Signz

Sign Language Translating Website

Senior Project

by

Hiba Alawieh & Sara Raii

Submitted to school of Arts & Science of the

Lebanese International University

In a partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervised By: Dr. Youssef Roumieh

Spring 2023-2024

Abstract

Deaf people can communicate with each other using their language, with gesture (sign language) being a particularly common one. However the communication with abled people is kind of rare due to the lack of laxatives, so here comes SIGNZ. The translation process methodically replaces the original text in the live camera feed, aligning the background and foreground colors derived from the source images. Numerous tests have been conducted to identify the optimal configurations for development. The web-based sign language translator allows users to display a few numbers (from 0 to 10) in American Sign Language to a webcam, which then detects and translates these signs into text. This tool helps bridge the communication gap between individuals with hearing impairments and those without. Its main goal is to enhance communication with hearing-impaired individuals, addressing a long-standing challenge. The translator mainly employs object detection principles, recognizing the hand in the camera view and converting the numbers in sign language into numbers in text.

Acknowledgment

The fulfillment of successfully completing this project would be incomplete without acknowledging the individuals whose invaluable support and encouragement were instrumental throughout our journey. We also recognize that none of this would have been possible without the will and blessing of God.

We extend our deepest gratitude to all those who guided and inspired us in bringing this project to fruition.

We are profoundly grateful to our project advisor, Dr. Youssef Roumieh, for his unwavering support, insightful guidance, and wealth of knowledge. His mentorship was crucial to the successful completion of our project.

Our heartfelt thanks also go to all the doctors who have taught and supported us throughout our academic journey. Their dedication and expertise have been foundational to our achievements, and we deeply appreciate their contributions.

Lastly, we owe an immense debt of gratitude to our parents for their constant inspiration and encouragement, which empowered us to pursue our studies and achieve our goals. Their support has been our driving force.

Table of Contents:

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

Sign Language has been the most ignored language since too long. As it’s the only way for communication between individuals who are speech and hearing impaired with people who are not. The most often used sign language is the Pidgin Signed English. In Lebanon, Lebanese deaf people express themselves with the Lebanese dialect of Levantine Arabic Sign Language. Sign language in Lebanon started with deaf people and their families communicating using gestures. Schools like the Father Andeweg Institute for the Deaf, founded in 1957, and the Lebanese School for the Deaf, started in 1959, helped in making the sign language more organized. Lebanese Sign Language (LSL) was influenced by other sign languages because Lebanon has many different cultures. Groups like the Lebanese Association of the Deaf work to help deaf people, but sign language still doesn't have full recognition by the law. Nowadays, technology and the internet are making sign language more available to everyone. But there are still problems, like needing more help from the government and better education for deaf people. In Lebanon, there are around 35,000 deaf people, and unfortunately, many of them don't know sign language.

Technology has really helped deaf people in many ways. For example, special phones and apps let them talk to others through text or video calls.

Watching TV and movies is easier too because of captions and subtitles. There are also devices like personal amplifiers that make sounds clearer and reduce background noise. Some deaf people can even hear better with cochlear implants. Speech-to-text software changes spoken words into written text, helping deaf people understand conversations and meetings. With educational tools and online courses, learning is more accessible.

1.2 Background of the project

The background of the Sign Language Translating Website project encompasses several key areas, including the research conducted and the rationale for its development. The primary goal of this project is to create a detailed explanation of the technology used. Named Signz, this project aims to capture images of numbers in sign language gestures and convert them into the corresponding numbers. The research focuses on developing the sign language translator application using TensorFlow and Python. Studies by the Informatics Department at Politeknik Negeri Semarang indicate that fingertip coordinate methods can effectively recognize hand gestures with open arms, while Hu Moments value methods are suitable for recognizing gestures with clenched hands (Triyono et al., 2017).

The translator utilizes a webcam, typically pre-installed in laptops or computers, to capture hand movements through image detection technology. Image detection occurs when the camera lens captures and identifies an object in its field of view. This technology allows the system to recognize specific hand gestures and translate them into text. The Sign Language Translating Website uses these principles to facilitate communication by interpreting sign language gestures into relative numbers, enhancing accessibility for the hearing impaired, with another extra characteristic which is the normal translator for all people.

1.3 Problem Statement

The difficulty some people experience when trying to communicate with hearing-impaired individuals. The core problem identified is the communication barrier between those who are hearing and those who are hearing impaired. This project seeks to eliminate this barrier and improve understanding between the two groups. It involves researching how effectively the sign language translator works in contemporary society. There is a pressing need for a tool that enhances the ability to communicate and understand sign language more effectively than ever before. The goal of the sign language translator is to facilitate meaningful communication between speech-impaired and non-speech-impaired individuals. Ensuring the project's efficiency and effectiveness is crucial.

1.4 Project Objective

The aim of this project is to assist individuals with speech and hearing impairments in communicating with others. The objectives include:

- Developing a machine learning system that recognizes hand signs using image detection technology.
- Integrating TensorFlow API to implement and enhance image detection capabilities.

- Utilizing machine learning algorithms to improve the accuracy and efficiency of hand sign recognition.

1.5 Project Scope

This project involves several key principles, starting with real-time image detection and recognition using TensorFlow, programmed in Python. The system detects hand signals, representing sign language gestures, and converts them to text displayed on the screen. This image recognition is essential for the project's success.

The hardware required is a laptop or desktop computer with a camera. The camera captures hand movements, enabling the image detection necessary for translating sign language.

The process involves detecting and recognizing hand patterns, then displaying the corresponding numbers on the screen. The system focuses on essential data while ignoring irrelevant information, using edge-finding methods to simplify image recognition.

TensorFlow processes these images by submitting them to a neural network, which categorizes and labels them based on predefined classes. This real-time translation of hand movements into text facilitates effective communication for speech and hearing-impaired individuals.

Chapter 2

Literature Review

2.1 Artificial Intelligence

Artificial Intelligence (AI) is a pivotal technology impacting daily life and business operations. It significantly contributes to Japan's long-term economic development and addresses various societal issues. Recently, AI has garnered substantial attention as a catalyst for growth in both industrialized nations like the United States and Europe, as well as emerging economies such as China and India. The focus has largely been on the development of modern AI information communication technology (ICT) and robot technologies (RT). AI's computational prowess, enhanced by the advent of Big Data and technological advancements, has entrenched it firmly in contemporary business practices and public discourse. AI can be categorized based on its exhibited intelligence—analytical, human-inspired, and humanized AI, as well as Artificial Narrow, General, and Super Intelligence—encompassing cognitive, emotional, and social intelligence (Haenlein and Kaplan, 2019).

AI also involves the simulation-based evaluation of intellectual abilities. It plays a crucial role in understanding and performing intellectual tasks such as reasoning, learning new skills, and managing various situations and challenges. Numerous AI methodologies have been developed, including Neural Networks, Fuzzy Logic, Evolutionary Computation, and Hybrid Artificial Intelligence (Haenlein and Kaplan, 2019).

