American Sign Language Recognition using Computer Vision and Deep Learning

MSc Research Project
Data Analytics

Aayush Aggarwal
Student ID: x21232911

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aayush Aggarwal
Student ID:	x21232911
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	14/08/2023
Project Title:	American Sign Language Recognition using Computer Vision and Deep Learning
Word Count:	8402
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Aayush Aggarwal
Date:	14th August 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

American Sign Language Recognition using Computer Vision and Deep Learning

Aayush Aggarwal
x21232911

Abstract

Sign language recognition system helps to understand the hand gestures made by speech and hearing-impaired community that involves movement of fingers and different palm orientations. This framework has experienced significant growth in the field of computer vision and deep learning. The researcher has investigated various hardware and software approaches for accurate sign recognition. American sign language dataset was used to achieve the goal of this study. The author has performed exploratory data analysis to get insights into data and applied pre-processing techniques such as image resizing, smoothing, and re-scaling. Two deep learning models were implemented for this research, namely Convolutional Neural Network (CNN) and Residual Network 50 (ResNet50). A 2D CNN which consists of optimized hyper-parameters outperformed the other model and achieved an accuracy of 98.76%. A learning curve was also demonstrated for accuracy and loss which was considered during model evaluation.

1 Introduction

1.1 Background

Communication plays an important role when an individual has to express their thoughts, emotions, or opinions to one another. Different people adapt to different languages to pursue the same. However, difficulty arises with deaf-mute people and thus they rely on sign language to communicate. Different Sign Languages for different countries are the predominant form of communication for hard of hearing and speaking individuals. A few such languages include American Sign Language (ASL), Bengali Sign Language (BSL), and Indian Sign Language (ISL) Sharma and Singh (2020). On one hand where oral languages involve the mouth for communication, whereas sign languages require the use of hand gestures to convey some meaningful information. These sign languages differ from each other in terms of their diverse orientations of palm, fingers, and thumb which include gestures for digits, alphabets, and some special letters. However, traditional methods used by the deaf-dumb community include the use of hardware devices such as sensors and gloves Nimisha and Jacob (2020). These approaches have proven to be less meaningful in recognising sign language due to a complication in hand gestures. It also involves multiple arrangements of wires and is costly as well. But on the other hand, tremendous growth has been observed in machine learning and deep learning techniques which are entirely software-based and can offer more accurate and robust sign language recognitions.

1.2 Motivation

Rahman et al. (2019), states that the population of people with hearing disability have drastically increased by around 200 million from 2005 till early 2018 and this number is expected to be doubled by 2050. The main motivation behind this technical report was to break the communication gap for hearing-impaired communities and give them equal opportunities in every sector. However, the lack of knowledge in sign language proficiency led to a barrier in communication, which lead to limited opportunities and a lack of social interactions. Sign language recognition using deep learning was challenging in terms of capturing diverse positions of hand shapes and their movements. The 29 classes (A-Z, space, delete, and nothing) have different types of hand gestures which makes it complex for deep learning models. Earlier, human interactions were possible with the help of an interpreter, who used to convert hand gestures into meaningful sentences. But nowadays, the advancement of Artificial Intelligence has opened a new and more effective way of conveying one's opinions and thoughts. This has allowed them to engage with the broader society and unlock new opportunities in the field of education, employment, and healthcare. Sign Language Recognition will help deaf and dumb people to connect easily with medical professionals, thereby ensuring proper care for them. In addition to it, they can also have a quick understanding of lectures, which will eventually pave the way for better employment. These social impacts have led the researcher to focus on this area by using computer vision and deep learning techniques.

1.3 Research Question

How well can a CNN-based approach achieve accurate recognition of American Sign Language (ASL) hand gestures, and how does it compare to other existing methods?

This Research Question mainly focuses on the implementation of the Convolutional Neural Network (CNN) and its pre-trained model for accurately recognising the hand gestures of American Sign Language. In addition, the author also aims to perform a comparison between various hardware and software approaches used for sign language recognition.

1.4 Research Objective

The primary objective of this technical report is to implement a deep learning model using a computer vision for American Sign Language (ASL) dataset. To achieve this, the project undergoes an extensive use of computer vision to analyse and track hand gestures and finally create a deep learning model that can categorise different sign languages into different classes. This report employs different stages to achieve the proposed research question, thereby it is divided into various components and objectives, which are stated below in Table 1

1.5 Outline Of Research paper

The content of the technical report is divided into the following sections. Section 2 is Related Work, Section 3 is Methodology, Section 4 is Design Specification, Section 5 is Model Implementation, Section 6 is Evaluation, Results, and Discussion, and last Section 7 is Conclusion and Future Work.

Table 1: Overview of Research Objectives

Name	Description
Obj. 1	To critically review similar papers for Sign Language Recognition based on software and hardware techniques
Sub Obj. 1.1	Hardware based Sign Language Recognition
Sub Obj 1.2	Machine Learning based Sign Language Recognition
Sub Obj 1.3	Deep Learning based Sign Language Recognition
Obj. 2	To collect an American Sign Language dataset that represents a wide variety including hand orientations, lightning conditions, and different backgrounds
Obj. 3	To perform exploratory data analysis (EDA) on the dataset
Obj 4	To implement data pre-processing techniques
Obj 5	To build an efficient model that is capable of recognizing the hand gestures of American Sign Language.
Obj 6	To evaluate the performance of the model based on evaluation matrices.

2 Related Work

The use of Sign Language has emerged significantly over the past few years due to an increased number of people with hearing and speaking disability. Due to recent advancements in technology, Sign Language Recognition involves an intersection of machine learning, computer vision, and deep learning techniques. The main goal of SLR is to enable effective communication for hard-of-hearing and speaking individuals. Over the past few decades, many researchers have explored this field and have developed different ways for recognition and translation. However, there are many challenges in sign language detection that are observed by authors, and they are yet to be resolved. Firstly, complications arise in choosing adequate sign language among many languages which can capture a large audience and solve the problem in mass. Secondly, selecting the dataset that is prone to all factors and conditions such as the use of both hands for gestures, different lighting conditions, and the involvement of some background image in the frame. Also, choosing a suitable machine learning or deep learning model could be a big challenge as well.

In this technical report, the researcher proposed to divide the literature review into three sub-sections. The first section describes sign language recognition using hardware devices such as gloves, sensors, and Arduino. The second and third section critically reviews all the similar papers where different sign languages were recognised by using various machine learning and deep learning techniques. Most of the articles show the use of smart gloves that are used to collect the data, which is then transmitted to a computer for recognition and translation. With the technology change, various machine learning-based SLR techniques were presented. They had made the big transitions from a hardware-based approach to simply software use. In these approaches, authors examine various pre-processing and feature extraction techniques which help to determine the best-fit model using KNN, Random Forest, SVM, etc. Later, a more powerful technique called deep learning is introduced which can handle complex problems and is considered most significant in recognising sign languages. Overall, this chapter critically examines the evolution of the above three methods and provides a comparison between their results.

2.1 Review of Literature for Hardware based Sign Language Recognition

An artificial Intelligence-based smart glove called a triboelectric smart glove was used to detect sign language recognition, especially sentence recognition. Wen et al. (2021)introduces a novel approach that uses a smart glove that can interact with non-signers by understanding sign language. This glove was equipped with a variety of sensors that can detect hand gestures and movements including bending of fingers, wrist motion, fingertips, and palm interaction. The author proposes two methods for recognition namely, segmentation and non-segmentation that consists of words and sentences. The first method segments all the signals into portions of words, whereas no splitting of words was observed in the non-segmentation technique. These words are then trained through a convolutional neural network (CNN) model, with an adjustment in hyper-parameters to achieve higher results. The model achieved an average accuracy of around 86%. In addition to it, the authors used this glove to connect with the virtual world using AI signals that will help the deaf-dumb community. One of the major drawbacks that were seen in this study is the limited collection of datasets which might have caused the limited generalisation.

