Graphical user interface, text, application, email

Description automatically generated

**Artificial Sign Language: A Mobile Assistant for ASL (American Sign Language) Users**

Current solutions of communication for American Sign Language (ASL) involve a physical translator who palliates the language barrier between both interlocutors. Such a system fails to encapsulate many situations. With 500,000 users of ASL in Canada and in the United States, the independence of this community is significantly affected by the lack of reliable translating means.[[1]](#footnote-2)

We have tested two machine learning models, in half a dozen different settings*.* The general design of our system remained the same. First, a visual transcript of hand signs and other physical features must be established, before accurately returning a string of words. Yet, such strings are not grammatically legible for English speakers, since “ASL is a language completely separate and distinct [in] word formation, and word order.”[[2]](#footnote-3) A second machine learning model was therefore implemented to translate between both languages.

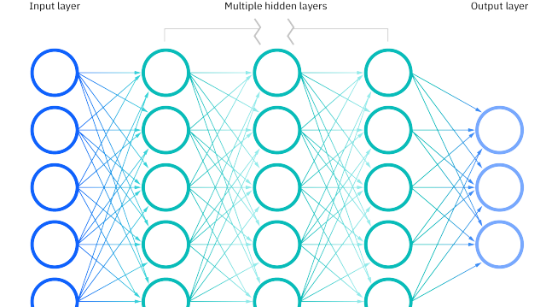
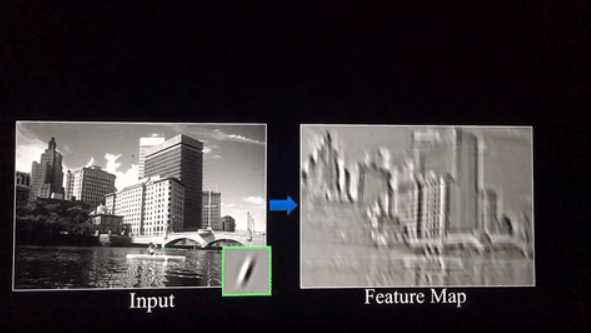
One common aspect of the models tested was the use of deep learning image classification algorithms. Deep learning involves the use of neural networks (Image 1) which contain *dense layers*, which are interconnected nodes (also known as neurons) interacting through weights.[[3]](#footnote-4) Our models use a class of artificial neural networks, called convolutional neural networks, which can be fed input data, in the shape of matrices (our images). When an image in the form of a matrix of numbers is fed to a convolutional layer, a convolution operation is done, which is an operation that uses filters, to extract relevant *features*, stored in a *feature map* (a smaller matrix, containing information about the previous layer).[[4]](#footnote-5) This process reduces the image until it reaches the last layer of the network, which contains probabilities of each label that it was fed (Image 2).[[5]](#footnote-6) The process of going from the initial image to the output layer is an *epoch*.

Image 2: Extracting Features (to reduce the matrix into its essential components)

Image 1: Deep Learning Neural Network: Each node (circle) in a layer (vertical) is connected by weights (lines)

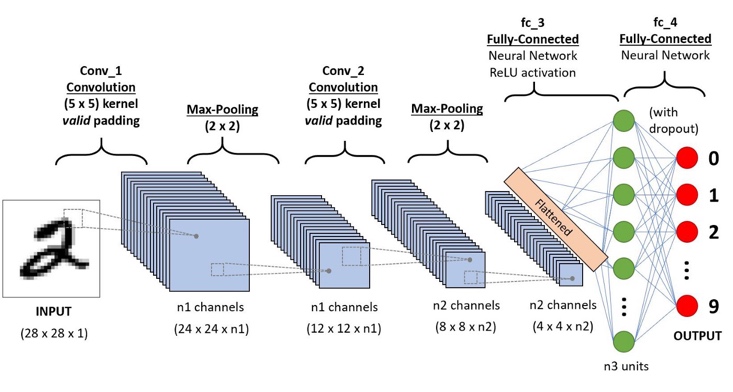
By iteratively reducing the size of an image through a filter, and comparing the resulting output with the expected output, the model learns to recognize important features in an image. To calculate the difference between expected and resulting output, the model uses a *loss function*. [[6]](#footnote-7) The model then adjusts the weight in function of the *loss*, through a technique called *backpropagation*.[[7]](#footnote-8) Simply put, if the loss is high, the model’s predictions were *very* wrong. Therefore, the weights are adjusted drastically. If the loss function is low, the model’s predictions were *close* to the right answer, and the weights are slightly adjusted. [[8]](#footnote-9) At the end of such a process, the model saves the weights that were *trained*. When given a new image, the model runs through all the weights, and a new output is given (Image 3).