2.1.1 Neural Networks of AI

Neural Networks, also known as artificial neural networks, represent a subset of artificial intelligence and form the foundation of deep learning algorithms. Inspired by the human brain, the concept of neural networks mimics the way biological neurons communicate with each other. These networks consist of layers of nodes, comprising an input layer, one or more hidden layers, and an output layer.

2.2 Deep Learning

Deep learning allows computational models with multiple processing layers to learn and represent data at various levels of abstraction, mimicking the brain's ability to process multimodal information and capture complex data structures. This includes neural networks, hierarchical probabilistic models, and various supervised and unsupervised feature learning algorithms. Researchers have been particularly interested in deep learning recently due to its ability to outperform previous state-of-the-art methods in numerous tasks and handle complex data from various sources, such as visual, auditory, medical, social, and sensor data (Voulodimos, Doulamis, Doulami, and Protopapadaki, 2018).

The early development of neural networks was driven by the desire to create technology that could replicate the human brain. McCulloch and Pitts aimed to describe how the brain could form complex patterns using interconnected basic cells called neurons. In 2006, Hinton and colleagues achieved a significant breakthrough in deep learning with the Deep Belief Network, composed of several layers of Restricted Boltzmann Machines trained one layer at a time in an unsupervised manner. This approach of guiding intermediate representations through unsupervised learning at each layer contributed to the recent surge in deep learning research (Voulodimos et al., 2018).

Deep learning has made substantial progress in various computer vision tasks, including object recognition, motion tracking, behavior recognition, human pose estimation, and semantic segmentation.

2.3 Object Detection

Object detection involves identifying instances of specific visual object classes, such as people, animals, or vehicles, in digital images. According to Zhao, Zheng, Xu, and Wu (2019), the goal of object detection is to develop computational models and methods that provide essential information for computers. Object detection, a fundamental problem in computer vision, underpins many other tasks, including instance segmentation, image captioning, and object tracking. Research in object detection can be divided into "general object detection," which explores methods for detecting various object types within a unified framework to simulate human vision and cognition, and "detection applications," which focus on specific scenarios like surveillance (Zhao et al., 2019).

The rapid advancements in deep learning techniques have revitalized object detection, leading to significant breakthroughs and drawing unprecedented attention. Object detection is now widely used in real-world applications, including autonomous driving, robot vision, and video surveillance (Zhao et al., 2019).

2.3.1 Object Detection with Deep Learning

Deep learning has enabled the development of more adaptive tools that can learn semantic, high-level, and deep features, overcoming many limitations of traditional methods. These models differ significantly in network architecture, training methods, and optimization functions. To fully understand a variety of images, it is essential not only to identify objects but also to accurately estimate their positions and relationships within each image. This task, termed Object Detection, has been thoroughly studied by Liu, Ouyang, Wang, Fieguth, Chen, Liu, and Pietikäinen (2020).

Object detection is one of the most fundamental and challenging problems in computer vision, aiming to recognize object instances from predefined categories in natural images. Deep learning has emerged as a powerful tool for learning feature representations directly from data, resulting in significant advancements in general object detection (Liu et al., 2020).

Deep learning has transformed numerous machine learning applications, including image recognition, video analysis, and natural language processing. Convolutional Neural Networks (CNNs) leverage real signal characteristics such as translation invariance, local connectivity, and hierarchical composition. Effective training of CNNs requires extensive data, substantial computational resources, and the ability to select appropriate learning parameters and network architectures. The basic layers of a CNN consist of a series of feature maps, each acting as a neuron, interconnected in a manner that enables effective feature extraction and learning.

2.4 Datasets

Datasets are crucial for advancing object recognition research, serving as benchmarks to evaluate algorithm performance and pushing the field into increasingly complex scenarios. The success of deep learning in visual recognition owes much to the availability of large, annotated datasets, which capture the immense variation found in internet images. Prominent datasets include PASCAL VOC, ImageNet, MS COCO, and Open Images. Creating these datasets involves three steps: selecting target categories, gathering candidate images, and annotating them, often via crowdsourcing methods (Liu et al., 2020).

The Open Image Challenge Object Detection dataset, based on Open Images V5, is the world's largest public dataset for object detection. It distinguishes itself from predecessors like ILSVRC and MS COCO by significantly expanding the number of classes, images, and bounding box annotations, and employing a unique annotation process. Unlike earlier datasets that fully annotated images, Open Images V4 only submitted labels with high scores for human verification, ensuring that OICOD includes only verified instances.

2.5 Detection Frameworks

There has been substantial progress in developing object attribute representations and classifiers, shifting away from handcrafted features. Despite this, the basic "sliding window" approach remains prevalent for object localization, though it's been modified to reduce exhaustive searches. Due to the sheer number of pixels and the need for scanning, the number of potential windows is enormous, growing quadratically. Different sizes and aspect ratios further complicate the process, highlighting the need for efficient and robust detection systems to balance computational cost and performance (Liu et al., 2020).

2.6 Sign Language

Contrary to popular belief, sign language interpretation has historical roots dating back to the early 1800s. Laurent Clerc's address to the U.S. government in sign language in 1818 is an early example. Significant progress was made in the 1950s when William C. Stokoe, an English professor at Gallaudet College, began researching American Sign Language (ASL). Stokoe's pioneering work led to the recognition of ASL as a legitimate language with its own phonology, morphology, syntax, and semantics, culminating in the first ASL dictionary in 1965 (Ball, 2017).

Before federal laws mandated interpreting services, the demand for interpreters far exceeded supply, with children of deaf adults often filling the gap informally. The federal mandate increased the need for trained interpreters, leading to professionalization efforts despite initial funding challenges. The establishment of training programs helped address the shortage and improve service quality (Ball, 2017).

2.6.1 American Sign Language

American Sign Language (ASL) is a fully developed natural language with unique syntax distinct from English. It uses hand and facial gestures for communication and is primarily used by the deaf and hard of hearing in North America. ASL includes fingerspelling for representing English words, where each handshape corresponds to a letter. Regional variations exist in ASL, similar to accents in spoken languages, and its usage is influenced by factors such as age and gender.

2.6.2 Sign Language Translators

Sign language utilizes various body parts to convey information, but it's not widely understood by the hearing population, creating communication barriers. Sign language recognition (SLR) research, influenced by gesture recognition, aims to bridge this gap by capturing and translating gestures into meaningful language. Early techniques like Dynamic Time Warping

(DTW) from speech recognition were adapted for gesture recognition (Bantupalli & Xie, 2018).

SLR systems can be signer-dependent, where the same individuals are used for training and testing, or signer-independent, which is more challenging as it involves different individuals. These systems can be sensor-based or image-based, with visual-based methods using cameras to capture gestures. Despite their affordability, visual-based methods face challenges like lighting variations and occlusion, which can be mitigated by using depth sensors to create 3D representations (Ibrahim et al., 2018).

2.6.3 Sign Language Translator with Image/Object Detection

Hearing disability, ranging from mild to severe, affects over 360 million people globally, according to the WHO. Sign Language Recognition Systems (SLRS) can help by translating sign language into written or spoken language. SLRS can be discrete, recognizing one sign at a time, or continuous, recognizing complete phrases. They can also be signer-dependent or signer-independent, with visual-based methods being more natural and affordable but challenged by issues like hand occlusion (Ibrahim et al., 2018).

