HandVoice
Vedant Sheel
Waterdown District High School
Waterdown, Ontario
FBLC Canadian National Leadership Conference
Coding & Programming
2024-25

Table of Contents

Social Issue

Effective communication is essential in our daily lives. However, for over 5% of the world's population, or the estimated 450 million human population who have hearing difficulties according to the World Health Organization (WHO), sign language is the primary means of communication. While ASL is widely used by the Deaf community

5% According to the World Health Organization, there are 466 million deaf people in the world (432 million adults and 34 million children)

90% Although it is believed that the majority of Deaf people learn Sign Language from their Deaf mothers and fathers, the fact is that 90% of Deaf people come from hearing families and learn Sign Language outside the family

worldwide, not everyone is proficient in it, resulting in a language barrier that can lead to exclusion, misunderstanding, and isolation. In many social, educational, and professional contexts, communication is taken for granted. When this fundamental aspect of interaction is disrupted, those who rely on sign language often face systemic disadvantages. In classrooms, for example, a lack of accessible communication can hinder a Deaf student's ability to participate fully, potentially impacting their academic performance and self-esteem. In the workplace, miscommunication may lead to missed opportunities for collaboration or career advancement, perpetuating cycles of economic and social marginalization. Even in healthcare settings, where precise communication is critical, Deaf individuals may struggle to convey symptoms or understand medical advice, resulting in suboptimal care. This pervasive language barrier is not just a technical or educational issue. It is a significant social justice concern. Communication is intrinsically tied to human rights.

The inability to communicate effectively can marginalize individuals from civic participation, limit their access to essential services, and erode their overall quality of life. When members of the Deaf community are excluded from critical conversations, whether in policy-making, community engagement, or daily interpersonal interactions, society as a whole loses out on diverse perspectives and talents.

Addressing this challenge is not solely about creating more inclusive spaces; it is also about recognizing and valuing the unique cultural identity of the Deaf community. Sign language is a vibrant, expressive medium that conveys nuances and emotions that spoken language sometimes cannot capture. However, the benefits of this rich communication mode are only fully realized when there is mutual understanding. The reality is that many hearing individuals lack the necessary skills to interpret sign language, which can inadvertently lead to social isolation for Deaf individuals. This isolation reinforces stereotypes and misunderstandings, creating a cycle that is difficult to break without innovative intervention.

This project, HandVoice, addresses this challenge by providing a tool that can translate ASL into text and subsequently to voice in real-time. This has the potential to transform

the way people communicate with Deaf individuals, making communication easier, efficient, and more effective.

Introduction

American Sign Language (ASL) is a natural and ancient form of human communication. Computer vision research has taken an interest in recognizing human gestures from camera images. A real-time translating Python program that can translate ASL characters into computer-generated speech in real-time was created, using Machine Learning (ML) frameworks and Convolutional Neural Networks (CNN).

A neural network is a computer program that learns from data, similar to how the human brain learns. It has layers of processing nodes that work together to predict/recognize patterns in new data. It is useful for image recognition. A CNN is a machine learning algorithm that is great for image classification, with multiple layers that learn to detect patterns/features in the input data. Starting with raw images, the CNN detects low-level features like edges, and gradually learns to recognize more complex shapes and textures. The output layer then produces a prediction based on patterns and features identified in the input data.
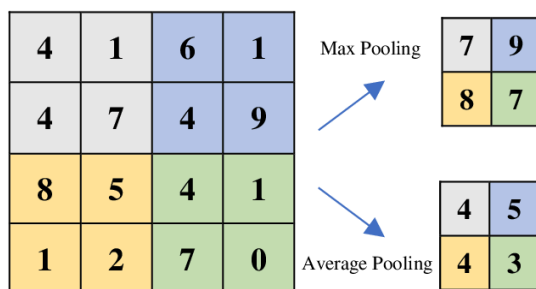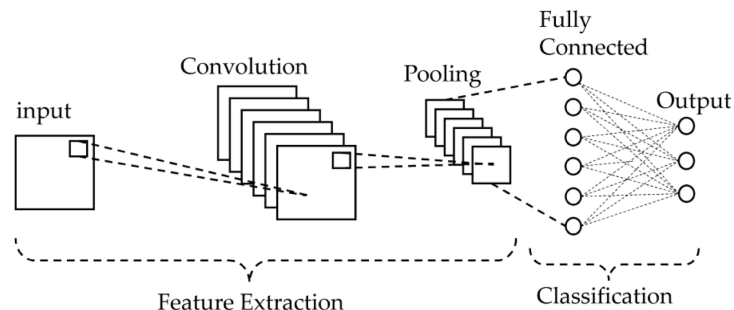
A pooling layer in a CNN down samples the output of the previous layer, using either max or average pooling. Max pooling selects the maximum value from each region. Average pooling takes the average from each region. This reduces the spatial dimensions of the feature maps, improving efficiency and performance by reducing the parameters.

