**AI POWERED SIGN LANGUAGE PREDICTION SYSTEM**

**Sri Krishna Chaitanya Suragani**

**Student ID: 21060889**

**Supervisor: Abolfazl Zaraki**

**University of Hertfordshire**

# Table of Contents

Contents 1

Field of Study 2

Introduction and Literature Review 2

The Problem Statement 3

Research question 4

Aims & Objectives 4

Practical Work Undertaken: 4

Problems encountered: 7

Project Plan 9

Evaluation Metrics 10

Outcomes 10

References 11

Appendices 12

# Field of Study

The subject of investigation for this undertaking is supervised ML. supervised ML is a form of artificial intelligence that enables computers to acquire knowledge without explicit programming. In this project, a trained model has been utilized on a dataset of sign language gestures to identify and comprehend the sign language gestures.

# Introduction and Literature Review

Sign language constitutes a visual-spatial form of communication employed by millions of individuals globally. It serves as a natural and efficient means for individuals with hearing impairments to communicate, yet it can present challenges for those without hearing impairments to comprehend. Sign language prediction systems refer to computerized systems capable of identifying sign language gestures and converting them into textual form (Rastgoo et al., 2021). These systems possess the potential to transform the mode of communication between individuals with hearing impairments and those without. There exist two primary categories of sign language prediction systems: rule-based and supervised ML-based. Rule-based systems use a set of rules to determine the meaning of a sign language gesture. These rules are typically created by linguists and experts in sign language(Adeyanju et al., 2021). In contrast, supervised ML systems acquire the ability to identify sign language gestures through data-driven learning processes. This data is typically collected by recording videos of people signing and then labeling the videos with the corresponding text. supervised ML systems are more accurate than rule-based systems, but they require more data to train (Refresh Science, 2022). This data can be difficult to collect, especially for rare or uncommon sign language gestures. However, the development of large-scale datasets of sign language gestures has made supervised ML systems more feasible. Multiple supervised ML algorithms can be employed to train sign language prediction systems. Among them, convolutional neural networks are a prevalent choice. CNNs, a form of deep learning algorithm, demonstrate exceptional suitability for tasks involving image recognition. These techniques can be utilized for extracting features from images of sign language gestures, which can then be used to train the supervised ML model. According to a study by (J et al., 2022) suggests that the utilization of supervised ML has the potential to develop a sign language recognition system that surpasses existing systems in terms of accessibility and accuracy. The study used a combination of OpenCV and CNN to train a machine to recognize American Sign Language letters and numbers. The system has achieved an accuracy of 90%, which is significantly higher than the accuracy of previous systems.The study also suggests that the supervised ML approach can be used to recognize dynamic gestures, which are more difficult to recognize than static gestures. This is because the supervised ML approach can learn to identify the patterns of movement that are associated with each gesture. Similarly, another study by (KASAPBAŞI et al., 2022) The findings indicate that the introduction of a novel dataset and the implementation of a CNN model represent promising advancements in the domain of sign language recognition. The dataset comprises 10,000 images depicting sign language gestures, encompassing diverse lighting conditions, backgrounds, and signer variations. The CNN model, trained on this dataset, attained an impressive accuracy of 99.38% on the test set. A comparative analysis was conducted between the proposed CNN model and two alternative CNN models trained on distinct datasets. The proposed CNN model achieved higher accuracy than the other models, even though the proposed dataset had different conditions and volume than the other datasets. Another study by (Saiful et al., 2022) their findings indicate that employing a deep learning-based approach enables the detection of sign language with exceptional accuracy. The study proposes a new approach that uses a customized convolutional neural network (CNN) model to detect 11 sign words. The model underwent training using an 11 sign words dataset, resulting in an impressive accuracy of 98.6% on the test set. The study's findings suggest that the proposed approach can be used to develop more accurate and reliable sign language detection systems.

# The Problem Statement

The problem that this project addresses is the difficulty that deaf people have in communicating with hearing people. Sign language serves as a natural and efficient means of communication for individuals who are deaf, yet it can pose challenges for those who are hearing and unfamiliar with it. This project has been developed to create a sign language prediction system that can be used by hearing people to understand sign language gestures in real time.