Advances in hand segmentation, crucial for SLRS, involve techniques like skin detection and context removal to isolate hands in video frames. These methods have evolved to reduce false positives and improve real-time application performance by dynamically adjusting to different lighting and skin tones, making them suitable for diverse user groups (Ibrahim et al., 2018).

Chapter 3

Methodology

3.1 Background

This section of the report details the core functionalities of the project. Each project comes with its own distinct set of requirements, which are typically divided into functional and non-functional categories. For this particular project, the primary requirement is to create a system capable of recognizing numbers from American Sign Language and translating them into text. The project employs the spiral development methodology.

3.2 Methodology

The spiral methodology consists of four distinct phases. The initial phase, known as the identification phase, involves planning for the sign language translation system. During this phase, the specific requirements are defined, and all essential information is gathered to ensure the project's successful completion. Additionally, research is conducted on OpenCV and TensorFlow to evaluate their efficiency for the project.

The second phase, also termed the identification phase, focuses on gathering the system's requirements. The third phase, referred to as the construction or implementation phase, encompasses the development of the sign language translator, where all errors are rigorously tested and corrected. The final phase is the evaluation phase, in which the completed system is ready for use by others, and the program is thoroughly evaluated and documented for potential future improvements.

3.3 Proposed Network Architecture:

3.3.1 CNN overview:

Convolutional Neural Networks (CNNs) are a type of deep neural network extensively used for image recognition tasks. They are particularly powerful for image classification due to their hierarchical structure and efficient feature extraction capabilities. CNNs convert input images into a format that computers can process by transforming them into matrix form. During this transformation, the system identifies which images correspond to which labels by analyzing differences in the matrices. Throughout the training phase, the network learns how these differences impact the labels, enabling it to predict labels for new images.

A CNN typically comprises three main types of layers: convolutional layers, pooling layers, and fully connected layers. The convolutional and pooling layers handle the feature extraction, while the fully connected layer is responsible for classification. This layered approach ensures effective processing and accurate image classification. The structure and flow of these operations are illustrated in the block diagram below fig(1):

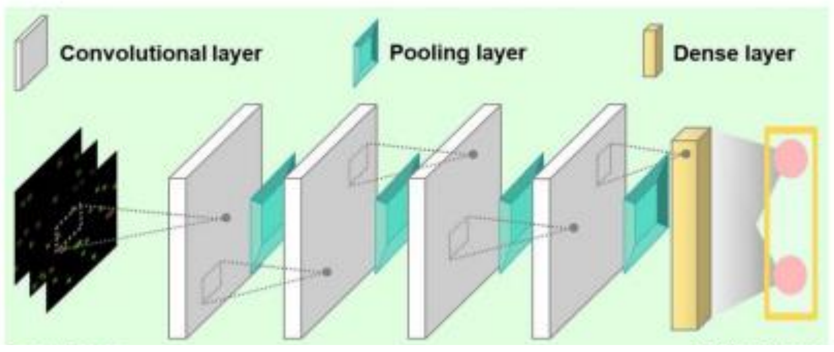


Figure 1

My proposed architecture consists of two main components: the feature extraction block and the classification block. Each component is discussed in detail below.

I. Feature Extraction Block:

a. Convolution Layer:

The convolutional layer is fundamental to CNNs, tasked with identifying features within the input patterns. This layer processes the input image through a filter, resulting in a feature map. It applies kernels that move across the image to extract both low- and high-level features. The stride parameter determines the steps taken while shifting over the input matrix.

Figure 2 illustrates how the kernel or filter extracts features when convolved with a
padded image.

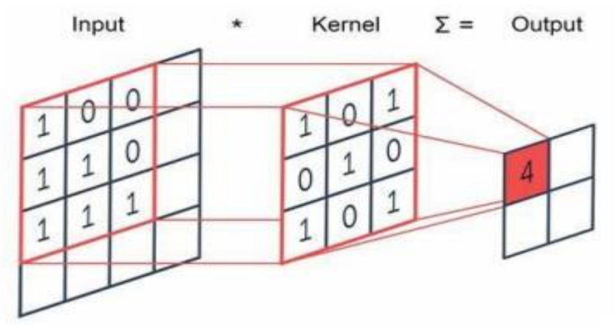


Figure 2

b. Pooling Layer:

Following the convolutional layer, the pooling layer is applied to the generated feature maps. The primary purpose of this layer is to reduce the spatial dimensions of the feature maps and the number of network parameters through specific mathematical operations. In this study, max pooling is utilized. Max pooling works by selecting the maximum value within a defined matrix size in each feature map, thereby decreasing the number of output neurons. This process is depicted in Figure 3.

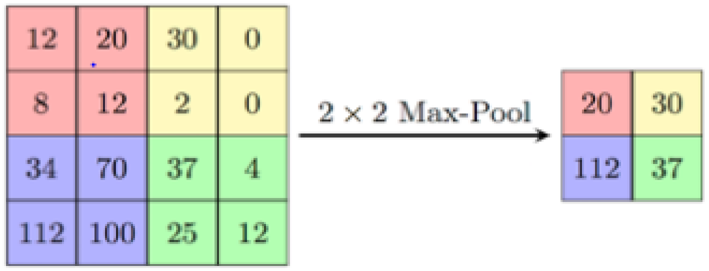


Figure 3

c. Rectified Linear Unit (ReLU):

The Rectified Linear Unit (ReLU) activation function is employed to learn complex mappings between inputs and response variables. ReLU is widely used in the hidden layers of CNNs due to its non-linear properties, which enable the network to extract and learn intricate features from the input data. Its simplicity and computational efficiency make it particularly suitable for deep networks with numerous layers, such as in this project.

ii. Flatten Layer:

The flatten layer (Figure 4) serves to convert a multidimensional array into a single-dimensional array, facilitating the transition to subsequent layers. This layer reshapes the output from the convolutional and pooling layers into a one-dimensional vector. This transformation is crucial as it prepares the data for the fully connected layers, which require a flattened feature representation as input. In this project, the output of the convolutional layer is flattened into a long feature vector, which is then fed into the final classification model, known as the fully connected layer.

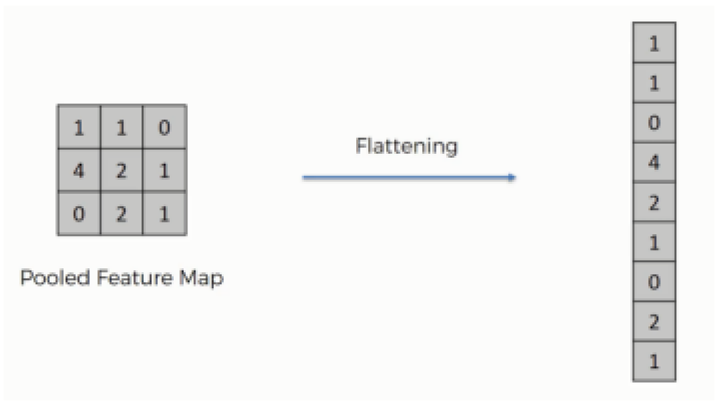


Figure 4

3.4 Project Design

The subsequent phase is the design phase. This phase involves the conceptual and architectural planning of the proposed system, which, in this case, is the sign language translator. In this phase, a near-accurate design of the sign language translator is developed. This step is crucial because a well-defined design ensures consistency and coherence throughout the project. Without a clear design to follow, the development process could become disorganized and inefficient. A specific design framework helps streamline the entire process, making it more precise and effective.