In this project, three pooling layers are tested for which the CNN is trained on an image dataset of hand positions and orientations, enabling real-time recognition of finger spelling in ASL. Data image-inputs are then filtered and classified based on predicted hand gestures. The system further exports the recognized alphabet as a string with dictionary suggestions with a "speak" button for voice output.

Materials

Python 3.10, PyCharm 22.1, OpenCV 4.2, Mediapipe 3.1, Numpy 2.3, Pyttsx3 2.6, Math 1.3, Keras 2.2.5, TensorFlow 2.12, Pyenchant 3.2.2, Tkinter 8.6, Pillow 9.4, React,

HTML, CSS, JavaScript, React Native, Node.js, Git, Figma, Firebase, computer, keyboard, mouse, webcam, monitor, power supply, HDMI cord.

Methodology
Two approaches to predict a character with high accuracy were suggested:

A. To create a model using a dataset of each character with a very large repository. For instance creating a model using 2000-3000 images per character.The issue with this method is that the overall program will occupy large memory space which does not suit the future goal which is to inherent this in a mobile application.

B. To use a small sample size of dataset and classify each similar looking gesture into their own classes. Then to further extract the character within that class using a novel approach wherein a distance between various landmarks will be calculated and utilized to determine the character.

Since option B uses very small memory space in contrast to option A, this was the chosen method. Two big challenges:

I. How to recognize a hand in the eyes of a computer? Humans can easily spot a hand, but how can a computer without any self-intelligence?

II. How to make computer classify what gesture the user is performing?

To address this, a five-stage methodology approach was created for an ASL character to text real-time translator using OPTION B.
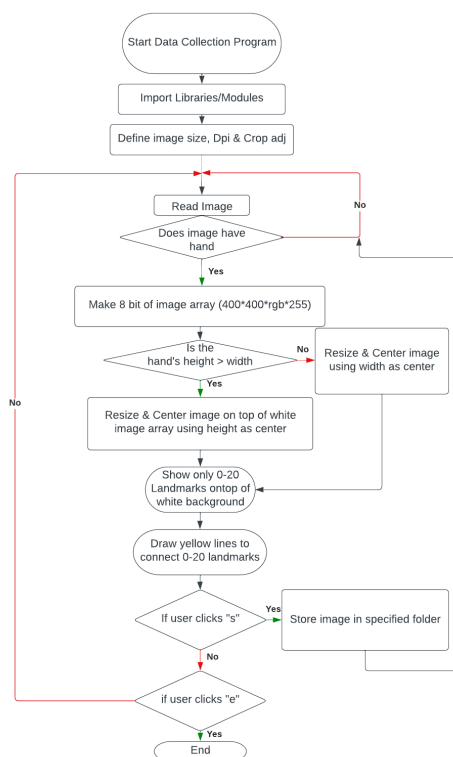
1. Data pre-processing & image segmentation
In this Python language project, a frame-by-frame image is captured from the webcam using the OpenCV open-source library. The hand in the image is then identified and tracked with 21 landmarks with the help of cvzone and mediapipe libraries. This addressed the first problem of detecting the hand. Once a hand is detected, the hand image is cropped and the size is calibrated so that each hand image is 400x400 pixels. To minimize the memory size and to keep only useful information needed for prediction, three separate pooling layers are created.