Saquib and Rahman (2020), proposes a system that can capture and recognise American Sign Language (ASL) and Bengali Sign Language (BSL) using data gloves. The authors in this paper extract the words and letters that are performed using hand gestures and then multiple letters are signalled in a sequence. This was performed using a glove that has some sensors attached to it, which are used to track the detection of bending, contact, orientation, and motion of the hand. Now, the data collected is passed to the Arduino microcontroller board, which was connected to a desktop using a USB port. The researchers were able to detect both the static and dynamic letters in the alphabet after splitting the dataset into 80-20%. An accuracy of around 96% was achieved by using an artificial neural network (ANN) with the help of K-Fold Cross-Validation. In this study, a detailed mechanism of system design and connectivity of smart gloves and computers is explained, but the involvement of users for longer periods of time to record the movements can cause inconvenience.

The main aim of the researchers Khomami and Shamekhi (2021) was to develop a system that helps non-signers to understand and translate the Persian sign language (PSL) into meaningful sentences for better communication. This study focuses on two types of sensors for gesture recognition, Inertial Measurement Unit (IMU) and Surface Electromyography (sEMG) which measures data related to motion by observing the user's hand and arm movements. Once the data is captured using two sensors, they are segmented and then features are extracted from each segment using the RELIEFF feature selection algorithm. This method extracts features such as variance, integration, mean, etc., along with a pre-processing technique where the data becomes synchronized, and relabelling is performed. Further, a few machine learning algorithm was applied to the dataset, where KNN outperforms with the highest average accuracy of 96% by using a 10-fold cross-validation technique. However, the arrangement of hardware devices and wires on the arm makes the system complex and can be uncomfortable. Also, the performance of the model can be hindered due to external noises.

Zhou et al. (2020) proposed a framework consisting of a machine learning algorithm that is assisted by wearable sensors. This hybrid approach mitigates the issue of communication between signers and non-signers. The author of this study converts American sign language into speech and offers a quick response time. The functioning of this system involves an installation of Wireless PCB and YSSA on the glove which can detect the motion of fingers and palm movements. All of the analog signals generated are converted into digital signals with the help of signal processing and then wirelessly transmitted to the terminal display for processing. PCA was applied to the dataset to extract the primary features and eliminate the redundant ones. A multi-class SVM algorithm was implemented, and the model attained an average accuracy of around 98%, along with a recognition time of less than 1s. Even though the researchers were able to manage a high recognition rate, the complexity and high cost of hardware devices cannot be neglected.

2.2 Critical Analysis of various Machine Learning based Sign Language Recognition

A machine learning based sign language detection is performed by Amrutha and Prabu (2021), which can interpret the signs and translate them. Sign language is the foremost mode of communication for hearing-impaired people, and this study will empower them. They will not be dependent on human translators for communication and a large population will be able to understand them. The authors use the camera to capture the hand movements, gestures and signs made by people, which will then be fed into the interpreter model. Further, a pre-processing technique was applied that will unblur the image and adjust the lighting conditions. Along with this, Colour-based and Edge-based segmentation was used. Besides this, feature extraction techniques called PCA, and convex hull were able to remove the redundant noise and prepare the data for modeling. KNN was used for classification which attained an accuracy of 65% on the testing dataset. The performance of this model can be improved by increasing the number of participants and using a different classification algorithm. Along with this, the issue of distance between the camera and the object should be also resolved.

Raghuveera et al. (2020) introduces an efficient algorithm that can recognise Indian Sign Language (ISL) gestures through Microsoft Kinect which can solve the problem of bad lighting conditions and object colour. A total of 4600 images were captured using it that consists of 140 static signs and this dataset was then divided into a ratio of 80:20. A hand segmentation is performed using K-mean clustering to speed up the process, with further addition of three methods (HOG, LBP, and SURF) for feature extraction. These approaches are used to detect the boundaries of hands and form a cluster and bounding box. These extracted features are then trained using a Support Vector Machine (SVM) where a model was able to identify around 71% of total hand gestures. A comparative analysis was also performed with state of art paper and a significant difference in accuracy was observed. The author proposes to improve the data dictionary and re-implement the model so that it can be useful in real-time applications.

An innovative approach was adopted by Athira et al. (2022) to understand single and double-handed static and dynamic hand gestures from live videos of an Indian Sign Language dataset. The main focus of the authors in this research was to minimise the effect of co-articulation in videos and develop a model that is trained on a large dataset and performed by many individuals. This study mainly comprises of three steps, pre-processing, feature extraction, and classification of hand gestures. During the pre-processing stage, videos are converted into a set of frames and later segmentation removes all the unnecessary regions from them. Further, the ROI algorithm is applied to extract the palm region. The second stage involves separating the relevant features such as hand shape, palm orientation, and average speed in the case of dynamic gestures. Finally, a SVM model is executed on the dataset which shows an accuracy of 91% for finger-spelling alphabets and 89% for dynamic words. However, researchers here provided light on real-world experience, but it can simultaneously cause an adverse effect due to the involvement of background objects and adversarial lighting conditions.

In this research, the authors intend to develop a model for hearing-impaired children which can enhance their educational background. Ibrahim et al. (2018) used a dataset that contains a set of 30 words that are used in day-to-day conversations during school. The aim of this paper was to translate the Arabic sign videos in the form of voice or text using four primary steps. The first stage involves the removal of hands from videos using accumulated difference (AD) which identifies the colour of skin based on the facial colour. The next step is head and hand tracking using the extracted skin blobs. These features are then used to create a feature trajectory. The authors here focused on extracting geometric and statistical features that include the hand projections and their shapes and this helped them to attain an overall recognition rate of about 97%. However, a major drawback observed in this study is the low classification rate for similar gestures.

2.3 Critical Analysis of various Deep Learning based Sign Language Recognition

Shahriar et al. (2018) presents a study on real-time recognition and categorisation of the American Sign Language (ASL) dataset based on skin segmentation and image category classification. The authors chose two methods for identifying colour on the objects, i.e., RGB and YCbCr, which is then followed by the removal of the background from the frame. Mask Filtering and Bounding Box step during the pre-processing helps in denoising of external noises and creates a box around each skin mask by identifying their shape. A total of 150 images were captured for each of the four classes (face, palm, A, V) and then fed into Convolutional Neural Network (CNN) for training and testing. The images are classified and categorised based on the classifier, which achieved an average accuracy of around 94% on the testing dataset. The researchers performed a comparative analysis of their result with other papers, and establish the novelty based on their real-time recognition. Though, the result could have been improved by the integration of large data with diverse conditions.

This research paper used a deep learning algorithm to detect and categorise the IISL2020 dataset for the speaking or hearing-impaired community. The authors in this report have created their own dataset with the help of 16 participants that consist of 1100 video samples in normal condition. Kothadiya et al. (2022) proposed a methodology that is a combination of LSTM and GRU and achieved an accuracy of 97%. The researchers extracted the feature vectors and passed them to the model after splitting the dataset into an 80:20 ratio of training and validation sets. The first layer comprises the combination of the above two models as they have the tendency to remember past frames by using the gates. Their model has a batch size of 16 and a softmax activation function. The dropout is also used to reduce the problem of overfitting and improve the model's performance. Finally, K-Fold cross-validation is used with 10 different combinations, but the author's intent to collect the dataset under a controlled environment could be a major limitation.