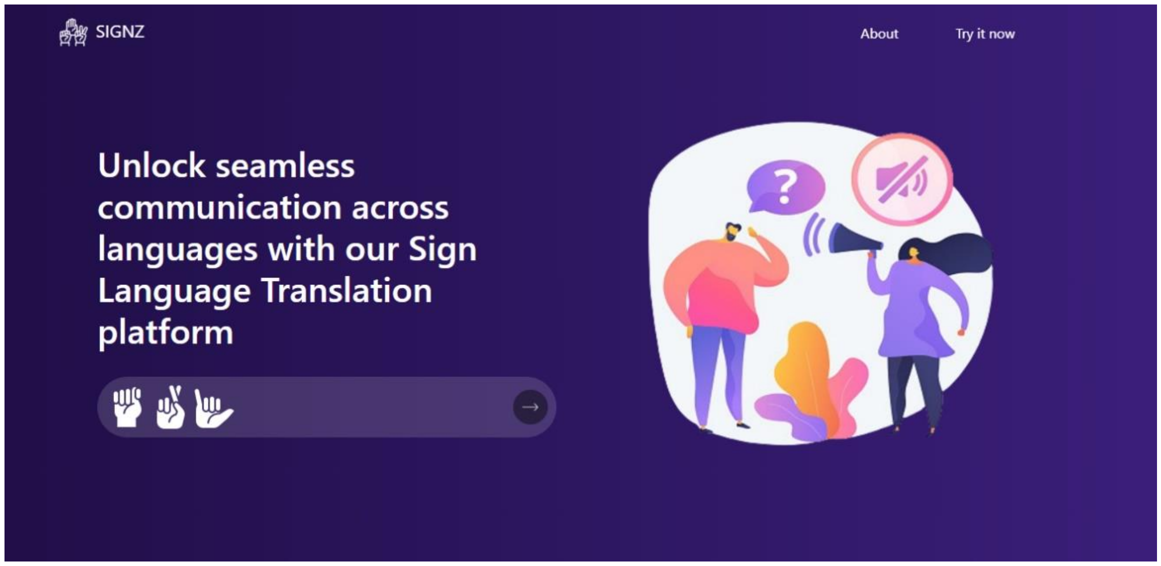

SIGNZ
About
Try it now
Unlock seamless communication across languages with our Sign Language Translation platform

Figure 5

3.3 Hardware and Software Requirements

- Software Requirements

The Sign Language Translation website harnesses various software tools and libraries, including TensorFlow, Keras, OpenCV, NumPy, Matplotlib, Scikit-learn, Google Colab, Visual Studio Code, and Flask. These components collectively enable the development of deep learning models, image processing, numerical computations, visualization, and web application deployment, ensuring efficient development and seamless user interaction.

- Hardware Requirements
  Computer - Necessary for developing and running the project.
  Webcam - Required for capturing images to translate sign language.

3.6 Network's Block Diagram:

1. Data Collection:

The first step in the workflow involved gathering relevant data to train and evaluate the model. The dataset was sourced from Kaggle, consisting of 11 classes (0 to 9 and Unknown) , 11 * 1500 = 16500 images, each class has 1500 images representing 10 different numbers in hand signs. This dataset was divided into a training set and an independent test set to later re-evaluate the model's performance on unseen data.

2. Data Pre-processing:

The training dataset was pre-processed to make it suitable for training the deep learning model. Firstly, the colored images were converted to grayscale to reduce data complexity and enhance processing speed. Secondly, the images were transformed from their original format into numeric arrays containing pixel intensity values. These images and their corresponding labels were stored in lists, which were then converted into NumPy arrays for model training. All images were resized to 100x100 pixels to ensure uniform dimensions. The pixel intensity values were normalized to a range of [0, 1] to homogenize the input data and avoid bias. The dataset was split into 80% for the training set and 20% for the validation set.

3. Model Selection and Training:

- Introduction:

The VGG16 model is a convolutional neural network architecture that has been widely used for image classification tasks. Developed by the Visual Geometry Group at the University of Oxford, VGG16 is known for its simplicity and effectiveness in extracting hierarchical features from images. In the context of sign language recognition, the VGG16 model serves as a powerful tool for accurately identifying hand signs and translating them into corresponding text or numerical representations.

- How VGG16 Works:

The VGG16 model consists of multiple convolutional layers followed by max-pooling layers, which progressively extract features from input images while reducing spatial dimensions. These features capture various visual patterns such as edges, textures, and shapes, enabling the model to learn discriminative representations of sign language gestures. The hierarchical nature of the architecture allows for the extraction of both low-level and high-level features, enhancing the model's ability to recognize complex hand movements.

- Training Process

During the training process, the VGG16 model learns to adjust its parameters by minimizing a predefined loss function. This optimization is achieved through backpropagation, where gradients are calculated with respect to the loss and used to update the model's weights. By iteratively optimizing its parameters on a labeled dataset of sign language images, the model becomes increasingly proficient at classifying different hand signs with high accuracy.

- Fine-tuning for Sign Language Recognition

To adapt the VGG16 model to the task of sign language recognition, it is fine-tuned on a specialized dataset containing images of sign language gestures. By leveraging transfer learning, the model initializes its weights with pre-trained values learned from a large-scale dataset such as ImageNet. These pre-trained weights serve as a valuable starting point, allowing the model to quickly learn relevant features specific to sign language gestures with minimal training data.

4. Model Evaluation:

Once the model is trained, I re-evaluate the model's performance on an independent test set (unseen data) to determine how it will generalize to this data. A dataset of 16500 images distributed among the 11 classes was used to re-evaluated a model and a promising outcome was
shown.

5. Model Classification:

When an input image is loaded into the system, the trained model classifies it as one of the eleven classes representing the digits 0 to 9, with an additional "unknown" class that is not used.

Chapter 4

In this chapter, we will present the pre-processed data and discuss the model's performance during the training phase by providing evaluation metrics. Additionally, we will showcase the results of several experiments conducted during the training process.

4.1 Reading and Loading Data:

The dataset collected contains images belonging to four different classes was downloaded as a zip folder on the desktop. Then this dataset was uploaded and unzipped on the drive to a folder called Dataset. After that, we mounted the drive to the Google Collab directory. we read the folder path and assigned labels to each class (0 for 0, 1 for 1 and so on). we resize the images to 100x100 and then normalized to a range [0, 1]. Lastly, I split the dataset to 80% train, 20% validation.