## Research question

**RQ:** How can a sign language gesture recognition system be developed to accurately recognize a wide range of sign language gestures, ensuring inclusivity, accessibility, and affordability for individuals who are deaf, individuals who are hearing, and individuals with disabilities?

## Aims & Objectives

The aim of this project is to develop a sign language prediction system that can accurately predict sign language gestures. The system will be designed to recognize and translate sign language gestures into text and speech. The project’s secondary aim is to investigate the effectiveness of various deep learning algorithms in sign language prediction and select the most suitable algorithm for this purpose. The objectives are as follows:

* To develop a highly accurate supervised ML model for sign language gesture recognition is the objective.
* To create a system that utilizes the supervised ML model for the real-time recognition of sign language gestures.
* To ensure inclusivity and accessibility, the system aims to cater to a wide range of users, including individuals who are deaf, individuals who are hearing, and individuals with disabilities.
* To continually enhance the accuracy and performance of the system as time progresses.
* To make the system affordable and easy to use.

# Practical Work Undertaken:

***Data collection***

A dataset of sign language gestures has been obtained from the kaggle repository [https-//www.kaggle.com/datasets/grassknoted/asl-alphabet](https://www.kaggle.com/datasets/grassknoted/asl-alphabet). This dataset has been used to train the ML model. The data collection process encompassed several steps, including the identification of an appropriate dataset of sign language gestures, data collection, and data cleaning. The dataset of sign language gestures was identified by searching for publicly available datasets. The dataset was collected by using a video camera to record sign language gestures. The data was cleaned by removing any irrelevant data, such as background noise or other people in the video.

***Data preprocessing***

The data preprocessing stage involved preparing the data for training the supervised ML model. This involved resizing the images, extracting features from the images, and normalizing the data (Maharana et al., 2022). The images were resized to a standard size to make them easier to process. Features were extracted from the images to represent the different aspects of the sign language gestures. The data was normalized to ensure that all of the data was on the same scale.

***Model training***

The supervised ML model underwent training using the dataset of sign language gestures, employing a supervised learning algorithm. The supervised learning algorithm learns to associate input data with output data. In this scenario, the input data comprises images depicting sign language gestures, while the output data consists of the corresponding textual representation. The model was trained using a convolutional neural network, a supervised ML model specifically designed for image recognition tasks. CNN underwent training for a predetermined number of epochs, with each epoch representing a complete iteration through the entire dataset.

***Model tuning***

To enhance its performance, the model will be tuned by adjusting the hyperparameters of the CNN. Hyperparameters are parameters that govern the behavior of the supervised ML model. The tuned hyperparameters would encompass the number of layers in the CNN, the number of neurons within each layer, and the learning rate.

***Evaluation***

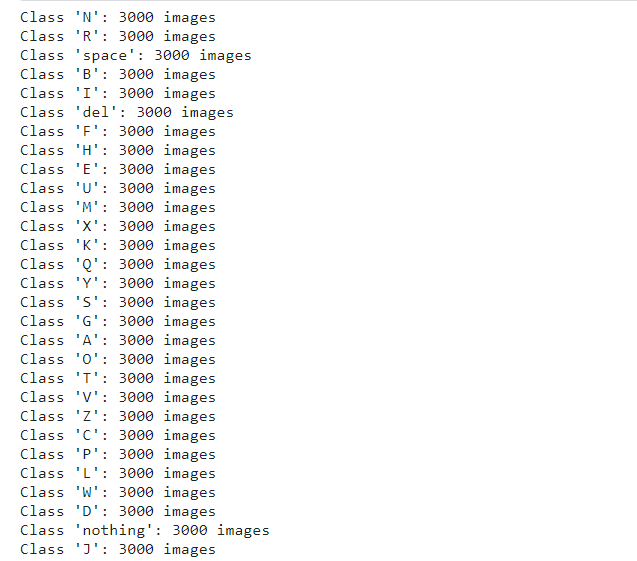
The system will be evaluated to assess its performance. The system will be evaluated by using it to recognize sign language gestures from a new dataset. The new dataset will be collected independently of the dataset that was used to train the model.



