Sign Language to SpeechConversion(Sign2Speech)

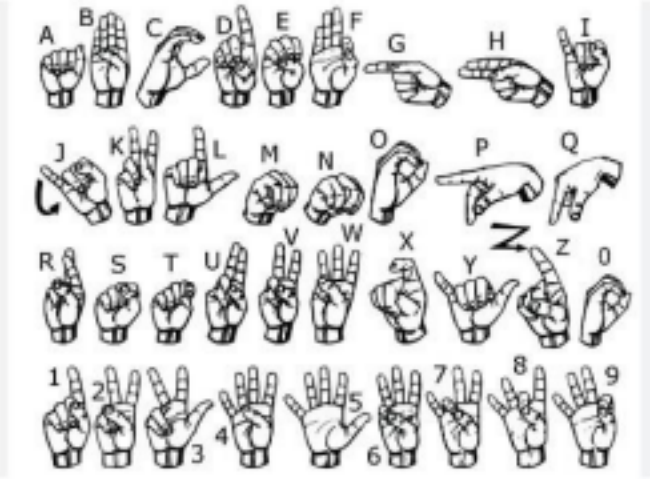
Sakshi Sonawane , Roll No. 210100150 , Mechanical Engineering Department., IIT Bombay

***Abstract*—Sign2Speech project aims to develop a model based on deep learning which aims to bridge the gap between hearing impaired community and general people. The project aims to develop a model capable of translating the American Sign Language (ASL) hand signs to spoken English. The system uses CNNs (Convolutional Neural Networks) that are present in the model to train on a large dataset and ultimately learns to predict a particular alphabet based on the hand sign provided to the model with a very high accuracy. After the model correctly predicts each alphabet, then TTS (Text to Speech) technology is used to convert the predicted text into speech. If combined with a good enough User Interface (UI), the project shows a lot of potential in reducing the communication gap between hearing impaired people and general public.**

I. Introduction (*Heading 1*)

The ability to effectively communicate is fundamental to human interaction and plays a vital role in social integration and personal development. However, for individuals with hearing impairments, traditional modes of communication such as spoken language may pose significant challenges. American Sign Language (ASL) serves as a primary means of communication for the deaf and hard of hearing community, allowing individuals to express themselves through a rich system of hand gestures, facial expressions, and body movements.

Despite its widespread use, ASL presents barriers to communication with the broader population, as many individuals are not proficient in sign language. This communication gap can lead to feelings of isolation and exclusion for individuals with hearing impairments. To address this challenge, the Sign2Speech project aims to develop an innovative solution that leverages advances in deep learning technology to facilitate communication between the hearing-impaired community and the general public. Below are the gestures for different alphabets according to ASL.



The model developed in the project takes an ASL gesture as an input outputs an alphabet that is associated with it. This in turn generates a text string if a bunch of images are passed as an input. This text string is then further converted into speech using Google TTS (Text to Speech) technology. The project aims to assist the hearing impaired community and motivate them to engage in social interactions with ease.

In this report, we provide an overview of the Sign2Speech project, including the methodology, implementation details, experimental results, and future directions. We discuss the significance of the project in addressing communication barriers for the hearing-impaired community and its potential impact on promoting inclusivity and diversity in society.

II. Data Description

The dataset used for training and testing the model is available on Kaggle and the data set is a collection of images of alphabets from the American Sign Language, separated in 27 folders which represent the various classes. Originally the data had 87,000 images which are 200x200 pixels. There are 27 classes, of which 26 are for the letters A-Z and SPACE.

Because of computational limitations, we used only 100 images per alphabet for training process but still, we were able to achieve a high enough accuracy because the images were cropped appropriately and were very clear. Some of the sample images I used are:

Below images are for letters F and H respectively. Clearly the images are cropped properly and background is same for all the images.

The labels used in the model project are 0 through 26 where A is given label 0 , B is given 1 and so on. Also the label 26 is provided for the space so that entire sentence can be created.

III. DAta Visualisation

The procedure started with me obtaining the dataset from the designated source and uploading it to my Google Drive, where I was working on a project using Google Collaboratory. The pictures were then read as NumPy arrays, and the described careful tagging was applied. Accurate label mapping to matching training images was ensured by this phase. The labeled data was then contained inside a Pandas Data Frame to preserve data organization and integrity. This data frame has 2 columns named images and labels. This methodical approach improved the project workflow's clarity and consistency in addition to facilitating more efficient data handling.

