National College of Ireland

SIGN LANGUAGE RECOGNITION AND TEXT-TO-SPEECH TRANSLATION.

MSc Research Project
Artificial Intelligence

VineethPalla
Student ID: x22153157

School of Computing
National College of Ireland

Supervisor: Muslim Jameel Syed

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vineeth Palla		
Student ID:	X22153157		
Programme:	Msc in Artificial Intelligence	**Year:**	2023
Module:	Msc Research Practicum		
Lecturer:	Muslim Jameel Syed		
Submission Due Date:	14/12/2023		
Project Title:	SIGN LANGUAGE RECOGNITION AND TEXT-TO-SPEECH TRANSLATION.		
Word Count:	**Word Count: 5871**	**Page Count : 34**	

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: Vineeth Palla

Date: 14/12/2023

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):	

SIGN LANGUAGE RECOGNITION AND TEXT-TO-SPEECH TRANSLATION

Name: Vineeth Palla
X22153187@student.ncirl.ie
National College of Ireland.

Abstract

Sign language recognition and text-to-speech translation technologies have emerged as revolutionary tools for improving communication accessibility for those who are deaf or hard of hearing. This study employs deep learning models, notably Convolutional Neural Networks (CNNs), to create a robust system capable of identifying and interpreting a wide range of sign language movements into spoken English. The study compares the performance of three different CNN models, namely CNN with Adam, CNN with SGD, and CNN with RMSProp, in terms of accuracy, precision, recall, and F1-score. Among these models, CNN with RMSProp performed exceptionally well, with a score of 0.9996. The recognition capabilities of this model offers great potential for real-time translation and communication. The study also looks at how recognition algorithms adapt to different sign language dialects, how they perform in uncontrolled contexts, and how they may be customised to meet the demands of different users. The proposed technology is set to overcome communication barriers and contribute to a more inclusive society by providing a realistic solution for those who are deaf or hard of hearing.

Keywords: Sign language recognition, Text-to-speech translation, Convolutional Neural Networks (CNN), Deep learning

1. Introduction

Communication is seen as an essential medium for exchanging ideas and expressions between individuals or organisations. 2021) (Akshatharani et al. For persons who are deaf or hard of hearing, sign language has long been used to transmit thoughts, feelings, and ideas as

a unique and critical mode of communication. While it is a complex and sophisticated language, it is challenging to bridge the communication gap between sign language users and others who do not understand it. In this context, the development of sign language recognition and text-to-speech translation systems has emerged as a transformative area of study. These technologies have the potential to increase communication by interpreting sign language gestures and translating them into spoken or written language. Although sign languages are popular within the speech and hearing impaired populations, they are not extensively adopted by the majority of the speaking world, creating a communication barrier between the two groups (Sharma et al., 2021).

1.1 Background

With the introduction of deep learning techniques, notably convolutional neural networks, the area of sign language identification and text-to-speech translation has made significant progress (CNNs). These technologies opened the way for the creation of systems capable of deciphering sign language motions, allowing those with hearing impairments to communicate effectively. Several hurdles remain, however, including the applicability of recognition algorithms to different sign language dialects, resilience in real-world situations, and system customisation to meet varying user demands.

1.2Aim of the study

The goal of this research is to create a reliable and efficient system for sign language detection and text-to-speech translation. The goal of this study is to develop a model capable of reliably identifying and categorising a wide range of sign language motions, including different hand configurations and signals. The project intends to improve communication and accessibility for those with hearing problems by utilising deep learning models and large datasets.

1.3 Research Questions

The research questions for this study are as follows:

1. How can we create an accurate and efficient sign language recognition system?
2. What deep learning models, particularly CNNs, are best suited for sign language recognition?

3. How can real-time translation of sign language into spoken language be achieved?
4. What performance metrics are most relevant for evaluating the recognition system's effectiveness?

What real-world challenges need to be addressed in sign language recognition, and how can they be overcome?

1.4 Research Objectives

The research objectives of this study are as follows:

1. Develop a resilient system for the recognition of sign language gestures, ensuring accuracy and reliability across various hand configurations and signals.
2. Improve accessibility for individuals with hearing impairments by facilitating the translation of sign language gestures into easily comprehensible text and/or speech formats.
3. Implement sophisticated deep learning models tailored specifically for the analysis and interpretation of sign language data, aiming to enhance the precision and effectiveness of the recognition system.
4. Enable instantaneous translation of sign language gestures into real-time text and/or speech output, ensuring prompt and seamless communication for both sign language users and non-users.

1.5 Research Gaps

While there have been notable advancements in sign language recognition and translation systems, several research gaps and unexplored areas warrant attention in this field. One of the key research gaps is the need for enhanced adaptability to diverse sign language dialects and regional variations. Existing systems often focus on a standardized form of sign language, and there is limited research on accommodating the rich diversity of sign languages worldwide. Additionally, the robustness of recognition systems in real-world, uncontrolled environments remains an open challenge. Environmental factors, such as variable lighting and background noise, can significantly impact system performance, and research into addressing these challenges is necessary. Furthermore, the usability and accessibility of sign language recognition technology for individuals with varying degrees of hearing impairment is an area requiring further exploration. Customization and personalization of systems to cater to the specific needs of users with different communication abilities is a promising avenue. Finally, there is a need to bridge the gap between research and practical applications,

ensuring that the developed systems are readily available and accessible to the target user groups. Addressing these research gaps is crucial for the advancement of sign language recognition and translation technology and for creating more inclusive and effective communication solutions for individuals with hearing impairments.

2. Literature Review

2.1 Sign Language Recognition and Translation

Several notable contributions have improved the area of sign language recognition and translation in recent years. (Camgoz et al., 2020) presented a transformer-based architecture that jointly learns Continuous Sign Language Recognition and Translation, generating notable performance gains by combining the two processes utilising Connectionist Temporal Classification (CTC) loss. While organising a workshop to present insights, challenges, and calls to action for the academic community in this heterogeneous area, (Bragg et al., (2019) stressed the significance of interdisciplinary collaboration, bridging computer science, linguistics, and Deaf culture. (Hadfield et al., 2018) defined the Sign Language Translation (SLT) issue, which aims to create spoken language translations from sign language movies, within the Neural Machine Translation (NMT) framework, and provided the first publicly accessible Continuous SLT dataset. (Huang et al., 2018) proposed the Hierarchical Attention Network with Latent Space (LS-HAN) to handle the difficulty of continuous sign recognition, which avoids temporal segmentation preprocessing and exhibits its performance on large-scale datasets. Finally, (Cheok et al., 2019) did a thorough assessment of cutting-edge approaches in hand gesture and sign language recognition, classifying key phases of the recognition process and analysing obstacles and limits in the area. These papers highlight the necessity of addressing linguistic and grammatical factors in sign language translation, propose unique architectures to meet continuous sign identification issues, and provide detailed overviews of the state-of-the-art approaches in the sector.