The authors Mittal et al. (2019) in this study introduce a modified long short-term memory (LSTM) model that can recognise a continuous sequence or gestures of Indian sign language. It is a non-verbal form of communication for the speech and hearing-impaired community that uses facial expressions, hand and mouth movements, and gestures. An ISL dataset is chosen for this research project that comprises of 35 unique sign words which altogether form 942 sentences. The researchers here proposed a new way for human-computer interaction where a leap motion sensor is used to capture the signs. The data detained through them is then stored and processed using the sensor's API. A pre-processing technique of feature extraction and size normalisation is applied where 12 dynamic features of both hands are extracted and are then scaled to make them uniform. These features are then trained using LSTM with ReLU activation function and Adadelta optimizer. After training, it was observed that the model achieved an average accuracy of around 72% and 89% for signed sentences and isolated signed words. The performance of the model could be increased if researchers would have used more training data.

The authors propose a novel approach for Bengali sign language recognition that is entirely based on a deep learning algorithm. A total of around 1100 images were created during this study that comprises 37 static Bengali signs. This dataset is formed under diverse lighting conditions, different skin colours, and the involvement of some background objects. Due to a limited dataset, images were rotated, flipped, and zoomed before training so that the data becomes generalised. The model architecture consists of a pre-trained VGG16 layer, followed by a Convolutional layer and a max-pooling layer. Finally, this network is fine-tuned to avoid overfitting and then connected to a fully connected layer that consists of 256 nodes. Hossen et al. (2018)were able to achieve a validation loss of 0.35 and a validation accuracy of around 85%, which is considerably high due to an inadequate dataset. The researchers aim to establish better results with the help of a more powerful deep learning framework.

2.4 Identified Gaps and Conclusion

To conclude, the above section critically reviews the recognition and translation of various sign language datasets using different approaches. These methods include a hybrid approach of hardware and software applications and a purely software based approach that includes different machine learning and deep learning algorithms. These processes have interpreted different hand gestures based on diverse conditions and proved to be successful in their domain. However, there are a few limitations observed in these studies which include the hardware costs, complex structures, and external interference. In addition, some of the researchers faced the issue of large data integration, the problem of overfitting, and the lack of pre-processing techniques. So, this technical report considered these drawbacks and the author successfully integrated them into the model.

Objective 1, along with three sub-objectives (1.1, 1.2, and 1.3) had been achieved, which also partially answers the proposed Research Question (1.3). The next Section in this technical report describes the research methodology.

3 Methodology for Sign Language Recognition

Sign Language is a non-verbal form of communication that is used by speech and hearing-impaired people to communicate with non-signers. This helps them to express their emotions, and opinions, and improve their language skills. They will be able to gain knowledge in various fields like education, private and government sectors and will have equal access to information as the rest of the world. This will not only make them independent but will also help them build a cultural identity. Different sign languages are used in different countries like Indian Sign Language (ISL) in India, Bangla Sign Language (BSL) in Bangladesh, American Sign Language (ASL) in the United States and many more. This section of the technical report will give a brief description of all the equipment and technologies used in the model building, an overview of the dataset and its exploratory data analysis, data pre-processing techniques applied, along with the evaluation matrices for sign language recognition. Below Fig.1 gives an overview of the methodology starting from the data collection; to its pre-processing; and finally building an efficient model.

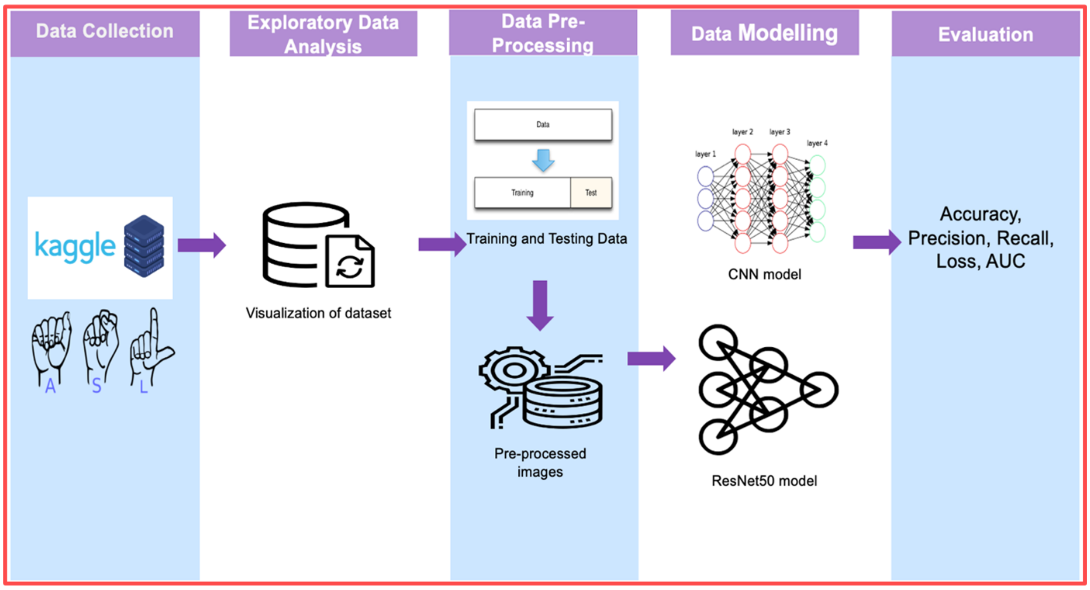

Figure 1: Overview of the Methodology

3.1 Data Acquisition

The dataset used in the model implementation is gathered from an open-source platform called Kaggle which has obtained the American Sign Language (ASL) dataset from National Institute on Deafness and Other Communication Disorders (NIDCD). This dataset consists of around 200 hundred thousand images of alphabets (A-Z) and three other special characters (Space, Delete, and Nothing). These three extra characters produce a uniqueness which is really significant during communication. The images in this dataset are captured in 29 different folders, each representing one class. The dataset consists of hand gestures that symbolise different characters, each with a unique orientation of fingers and palms. It also provides diversity in terms of adverse lighting conditions, its subjection to different skin colours, and the inclusion of varied background objects which is evident in below Fig.2 as well. The major challenge faced during the project was the collection of large dataset that includes diverse conditions, which can incorporate deep learning models and mitigate the issue of biases. The image data obtained from Kaggle is downloaded and stored on the desktop, which is then transferred to Google Colab. A link is established between them by mounting the Google Drive with Colab so that the images are uploaded and can be further used for pre-processing and modelling.

The collection of the multi-characteristic American Sign Language hand gesture dataset has fulfilled objective 2.