Total number of files that were read from the drive are 2730 files. Their detailed distribution is present in the report ahead.

The folders containing images of each alphabet had 100 files each:

```
Folder: A, Number of files: 100
Folder: B, Number of files: 100
Folder: C, Number of files: 100
Folder: D, Number of files: 100
Folder: E, Number of files: 100
Folder: F, Number of files: 100
Folder: G, Number of files: 100
Folder: H, Number of files: 100
Folder: I, Number of files: 100
Folder: J, Number of files: 100
Folder: K, Number of files: 100
Folder: L, Number of files: 100
Folder: M, Number of files: 100
Folder: N, Number of files: 100
Folder: O, Number of files: 100
Folder: P, Number of files: 100
Folder: Q, Number of files: 100
Folder: R, Number of files: 100
Folder: S, Number of files: 110
Folder: T, Number of files: 110
Folder: U, Number of files: 100
Folder: V, Number of files: 100
Folder: W, Number of files: 110
Folder: X, Number of files: 100
Folder: Y, Number of files: 100
Folder: Z, Number of files: 100
Folder: space, Number of files: 100
```

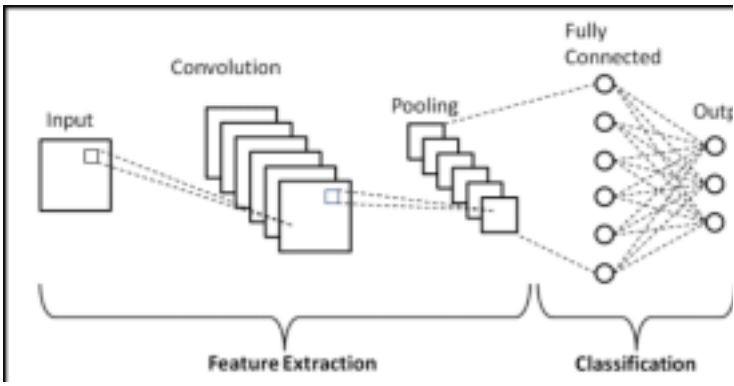
Data augmentation was not required because I was getting high accuracy without it. Here is how pandas data frame looks like

	Images	Labels
0	[[[0, 2, 254], [1, 2, 253], [6, 0, 247], [9, 0...	2
1	[[[0, 4, 253], [0, 6, 252], [0, 2, 241], [3, 4...	19
2	[[[0, 3, 252], [0, 2, 251], [4, 3, 247], [5, 0...	14
3	[[[0, 3, 252], [0, 9, 255], [0, 0, 239], [3, 2...	7
4	[[[0, 2, 252], [0, 6, 254], [0, 0, 239], [6, 3...	20

IV. Model Architecture

I used CNN for image feature extraction and recurrent neural network for sequence modelling. My model consists of 5 layers

- Input layer- Input layer accepts image data with dimensions (200, 200, 3), representing a height of 200 pixels, width of 200 pixels, and 3 color channels (RGB).
- Convolutional Layers - It begins with 32 filters of convolutional layer of everyone's size (3,3). It introduces non-linearity by applying Rectified Linear Unit activation function . Purpose is to extract low-level features from the input images. After this I applied max-pooling layer of pool size (2, 2) for down sampling the feature maps and reduce computational complexity. After the above convolutional layer, another convolutional layer is added with 64 filters of size (3, 3), followed by ReLU activation and max-pooling. This layer helps model to refine the extracted features
- Dense Layers- The features that we have extracted will now go through two layers that are densely packed and totally connected. As a hidden layer to identify intricate patterns in the data, the first dense layer is made up of 128 neurons with ReLU activation. The last dense layer uses the SoftMax activation function to create class probabilities, and its num_classes neurons correspond to the number of output classes—in this case, 27.
- Output Layer: The output layer generates a probability distribution over the 27 classes, signifying the probability of every class in relation to the input image. Subsequently, we employed the Adam optimizer to compile the model, which modifies the training rate in order to maximize convergence. We computed a loss function that yields integer-labeled multiclass classification jobs. For the model's performance during training, I used the accuracy metric.