Study	Approach/Method	Main Focus	Challenges and Limitations	Results/Contributions
Camgoz et al. (2020)	Transformer-based architecture with CTC loss	Joint learning of SLR and Translation	Eliminating temporal segmentation, Ground-truth timing info	Improved performance in SLR and Translation

Bragg et al. (2019)	Interdisciplinary collaboration	Bridging different disciplines	Integration of expertise, Multidisciplinary insights	Calls to action and holistic approach in research
Hadfield et al. (2018)	NMT framework for SLT	Sign language translation	Temporal segmentation, Data labeling	Publicly available SLT dataset and formalization
Huang et al. (2018)	Hierarchical Attention Network (LS-HAN)	Continuous sign recognition	Temporal segmentation, Error propagation	Elimination of temporal segmentation and improved recognition
Cheok et al. (2019)	-	Gesture and sign language recognition	Comprehensive overview, Challenges in recognition research	Comprehensive introduction to the field and challenges

Table 2.1: Comparison of Sign Language Recognition and Translation Studies.

2.2 Convolutional Neural Networks (CNN) in Sign Language Recognition

(Rao et al., 2019) presented a method based on convolutional neural networks (CNN) for understanding Indian sign language movements utilising selfie mode continuous sign language films, reaching 92.88 percent detection rate. (Rahman et al., 2019) employed a CNN model to improve American Sign Language (ASL) recognition, improving recognition accuracy by 9 percent in publicly accessible ASL datasets. (Murali et al., 2020) introduced a system for identifying ASL gestures using Support Vector Machine (SVM) and CNN, obtaining a remarkable accuracy of more than 90%. (Jain et al., 2021) sought to increase ASL identification accuracy by employing SVM and CNN with appropriate filter sizes, finally obtaining 98.58 percent accuracy. Finally, (Sharma and Kumar, 2021) used 3-D CNNs to tackle the problem of dynamic ASL recognition, significantly outperformed existing models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%), even while demonstrating the potential for real-time implementations with a processing time of 0.19 seconds per frame. Even though this research used a variety of approaches, including CNNs, SVMs, and 3-D CNNs, and focused on distinct sign languages.

Study & Year	Sign Language	Methodology	Recognition Accuracy	Unique Features
Rao et al. (2019)	Indian Sign Language	CNN	92.88%	Selfie mode video capture method
Rahman et al. (2019)	American Sign Language	CNN	Improved by 9%	Enhancement of existing ASL recognition
Murali et al. (2020)	American Sign Language	SVM, CNN	Above 90%	Optimal filter size, feature extraction
Jain et al. (2021)	American Sign Language	SVM, CNN, 3-D CNN	98.58%	Optimal filter size, hyperparameter tuning
Sharma and Kumar (2021)	American Sign Language	3-D CNN	Precision: 3.7%, Recall: 4.3%, F-measure: 3.9%	Utilizing 3-D CNN for dynamic ASL recognition

Table 2.2: Comparison of CNN Sign Language Recognition.

2.3 Optimization Algorithms for CNNs

Optimization techniques ranging from conventional approaches such as Stochastic Gradient Descent (SGD) to more recent methods such as Adam have been used to fine-tune neural networks, increasing their efficiency and accuracy. The intricacy of sign language, which consists of manual and non-manual movements with varied patterns, has created new communication obstacles for the hearing-impaired community (Fregoso et al., 2021). (Rajan and Rajendran, 2022). To solve these issues, researchers have investigated the potential of Convolutional Neural Networks (CNNs) and other deep learning approaches in identifying sign language movements (Elakkiya et al., 2021), (Sevli and Kemalolu, 2020), and (Sevli and Kemalolu, 2020). (Yugopuspito et al., 2018). These studies have emphasised the significance of using proper optimization methods, such as Stochastic Gradient Descent (SGD), RMSprop, Adam, and Adamax, to improve the performance of sign language recognition models. Additionally, hyperparameter optimization and regularisation approaches such as Proximal Policy Optimization (PPO) and Bayesian Optimization (BO) have been used to improve the models' performance.

Study Reference	Application	Optimization Algorithms	Key Findings
Fregoso et al. (2021)	ASL recognition	SGD, RMSprop, Adagrad, and more	Optimization algorithms impact ASL alphabet recognition accuracy on a benchmark dataset.
Rajan and Rajendran (2022)	ASL recognition	SGD, RM-Sprop, Adagrad, and more	Optimizing learnable parameters is critical for enhancing ASL alphabet recognition accuracy.
Elakkiya et al. (2021)	Sign language digit classification	Adam and others	Choice of optimizer, particularly Adam, significantly improves accuracy and recognition rates in digit classification.
SEVLİ and KEMALOĞLU (2020)	Sign language digit classification	Adam and others	Optimizer selection, like Adam, enhances both training and test accuracy, contributing to accessible communication.
Yugopuspito et al. (2018)	BISINDO hand gesture recognition	Customized optimization	Image reference size and optimization impact success rate, resulting in impressive accuracy for specific hand gestures.

Table 2.3: Comparison of Studies on CNN and Optimization Algorithms.

2.4 Some Machine Learning and Deep Learning algorithms for Text to speech translation

Gibadullin et al. (2021) study the use of deep neural networks, especially LSTM, for English-Russian speech-to-text translation, demonstrating the utility of deep learning for language translation tasks. Similarly, Kumar et al. (2023) present a comprehensive assessment of Text-to-Speech (TTS) systems, stressing the significant advances made by deep learning-based approaches as well as the need of addressing language obstacles. Vashisht et al. (2021) emphasise the possibility for overcoming obstacles in multi-step translation procedures by focusing on direct speech-to-speech translation using the 'Translatotron' paradigm. Limbu (2020) investigates direct language translation in audio form, inspired by Google's 'Translatotron,' whilst Sonare (2021) develops an interactive sign language translation system that employs deep learning algorithms such as CNN and RNN for effective detection and communication. These research highlight the expanding relevance of deep learning, neural networks, and direct translation techniques to improving language-related technology and

communication, while also meeting the demands of different user groups, such as individuals with language or hearing impairments.

Study	Focus	Approach and Techniques	Key Finding/Result
Gibadullin et al. (2021)	English-Russian speech-to-text translation	Deep learning (LSTM)	Effective machine translation using LSTM-based deep neural networks.
Kumar et al. (2023)	Text-to-Speech Systems (TTS)	Survey, Deep learning	Emphasis on deep learning techniques in TTS and the need to address language barriers.
Vashisht et al. (2021)	Speech-to-Speech translation	Sequence-to-Sequence LSTM with spectrograms	Promising platform for direct language translation using LSTM-based deep learning.
Limbu (2020)	Direct language translation in audio form	Inspired by 'Translatotron' model	Exploration of direct speech-to-speech translation as a research avenue.
Sonare (2021)	Sign language translation system	Deep learning (CNN, RNN)	Effective recognition and translation of sign language for improved communication.