Image 3: The layers transform an image of a drawn number, into a single output (a single digit from 1 to 9)

The most difficult part of our solution was in judiciously choosing a model’s architecture, and in balancing our limited computing resources. At first, we believed that we could use an object detection model, named YOLO (*You Only Look Once)*, which is a pretrained neural network that detects the location of chosen objects (such as hands) in each image. [[9]](#footnote-10) However, such an approach was not optimal since ASL does not simply involve tracking the position of the user’s arms but also tracking the movements of their hands and fingers.

In addition, the training of such models is very time-intensive, and requires a lot of computational power, which we did not have access to. Instead, we reused existing publicly available CNN architectures, which were trained by wealthy institutions, and we adjusted them for our purpose, a process known as *transfer learning*. By taking models already partially created and adding our own layers on top (thereby modifying the purpose of the models themselves), we significantly reduced the amount of preprocessing power needed to train the model. [[10]](#footnote-11)

For our project, we took advantage of two architectures: VGG-16 (Visual Geometry Group) and ResNet-50 (Residual Neural Network). VGG is a CNN architecture that was created by researchers at Oxford, which serves to detect thousands of different features, while ResNet was introduced by researchers at Microsoft in 2015. [[11]](#footnote-12) [[12]](#footnote-13) They are known to be robust models used in a variety of image classification tasks. They have been previously trained on a dataset called ImageNet, consisting of more than 14 million images. [[13]](#footnote-14)

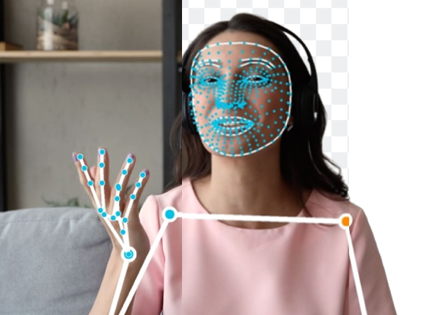
We used the MS-ASL dataset, which contains 16,000 entries of recorded video gestures that were available on YouTube, representing 1000 distinct words (classes). For each entry, a timestamp was given, indicating the relevant time frame when the gesture was done. [[14]](#footnote-15) We first proceeded to *scrape the web* (to download the videos on a computer). [[15]](#footnote-16) We later treated the data to keep relevant time frames to train and test the model, by reducing the size of the images, through different compression algorithms, and saving everything into 3D matrices, containing width, height, and RGB values. These frames were then randomly shuffled and split into two distinct datasets: training and testing. Our loss and accuracy metrics on the models were taken by evaluating the test dataset, which the model had never interacted with beforehand.

Due to limited computation, our first models only classified 25 possible classes (words), instead of 1000 words. The frames from the videos were saved at a rate of 1 frame per second. Our best accuracy with this approach was 8.06%. We narrowed it further by training models that could classify the data into 2 and 4 possible classes (words) respectively (Image 4).

Table

Description automatically generatedWhen a user tries our program, the computer records the movement using its webcam and decomposes the video into frames (10 fps). Each frame is then being fed to our model and outputs a prediction of the gesture represented. At the end, based on the predictions from all frames, the model concludes, through a weighted voting system by the majority, the nature of the gesture represented in the video.

Image 4: By training our model on different architectures, we observe that the ResNet50 architecture could best predict 4 different words. Yet the training time for 4 labels hovered around 45 minutes. To train 1000 labels, we would need to further the efficiency of our model training and increase the number of frames used (we only used 10 frames out of 30 per second).