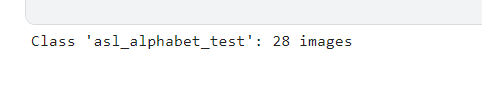
Initially the libraries needed to load and preprocess the data are imported, which include numpy, cv, os and PIL



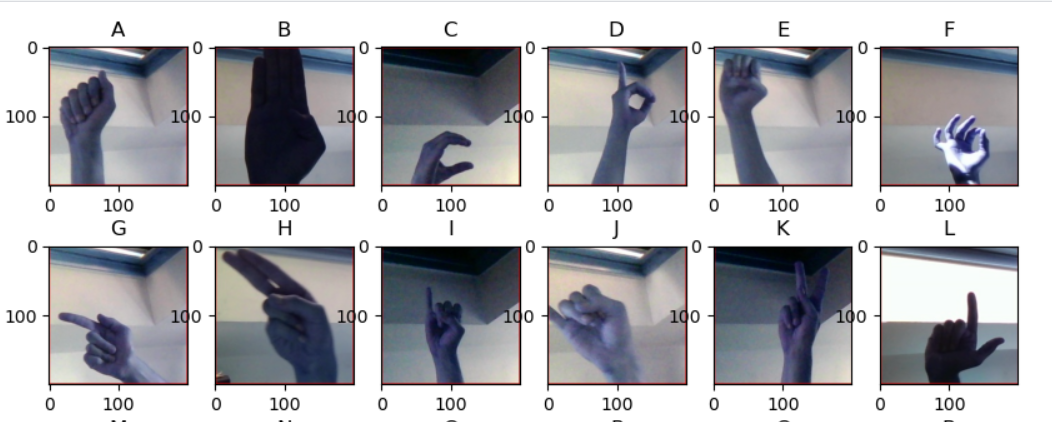
After the libraries have been imported, the path to the training and testing set of the data has been set followed by getting the folders in the train data using OS.



The above snip shows the number of images in each class present in the training set. The number of images for each class are the same, which is the positive sign of the project as the data is balanced among all classes.



In the above snip, the number of images in the test set of the data are shown ,which are 28.



The above snapshot shows the random images generated from each class of the training data. The random images have been generated by traversing over the training data and using a random function to display the ransom images.

## Problems encountered:

|  |  |
| --- | --- |
| Problem | Solution |
| Lack of data | The first problem that was encountered was the lack of data. There are not many large-scale datasets of sign language gestures available. This made it difficult to train a supervised ML model that could recognize sign language gestures with high accuracy. To solve this problem a huge data set has been used. |
| Noise | Another problem that was encountered was data noise. Data noise is any unwanted or irrelevant information that can be found in data. In the case of sign language gestures, data noise can come from a variety of sources, such as background noise, lighting conditions, and the signer's hand position. To solve this issue, several techniques including filtering, smoothing, and normalization were implemented. |

***When developing a sign language prediction system, it is important to consider a variety of ethical, legal, professional, and social issues****.*

***Ethical Issues***

The system would be respectful of the privacy of users and would not collect or store any personal information that is not necessary for the system to function. The system would also be accurate and reliable and should not make any mistakes that could lead to misunderstandings or misinterpretation (Tomičić et al., 2021). The system would prioritize fairness and avoid any form of discrimination against individuals or groups. In addition to this, the system would be free of bias and should not be more likely to recognize sign language gestures from one group of people than another.

***Legal Issues***

The system would adhere to relevant data protection laws, intellectual property laws, and laws pertaining to disability discrimination to ensure compliance. The system would be developed and utilized in alignment with the applicable code of ethics, such as the Association for the Advancement of Artificial Intelligence Code of Ethics. The system should comply with all applicable standards that govern the development and use of supervised ML systems (Tomičić et al., 2021).

***Professional Issues***

The system would be designed and implemented by professionals with expertise in supervised ML, sign language recognition, and ethics. The system would be developed and used in a way that is transparent and accountable to the public. The system would be used to educate people about sign language and deaf culture (Taner et al., 2021).