4.2 Training Progress

In this section, we will present and discuss the results of my training on the prepared dataset. Figure 3.2-1 shows the validation/training accuracy and the validation/training loss at several epochs, reaching the final epoch at 10/10 during the training process. The choice of 10 epochs as the endpoint of the training process was based on experiments. we monitored the performance over multiple epochs and assessed the training progress to determine the optimal stopping point. This experiment showed that up to epoch 10, the accuracy of both validation and training datasets was increasing, and the loss for both was decreasing. This led to a training accuracy of 94.67%, a validation accuracy of 96.43%, a training loss of 15.87%, and a validation loss of 12.11%.

4.3 Iterative Improvements to Model Accuracy

Throughout the development and training of the model, various techniques were iteratively applied to enhance its performance and achieve higher accuracy. Initially, the model underwent modifications aimed at improving accuracy by addressing potential limitations. Below are the key modifications implemented during this iterative refinement process:

1- Increasing Model Complexity:
The model architecture was augmented by increasing the number of trainable parameters and adding additional layers. For example, the number of neurons in the dense layer was increased from 64 to 256.

2- Expanding Image Size:
Input images were initially resized to smaller dimensions (e.g., 50x50 pixels) to reduce computational complexity. However, as part of the optimization process, the image size was increased to provide the model with more detailed information (e.g., 100x100 pixels).

3- Fine-tuning Regularization:
The regularization techniques such as dropout and L2 regularization were fine-tuned by adjusting their parameters. For example, the dropout rate was decreased from 0.5 to 0.3.

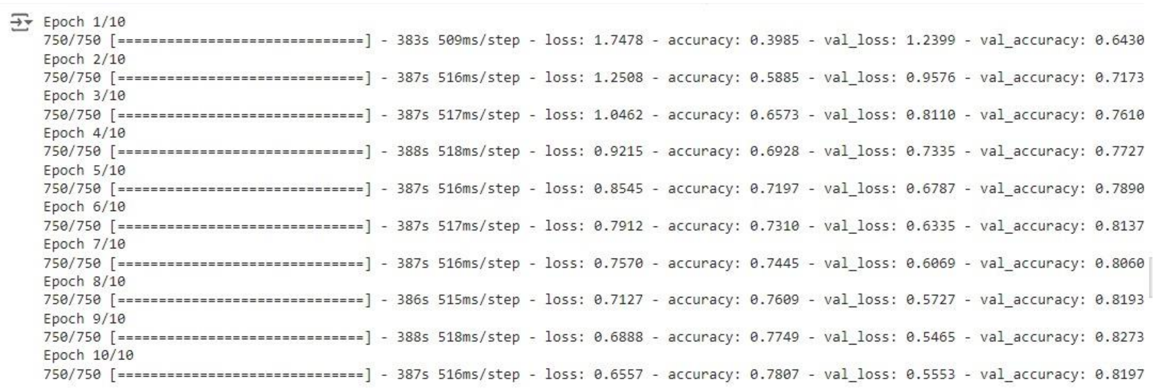
4- Enhancing Data Augmentation:

Data augmentation techniques and parameters were optimized to generate more diverse and representative data. For instance, the range of rotation angles for image rotation was expanded from ±10 degrees to ±20 degrees.

Through iterative experimentation and fine-tuning of these techniques, the model gradually improved its accuracy and performance. By carefully adjusting model complexity, data representation, and regularization strategies, significant enhancements were achieved, resulting in a highly accurate and reliable model.

4.4 Testing and Experiments

During the initial phase of testing our model, the performance metrics indicated substantial room for improvement. The model achieved a training loss of 65.57% and an accuracy of 78.87%, with a validation loss of 53.33% and a validation accuracy of 81.97%. Recognizing the need for enhancement, we undertook a series of optimization experiments. These included refining the model architecture, tuning hyperparameters, and implementing data augmentation techniques. After these improvements, the model exhibited significantly better performance. The training loss was reduced to 15.87%, and the training accuracy increased to 84.67%. Similarly, the validation loss dropped to 12.11%, and the validation accuracy soared to 96.43%. These results demonstrate the effectiveness of the optimization strategies in enhancing the model's accuracy and generalization capability.



```
Epoch 1/10
750/750 [==============================] - 383s 509ms/step - loss: 1.7478 - accuracy: 0.3985 - val_loss: 1.2399 - val_accuracy: 0.6430
Epoch 2/10
750/750 [==============================] - 387s 516ms/step - loss: 1.2508 - accuracy: 0.5885 - val_loss: 0.9576 - val_accuracy: 0.7173
Epoch 3/10
750/750 [==============================] - 387s 517ms/step - loss: 1.0462 - accuracy: 0.6573 - val_loss: 0.8110 - val_accuracy: 0.7610
Epoch 4/10
750/750 [==============================] - 388s 518ms/step - loss: 0.9215 - accuracy: 0.6928 - val_loss: 0.7335 - val_accuracy: 0.7727
Epoch 5/10
750/750 [==============================] - 387s 516ms/step - loss: 0.8545 - accuracy: 0.7197 - val_loss: 0.6787 - val_accuracy: 0.7890
Epoch 6/10
750/750 [==============================] - 387s 517ms/step - loss: 0.7912 - accuracy: 0.7310 - val_loss: 0.6335 - val_accuracy: 0.8137
Epoch 7/10
750/750 [==============================] - 387s 516ms/step - loss: 0.7570 - accuracy: 0.7445 - val_loss: 0.6069 - val_accuracy: 0.8060
Epoch 8/10
750/750 [==============================] - 386s 515ms/step - loss: 0.7127 - accuracy: 0.7609 - val_loss: 0.5727 - val_accuracy: 0.8193
Epoch 9/10
750/750 [==============================] - 388s 518ms/step - loss: 0.6888 - accuracy: 0.7749 - val_loss: 0.5465 - val_accuracy: 0.8273
Epoch 10/10
750/750 [==============================] - 387s 516ms/step - loss: 0.6557 - accuracy: 0.7807 - val_loss: 0.5553 - val_accuracy: 0.8197
```

Figure 6

4.5 Classification

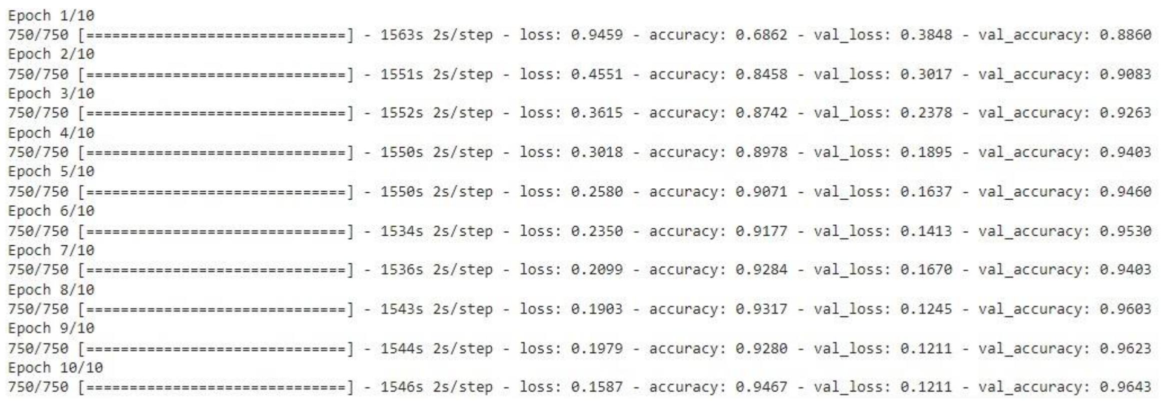
For classification, I used the Softmax activation function in the final layer of the VGG16-based CNN model. The Softmax function takes a vector of real numbers as input and converts them into a probability distribution, ensuring that the sum of the output probabilities equals 1. This makes it ideal for multi-class classification tasks where each input sample belongs to one of several classes.