The above summary helps in understanding the complexity and structure of the Sign2Speech model, facilitating model interpretation and debugging.

V. Training and Evaluation

The training process was done by reading the images of the data frame as tensors and passing them to the model. For this we used 5 epochs and batch size of 32. This means 5 iterations were done for the entire data and in each iteration, the model was trained 63 times suing a batch of 32 images.

The model history was also printed to get better understanding of the training process:

poch	Loss	Accuracy	Val_loss	Val_accuracy
1	205	0.57	0.1537	96.3

2	0.0362	99.1	0.0247	99.3
3	0.0361	99.05	0.0989	97.53
4	0.0326	99.40	0.0995	97.67
5	0.0155	99.5	0.0906	97.72

After training, the accuracy of model was checked on the entire test data, which had almost 730 images and an accuracy of about 97.27% was observed.

Finally, I made a final test folder on drive and uploaded a bunch of images in sequence , which ultimately form a meaningful sentence. After that I predicted the labels of these images using the model. These numerical labels were further converted into alphabetical labels and a text string was generated. This text string was then converted speech using Google Text to Speech Technology. The code for this part was taken from online resources. TTS code I used is given below:

```
 from these alphabats which then needs to be converted to speech
(alphabet_labels)

generate speech
tring, lang='en')

-memory file-like object

)
```

The speech was exactly what we expected from the model. So ultimately the project did a good job.

VI. Discussion

The Sign2Speech project demonstrates promising results with an accuracy of approximately 97.26% on the test data. This high accuracy suggests that the model is proficient in recognizing sign language gestures and converting them into spoken language, thus facilitating communication accessibility for individuals with hearing impairments. The project's success underscores the potential of deep learning models in bridging communication gaps and enhancing inclusivity in society. One of the strengths of the Sign2Speech model is its robust performance in accurately interpreting sign language gestures, even in diverse and complex scenarios.

There is however a significant limitation to our project. The model is trained using images that are of size 200,200,3 and are properly cropped. So currently the model is not capable of working in real time that is using webcam and producing the results in real time. If the model was capable of focussing only on the hand signs of images captured by the webcam, then the effectiveness of model would skyrocket.

To enhance the Sign2Speech model's performance and address its limitations, several avenues for improvement can be explored. This includes collecting more diverse and representative training data to improve the model's ability to generalize across different sign language variations and contexts. Furthermore, incorporating techniques such as data augmentation, transfer learning, and attention mechanisms could help improve the model's robustness and accuracy. Additionally, leveraging advancements in hardware accelerators and optimization techniques can enhance the efficiency and scalability of the model, making it more accessible and cost-effective to deploy in real-world settings.

References

Most of the resources that I have used for the project are online and I will be sharing the link to those sources ahead: • Purnomo, A., & Anshary, A. (2021). Fingerspelling in American Sign Language. [Image]. ResearchGate. Retrieved from https://www.researchgate.net/figure/Fingerspelling-in-American-Sign-Language-which-represents-26-letters-and-10-digits-with_fig1_346023992

- National Institute on Deafness and Other Communication Disorders (NIDCD). (n.d.). American Sign Language (ASL). Retrieved from https://www.nidcd.nih.gov/health/american-sign-language#:~:text=American%20Sign%20Language%20(ASL)%20is,of%20the%20hands%20and%20face.
- Wikipedia contributors. (2022, April 10). American Sign Language. In Wikipedia. Retrieved April 15, 2024, from https://en.wikipedia.org/wiki/American_Sign_Language
- Mooney, P. (2018). Interpret Sign Language with Deep Learning. [Data set]. Kaggle. Retrieved from https://www.kaggle.com/code/paultimothymooney/interpret-sign-language-with-deep-learning/input
- ChatGPT. OpenAI. (2022). [Software]. Retrieved from https://openai.com/chatgpt
- Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6645-6649). IEEE. DOI: 10.1109/ICASSP.2013.6638947
- Zhang, Y., Guo, Y., Wu, Y., & Gao, J. (2020). A Survey of Sign Language Recognition: Image, Video, and Wearable-Based Approaches. IEEE Transactions on Human-Machine Systems, 50(6), 525-543. DOI: 10.1109/THMS.2019.2920703