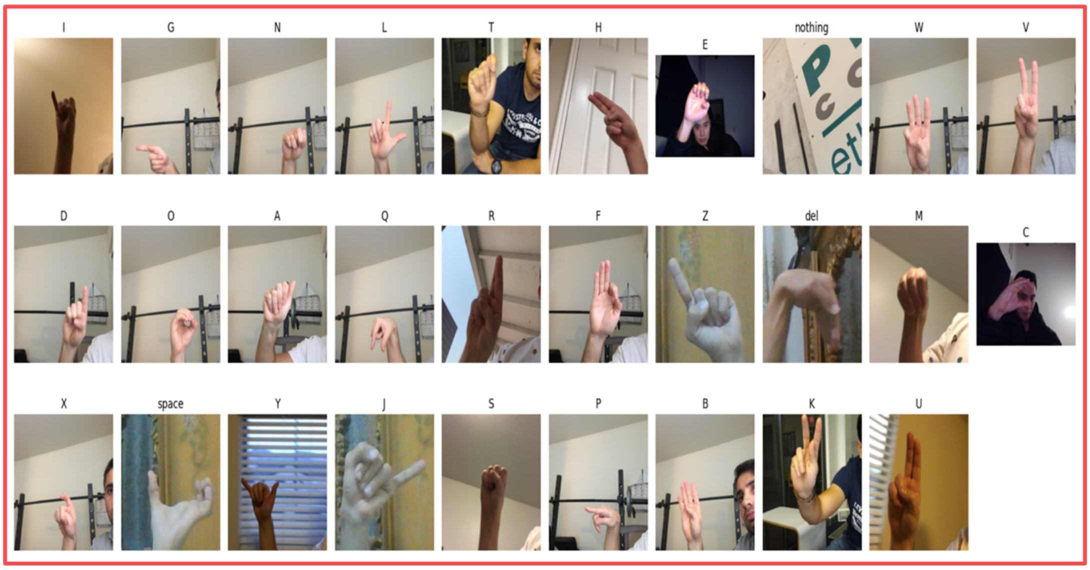

Figure 2: Sample images of ASL dataset

3.2 Exploratory Data Analysis

The first stage that provides statistical and graphical analysis of the dataset chosen for implementation is exploratory data analysis (EDA). This helped the researcher to find the structure of the dataset such as the total number of images present for each alphabet, any discrepancy like the excess images of one alphabet or class over another class, and severe irregularities in the dataset. EDA also assisted the author in selecting the appropriate model, and decision-making. In addition, it has not only been directed towards data analysis but has also helped to establish the proper resource allocation required in pre-processing. The insights gained from visualisation is the presence of generalised data which helped the researcher to interpret, manage, and analyse the data rapidly. The dataset collected from Kaggle has two folders which have training and testing images. But it was further split into validation sets which helped to evaluate the performance of a model during the training time. Fig,3 represents a total distribution of 29 classes (A-Z, space, delete, and nothing) for each set.

The accomplishment of objective 3 which is listed in Table 1 (1.4) has been made possible by this subsection.

3.3 Data Pre-Preprocessing

Once the data analysis had been performed and all the precautionary measures had been considered, the author prepare the dataset for pre-processing. Firstly, the dataset is split into training, validation, and testing sets in a ratio of 60:20:20 for the implementation of the first model. In this next step, the American Sign Language image dataset is pre-processed and managed for a deep learning framework using a TensorFlow API.

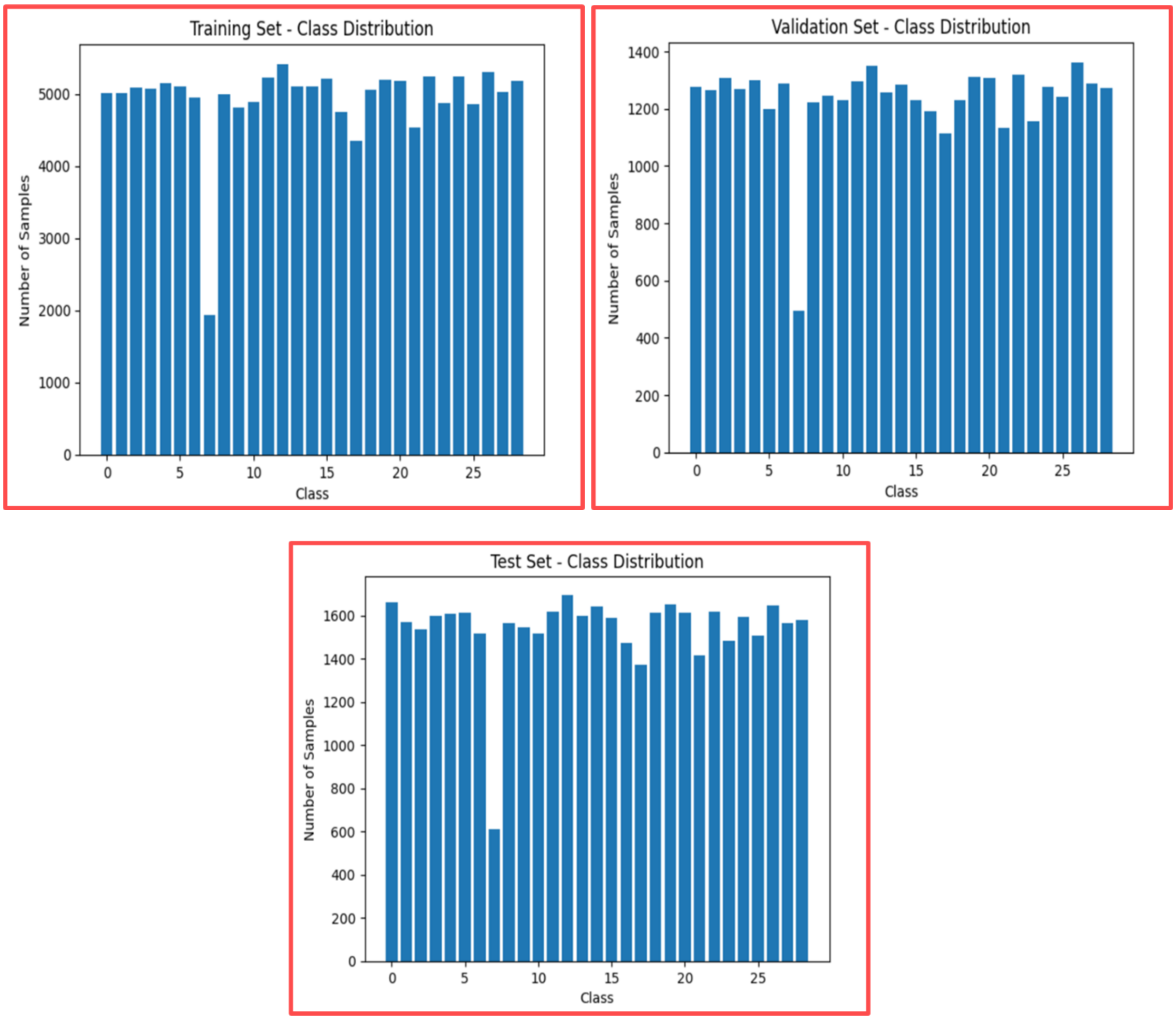

Figure 3: Distribution of Training, Validation, and Testing sets

It is highly efficient and capable of handling large amounts of data, with optimised performance. Here, TensorFlow's functions are applied which takes the image path and their corresponding label as inputs. These images are firstly read from the specified path which decodes the 3 colour channels (RGB) in the images using tensors image. Further, pre-processing takes place in the form of resizing where all the images are resized into the same format of 64x64 pixels. Resizing allows the images to have a consistent size which is necessary for a deep-learning neural network. The pixel values of images are then normalised by dividing them by 255.0 and scaling them to the specified range of [0,1]. This will help to stabilise the training process and give the same magnitude to all the images.

Now, the TensorFlow training dataset is sliced to form two input data (X_train) and (y_train) which consist of the image paths and their labels. After this, all the image files are loaded and pre-processed individually before they are fetched for modeling. Followed by this, it involves autotuning as well which allows for parallel processing of all images. The next step of pre-processing includes the shuffling of images so that randomness is created in the dataset. In addition to it, a buffer size parameter would control the random sampling during the shuffling process. A batch size of 32 was set initially which is useful for training the model in small batches. It takes into account the total number of images that are processed simultaneously in each iteration. In the last stage of pre-processing, the author applied an optimisation technique that utilizes both CPU and GPU or TPU at the same time. This happened by allowing data loading and model computation to work simultaneously; when GPU or TPU is performing the pre-processing, the CPU can load the next batch of data at the same time resulting in increased efficiency. By using this, the training speed of the data is increased and the idle time is minimal. A similar procedure is then followed for validation and testing sets. But shuffling is not required in these sets, so the author discarded this step.