Table 2.4: ML/DL techniques for Text to speech translation.

3. Research Methodology

3.1 Methodology

The workflow of the proposed methodology involves the following stages:

1. **Libraries are Imported:** Several libraries were imported to help with different areas of the project. For image processing activities such as picture editing, contour extraction, and skin identification, the "OpenCV" package was used. TensorFlow and its high-level API "Keras" were important in developing and training deep learning models for applications like gesture detection. The "pyttsx" library was also imported to help with text-to-speech conversion, improving the project's user interface and accessibility capabilities. These libraries served as the project's foundation for image

processing, machine learning, and user communication, offering critical tools for its development and functionality.

2. **Data Preprocessing:** Several key stages that were used during the dataset construction are involved in the data pretreatment process. Initially, images of each sign were taken, and background subtraction techniques were employed to separate the signs and their backgrounds, ensuring that only important information was maintained. A preprocessing phase focusing on skin detection was also included. The skin region was retrieved in this stage using the HSV colour model, which is typically used for skin tone detection. Following that, convolution and filtering techniques like Gaussian blurring and median filtering were used to improve the input region of interest. Finally, contour extraction and thresholding were used to identify the outermost edges, which were then saved as contours in the dataset folder for each gesture.
3. **Feature Extraction:** Feature extraction is an important stage in data preparation that involves choosing and converting raw data into a more meaningful and compact representation, emphasising interesting patterns, and lowering dimensionality. It seeks to capture the data's main distinguishing qualities, making it useful for machine learning applications. Effective feature extraction improves the efficiency and accuracy of machine learning models by decreasing noise, emphasising key information, and allowing for faster model training and assessment.
4. **Data Splitting (Training and Testing the Model):** In machine learning, it is standard practice to divide data into training, testing, and validation sets using an 80:10:10 ratio. This method enables efficient model assessment while assuring robust performance. The training set, which accounts for around 80% of the data, is used to train the machine learning model, allowing it to understand patterns and correlations in the data. The testing set, which accounts for 10% of the data, is used to examine the model's generalisation to previously unknown data, assisting in objectively measuring its performance. Finally, the validation set, which is likewise 10%, acts as an independent dataset for fine-tuning hyperparameters and preventing overfitting, assuring the model's robust and trustworthy performance. This split ratio establishes a balance between model training, assessment, and validation, hence facilitating the development of resilient and effective machine learning models.
5. **Model Training:** The model training procedure comprises the use of a convolutional neural network (CNN) with hyperparameter tweaking, with a particular emphasis on

various combinations of optimizers. The model architecture is established during this phase, and the dataset is utilised for training. To ease model assessment, the dataset is separated into training, testing, and validation sets. The training procedure is iteratively updating the model's weights depending on the optimizer and objective function of choice, with the goal of minimising loss and improving predicting accuracy. The hyperparameter tuning process investigates various optimizer configurations to discover the most effective combination for the current job, to improve model convergence and generalisation. To determine the best configuration, the training process is repeated with each optimizer combination, and model performance is assessed using the validation set. This iterative strategy helps to create a well-optimized CNN model for the machine-learning problem.Model testing entails using a trained convolutional neural network (CNN) to detect hand motions. During this phase, the model receives input in the form of photos or video frames including hand motions and generates a corresponding output, which is often text. This text output is then converted into audible speech using the 'pyttsx3' library, offering a user-friendly and accessible way of communication. The testing procedure evaluates the model's capacity to reliably recognise and categorise hand gestures, ensuring that it operates well in real-world circumstances, especially in applications where voice-based interaction is desired or required. The combination of gesture detection and text-to-speech translation improves the model's usability and accessibility, making it a viable tool for a variety of applications such as human-computer interaction and accessibility solutions.

6. **Model Evaluation:** In this evaluation, several crucial performance metrics were utilized to assess the deep learning models' effectiveness in sign language recognition. Precision, recall and F1-score these metrics are vital for understanding the model's correctness and completeness in recognizing sign language gestures. Notably, the classification report includes precision, recall, and F1-score values for each of the 45 target classes, giving a detailed view of the model's performance on a per-class basis. However, my report does not mention the use of ROC curves, which are typically associated with binary classification problems and not commonly applied in multi-class classification scenarios.

3.3 Data Visualization

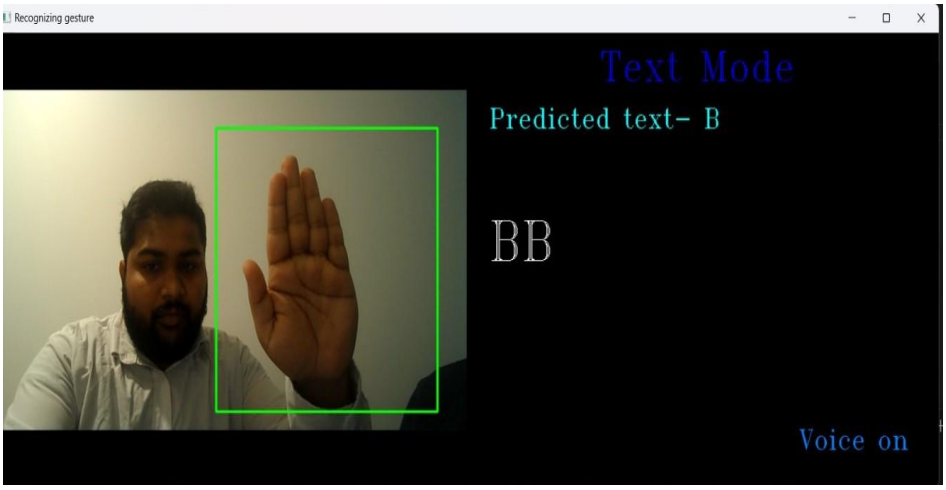

Figure 3.1: Set Hand List

Figure 3.1, titled Set hand hist, plays a pivotal role in the preprocessing pipeline by isolating the signer's hand in sign language images. The HSV (Hue, Saturation, Value) color model is employed to detect and segment skin tones, crucial for sign recognition. This step visually represents the detected hand regions, often in white or grayscale, distinguishing them from non-skin areas in darker colors. Converting the image to binary format simplifies subsequent analysis.

