

American Sign Language (ASL) Fingerspelling Detection

Shoaib Mohammed

01 June 2021

Abstract

Approximately 70 million people around the world are deaf-mute. While translation services have become easily accessible for about 100 languages, sign language is still an area that has not been explored. Our goal is to detect & translate the letters of ASL in real-time.

Keywords — ASL, fingerspelling, Convolutional Neural Network (CNN), transfer learning, computer vision

1 Introduction

1.1 Background

According to the Communication Service for the Deaf (CSD) [1], there are 360 million deaf people worldwide. Another report by the World Health Organization (WHO) [2] bumps up the number to 466 million people suffering from disabling hearing loss. Future projections estimate 630 million people by 2030 and over 900 million people by 2050. But even in this age of technology and communication, we are yet to see a universal translation system that helps bridge the gap between people that can and cannot speak. The goal of this project is to detect and accurately translate the letters in ASL.

1.2 Scope & Limitations

The bottom line is that there is no universal sign language. For instance, the British Sign Language (BSL) differs by a great margin from the ASL. Generally speaking, a person in the US can understand spoken English in the UK but this is not the case with sign language.

It is interesting to note that there are more than 200 sign languages that are used across the world. However, if an accurate model was developed to recognize a sign language, in our case, ASL, the same methodology could be applied to recognize other sign languages. Perhaps the major limitation is classifying each sign from the sheer corpus of signs in ASL. However, since our focus is classifying the alphabets alone, our range is limited to 24 alphabets since we exclude the alphabets J & Z since these require motion.

2 Literature Survey

2.1 Population Statistics

A combined study in [2] estimates there were more than 250,000 deaf people and as many as 500,000 people who used ASL in 1972. Over the years, this number has been on the rise.

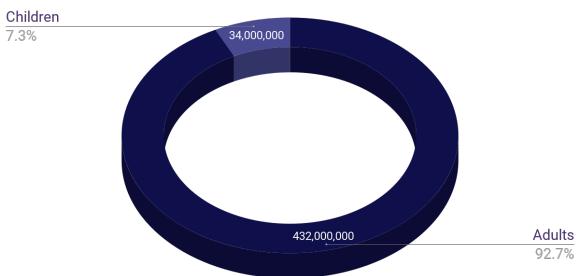


Figure 1: Distribution of the population

The Survey of Income and Program Participation (SIPP) collects data for the US population. From the research of Ross E. Mitchell [2] in the year 2006, fewer than 1 in 20 Americans or 10,000,000 people suffered from hard of hearing and close to 1,000,000 were classified as functionally deaf.

References

- [1] C. Soukup, “Communication service for the deaf,” 2021. <https://www.csd.org/>, Last accessed on 2021-05-31.
- [2] R. E. Mitchell, T. A. Young, B. Bachelder, and M. A. Karchmer, “How many people use asl in the united states? why estimates need updating,” *Sign Language Studies*, vol. 6, no. 3, pp. 306–335, 2006.

List of Figures

1	Distribution of the population	1
2	Process of serialization & deserialization . .	2
3	ASL alphabets from our dataset	2
4	Training from scratch vs. transfer learning	2

5	Transfer learning process	2
6	Max pooling vs. Average pooling	3
7	Softmax function pushing high scores close to 1 and low scores close to 0	3
8	Process of training the model	3
9	Process of testing the model	3

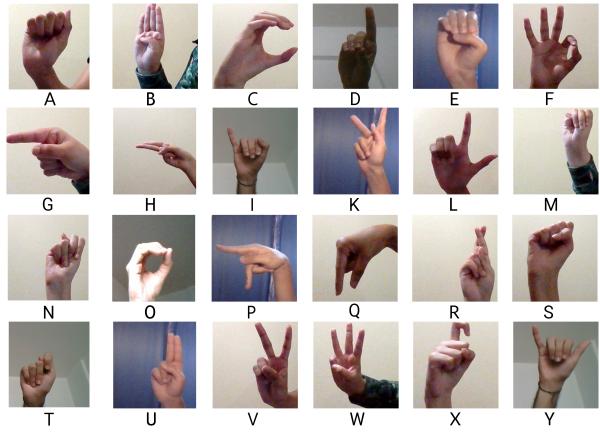


Figure 3: ASL alphabets from our dataset

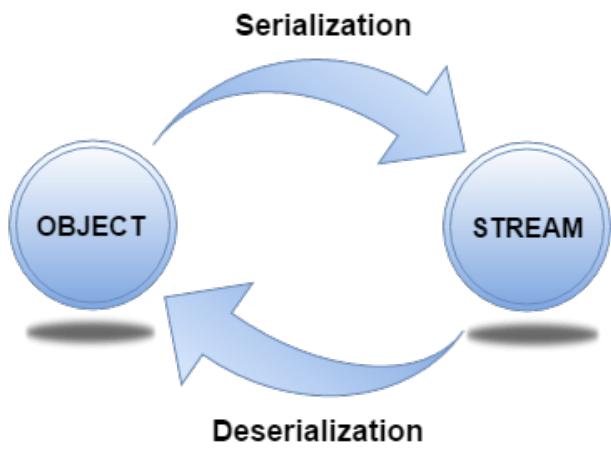


Figure 2: Process of serialization & deserialization

Model	Size	Accuracy	Parameters
MobileNetV2	14 MB	0.713	3,538,981
InceptionV3	92 MB	0.779	23,851,784
Xception	88 MB	0.790	22,910,480
InceptionResNetV2	215 MB	0.803	55,873,736