i. The Grayscale layer: This pooling layer converts the image to black and white, removing saturation.

ii. Gaussian blur over grayscale image. This filter will blur all the background from the image leaving only the hand's important outlines in black and white.

iii. The landmark segmentation. In this filter the camera input is not shown, only displaying the 21 landmarks on a white 400x400 image-array. A code is further written to connect each landmark, outlining the hand shape using a yellow coloured line, solving lighting and background confusion for the computer.
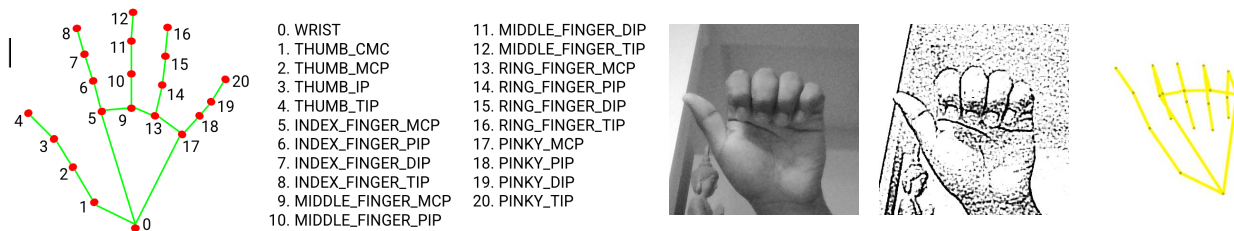
2. **Data collection:**
In this step a dataset of images of each ASL letter was created. One hundred and eighty sample images of each letter were collected and stored in a directory for each letter. This dataset was used for the CNN training model, providing various samples and possibilities of each ASL letter gesture.

3. **Data training and modeling**

After collecting a comprehensive dataset of ASL images, the next step was to train a robust deep learning model capable of accurately classifying each gesture. Rather than relying on pre-existing tools such as Google's Teachable Machine, I developed a custom neural network using TensorFlow and Keras in Python. This approach allowed for greater flexibility in model architecture design and fine-tuning, ensuring that the system could meet the specific challenges presented by the dataset. Initially, a single monolithic model was implemented to classify all twenty-six ASL characters directly. However, this approach proved inefficient, as the model struggled to manage the high variability in the input data. Each character was represented by approximately 180 images captured under diverse conditions, which resulted in a model that was overburdened and prone to reduced accuracy and generalization issues.

To address these challenges, the classification task was restructured by grouping the ASL alphabet into eight distinct categories based on the visual similarity of the hand gestures. This modular approach allowed the model to focus on smaller

subsets of characters, thereby simplifying the learning process and improving overall performance. The groups were defined as follows:

- Group 1: [A, E, M, N, S, T] – Characters that typically involve closed hand positions, similar to a fist.
- Group 2: [B, D, F, K, L, R, U, V, W] – Letters that are characterized by flat, open hand positions.
- Group 3: [C, O] – Gestures in which the hand forms a curved or circular shape.
- Group 4: [G, H] – Letters represented by a bent hand position, often resembling a gun shape.
- Group 5: [I] – The letter "I," which is distinct enough to be classified independently.
- Group 6: [J, Y] – Gestures where the hand is mostly closed with the pinky extended.
- Group 7: [P, Q, Z] – Characters involving a bent hand with two open fingers.
- Group 8: [X] – The letter "X," which forms its own unique category.

Each group was modeled separately, allowing the network to specialize in distinguishing subtle differences among similar gestures. The custom neural network was trained using backpropagation with carefully selected hyperparameters to minimize classification errors. Through iterative training and validation, the network achieved a significant improvement in accuracy, ultimately providing a robust and reliable ASL translation system. This group-based training strategy not only enhanced the model's performance but also improved its efficiency, making it well-suited for real-time ASL translation in practical applications.

4. Prediction

This stage is divided into a two-step process.