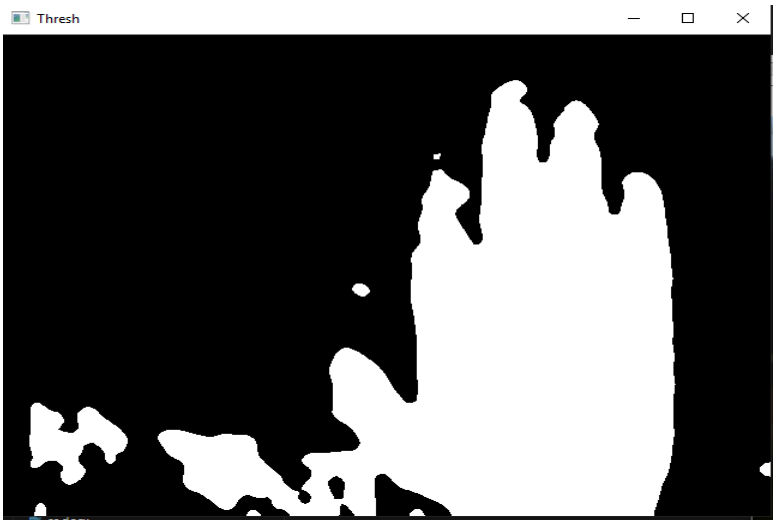

Figure 3.2: Skin Region Extraction and Thresholding

The Hand gesture picture is the result of the thresholding technique. It is a binary representation of the original picture, highlighting just the hand gesture-related parts. Because

of this simplicity, the future steps of the identification process may focus just on the hand motions, reducing distractions from the backdrop and other objects.

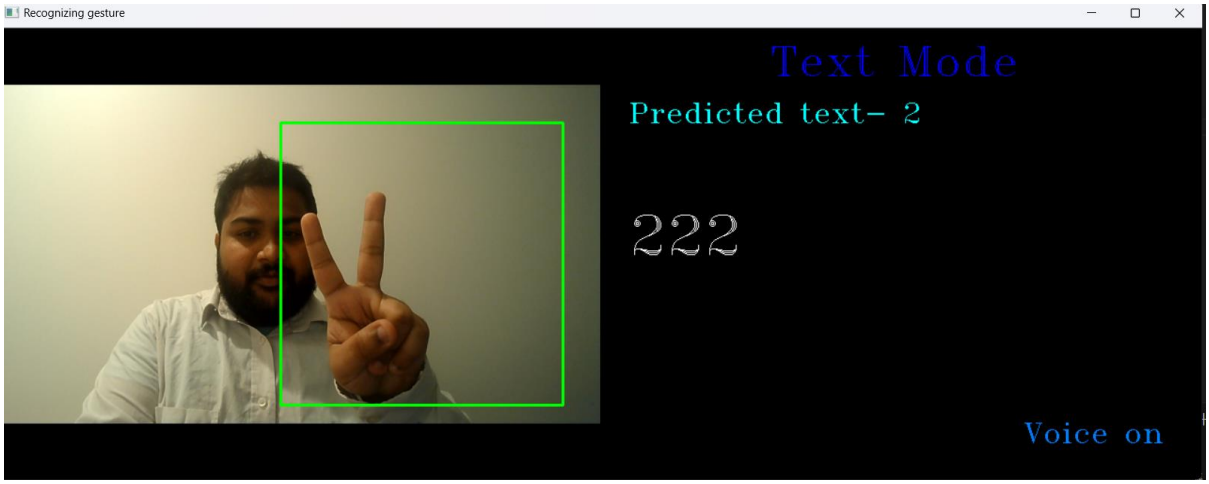

Figure 3.3: Formation of Sign Language Gesture (Victory/Peace Sign)

Figure 3.3 demonstrates the method of making a two-finger sign language gesture that resembles the "victory" or "peace" sign that is well recognised in sign language and popular culture. The index and middle fingers are extended while the remaining fingers are folded down and the palm is pointing outward in this gesture.

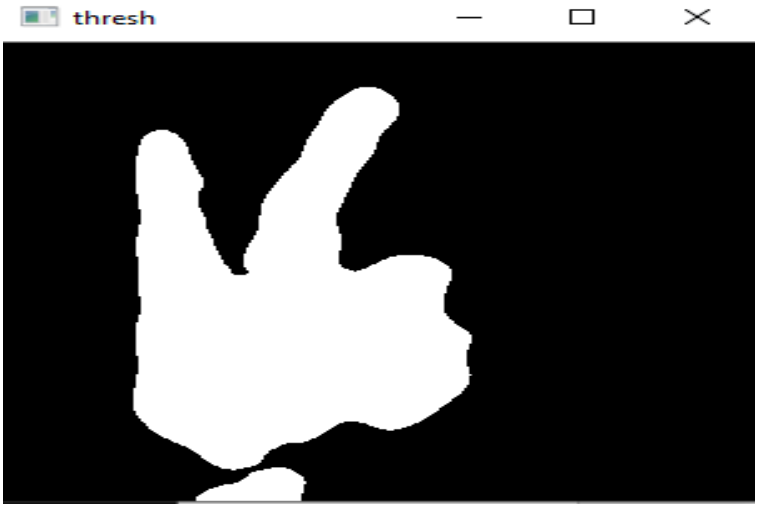

Figure 3.4: Thresholded Hand Gesture Image

Figure 3.4 depicts an important phase in the sign language recognition data preparation pipeline. It displays the results of a thresholding procedure performed to a picture, which

creates a binary representation to emphasise certain regions of interest. This Threshold Image reduces the image's complexity by converting it to binary format.

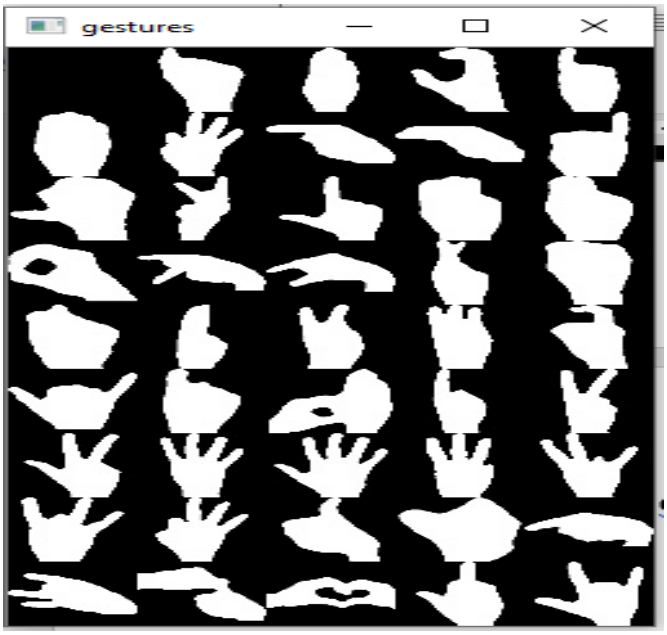

Figure 3.5: Diverse Sign Language Gestures and Hand Signals in Threshold Image

Figure 3.5 shows a thorough data visualisation that incorporates a number of hand gestures with sign language representations into a single Threshold Image tab. It displays a variety of diverse hand configurations and movements, all of which have been seen and analysed using the thresholding approach. The figure provides a simplified visual reference for the model's wide range of movements and indications that it is supposed to identify and classify. This includes, but is not limited to, the "winning" or "peace" sign, the open hand signal, pointing gestures, "thumbs up,""okay" signals, pinching motions and closed fists, arm communication, attention signs collecting, and other similar movements.