For the pre-processing of the second model, the author first created separate folders for all 29 classes (A-Z, space, delete, and nothing) of the validation dataset and then moved the random images from the training dataset into these folders. The dataset is split into a ratio of 80:20 which is stored in a different directory and then pre-processing techniques were applied. Here, a function called 'ImageDataGenerator' was created. This function performs data augmentation on original images that include some transformations like rescaling, rotation, zooming, height and width adjustments, flipping, etc. All the training images in the dataset are rotated up to 20 degrees and pixel values are normalised to 0 and 1. Further, a horizontal or vertical shift of 10% is applied, along with zooming in or out of 10%. The images can be horizontally flipped, if required during the training and their pixel values are adjusted to the nearest pixel. A new generator function called 'flow_from_directory' load the images from a specified directory and accounts for batch size and their corresponding labels. This process is then repeated for the validation dataset, which is suitable for training deep learning models.

Objective 4 has been achieved, and the next sub-section presents the model implementation for sign language recognition.

3.4 Model Implementation

Sign Language Recognition can be performed by using various machine learning and deep learning algorithms. As already observed in the Literature review section, deep learning methods outperformed traditional methods due to their feature learning capacity and their ability to handle complex relationships. After critically analysing all the related papers, the researcher uses Convolutional Neural Network (CNN) and ResNet50 to implement sign language recognition. Both models have shown tremendous performance in computer vision related problems. CNN tends to recognise shapes from different hand gestures which involve complex hand movements and classifies them into 29 classes. The images are fed onto the convolutional layer in the form of grid-like data, thus enabling it to learn different patterns. On the other hand, ResNet50 which is a pre-trained CNN model includes residual networks. This algorithm skips some connections which allows the model to bypass some layers, thus making the training process smooth. Further in-depth architecture and process flow for both deep learning models are explained in Section 5.

3.5 Model Evaluation

Once the model is built, the next step is to assess its effectiveness. This process is critical in a deep learning neural network since it helps to evaluate the performance of a trained model and find its efficiency when explored to the unseen testing data. There are different parameters used by the author in both modeling processes. Some of the matrices include Precision, Recall, and AUC. Along with these, training and validation losses are also calculated. The researcher displays the accuracy and loss plots for training and validation in order to identify the signs of overfitting. These plots also help to find the impact of different hyper-parameters used while training the model. Besides these, training, validation, and testing accuracies are also measured which is commonly used in these types of classification problems. A detailed description through graphs and tables for both models is mentioned in Section 6 of this technical report.

3.6 Conclusion

All the steps performed during the implementation of sign language recognition are explained thoroughly in this section. The model achieved reliable results by using deep learning based CNN and ResNet50 which were able to recognise hands and classifies them into different classes. This was possible due to the appropriate selection of dataset, exploratory data analysis, detailed pre-processing techniques, and befitting neural network models, thus achieving objectives 2, 3, and 4 stated in Table 1. However, during the development of the project, the author faced difficulties in capturing different hand orientations, especially the ones having similar hand shapes and sizes such as M, N, and U. Also, some of the letters include overlapping of fingers and some are closely spaced together which makes it difficult to recognise few signs.

A design process flow or architecture involving three different layers for sign language recognition is illustrated in next Section 4.

4 Design Specification Flow for Sign Language Recognition

The design flow for American Sign Language Recognition using deep learning based CNN and ResNet50 model is depicted using three-layer architecture which consists of a Business Presentation layer, Logic layer, and Data Persistent layer as shown below in Figure 4. The first layer depicts the visual representation of input data for exploratory data analysis, information on pre-processed images, and the results obtained during evaluation. These tasks were achieved by using a powerful data visualisation library called Matplotlib. The logic layer is the critical layer in system flow as it defines the core application of both deep learning models, Convolutional Neural Network (CNN) and ResNet50. These models perform the efficient recognition and categorisation of sign language by using Keras and TensorFlow. They are powerful deep learning frameworks that are essential for building deep learning neural networks. The third data persistent layer includes the data collection from Kaggle, loading the images to a Google Colab cloud platform, splitting, and pre-processing them, and performing feature extraction. This pre-processed data is then sent back to the previous logic layer for model implementation.

In summary, this is a three-tier process that involves different layers, starting from visualisation through Python libraries, to implementing an efficient model that is capable of recognising accurate sign language, to loading and pre-processing the image data using the data augmentation method. This is a cyclic process where the logic layer receives the data created in the third layer for model implementation, which is then again sent to the first layer for presentation.

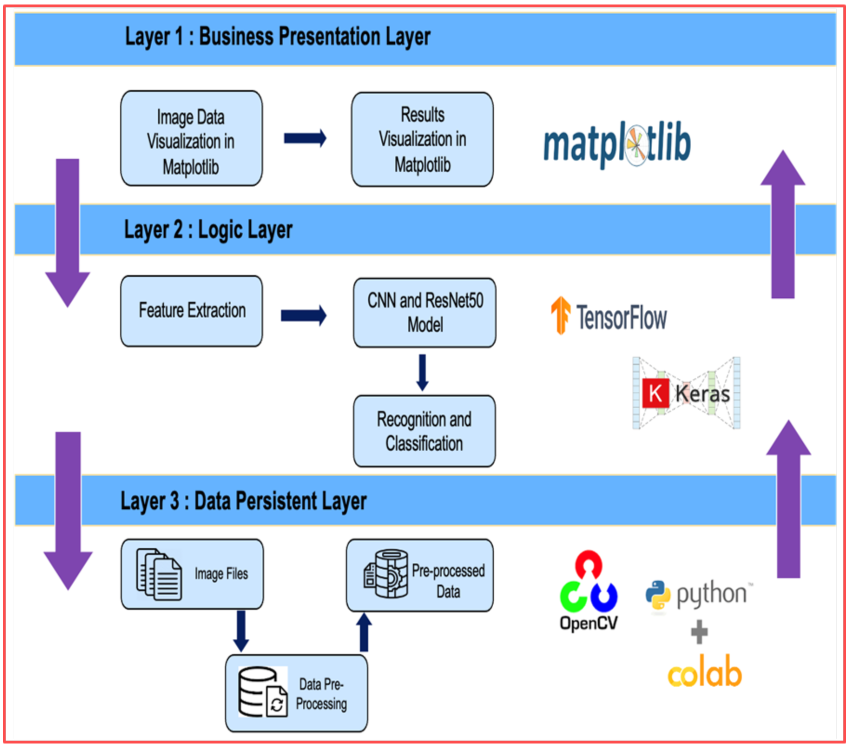
Layer 1 : Business Presentation Layer
Image Data Visualization in Matplotlib
Results Visualization in Matplotlib
matplotlib
Layer 2 : Logic Layer
Feature Extraction
CNN and ResNet50 Model
Recognition and Classification
TensorFlow
Keras
Layer 3 : Data Persistent Layer
Image Files
Pre-processed Data
Data Pre-Processing
OpenCV
python
colab

Figure 4: Design Process Flow for Sign Language Recognition

The core feature extraction and implementation of CNN and ResNet50 deep learning models are explained thoroughly in next Section 5.

5 Model Implementation for Sign Language Recognition

This section discusses the tools and technologies used in the research project, along with the specification of feature extraction and model implementation. The author has used Google Colab which is an open source cloud platform for writing and executing the code. Python language provides a wide variety of tools and libraries such as OpenCV, Keras, and TensorFlow that were used for sign language recognition and classification. OpenCV is a powerful computer vision library that was used to analyse hand gesture images and supports feature extraction. TensorFlow is an open-source framework that is used for training and building deep learning models, while Keras is an API that is built on the architecture of TensorFlow, and it simplifies the process of feature extraction and model building. Other than these, ImageDataGenerator was also used to pre-process the image data and Matplotlib for generating visualisations and plots. This section is further divided into two sub-sections, each explaining the feature extraction and model implementation using Convolutional Neural Network (CNN) and ResNet50. These models are chosen for sign language recognition as they can handle large image datasets and excel in capturing features such as hand movements, shapes, and orientation.