A. Predicting the class
There are 8 classes which store the 26 characters. In order to predict the character one must first determine the class holding it. By calling the keras model, the train images are called upon and compared with the convoluted input which is stored as frames. Using the CNN created, 186 predicted readings for each letter were collected on a spreadsheet with two charts; Chart1 (ch1) and Chart 2 (ch2). Chart 1 held the most predicted class's index which would range from zero to seven while ch2 held the second most predicted class. Several 2 dimensional arrays were written for each of eight classes based on the most common occurrences

of ch1 and ch2 combinations for each letter in their class. The CNN and ch1,ch2 combination gives out a 98.87% accurate prediction to determine the correct class.

B. Predicting the character

After the class is determined, all that's left is to distinguish between the characters within the class itself, and present the highest confidence value as output. To further distinguish between characters, one may suggest using a CNN once again as it had such high accuracy for the 8 classes. However, when CNN is attempted, the computer provides incorrect and inaccurate outputs as it is very complicated to distinguish between a few letters in common. To solve this problem a different approach was taken. Rather then using a CNN to search within the class for the most accurate character, distance calculations were implemented.

To solve for distance from a point to point, Euclidean distance formula was used for which two landmark locations were calculated. To sort within [A,E,M,N,S,T] class,

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
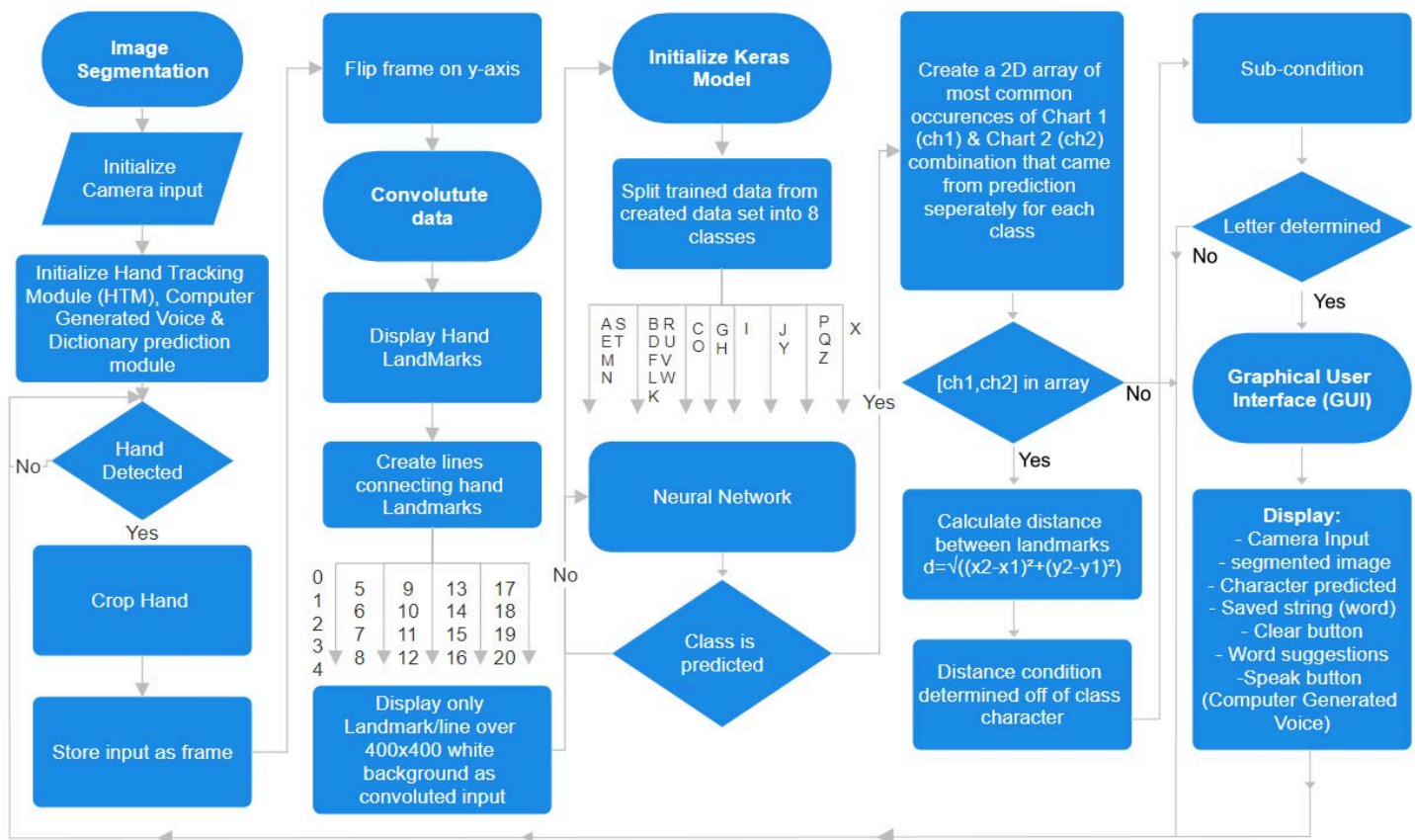
5. Frontend & Additional features