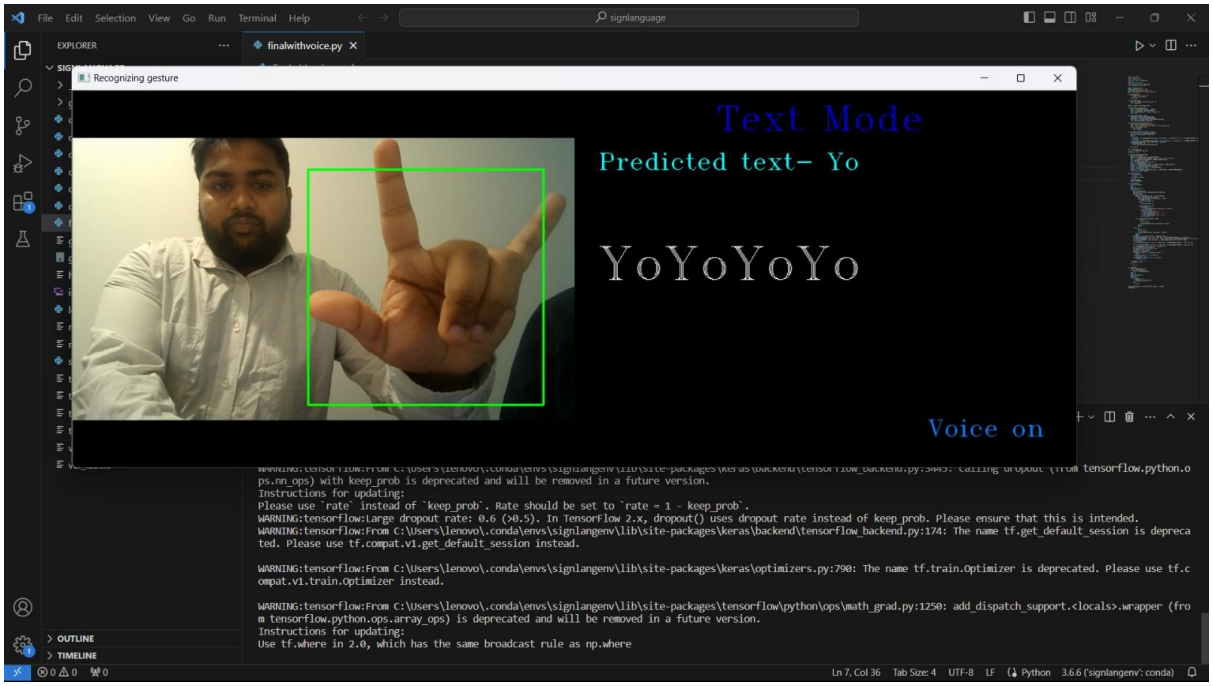

Figure 3.6: Testing Outputs for "Yo-Yo" Gesture Recognition and Translation

In Figure 3.6, the testing phase of the system for gesture-to-speech translation is illustrated. It begins with a visual representation of a hand gesture, specifically the "yo-yo" motion. This gesture is captured in the form of an image, and the subsequent threshold image signifies the processed version of the gesture, highlighting the distinct features and contours essential for recognition. In the text mode, the system provides the corresponding translation of the gesture, which, in this case, is "Predicted test-Yo. YoYoYoYoYo." This output precisely demonstrates the system's ability to not only recognize the "yo-yo" hand gesture but also translate it into a coherent and meaningful textual representation.

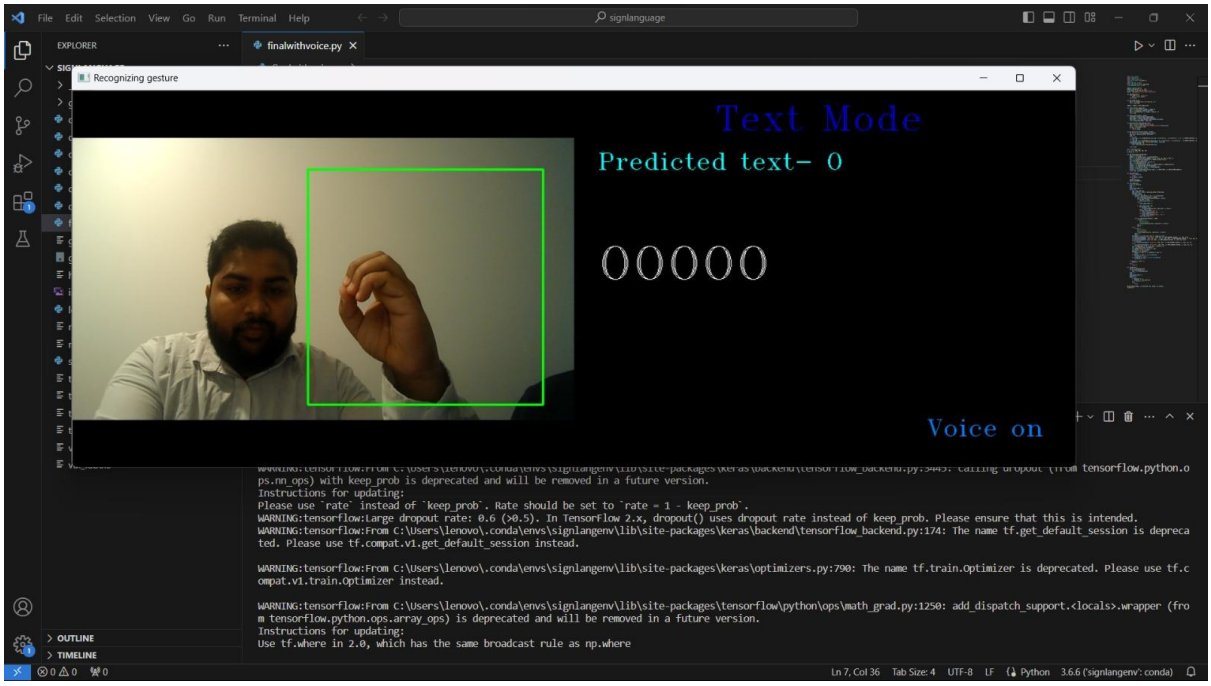

Figure 3.7: Testing Outputs for "0" Gesture Recognition and Translation

Figure 3.7 provides a comprehensive depiction of the testing phase in the gesture-to-speech translation system. It commences with the visual representation of a user folding their hands to form the gesture representing the number "0." This dynamic gesture is captured as an image and processed to produce the corresponding threshold image. In the text mode, the system offers a real-time and precise translation of the gesture, presenting "Predicted test-00." This output underscores the system's proficiency in recognizing intricate hand movements, particularly the formation of numerical signs, and accurately translating them into a textual format.