After such a task, our model returns bribes of words, which are not legible for the common English speaker, as the syntax present for ASL users differ slightly in comparison to the syntax of the spoken and written English. It is therefore the task of another Natural Language Processing model (NLP) to translate the different words into sentences. We used GPT-3, which is a generative language model, developed by OpenAI, that can infer different sentences out of given words. [[16]](#footnote-17)

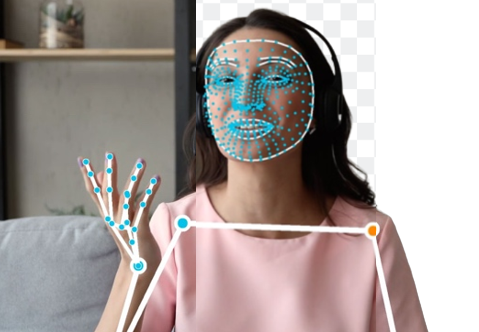
There are multiple possibilities to improve the efficiency of our models. First, using MediaPipe, a computer vision tool created by Google, we can manipulate our video data to remove the backgrounds and only keep the users making the gesture. This would significantly reduce irrelevant noise and distraction in the data. Second, MediaPipe is also capable of tracking a person’s pose in a video and returning an efficient matrix containing important features (Image 5).[[17]](#footnote-18) We are also investigating more efficient compression algorithms, such as grayscale transformation, jpeg compression algorithms, and noise reduction for our images and videos, to further reduce the size of our dataset, all in our goal of training 1000 words! Third, by only using images, our model loses the temporality and continuity in the frames present in videos. We are therefore investigating architectures that can process and train videos directly. Finally, bring our product to the general audience, we are also working on a mobile application that would directly enable a user to use a phone camera to track surrounding people’s speech. It would therefore also be possible to include a speech recognition transcript, so that the inverse conversation may be possible (to enable English speakers to talk directly to the camera and enable the deaf person to engage in a conversation).

Image 5: MediaPipe can remove the background and track essential features in a *person* (instead of the whole image, we can keep track of only the hand through their coordinates), thereby significantly reducing the size of our dataset. (We can train with MORE data!)

Our objective was to automate the process of translating American Sign Language into English using machine learning, with our current model being able to accurately classify 4 words. We sincerely believe that our project will improve inclusivity of the deaf community. Such a camera system could be placed in supermarkets, service centers, and schools, helping deaf people forge their independence!

Bibliography:

Akash. ASL Alphabet, 2018.

American sign language [Internet]. National Institute of Deafness and Other Communication Disorders. U.S. Department of Health and Human Services; [cited 2023Feb26]. Available from: https://www.nidcd.nih.gov/health/american-sign-language

Dima. ASL Benchmark Dataset (YOLOv5 PyTorch Format), 2021.

Goodfellow I, Bengio Y, Courville A. Deep learning. The MIT Press; 2016.

He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016;

Hosna A, Merry E, Gyalmo J, Alom Z, Aung Z, Azim MA. Transfer learning: A friendly introduction. Journal of Big Data. 2022;9(1).

Kevin C. Feature maps [Internet]. Medium. Medium; 2018 [cited 2023Feb26]. Available from: https://medium.com/@chriskevin\_80184/feature-maps-ee8e11a71f9e

Keshari N, Srikantham AK, Gomez Martinez G. Real-time ASL to English text translation [Internet]. Stanford.edu; 2021 [cited 2023 Jan 26]. Available from: http://cs230.stanford.edu/projects\_winter\_2021/reports/70749752.pdf.

Louridas P. Algorithms. Cambridge, MA: The MIT Press; 2020.

MS-ASL [Internet]. Microsoft Research. 2019 [cited 2023Feb26]. Available from: https://www.microsoft.com/en-us/research/project/ms-asl/

Object detection: Models, Architectures & Tutorial [2023] [Internet]. Object Detection: Models, Architectures & Tutorial [2023]. [cited 2023Feb26]. Available from: https://www.v7labs.com/blog/object-detection-guide

OpenAI API. [cited 2023Feb26]. Available from: https://platform.openai.com/docs/introduction

Pere C. What are loss functions? [Internet]. Medium. Towards Data Science; 2020 [cited 2023Feb26]. Available from: https://towardsdatascience.com/what-is-loss-function-1e2605aeb904

Reza H, Joze V, Koller O. MS-ASL: A large-scale data set and benchmark for understanding American sign language [Internet]. Arxiv.org; 2018 [cited 2023 Jan 26]. Available from: https://arxiv.org/pdf/1812.01053v2.pdf.