In the early phases, the GUI was built using Tkinter, Python's standard library, which enabled rapid prototyping and testing of the core ASL translation functionality. This initial interface integrated directly with the backend, allowing users to view a live video feed, processed images with highlighted landmarks, and real-time text translation, along with basic controls such as "Clear" and

"Speak." However, Tkinter's limited design flexibility and scalability became apparent as the project progressed.

To overcome these limitations, the interface was transitioned to a web-based application developed with React, ensuring an intuitive, responsive design accessible from any device with a web browser. This new platform supports dynamic video processing, instantaneous translation updates, high-quality text-to-speech, and enhanced user controls. Additionally, a mobile application was developed using React Native to provide cross-platform compatibility for Android and iOS devices. The mobile solution leverages native camera APIs and cloud connectivity to maintain high accuracy and responsiveness, even on devices with limited processing power.

This evolution from a basic Tkinter interface to a sophisticated, cross-platform solution highlights our commitment to continuous improvement and innovation, ensuring that HandVoice remains a high-impact, accessible communication tool for the Deaf community.

6. Testing and Evaluation

User testing was conducted across various environments—including classrooms, healthcare settings, and community centers—to evaluate the real-time performance of HandVoice. Initial tests revealed prediction accuracies between 62-73% using plain hand images, which improved steadily with the application of a gray filter (raising accuracy to 67-72%) and further with a Gaussian blur (increasing accuracy to 77%). Finally, by implementing landmark segmentation and grouping strategies, accuracy reached an impressive 98%. These iterative evaluations confirmed that the system is reliable for real-world application, providing consistent performance across diverse lighting and background conditions.

Observations

During the data-preprocessing and feature extraction phase, the plain hand images were first trained. The prediction/confidence values obtained were in between 62-73% and varied depending on the background and lighting, resulting in very low accuracy. To eliminate the background and improve accuracy, a gray filter was applied, which raised the prediction values to a constant 67-72% which may have been a low accuracy, but more consistent. After applying the gaussian blur filter above the grayscale pooling layer, the prediction value average increased to 77%. After further segmenting the image and creating a landmark layer and distinguishing for the character within the eight classes by conditional statements, the accuracy increased to 98%.

Challenges and Limitations

Despite significant progress, several challenges remain. Data variability, stemming from inconsistent lighting, backgrounds, and hand orientations continues to impact image quality and requires further refinement of pre-processing techniques. Initially, a single monolithic model struggled with high variability and limited data, prompting a switch to a group-based classification strategy, which improved accuracy but added complexity. Real-time processing also poses challenges, particularly on mobile devices, where balancing low latency with high accuracy requires optimization and reliance on cloud services. Additionally, transitioning from the basic Tkinter prototype to a more dynamic web and mobile interface demanded extensive redevelopment to ensure a seamless user experience across platforms. These challenges are guiding ongoing improvements

and will be the focus of future enhancements to ensure HandVoice remains a reliable and accessible communication tool for the Deaf community.

Monetization Strategy

HandVoice is a cutting-edge ASL translation application that transforms American Sign Language into real-time text and voice outputs. As a socially-driven technology aimed at promoting inclusivity, HandVoice must remain accessible while also achieving financial sustainability. Our monetization strategy has been carefully designed to create multiple revenue streams without compromising our commitment to the Deaf community. This report details our revenue generation models, strategic partnerships, and future development plans.