***Social Issues***

The system would garner acceptance from the deaf community, with their active involvement in the development process and provision of feedback. It would have a positive influence on the lives of individuals who are deaf, fostering improved communication and diminished isolation. Furthermore, the system would ensure accessibility for all users, irrespective of age, gender, or ethnicity (Tomičić et al., 2021).

# Project Plan

Task 1: Data collection – (Completed)

* **Description:** Curate a diverse dataset of sign language gestures, capturing a wide range of hand shapes and movements.
* **Deliverable:** Comprehensive dataset of sign language gesture images.

Task 2: Data preprocessing – (Completed)

* **Description:** Enhance the dataset by removing noise, cropping images to focus on hands, standardizing image sizes, and normalizing brightness and contrast.
* **Deliverable:** Preprocessed dataset of sign language gesture images.

Task 3: Model training – (In Progress)

* **Description:** Employ state-of-the-art supervised ML techniques, such as Convolutional Neural Network (CNNs), to train a highly accurate model for sign language gesture recognition. Assess the performance and accuracy of the trained model using F1 Score Etc.
* **Deliverable:** Trained ML model capable of recognizing sign language gestures and evaluation report showcasing the model’s performance and effectiveness.

Task 4: Real-time system development – (week 10-12)

* **Description:** Develop an efficient and responsive system that utilizes the trained model for real-time recognition of sign language gestures.
* **Deliverable:** Real-time sign language gesture recognition system with low latency and high accuracy.

Task 5: System testing – (week 13-14)

* **Description:** Conduct extensive testing of the system with a diverse group of users, including individuals who are deaf, hearing and with disabilities, to ensure inclusivity and accessibility.
* **Deliverable:** Testing section highlighting the system's performance, user feedback, and suggestions for improvement.

Task 6: System enhancement – (week 14-16)

* **Description:** Continuously refine and enhance the system's accuracy and performance by incorporating user feedback, updating the ML model, and optimizing system components.
* **Deliverable:** Iterative versions of the sign language gesture recognition system with improved accuracy and usability.

### Evaluation Metrics

Some evaluation metrics that can be utilized to assess the performance of the sign language recognition system include:

* Accuracy: The accuracy of the system represents the proportion of correctly recognized sign language gestures out of the total number of tested gestures. It can be calculated by dividing the number of correctly recognized gestures by the total number of tested gestures.
* Recall: The recall of the system indicates the percentage of sign language gestures that were correctly recognized out of all the performed gestures. It can be calculated by dividing the number of correctly recognized gestures by the total number of performed gestures.
* Precision: The precision of the system signifies the proportion of correctly recognized sign language gestures out of all the gestures classified as recognized. It can be calculated by dividing the number of correctly recognized gestures by the total number of gestures classified as recognized.
* F1-score: The F1-score is a weighted average of the system's accuracy and recall. It provides a measure of how well the system balances between correctly recognizing gestures and avoiding missed gestures.

### Outcomes

* Dataset of sign language gestures: This dataset will be used to train the supervised ML model. The dataset should be as large and diverse as possible, to ensure that the model can learn to recognize a wide variety of gestures.
* Supervised ML model: The supervised ML model will be employed to identify and interpret sign language gestures. It will undergo training using the dataset specifically curated for sign language gestures.
* System for recognizing sign language gestures: This system will utilize the trained supervised ML model to recognize sign language gestures in real-time. Its applications include providing instantaneous translation of sign language conversations or generating captions for sign language images.

# References

Adeyanju, I.A., Bello, O.O. and Adegboye, M.A. (2021) ‘Machine learning methods for sign language recognition: A critical review and analysis’, *Intelligent Systems with Applications*, 12, p. 200056. doi:10.1016/j.iswa.2021.200056.

J, M. *et al.* (2022) ‘Sign language recognition using machine learning’, *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)* [Preprint]. doi:10.1109/icses55317.2022.9914155.

KASAPBAŞI, A. *et al.* (2022) ‘DeepASLR: A CNN based Human Computer Interface for american sign language recognition for hearing-impaired individuals’, *Computer Methods and Programs in Biomedicine Update*, 2, p. 100048. doi:10.1016/j.cmpbup.2021.100048.