To predict the class of each image, the model.predict() function is called. This function processes the input image and returns an output matrix containing probabilities for each class. For example, in this project, the model returns a matrix with ten probabilities, each corresponding to one of the digit classes (0 through 9).

To determine the predicted class for each image, the np.argmax() function is used. This function identifies the index of the highest probability in the output matrix, which corresponds to the predicted class.

4.6 Results

Overall achieved model’s accuracy on the training, validation, and testing dataset. The model scores 94.67% training accuracy, 96.43% validation accuracy.



```
Epoch 1/10
750/750 [==============================] - 1563s 2s/step - loss: 0.9459 - accuracy: 0.6862 - val_loss: 0.3848 - val_accuracy: 0.8860
Epoch 2/10
750/750 [==============================] - 1551s 2s/step - loss: 0.4551 - accuracy: 0.8458 - val_loss: 0.3017 - val_accuracy: 0.9083
Epoch 3/10
750/750 [==============================] - 1552s 2s/step - loss: 0.3615 - accuracy: 0.8742 - val_loss: 0.2378 - val_accuracy: 0.9263
Epoch 4/10
750/750 [==============================] - 1550s 2s/step - loss: 0.3018 - accuracy: 0.8978 - val_loss: 0.1895 - val_accuracy: 0.9403
Epoch 5/10
750/750 [==============================] - 1550s 2s/step - loss: 0.2580 - accuracy: 0.9071 - val_loss: 0.1637 - val_accuracy: 0.9460
Epoch 6/10
750/750 [==============================] - 1534s 2s/step - loss: 0.2350 - accuracy: 0.9177 - val_loss: 0.1413 - val_accuracy: 0.9530
Epoch 7/10
750/750 [==============================] - 1536s 2s/step - loss: 0.2099 - accuracy: 0.9284 - val_loss: 0.1670 - val_accuracy: 0.9403
Epoch 8/10
750/750 [==============================] - 1543s 2s/step - loss: 0.1903 - accuracy: 0.9317 - val_loss: 0.1245 - val_accuracy: 0.9603
Epoch 9/10
750/750 [==============================] - 1544s 2s/step - loss: 0.1979 - accuracy: 0.9280 - val_loss: 0.1211 - val_accuracy: 0.9623
Epoch 10/10
750/750 [==============================] - 1546s 2s/step - loss: 0.1587 - accuracy: 0.9467 - val_loss: 0.1211 - val_accuracy: 0.9643
```

Figure 7

Chapter 5

5.1 Functional and non functional requirements

❖ Functional requirements

- The website should provide the ability to translate sign language to text.
- It should also offer the capability to translate text to sign language.
- The website should support importing photos for translation.
- It should have an option to open the camera for real-time sign translation.
- It should support multiple sign languages and have a comprehensive sign language dictionary.

❖ Non-Functional requirements

- The website should have a user-friendly interface for easy navigation.
- It should have a fast and responsive performance.
- The website should be accessible and compatible with different devices and browsers.
- The translations should be delivered in a timely manner.
- The website should prioritize user privacy and data security.
- It should have a visually appealing and intuitive design.
- The website should be scalable to handle increasing user traffic.

5.2 Use Case Diagram

This website offers a platform that captures images from the webcam and converts the detected numbers into text. It also allows users to view previous tests conducted using the system. Additionally, users can download all the images previously uploaded to the database.

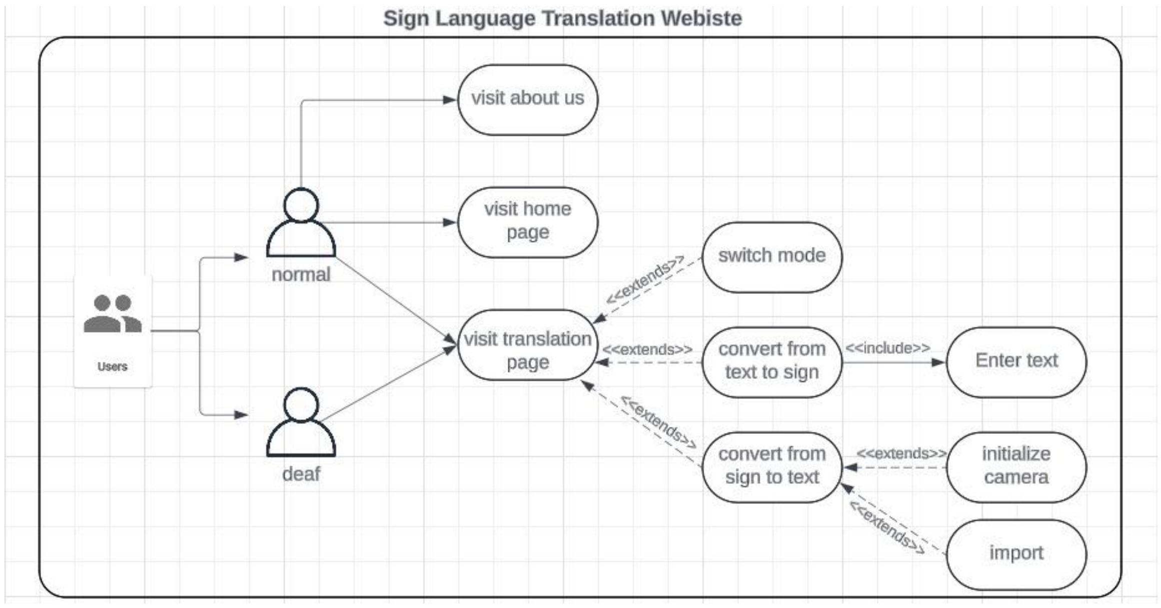
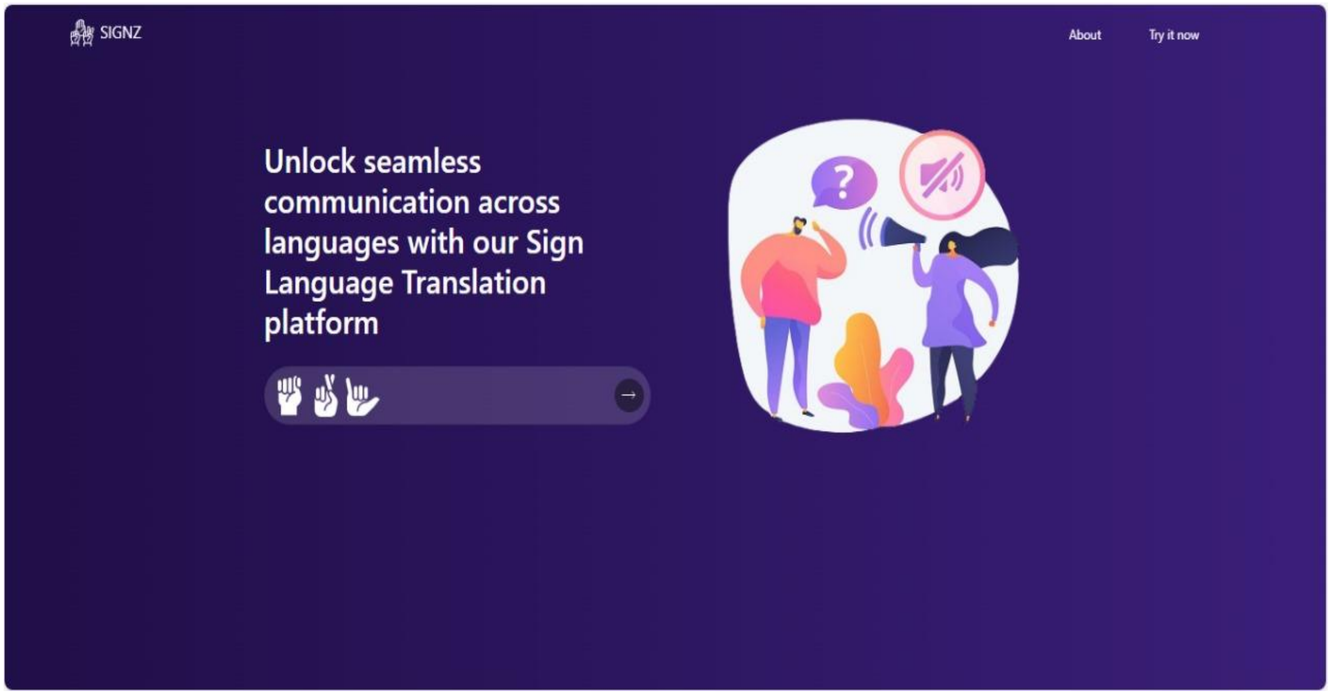