1. **Freemium and Subscription Models**
   At the core of HandVoice is a freemium model, where the essential functionality—real-time translation of ASL into text and voice—is provided free of charge. This approach ensures widespread accessibility and a broad user base, including individuals, educational institutions, and community organizations. To drive revenue, premium subscription tiers are offered, unlocking enhanced features such as higher translation accuracy, offline functionality, expanded vocabulary support, and customizable interfaces.
   - Individual subscriptions are projected at approximately $9.99 per month.
   - Enterprise or institutional tiers are anticipated to be priced around $29.99 per month.
     Based on conservative estimates, achieving a user base of 100,000 within the first year with a conversion rate of 10% to 30% could result in monthly subscription revenues ranging from $100,000 to $300,000.

2. **In-App Purchases**
   HandVoice also integrates an in-app purchase system, allowing users to buy one-time enhancements that enrich their experience. These enhancements include specialized modules such as advanced sign language tutorials, premium dictionary features, and custom themes. Priced between $0.99 and $4.99, these purchases provide an additional, scalable revenue stream. A conservative estimate of 5,000 in-app purchases per month could generate approximately $20,000 in monthly revenue.

3. **Advertising and Sponsorships**
A targeted advertising strategy is a key component of our monetization plan. HandVoice will feature unobtrusive, contextually relevant advertisements designed to promote products and services aligned with our mission of accessibility. Based on an estimated 500,000 monthly active users and a conservative cost per mille (CPM) of $3, advertising could generate roughly $4,500 per month. Additionally, sponsorship deals with organizations in the accessibility, education, and healthcare sectors will provide direct financial support and enhance market credibility.

4. **Enterprise Licensing and White-Label Solutions**
Enterprise licensing represents one of the most promising revenue streams. HandVoice's adaptable ASL translation engine can be integrated into existing systems across various sectors. For instance, educational institutions can implement HandVoice to support Deaf students, while healthcare providers can use it to improve patient communication. Large corporations, particularly in customer service and human resources, can license a white-label version to meet regulatory requirements and boost internal communication. Securing licensing contracts with 50 organizations at an average annual fee of $10,000 could yield approximately $500,000 in annual revenue.

5. **Data Analytics and Research Collaborations**
With strict adherence to privacy regulations and with proper user consent, anonymized data from HandVoice can be a valuable resource for research and market analysis. By analyzing usage patterns and communication trends, we can offer insights to academic institutions, technology firms, and policymakers. Collaborative research initiatives and data-sharing agreements are expected to generate an additional $100,000 to $200,000 annually, supplementing our revenue while driving innovation.

Social Impact and Future Effectiveness

HandVoice is designed to address the longstanding communication barriers faced by the Deaf community. By converting ASL gestures into real-time text and voice outputs, the application empowers Deaf individuals to interact more effectively in educational, healthcare, and professional settings.

In the future, HandVoice will enable Deaf students to follow lectures more clearly and participate more confidently in group discussions. In healthcare environments, it will support better patient-provider interactions, ensuring that Deaf patients can convey their concerns accurately and understand critical medical information. In the workplace, the tool will help reduce misunderstandings and foster smoother collaboration, thereby promoting a more inclusive environment. Furthermore, HandVoice will be developed with an intuitive, user-friendly interface that will allow users with varying levels of technical expertise to take full advantage of its features. As the application is deployed in real-world settings, it is expected to significantly reduce social isolation and enhance overall community engagement by connecting users with each other and with institutions that prioritize accessibility.

While formal impact studies will be conducted in the future to measure its effectiveness quantitatively, early qualitative feedback is anticipated to be positive. HandVoice is expected to serve as a transformative tool that not only bridges communication gaps but also drives social empowerment, thereby making a lasting contribution to inclusivity and improved quality of life for the Deaf community.

