. Conf. Inf. Sci. Cloud Comput. Companion, ISCC-C 2013. 629–634 (2014).
10. Lim, K.M., Tan, A.W.C., Tan, S.C.: A feature covariance matrix with serial particle filter for isolated sign language recognition. Expert Syst. Appl. 54, 208–218 (2016).
11. Lim, K.M., Tan, A.W.C., Tan, S.C.: Block-based histogram of optical flow for isolated sign language recognition. J. Vis. Commun. Image Represent. 40, 538–545 (2016).
12. Gaus, Y.F.A., Wong, F.: Hidden Markov Model - Based gesture recognition with overlapping hand-head/hand-hand estimated using Kalman Filter. Proc. - 3rd Int. Conf. Intell. Syst. Model. Simulation, ISMS 2012. 262–267 (2012).
13. Nikam, A.S., Ambekar, A.G.: Sign language recognition using image based hand gesture recognition techniques. Proc. 2016 Online Int. Conf. Green Eng. Technol. IC-GET 2016. (2017).
14. Mohandes, M., Aliyu, S., Deriche, M.: Arabic sign language recognition using the leap motion controller. IEEE Int. Symp. Ind. Electron. 960–965 (2014).
15. Enikeev, D.G., Mustafina, S.A.: Sign language recognition through Leap Motion controller and input prediction algorithm. J. Phys. Conf. Ser. 1715, 012008 (2021).
16. Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names (gombru.github.io)
17. Adam Optimizer paper