Figure 8

5.3 Implementation and Design

The implementation of our website focuses on delivering a user-friendly and accessible experience for deaf users, allowing them to easily translate sign language into text. The website is designed with simplicity and usability in mind, consisting of three main pages: Home, About, and Translation.
To enhance accessibility, we have incorporated sign language letters for the word "try" on the Translation page. This feature guides deaf users on where to click, ensuring they can navigate the site intuitively.

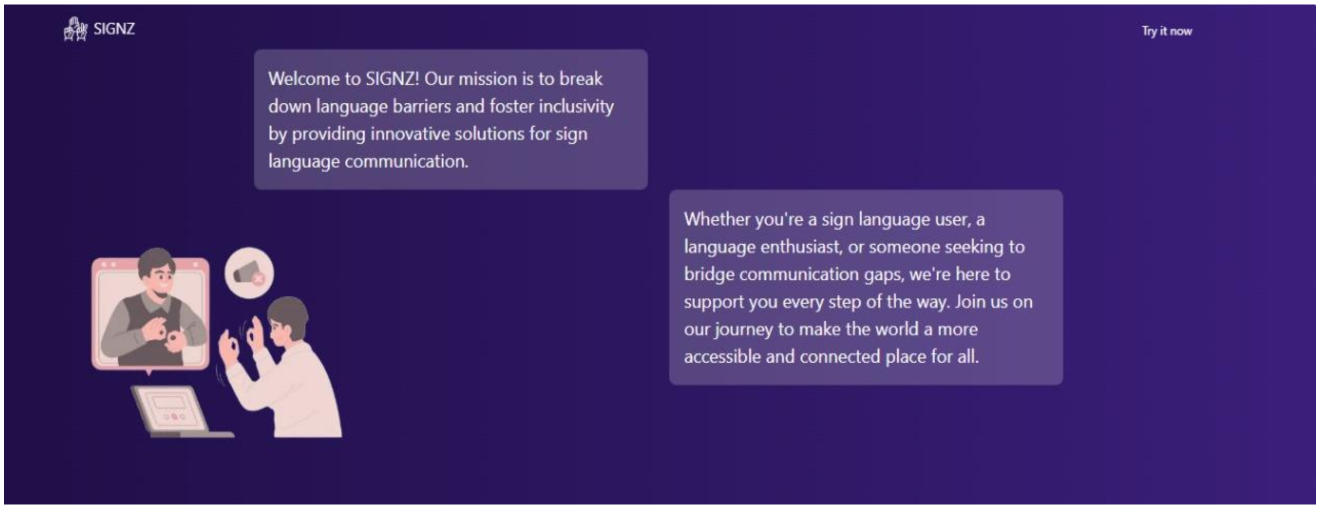
- Home Page: Offers a welcoming introduction and easy navigation to other sections of the site.



Unlock seamless communication across languages with our Sign Language Translation platform

Figure 9

- About Page: Provides information on the purpose and development of the project.



Welcome to SIGNZ! Our mission is to break down language barriers and foster inclusivity by providing innovative solutions for sign language communication.

Whether you're a sign language user, a language enthusiast, or someone seeking to bridge communication gaps, we're here to support you every step of the way. Join us on our journey to make the world a more accessible and connected place for all.

Figure 10

- Translation Page: The core feature of the site, enabling users to translate sign language in real-time.