Sign Language Translator [Internet]. Fun Translations; [cited 2023 Jan 23]. Available from: https://funtranslations.com/sign-language.

Tammina S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications. 2019 Oct [cited 2023 January 30]; p.9420. Available from: http://dx.doi.org/10.29322/IJSRP.9.10.2019.p9420.

TensorFlow Hub [Internet]. Tfhub.dev; [cited 2023 Jan 13]. Available from: https://tfhub.dev/sayannath/american-sign-language/1.

1. Sign Language Translator [Internet]. Fun Translations; [cited 2023 Jan 23]. Available from: https://funtranslations.com/sign-language. [↑](#footnote-ref-2)
2. American sign language [Internet]. National Institute of Deafness and Other Communication Disorders. U.S. Department of Health and Human Services; [cited 2023Feb26]. Available from: https://www.nidcd.nih.gov/health/american-sign-language [↑](#footnote-ref-3)
3. What are neural networks? [Internet]. IBM. [cited 2023Feb26]. Available from: https://www.ibm.com/topics/neural-networks [↑](#footnote-ref-4)
4. Kevin C. Feature maps [Internet]. Medium. Medium; 2018 [cited 2023Feb26]. Available from: https://medium.com/@chriskevin\_80184/feature-maps-ee8e11a71f9e [↑](#footnote-ref-5)
5. Saha S. A comprehensive guide to Convolutional Neural Networks - the eli5 way [Internet]. Medium. Towards Data Science; 2022 [cited 2023Feb26]. Available from: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53?gi=4673d99837c0 [↑](#footnote-ref-6)
6. Pere C. What are loss functions? [Internet]. Medium. Towards Data Science; 2020 [cited 2023Feb26]. Available from: https://towardsdatascience.com/what-is-loss-function-1e2605aeb904 [↑](#footnote-ref-7)
7. Louridas P. Algorithms. Cambridge, MA: The MIT Press; 2020. [↑](#footnote-ref-8)
8. Goodfellow I, Bengio Y, Courville A. Deep learning. The MIT Press; 2016. [↑](#footnote-ref-9)
9. Object detection: Models, Architectures & Tutorial [2023] [Internet]. Object Detection: Models, Architectures & Tutorial [2023]. [cited 2023Feb26]. Available from: https://www.v7labs.com/blog/object-detection-guide [↑](#footnote-ref-10)
10. Hosna A, Merry E, Gyalmo J, Alom Z, Aung Z, Azim MA. Transfer learning: A friendly introduction. Journal of Big Data. 2022;9(1). [↑](#footnote-ref-11)
11. Tammina S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications. 2019 Oct [cited 2023 January 30]; p.9420. Available from: http://dx.doi.org/10.29322/IJSRP.9.10.2019.p9420. [↑](#footnote-ref-12)
12. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016; [↑](#footnote-ref-13)
13. https://www.image-net.org [↑](#footnote-ref-14)
14. MS-ASL [Internet]. Microsoft Research. 2019 [cited 2023Feb26]. Available from: https://www.microsoft.com/en-us/research/project/ms-asl/ [↑](#footnote-ref-15)
15. Real Python. Beautiful soup: Build a web scraper with python [Internet]. Real Python. Real Python; 2023 [cited 2023Feb26]. Available from: https://realpython.com/beautiful-soup-web-scraper-python/ [↑](#footnote-ref-16)
16. OpenAI API. [cited 2023Feb26]. Available from: https://platform.openai.com/docs/introduction [↑](#footnote-ref-17)
17. Home [Internet]. mediapipe. [cited 2023Feb26]. Available from: https://google.github.io/mediapipe/ [↑](#footnote-ref-18)