5.1 Feature Extraction and Model Construction using CNN

In this deep learning based Convolutional Neural Network (CNN), the author imported two modules using Keras and TensorFlow, namely 'layers' and 'models'. The layer module provides various types of layers to be used in model construction and the models module is used to define the neural network. The main aim of the CNN model is to accurately recognise American sign language and classify them into different classes. To avoid the problem of overfitting, the researcher has applied an 'Early stopping callback' function. It takes two parameters, where first one is 'monitor', which will check for the loss during the training time. This parameter will observe the loss and stop the training if there is no improvement of a certain continuous number of epochs. The next parameter is 'patience' where a value of 3 has been passed, this signifies that the training time of the model will stop if the value of the loss is the same for 3 consecutive epochs. Rahman et al. (2019) also shows the use of an early stopping class where the training time was halted if there is no improvement of validation loss for thirty consecutive epochs. This also helped to save the training time of the model.

In the next step, the CNN model is defined which consists of three major layers, i.e., convolutional layers, pooling layers, and dense layers. A series of convolutional 2D layers were applied with three different filters (32, 64, and 128), followed by a ReLU activation function. These convolutional layers extracted the features from the images by adding some filters to them and prepared them for modeling. It takes an input shape of 64x64 pixels and 3 colour channels (RGB) which were formed during the pre-processing stage. This activation function captured complex patterns and features in the image data. Followed by every 2D Convolutional layer, the author applied a 2D MaxPooling layer with a (2, 2) pooling window that preserves only the most important features from the images. After this, a Flatten layer is applied which reduced the dimensions of all images from 2D to 1D vector. Finally, two fully connected dense layers are added with 128 and 29 units each. Here, the second dense layer takes a total number of classes as the units and the author included a softmax activation function to predict the probability of each class. This model is then compiled using an Adam optimizer and two evaluation matrices, accuracy and Top K accuracy are specified.

Finally, the neural network CNN model is trained using a 'fit' function. This model contains training and validation data, with a batch size of 32. An early stopping callback is provided during training. The model is trained for 5 and 10 epochs to monitor the entire training dataset. The next sub-section explains the feature extraction and modelling using ResNet50.

5.2 Feature Extraction and Model Construction using ResNet50

Residual Network 50 (ResNet50) is a pre-trained CNN model that can handle deep neural networks and contains skip connections that pass the information to many layers. Some Python libraries for Keras, TensorFlow, and ResNet50 are imported for modeling. Firstly, ResNet50 is loaded with weights of the ImageNet dataset and consists of input shapes with 224x224 pixels and 3 colour channels. A sequential model is created which consists of three 2D convolutional layers with 32, 64, and 256 filter sizes and ReLU activation function. These layers are followed by 2D MaxPooling layers that reduced the dimensions of image features. A flattened layer is also included, along with a dense layer of 512 units. Now, the author added a dropout rate of 0.5 to reduce the problem of overfitting and finally, a dense layer is added with a softmax activation for identifying 29 different classes of hand gestures. A similar approach was adopted by Rathi et al. (2020) for fingerspelling American Sign Language recognition.

A master model is created by adding a pre-trained ResNet50 model as input with a sequential model. This combined model is then compiled using the Adam optimizer and loss function which computes evaluation matrices such as accuracy, precision, recall, and AUC. Finally, this combined model is trained using training and validation datasets for 5 epochs. A batch size of 32 samples is processed in each epoch for both sets. A model summary was also displayed that shows the number of layers, shapes, and parameters.

5.3 Conclusion

Objective 5 has been achieved by successfully implementing the Convolutional Neural Network (CNN) and ResNet50 models for sign language recognition. Different kinds of modifications were performed for various parameters and many layers were added for feature extraction. The model performance was assessed by using evaluation matrices and plots of accuracy and loss for the training and validation dataset which is explained and depicted in Section 6.

6 Evaluation, Results and Discussion for Sign Language Recognition

Convolutional Neural Networks (CNN) and ResNet50 deep neural networks were used to implement American Sign Language recognition. There are different parameters used for feature extraction in both models, with a different number of layers. These models are compiled using an optimizer and trained using training and validation datasets. The results were then evaluated on a testing dataset which predicts the test loss, test accuracy, and recall. The visualisation plots of training and validation accuracy as well as for losses for each epoch were also depicted and the same is presented in the following sub-sections (6.1 and 6.2). A brief comparison of the author's best-fitted model with other existing models was discussed (6.3), along with the findings that were discovered in this research. The sub-section 6.4 will also discuss limitations and some modifications that should be applied to achieve better results.

6.1 Experiment 1 - CNN Model

A 2D CNN model was implemented for recognising sign languages performed by hand gestures. In this model, three convolutional layers along with three maxpooling were added. Two dense layers were also included for multi-class classification, along with an Early stopping callback method to solve the issue of overfitting. This model is trained on 5 and 10 epochs with a batch size of 32 samples. Further, the CNN model performance is measured by evaluating the testing dataset using the 'module. evaluate' function, which calculates loss, accuracy, top-k accuracy, and recall. The model shows a testing accuracy of 98.30% on 5 epochs. On the other hand, there is an accuracy of 98.75% when tested on 10 epochs. Here, Top-k accuracy says the correct predictions of classes if the prediction is made among the top k choices. Thus, a model shows better results in predicting the accurate class. A binary recall parameter will take the value of a specific class as a positive class and then determine the results. Table 2 shows the results of these parameters with 10 epochs which is the best-fit model.

Table 2: Evaluation matrices for CNN model

Accuracy	Loss	Top-k Accuracy	Binary Recall(class 10)
98.76%	0.083	99.86%	99.28%

The above results show the high efficiency of the CNN model with a loss value of less than 1. After evaluating the results, plots of loss and accuracy were visualised using a Python library called Matplotlib. The below Fig. 5 shows the plot of training and validation accuracy where the value of training and validation accuracy is increasing with each epoch. However, validation accuracy dropped slightly at epoch 5, but it then increases gradually. On the other hand, the second plot depicts the training and validation loss which indicates depreciation at each epoch. These graphs indicate the absence of overfitting in the CNN model.