Conclusion

In conclusion, HandVoice, a real-time ASL recognition system was successfully developed. Out of all three datasets, the binary hand-landmarks with white background were most successful with an accuracy of 97%. The use of CNN and pooling layers allowed for accurate hand gesture recognition. The project also collected a dataset of ASL character images for model training and implemented eight classes to improve prediction accuracy. The monetization strategy for HandVoice is comprehensive and forward-thinking, combining subscription models, in-app purchases, targeted advertising, enterprise licensing, and data analytics to create a diversified revenue base. With conservative revenue projections indicating strong potential, and a well-defined roadmap for future development, HandVoice is well-positioned for long-term success. Our balanced approach ensures that financial sustainability is achieved without compromising our social mission, ultimately delivering a transformative communication tool that empowers the Deaf community and drives inclusive growth.

Future Enhancements

In the coming phases, HandVoice will continue to evolve with a series of planned enhancements designed to further improve its functionality and broaden its impact on bridging communication gaps. Future updates will include support for additional sign languages, allowing the application to serve a more diverse, global user base. We will integrate more advanced machine learning algorithms that will enhance gesture recognition accuracy and reduce latency, ensuring that the system remains at the

forefront of real-time translation technology. HandVoice will also integrate with wearable devices, enabling seamless, on-the-go translation for users in dynamic environments. Enhanced offline capabilities will be developed, ensuring that the application performs reliably even in areas with limited internet connectivity. The user interface will be further refined and will offer greater customization options, making the tool even more intuitive and user-friendly. Additionally, robust analytics features will be incorporated, providing users with personalized feedback and usage insights. These analytics will help drive continuous improvements and allow for more targeted updates. As emerging technologies and user needs evolve, HandVoice will adapt by incorporating state-of-the-art advancements, ensuring that it remains a transformative tool for fostering inclusivity and effective communication within the Deaf community and beyond.

References & Bibliography

Shaw, Zed. *Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code*. Addison-Wesley, 2017.

Renotte, Nickolas, director. *Real Time Sign Language Detection with Tensorflow Object Detection and Python | Deep Learning SSD. YouTube*, YouTube, 5 Nov. 2020, www.youtube.com/watch?v=pDXdlXlaCco.
Renotte, Nicholas. *"Sign Language Detection using ACTION RECOGNITION with Python | LSTM Deep Learning Model."* Youtube, 19 Jun. 2021 https://www.youtube.com/watch?v=doDUihpj6ro.

Robotics and AI. , Murtaza's Workshop, director. *"Easy Hand Sign Detection | American Sign Language ASL | Computer Vision."* . *YouTube*, YouTube, 4 July 2022, www.youtube.com/watch?v=wa2ARoUUdU8.

YouTube. (2021). *Ai For Everyone Lesson 1: Introduction to Artificial Intelligence for Absolute Beginners. YouTube*. Retrieved April 26, 2023, from https://www.youtube.com/watch?v=gD_HWj_hvbo.

McWhorter, Paul, director. *AI for Everyone LESSON 2: Running Two Different Versions of Python on the Same Computer. YouTube*, YouTube, 27 July 2021, www.youtube.com/watch?v=I7GKSQeFGQs.

McWhorter, Paul, director. *AI for Everyone LESSON 3: Understanding and Using Python Virtual Environments. YouTube*, YouTube, 29 July 2021, www.youtube.com/watch?v=XCvsLMk4OHM.

McWhorter, Paul, director. *AI for Everyone LESSON 5: Installing OpenCV in Windows and Launching WEBCAM. YouTube*, YouTube, 3 Aug. 2021, www.youtube.com/watch?v=fGRe4bHRoVo.

McWhorter, Paul, director. *Managing Multiple Windows Size and Position in OpenCV. YouTube*, YouTube, 12 Aug. 2021, www.youtube.com/watch?v=VRRTglL9fN0.

McWhorter, Paul, director. *AI for Everyone LESSON 7: Understanding Pictures and Images as Data Arrays. YouTube*, YouTube, 17 Aug. 2021, www.youtube.com/watch?v=W43MpRroplA.