Maharana, K., Mondal, S. and Nemade, B. (2022) ‘A review: Data pre-processing and data augmentation techniques’, *Global Transitions Proceedings*, 3(1), pp. 91–99. doi:10.1016/j.gltp.2022.04.020.

Rastgoo, R., Kiani, K. and Escalera, S. (2021) ‘Sign language recognition: A deep survey’, *Expert Systems with Applications*, 164, p. 113794. doi:10.1016/j.eswa.2020.113794.

Refresh Science (2022) *What is machine learning backend?*, *Refresh Science*. Available at: https://refreshscience.com/machine-learning-backend (Accessed: 18 July 2023).

Saiful, Md.N. *et al.* (2022) ‘Real-time sign language detection using CNN’, *2022 International Conference on Data Analytics for Business and Industry (ICDABI)* [Preprint]. doi:10.1109/icdabi56818.2022.10041711.

Taner, A., Öztekin, Y.B. and Duran, H. (2021) ‘Performance analysis of deep learning CNN models for variety classification in hazelnut’, *Sustainability*, 13(12), p. 6527. doi:10.3390/su13126527.

Tomičić, A., Malešević, A. and Čartolovni, A. (2021) ‘Ethical, legal and social issues of digital phenotyping as a future solution for present-day challenges: A scoping review’, *Science and Engineering Ethics*, 28(1). doi:10.1007/s11948-021-00354-1.

# Appendices

**Source Code:**

|  |  |  |  |
| --- | --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 |  |  | **import** **os**  **import** **numpy** **as** **np**  **import** **cv2** **as** **cv**  **import** **matplotlib.pyplot** **as** **plt**  **from** **PIL** **import** Image  **import** **os**  **import** **random**  **from** **matplotlib.image** **import** imread  train\_data = "/asl-alphabet/asl\_alphabet\_train/asl\_alphabet\_train"  test\_data = "/asl-alphabet/asl\_alphabet\_test/asl\_alphabet\_test"  folders = os.listdir(train\_data)  folders.sort()  **import** **os**  # Set the path to the train dataset directory  train\_dataset\_dir = root+"/asl-alphabet/asl\_alphabet\_train/asl\_alphabet\_train"  # Initialize a dictionary for storing the count of images in each class  class\_count = {}  # Iterate the subdirectories in train dataset directory  **for** class\_name in os.listdir(train\_dataset\_dir):  class\_dir = os.path.join(train\_dataset\_dir, class\_name)  # Check if the current item is a directory  **if** os.path.isdir(class\_dir):  # Count the number of images in the current class directory  num\_images = len(os.listdir(class\_dir))  # Store the count in the dictionary  class\_count[class\_name] = num\_images  # Print the number of images in each class  **for** class\_name, num\_images **in** class\_count.items():  **print**(f"Class '{class\_name}': {num\_images} images")  **import** **os**  # Set the path to the train dataset directory  test\_dataset\_dir = "/kaggle/input/asl-alphabet/asl\_alphabet\_test"  # Initialize a dictionary for storing the count of images in each class  class\_count = {}  # Iterate the subdirectories in train dataset directory  **for** class\_name **in** os.listdir(test\_dataset\_dir):  class\_dir = os.path.join(test\_dataset\_dir, class\_name)  # Check if the current item is a directory  **if** os.path.isdir(class\_dir):  # Count the number of images in the current class directory  num\_images = len(os.listdir(class\_dir))  # Store the count in the dictionary  class\_count[class\_name] = num\_images  # Print the number of images in each class  **for** class\_name, num\_images **in** class\_count.items():  **print**(f"Class '{class\_name}': {num\_images} images")  **def** **plot\_random\_img**(images=None):  fig=plt.figure(figsize=(**10**,**10**))  **for** i,folder **in** enumerate(folders):  fig.add\_subplot(**5**,**6**,i+**1**)  images=os.listdir(train\_data+'/'+folder)  ind=random.randint(**0**,len(images))  image=cv.imread(train\_data+'/'+folder+'/'+images[ind])  plt.imshow(image)  plt.title(folder)  plt.show()  plot\_random\_img()  **import** **string**  labels = []  letters = list(string.ascii\_uppercase)  labels.extend(letters)  labels.extend(["del", "nothing", "space"])  **print**(labels) |