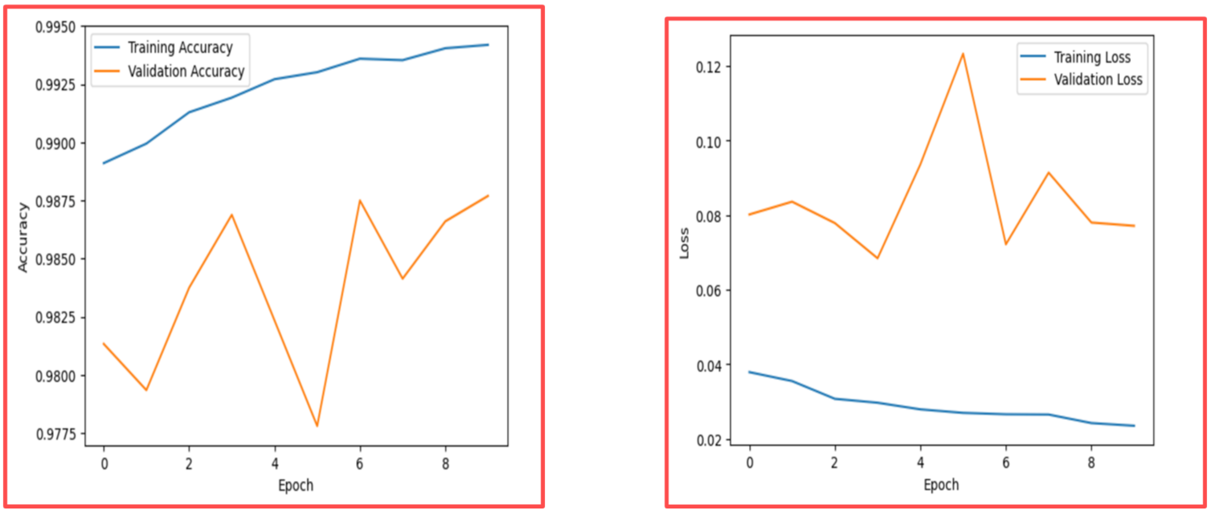

Figure 5: Plot Of Accuracy and Loss for CNN model

6.2 Experiment 2 - ResNet50 Model

A pre-trained CNN model called ResNet50 was implemented for this classification problem. A combined model is created with a combination of the base model and sequential model. This convolutional model has three convolutional and maxpooling layers. The model is then compiled using Adam optimizer and is trained on 5 epochs with a batch size of 32 samples. Further, a model is trained using training and validation data which will process the learned features and its result was evaluated based on different evaluation matrices such as accuracy, precision, recall, and AUC. Below Table 3 shows the testing results of each parameter with 10 epochs.

Table 3: Evaluation matrices for ResNet50 model

Accuracy	Loss	Precision	Recall	AUC
91.70%	0.26	94.48%	89.88%	99.61%

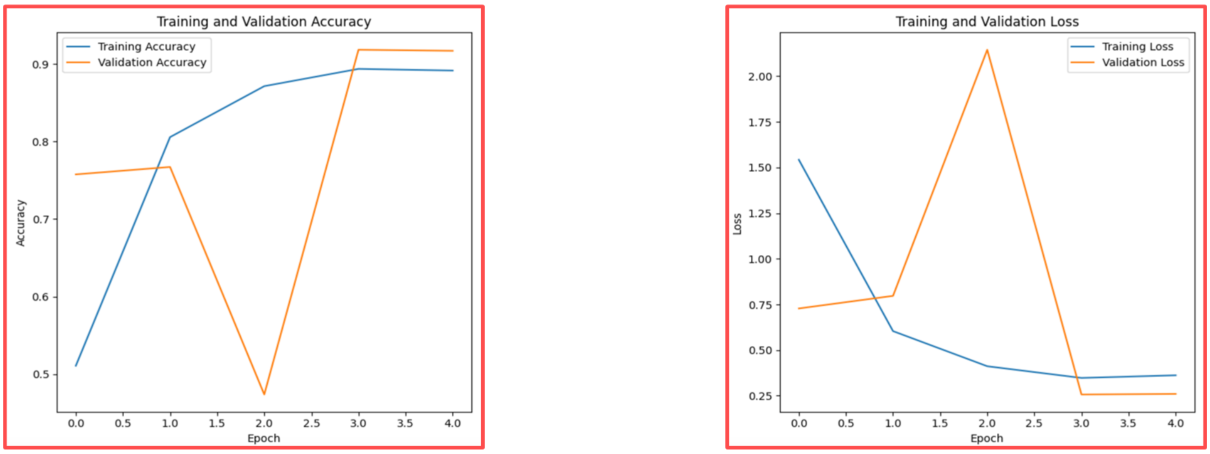

Figure 6: Plot of Accuracy and Loss for ResNet50 model

The above ResNet50 model shows good results, but its efficiency is less as compared to the CNN model. Now, the author implemented an accuracy plot for training and validation sets which shows a slight issue for validation accuracy at epoch 2, but then its value sharply increased till the last epoch to 91.70%. While the value of training accuracy was increased with an increase in epochs. The second plot shows the visualisation of training and validation loss. A similar trend was observed where the value of validation loss was highest at epoch 2 and then it decreased to 0.26 at epoch 5. A training loss is continuously decreasing at every epoch.Fig. 6 shows the visualisation plots for both.

6.3 Comparison of Developed Model with Existing Models

The author has successfully implemented both models for accurate recognition of hand gestures for American Sign Language and evaluated their performance based on evaluation matrices. This has accomplished the last Objective 6 of this technical report. However, Convolutional Neural Network (CNN) outperforms Residual Network 50 (ResNet50) in terms of accuracy and loss. This was verified by plotting visualisations and comparing them. A critical review of articles for sign language recognition using different machine learning and deep learning models was performed in Section 2. Table 4 shows the comparison of a few related papers with the author's implemented approach for this project.

6.4 Discussion

The main purpose of this research was to accurately predict the gestures such as hand movements, palm orientation, and finger positions for American Sign Language and classify them into different classes (A-Z, space, delete, and nothing). The dataset consists of hand images that are explored with diverse conditions which makes it suitable for model implementation. The modelling was performed using two deep learning based neural networks, namely, Convolutional Neural Network (CNN) and Residual Network 50 (ResNet50). A 2D convolutional network was created that contains three convolutional layers and three maxpooling layers for feature extraction. These layers add some filters to the pre-processed images and prepare them for modelling. Different dense layers were also added and used an activation function for predicting a multi-class classification. This

Table 4: Comparison of developed model with existing model

Author	Dataset	Feature Extraction	Model Name	Accuracy
Amrutha and Prabu (2021)	Random images	PCA, Convex hull	KNN	65%
Hossen et al. (2018)	BSL	Data Augmentation	CNN	85%
Kothadiya et al. (2022)	IISL2020	Hand gesture and edge detector	LSTM and GRU	97%
Shahriar et al. (2018)	ASL	RGB, YCbCr, Mask Filtering, Bounding Box	CNN	94%
Raghuveera et al. (2020)	ISL	HOG, LBP, SURF	SVM	71%
Developed model	ASL	Resizing, Normalisation	CNN	98.76%

model was able to achieve 98.76% accuracy and 0.083 loss. The second pre-trained ResNet50 model creates a base model with weights of ImageNet. This model is combined with a convolutional model that has three convolutional layers and three maxpooling layers, along with a ReLU activation function. A flatten and dense layer is also added. The master model that is created used ResNet50 as input and compiled the model using the optimizer. This model shows an accuracy of 91.70% and 0.26 loss. A visualisation plot of accuracy and loss for both models are constructed and the results are justified. Lastly, a comparison has been made between the author's most efficient developed model and the related models.

Both models performed reasonably well in predicting the gestures and classifying them into different categories. These models are selected for this classification problem as they can easily handle large image datasets and their feature learning capacities are high. They can detect the patterns in the image data such as different angles, and positions and can adjust to diverse sets of data. Thus, these models have shown prominent results in the field of computer vision and deep learning. However, in this research project, the performance of ResNet50 was less as compared to the CNN model. The author had implemented the ResNet50 model with 10 epochs to improve the efficiency of the model but faced runtime errors due to the maximum utilization of GPU and system RAM. The researcher could have made some changes in pre-processing techniques in such a way that images are more fine-tuned and take less time for training. Thus, some improvements in pre-processing were required to achieve higher results. However, the CNN model can also be implemented on other sign language datasets as performed by

Pigou et al. (2015) and Huang et al. (2015) with low lighting conditions, different types of objects in the background, and complicated shapes to detect its performance. The main challenges faced during the research project were to implement deep learning models that can work well in every condition and apply pre-processing techniques with different hyper-parameters.