Table 1: Transfer learning model statistics

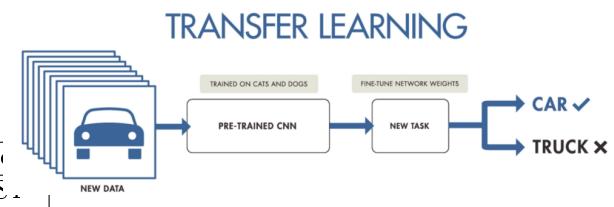
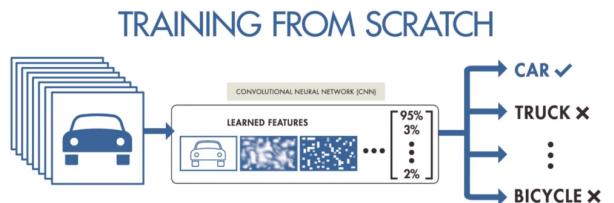


Figure 4: Training from scratch vs. transfer learning

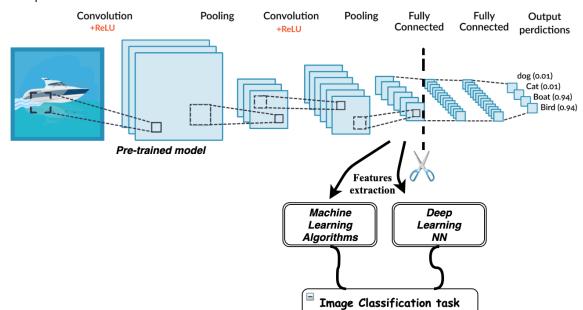


Figure 5: Transfer learning process

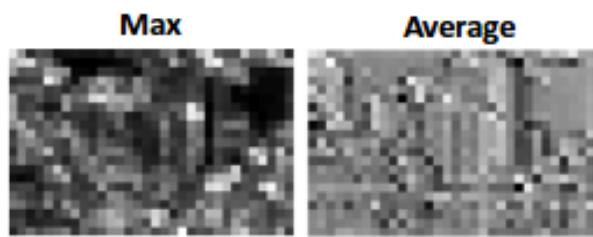
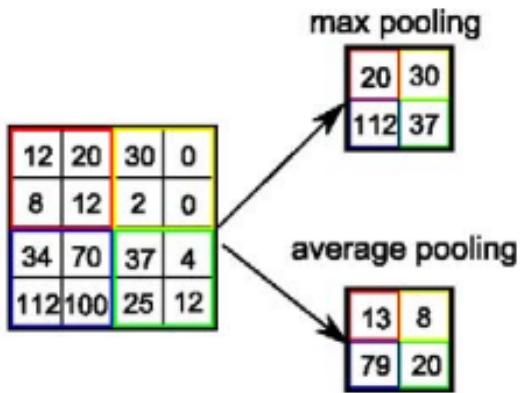


Figure 6: Max pooling vs. Average pooling

LOGITS SCORES	◦ SOFTMAX	PROBABILITIES
$y \begin{bmatrix} 2.0 \\ 1.0 \\ 0.1 \end{bmatrix}$	$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$	$\begin{array}{l} \longrightarrow p = 0.7 \\ \longrightarrow p = 0.2 \\ \longrightarrow p = 0.1 \end{array}$

Figure 7: Softmax function pushing high scores close to 1 and low scores close to 0

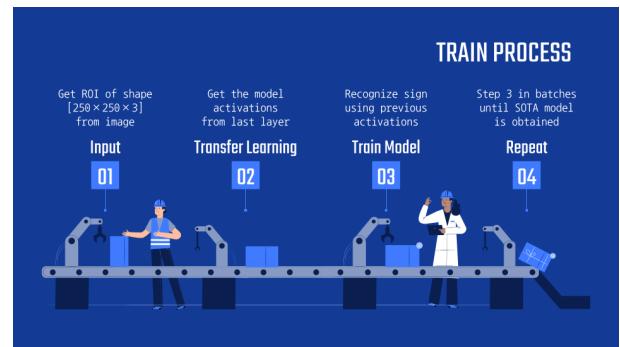


Figure 8: Process of training the model

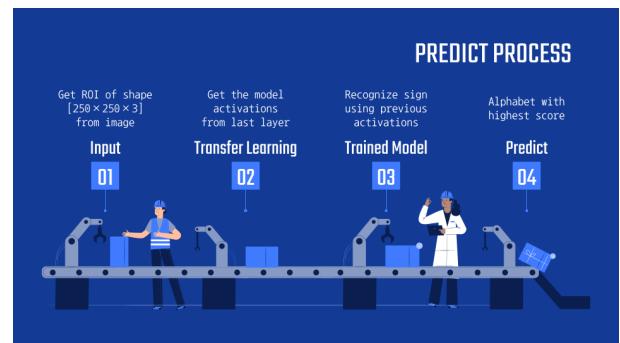


Figure 9: Process of testing the model