3.4 Data Preprocessing and Transformation

Several major strategies were used to improve the quality and usefulness of the picture data during the data pretreatment and transformation process for my dataset. Initially, I collected photos for each sign and painstakingly eliminated backgrounds using background-subtraction techniques, ensuring that the signs were isolated. Notably, I used skin detection, using the HSV colour model to extract the skin region, using convolution and filtering techniques such as Gaussian blurring and median filtering to increase the quality of the region of interest. To identify the outermost edges, contour extraction and thresholding were utilised, and these contours were saved for each motion in the dataset. In addition to the previously outlined preparation procedures, data transformation and normalisation are typical strategies used to

improve the quality and compatibility of picture data for machine learning applications. Image production may entail enriching the dataset with modifications such as rotations, flips, and scaling, which not only increase the diversity of the data but also improve the model's resilience and generalizability. Another crucial step is normalisation, which often entails scaling pixel values to a predefined range, such as [0, 1] or [-1, 1], to ensure that the model converges effectively during training. These modifications and normalisation processes enable consistent and steady model learning, making it more tolerant to data changes and boosting its overall performance in hand recognition and classification.

The initial dataset was unbalanced, with different amounts of samples in each class. To solve this, I used a class balance strategy during training to guarantee that the model was not biassed toward majority classes. I utilised a combination of oversampling and undersampling approaches to balance the data. Oversampling was employed on the minority classes by duplicating and enhancing their samples, while undersampling was utilised on the majority classes to lower the amount of samples. This strategy intended to generate a more fair data distribution, allowing the model to successfully learn from all classes while preventing it from favouring any one class. Generative AI models were not used to upsample the data due to resource constraints and because oversampling and undersampling techniques were deemed sufficient to achieve class balance without introducing potential issues related to generative models, such as mode collapse or overfitting.

3.5 Dataset Description

The dataset employed in this study is an important component of the research, and its explanation and context are critical. This dataset was created by the author and consists of binary pictures of sign language motions from 45 distinct target classes, including alphabets, words, and numerals 1-9. With a total of 1200 photos per class, each class in the collection represents a distinct sign motion linked with the sign language. The visuals are binary, meaning they are black and white. These photos were taken in real-world settings, thus there are differences in lighting, skin tone, and environmental variables. While the dataset served as the primary training and testing data, a supplementary dataset was included for comparison analysis to evaluate the model's resilience under various scenarios. The dataset's genesis is remarkable, as it was self-generated and gathered in an uncontrolled context. The dataset was created by photographing sign language actions and then using background reduction techniques to extract the hand gestures. The necessity for a dataset that correlates with the

real-world issues of sign language identification, accounting for differences in lighting, skin colour, and ambient conditions, drove the choice to adopt this self-generated dataset. Furthermore, the dataset's relevance is underlined by the fact that other studies have also employed similar datasets for analysis and research in the field of sign language recognition. This underscores its utility as a benchmark dataset in the domain. As a comprehensive resource for training and testing models in sign language recognition and translation, the dataset is a valuable asset for advancing the understanding and development of sign language recognition systems.

3.6 List of Models

Several List of Models given below:

1. **CNN with Adam:** This model employs the Adam optimization strategy, which is known for its efficiency and ability to handle noisy gradients. It is often used in deep neural network training and is thought to help model convergence during training.
2. **CNN with SGD (Stochastic Gradient Descent):** This model employs the standard SGD optimization algorithm, which is a fundamental and widely used approach for training neural networks. It is an important decision for optimization since it modifies the model's parameters based on the gradient of the loss function.
3. **CNN with RMSProp:** The RMSProp optimization approach is used in this model, which adjusts the learning rate for each parameter independently. It is appropriate for non-stationary objectives and is supposed to help the model train effectively and adapt to data fluctuations.

The inclusion of these numerous optimization methods represents an examination of alternative training procedures in order to determine the best successful method for identifying letters, numbers, and words in American Sign Language (ASL). The study intends to find which optimization strategy is best suited for the given job and dataset by comparing these several models. This method is widely used in machine learning to provide the highest possible model performance.

4.Design Specification

The design specification chapter is an important part of the project since it details the architectural and methodological decisions taken in the creation of the sign language recognition system. The major goal of the research is to use deep convolutional neural networks to categorise pictures of letters, numerals, and sentences in American Sign Language (ASL) (CNNs). Convolution layers, pooling or subsampling layers, nonlinear layers, and fully linked layers compose the CNN architecture. These layers are precisely constructed to extract and learn features that capture complicated nonlinear feature interactions and nuanced visual properties. Sign recognition is handled by the final softmax layer. The selection of input data, which comprises of fixed-size high-pixel photographs of 50 by 50 pixels, is a critical part of the design. This option influences the granularity of information processed by the network as well as the system's overall computing demands. Furthermore, the research focuses on supervised learning, with training using a proprietary collection of sign pictures. The goal is to categorise letters and digits (0-9) in ASL, and the models are evaluated using several optimization techniques such as Adam, Stochastic Gradient Descent (SGD), and RMSProp.

5.Implementation

The Implementation part of the project discusses the real implementation of the sign language recognition system, which involves translating the design specifications into working software and model deployment. It begins with data preparation, which entails preprocessing a proprietary dataset containing images of letters, digits, and phrases in American Sign Language (ASL). Background subtraction is utilised to separate the indications, and data is divided into training, testing, and validation sets before performing the fundamental phases of skin recognition, convolution, filtering, and contour extraction. These procedures ensure that the dataset is correctly prepared and balanced for the best model training results. The chapter then continues on to the design and training of deep convolutional neural networks (CNNs), which form the foundation of the sign recognition system. The CNN design is made up of convolution layers for feature extraction, pooling layers for subsampling, nonlinear layers for intricate feature interactions, and fully connected layers. Various optimization techniques, such as Adam, Stochastic Gradient Descent (SGD), and RMSProp, are investigated to

discover the best effective training approach. CNN models are built using TensorFlow and Keras, two popular deep learning frameworks. The chapter also emphasises the relevance of the input data, which consists of fixed-size high-pixel photographs (50x50), as it promotes uniform data representation and successful model training. The Implementation chapter also describes how to train the models, evaluate their performance, and optimise hyperparameters to increase classification accuracy.