There are many technical skills learned by the author during this research project. These include the use of Computer Vision to pre-process and analyse the image data, extract features, and apply data augmentation methods. In addition, this also helped the researcher to gain knowledge about the Deep Learning models and their frameworks such as Keras and TensorFlow. The learning outcomes also comprise handling large image data, applying various pre-processing techniques, and interpreting the model. Besides these, the author's critical thinking ability has also increased. There can be various ethical consideration and potential biases that can be considered into this project, especially the cultural sensitivity and respect for deaf and dumb individuals. It also preserves the diversity of sign languages by considering the data from diverse background.

6.5 Conclusion

Convolutional Neural Network (CNN) and Residual Network 50 (ResNet50) deep learning models have been successfully implemented on the American sign language dataset. CNN has shown better results with an accuracy of 98.76% and 0.083 loss. Objective 6 has been achieved by evaluating the robustness of the model on different evaluation matrices. Also, the author's proposed Research Question (1.3) has been completely answered. The final conclusion and future work are explained in Section 7.

7 Conclusion and Future Work

The main aim of this technical report was to fulfil the research question by implementing a robust model that can most accurately recognise different patterns and movements in hand and provide a comparative analysis of different hardware and software approaches for sign language recognition. There are various research objectives stated in Table 1 (1.4) and the author has successfully answered all of them, which also completes the proposed research question (1.3). Sign Language Recognition using deep learning has resulted in the enhancement of the field of practice which includes a diverse dataset, intriguing pre-processing techniques and high accuracy for both models. To start with, the author has first provided a critical review of all the related articles. The first section (2.1) of the literature review describes the use of devices such as smart gloves, Arduino, and sensors with some machine and deep learning techniques for sign language recognition. Whereas the next two Sections (2.2 and 2.3) give a description of entirely software-based approaches. There are different datasets and pre-processing techniques such as convex hull, mask filtering, data augmentation, etc, used by authors which were critical for model implementation, thus achieving objective 1 and sub-objectives (1.1, 1.2, and 1.3). It was discovered that the problem of classification requires a suitable choice of models and appropriate hyper parameters to accurately recognise gestures.

Followed by this, the next section 3 introduces the system methodology that includes data collection, data pre-processing, modelling, and evaluation, which fulfils the next research objectives (2, 3, and 4). The author has further implemented the model using two deep learning algorithms, namely, CNN and ResNet50. These models utilized different approaches for feature extraction and model building and their efficiency was measured by various evaluation matrices. Convolutional Neural Network (CNN) has more accurately predicted hand gestures with 98.76% accuracy and a loss of 0.083. Thus, accomplishing objectives 5 and 6. During the implementation of this research project, the author faced the issue of limited computational resources, which increased the training time of the models and sometimes resulted in runtime errors. Apart from this, sign language recognition has many implications, such as it can benefit deaf and hearing-impaired people to communicate with non-signers and support their education and employment opportunities. This can also ensure proper healthcare facilities for them and promote independent living.

For future work, a highly efficient deep learning model can be implemented that supports recognition of multiple sign languages which involves the use of diverse hand movements and facial expressions. The achieved accuracies in sign language for both models signify the recognition percentage of all 29 classes collectively. Its potential impact in the real world would help deaf and dumb individuals to most accurately convey their thoughts and feelings to non-signers. Additionally, the author can work on two follow-up project works for better live conversations. The first one includes real-time recognition of signs where a deep learning model will be able to predict and simultaneously translate the sign into text format. Another approach involves the continuous prediction of words and sentences by using a video interface. This will help the signers to communicate effectively with non-signers.

Acknowledgment

I would like to express my sincere gratitude to my supervisor Dr. Catherine Mulwa for her continuous support and mentorship during the course of the entire research project. The valuable insights and regular feedback have helped me to accomplish all the goals.

References

Amrutha, K. and Prabu, P. (2021). Ml based sign language recognition system, *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, IEEE, pp. 1–6.

Athira, P., Sruthi, C. and Lijiya, A. (2022). A signer independent sign language recognition with co-articulation elimination from live videos: an indian scenario, *Journal of King Saud University-Computer and Information Sciences* **34**(3): 771–781.

Hossen, M., Govindaiah, A., Sultana, S. and Bhuiyan, A. (2018). Bengali sign language recognition using deep convolutional neural network, *2018 joint 7th international conference on informatics, electronics & vision (iciev) and 2018 2nd international conference on imaging, vision & pattern recognition (icIVPR)*, IEEE, pp. 369–373.

Huang, J., Zhou, W., Li, H. and Li, W. (2015). Sign language recognition using 3d convolutional neural networks, *2015 IEEE international conference on multimedia and expo (ICME)*, IEEE, pp. 1–6.

Ibrahim, N. B., Selim, M. M. and Zayed, H. H. (2018). An automatic arabic sign language recognition system (arslrs), *Journal of King Saud University-Computer and Information Sciences* **30**(4): 470–477.

Khomami, S. A. and Shamekhi, S. (2021). Persian sign language recognition using imu and surface emg sensors, *Measurement* **168**: 108471.

Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A.-B. and Corchado, J. M. (2022). Deepsign: Sign language detection and recognition using deep learning, *Electronics* **11**(11): 1780.

Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R. and Chaudhuri, B. B. (2019). A modified lstm model for continuous sign language recognition using leap motion, *IEEE Sensors Journal* **19**(16): 7056–7063.

Nimisha, K. and Jacob, A. (2020). A brief review of the recent trends in sign language recognition, *2020 International Conference on Communication and Signal Processing (ICCSP)*, IEEE, pp. 186–190.

Pigou, L., Dieleman, S., Kindermans, P.-J. and Schrauwen, B. (2015). Sign language recognition using convolutional neural networks, *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*, Springer, pp. 572–578.

Raghuveera, T., Deepthi, R., Mangalashri, R. and Akshaya, R. (2020). A depth-based indian sign language recognition using microsoft kinect, *Sādhanā* **45**: 1–13.

Rahman, M. M., Islam, M. S., Rahman, M. H., Sassi, R., Rivolta, M. W. and Aktaruzzaman, M. (2019). A new benchmark on american sign language recognition using convolutional neural network, *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, IEEE, pp. 1–6.

Rathi, P., Kuwar Gupta, R., Agarwal, S. and Shukla, A. (2020). Sign language recognition using resnet50 deep neural network architecture, *5th International Conference on Next Generation Computing Technologies (NGCT-2019)*.

Saquib, N. and Rahman, A. (2020). Application of machine learning techniques for real-time sign language detection using wearable sensors, *Proceedings of the 11th ACM Multimedia Systems Conference*, pp. 178–189.

Shahriar, S., Siddiquee, A., Islam, T., Ghosh, A., Chakraborty, R., Khan, A. I., Shahnaz, C. and Fattah, S. A. (2018). Real-time american sign language recognition using skin segmentation and image category classification with convolutional neural network and deep learning, *TENCON 2018-2018 IEEE Region 10 Conference*, IEEE, pp. 1168–1171.

Sharma, S. and Singh, S. (2020). Vision-based sign language recognition system: A comprehensive review, *2020 international conference on inventive computation technologies (ICICT)*, IEEE, pp. 140–144.

Wen, F., Zhang, Z., He, T. and Lee, C. (2021). Ai enabled sign language recognition and vr space bidirectional communication using triboelectric smart glove. nat. commun. 12, 5378 (2021).

Zhou, Z., Chen, K., Li, X., Zhang, S., Wu, Y., Zhou, Y., Meng, K., Sun, C., He, Q., Fan, W. et al. (2020). Sign-to-speech translation using machine-learning-assisted stretchable sensor arrays, *Nature Electronics* **3**(9): 571–578.