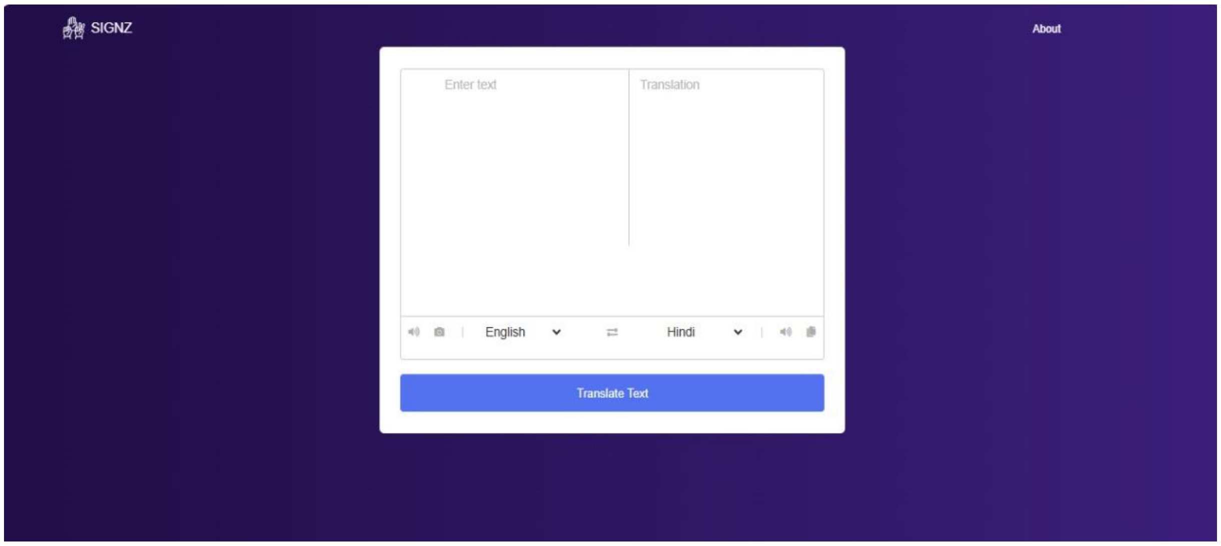


Figure 11

- Demo

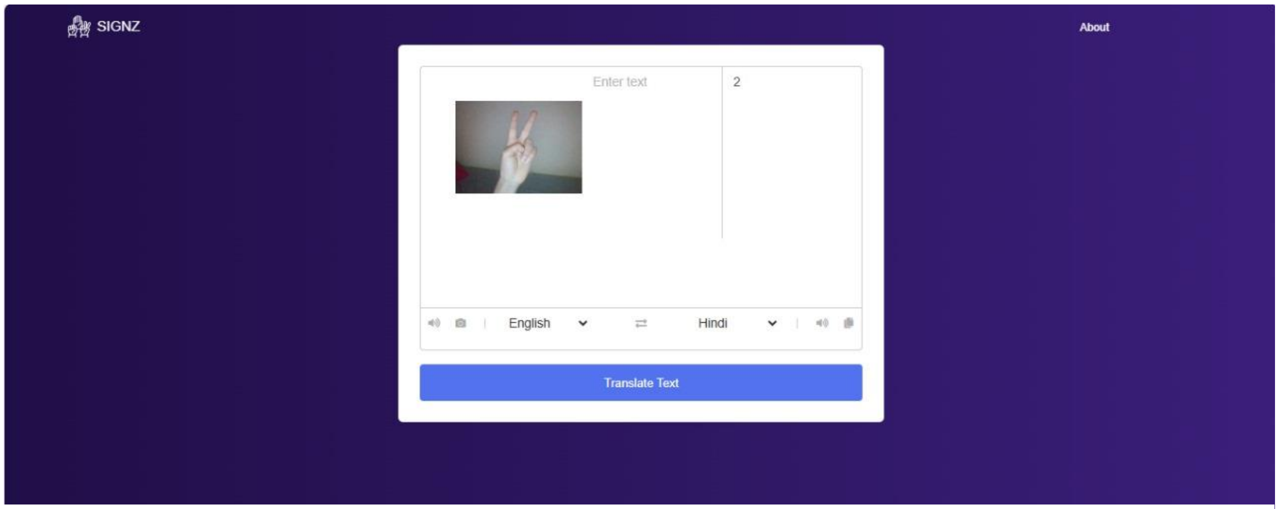


Figure 12

Chapter 6

6.1 General Conclusion

To conclude succinctly, the proposed system aims to address the language barrier by facilitating communication through sign language. Currently, only a small fraction of non-deaf individuals can communicate effectively using sign language. However, given the advancements in technology, there's an urgent need for a solution to bridge this gap. Just as there are tools available for spoken languages, sign language should also have its own assistance. With a sign language translator, individuals who are hearing impaired can communicate seamlessly with those who are not. Although there's still room for development, the rapid progress in technology suggests that a solution is within reach in the near future. It's high time to take this concept seriously and work towards its realization.

6.2 Recommendations

Looking ahead, there are several potential enhancements that could be incorporated into the system in the future. Firstly, expanding the functionality of the sign language translator to include mobile applications and online platforms would greatly benefit hearing-impaired individuals. This would provide greater accessibility, allowing users to utilize the translator on-the-go, rather than being confined to a desktop computer. Additionally, offering support for multiple sign languages beyond just American Sign Language (ASL) would be advantageous. Incorporating various sign languages from different countries and regions with distinct dialects would cater to a more diverse user base, ensuring inclusivity and accessibility on a global scale.

References

I. Chen , Kim, Lu H, Li Y, M, H. and Serikawa, S., (2017). “Brain Intelligence: Go beyond Artificial Intelligence,” Mobile Networks and Applications **23(2)** :368-375

II. A., Haenlein, Kaplan and M, (2019). “A Brief History of Artificial Intelligence: On the Past,Present, and Future of Artificial Intelligence,”California Management Review **61(4)** : pp.5-14

III. Hodes C and Miailhe, N, (2017). “The third age of artificial intelligence”: Field actions science reports. The Journal of Field Actions **(17)**: pp.6-11

IV. A., Doulamis, E. Voulodimos, N. Doulamis, and Protopapadakis, (2018). “Deep Learning for Computer Vision,” Computational Intelligence andNeuroscience : pp.1-13.

V. P. Zheng, S. Xu and X. Wu, Z. Zhao, (2019) "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, **30(11):** pp. 3212-3232

VI. Chen, Fieguth, Liu, L., Liu, X. Ouyang, Pietikäinen, X., and W. Wang (2020). “Deep learning for generic object detection,” International journal of computer vision, **128(2):** pp.261-318

VII. Ball, C., 2017. “The History of American Sign Language Interpreting,”
International Review of Studies in Applied Modern Languages
VIII. Jaward M., Jin Cheok M., Omar, Z. (2019). “A review of hand gesture and sign
language recognition techniques,” International Journal of Machine Learning
and Cybernetics **10(1–3**)
45
IX. Bantupalli, K. and Xie, Y., (2018). “American Sign Language Recognition
using Deep Learning and Computer Vision,” IEEE International Conference
on Big Data

X. Ibrahim, N., Selim, M. and Zayed, H., (2018). “An Automatic Arabic Sign
Language Recognition System,” Journal of King Saud University - Computer
and Information Sciences, **30(4):** pp.470-477
XI. Bengio, Y , Hinton, G and LeCun, Y, (2015). “Deep learning,” **521(7553):**
pp.436-444.

XII. Bengio, Y ., Courville, A. and Goodfellow, I., (2016). “Deep learning,”
(Vol.1,No. 2)

XIII. Armstrong, David F., and Michael A. Karchmer. William C. Stokoe. (2009).
“Study of Signed Languages." Sign Language Studies **9(4)**: 389-97.

XIV. Correll, Robyn. (2017). “Challenges That Still Exist for the Deaf Community.” Verywell Health.

XV. Annelies Braffort, Christophe Collet ,Christian Vogler, Efthimiou, Eleni, Jérémie Segouat, John Glauert , Petros Maragos, Richard Bowden, Stavroula-Evita Fotinea and Thomas Hanke,(2009). “Sign Language Recognition, Generation, and Modelling: A Research Effort with Applications in Deaf Communication.” Universal Access in Human-Computer Interaction. Addressing Diversity **(5614):** pp21–30

XVI. Steinmetz, Katy. (2014). “This Technology Could Change the Way Deaf People Live.”