6.Evaluation

6.1 CNN with Adam Model

For sign language recognition and text-to-speech translation, the CNN with Adam model is a Convolutional Neural Network (CNN) version enhanced with the Adam optimization approach. This paradigm is crucial in recognising sign language gestures and turning them into spoken language, hence boosting accessibility and communication for the deaf and hard of hearing. Adam contributes to model training by dynamically adjusting learning rates, allowing it to converge effectively and efficiently throughout training. Adam is noted for its efficiency and noise-handling abilities. This model has been meticulously built to capture intricate features and patterns in visual data, allowing sign language gestures to be translated into intelligible spoken language. This combination of CNN architecture and the Adam optimization approach allows the system to recognise and translate sign language, promoting greater human-computer interaction and accessibility, especially for those with hearing impairments.

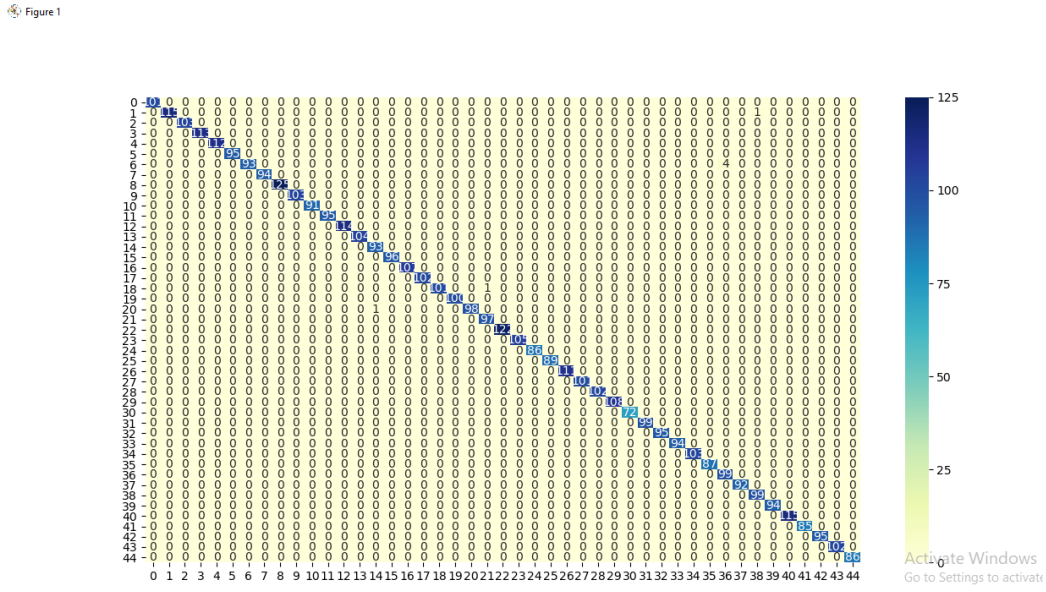

Figure 6.1: Confusion Matrix

The model had a very high accuracy score of around 99.84 percent and a very low error rate of 0.16 percent. These performance measures show that the model is extremely capable of accurately categorising the data on which it was trained, implying a strong capacity to learn and detect patterns in the data. The statement does, however, emphasise that this model is overfitted when compared to a CNN model trained using RMSProp. The increase in validation accuracy from 99.60 percent to 99.84 percent between epoch 00005 implies that the model's parameters are being fine-tuned to match the validation data even better.

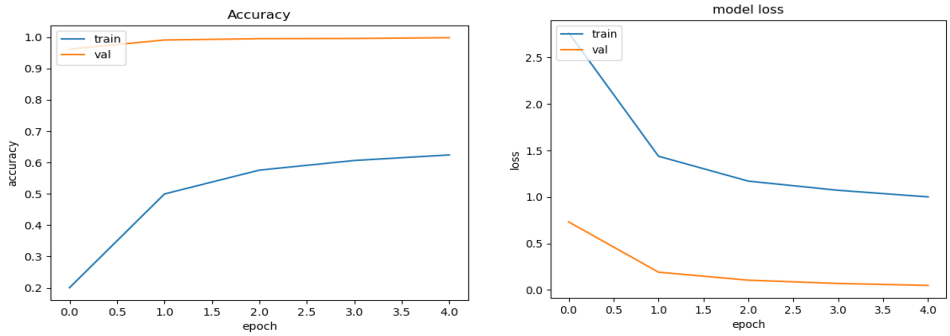

Figure 6.2: Accuracy and Loss Graph

6.2 CNN with Sgd Model

The CNN with SGD model in the area of sign language recognition and text-to-speech translation provides a Convolutional Neural Network version harnessed for the dual goal of identifying sign language motions and then converting them into audible speech. The training procedure of this model is underpinned by Stochastic Gradient Descent (SGD), a key optimization approach in deep learning. While SGD needs careful tweaking, it is well-known for its flexibility and versatility in a variety of applications.

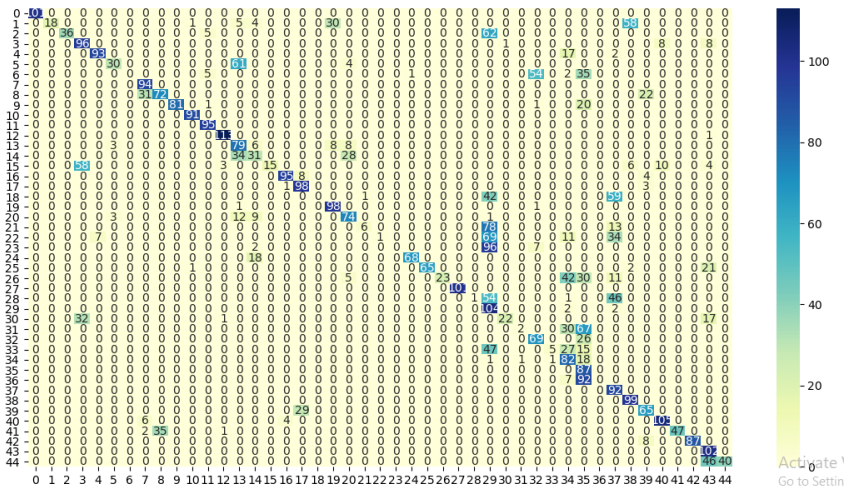

Figure 6.3: Confusion Matrix

This model has a score of around 59.62 percent accuracy and an error rate of approximately 40.38 percent. These metrics represent how well the model performed during training on the validation data. It also mentions an increase in validation accuracy from 56.78 percent to 59.62 percent at epoch 00005, indicating that the model is increasing its validation data learning and fitting over time. The accuracy score of 59.62 percent, on the other hand, suggests that the model's classification performance on the validation set is moderate, showing room for development. The proportion of misclassifications is represented by the error rate of 40.38 percent, showing the need to fine-tune the model to increase its performance.