McWhorter, Paul, director. *AI for Everyone LESSON 9: Understanding Region of Interest (ROI) in OpenCV. YouTube*, YouTube, 31 Aug. 2021, www.youtube.com/watch?v=E46B7NPWK38.

McWhorter, Paul, director. *AI for Everyone LESSON 16: Face Recognition with OpenCV*. *YouTube*, YouTube, 19 Oct. 2021, www.youtube.com/watch?v=BHYZ1xkTGi8.

McWhorter, Paul, director. *AI for Everyone LESSON 18: Mediapipe for Hand Detection and Pose Estimation*. *YouTube*, YouTube, 2 Nov. 2021, www.youtube.com/watch?v=xHK-wv2JG18.

Gannon, Madeline. "Madelinegannon/Example-Mediapipe-UDP: Connecting Openframeworks to Google MediaPipe Machine Learning Framework over UDP." *GitHub*, www.github.com/madelinegannon/example-mediapipe-udp.

Grov, Ivan. "Hand-Gesture-Recognition-Mediapipe." *GitHub*, 28 Jan. 2021, www.github.com/kinivi/hand-gesture-recognition-mediapipe.

"Deaf Canadians 'at Risk' in Times of National Emergency | CBC News." *CBCnews*, CBC/Radio Canada, 27 Sept. 2018, www.cbc.ca/news/health/deaf-canadians-at-risk-in-times-of-national-emergency-1.4832321#:~:text=The%20Canadian%20Hearing%20Society%20estimates%20there%20are%203.15,deaf%2C%20including%20an%20estimated%2011%2C000%20who%20are%20deaf-blind.

Raval, Devansh. "Devansh-47/Sign-Language-to-Text-and-Speech-Conversion: This Is a Python Application Which Converts American Sign Language into Text and Speech Which Helps Dumb/Deaf People to Start Conversation with Normal People Who Dont Understand This Language." *GitHub*, 11 Nov. 2022, www.github.com/Devansh-47/Sign-Language-To-Text-and-Speech-Conversion

Nine, Neural, director. *Tkinter Beginner Course - Python GUI Development*. *YouTube*, YouTube, 29 Sept. 2021, www.youtube.com/watch?v=ibf5cx221hk.

Galli, Keith, director. *How to Program a GUI Application (with Python Tkinter)! YouTube*, YouTube, 1 Feb. 2019, www.youtube.com/watch?v=D8-snVfekto.

CodeHub, Sam, director. *How to Create Advance Text to Speech Convertor App Using Python Tkinter Framework & PYTTSX3 Library. YouTube*, YouTube, 22 Sept. 2022, www.youtube.com/watch?v=0ldMgo8r9Ck.

Dedic, Sanjin, director. *Python Enchant English Dictionary + Caesar Cipher Decryption*. *YouTube*, YouTube, 6 July 2017, www.youtube.com/watch?v=vdhCzy4Ibps.

Saha, Sumit. "A Comprehensive Guide to Convolutional Neural Networks-the eli5 Way." *Medium*, Towards Data Science, 16 Nov. 2022,

www.towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

Most Popular JavaScript Front-End Frameworks in 2024. (2024). Pkgindex.com.
https://pkgindex.com/blog/best-javascript-front-end-frameworks
GeeksforGeeks. (2023, December 27). *React Tutorial*. GeeksforGeeks.
https://www.geeksforgeeks.org/react/

Guissous, Alla Eddine. "9: Example for the Max-Pooling and the Average-Pooling with a Filter ..." *ResearchGate*, Nov. 2019,
www.researchgate.net/figure/Example-for-the-max-pooling-and-the-average-pooling-with-a-filter-size-of-22-and-a_fig15_337336341.

"American Sign Language (ASL) Alphabet (ABC) Poster." *Gerarda Flague Collection*,
www.gerardaflaguecollection.com/products/american-sign-language-asl-alphabet-abc-poster.html

Google. (n.d.). MediaPipe Hand Landmark Model. Retrieved April 26, 2023, from
https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

Ferreiro, E. (2019, September 25). *Infographic: Sign Language Rights for All!*
Unusualverse.
https://www.unusualverse.com/2019/09/infographic-sign-language-rights-for-all.html