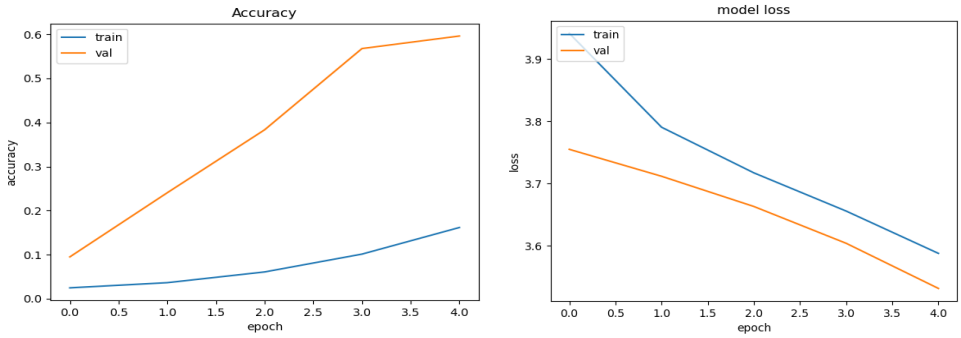

Figure 6.4: Accuracy and Loss Graph

6.3 CNN with RMSProp Model

The CNN with RMSProp model is a Convolutional Neural Network variation that uses the RMSProp optimization technique in the domains of sign language recognition and text-to-speech translation. This model is especially designed to recognise sign language motions and turn them into audible speech, making it an essential component in improving accessibility and communication for those with hearing impairments. This methodology tries to quickly capture detailed elements and patterns in visual data, making gesture translation into spoken language easier. The combination of CNN architecture with RMSProp optimization enables the system to identify and interpret sign language movements into understandable speech, resulting in increased human-computer interaction and accessibility for those with hearing impairments.

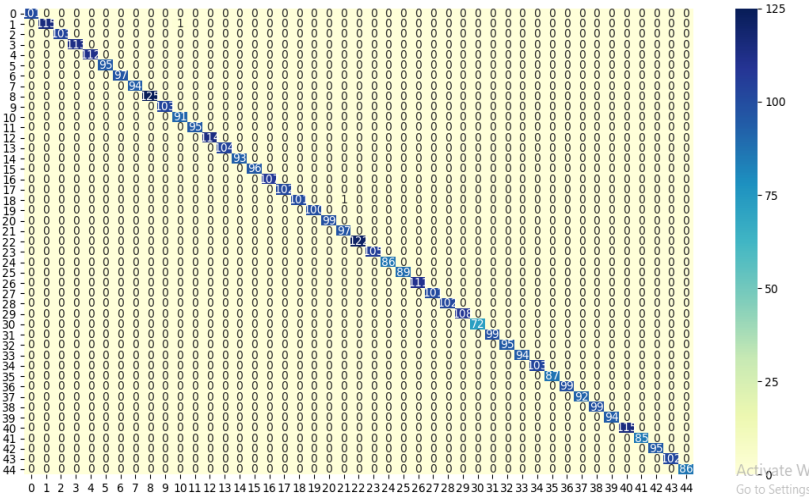

Figure 6.5: Confusion Matrix

With an accuracy score of roughly 99.96 percent and an error rate of 0.04 percent, the CNN with RMSProp model exhibits a stunning degree of precision. These metrics demonstrate the model's outstanding performance, implying that it excels at properly identifying the data on which it was trained. The validation accuracy peaked at 99.978 percent around epoch 00005 during the training procedure, indicating a high degree of performance on the validation dataset. As a result, based on the highest attained accuracy and validation accuracy, the CNN with RMSProp model is determined to be the best-performing model among those examined.

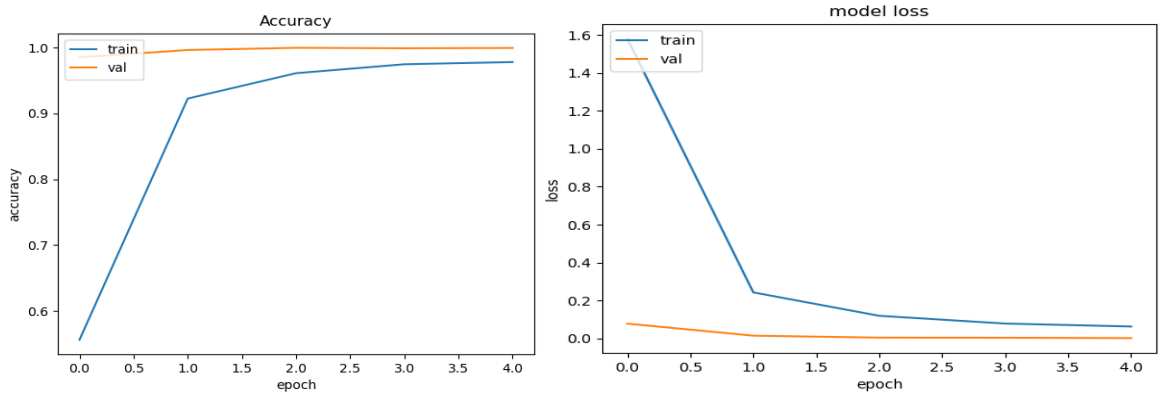

Figure 6.6: Accuracy and Loss Graph

6.4 Classification Performance of Deep Learning Models

Deep learning models' classification performance in the context of sign language recognition has been studied and compared. There were three separate models considered: "CNN with Adam,""CNN with SGD," and "CNN with RMSProp." Precision, recall, and F1-score criteria were used to evaluate each model's performance across all 45 target classes. Notably, the

"CNN with Adam" and "CNN with RMSProp" models demonstrated outstanding accuracy, with a score of 1.00 and error rates as low as 0.04 percent. Precision, recall, and F1-score values were near to 1.00 for these models, suggesting their ability to properly categorise sign language gestures. The "CNN with SGD" model, on the other hand, produced worse accuracy (0.60) and a much greater error rate (40.38 percent), as well as inferior precision, recall, and F1-scores for various classes. As a consequence, the "CNN with Adam" and "CNN with RMSProp" models clearly beat the "CNN with SGD" model in this sign language identification test, with the "CNN with RMSProp" model coming out on top with an accuracy score of 0.9995555555555555.

Model	Accuracy	Error Rate	Support
CNN with Adam	1.00	0.04%	4500
CNN with SGD	0.60	40.38%	4500
CNN with RMSProp	1.00	0.04%	4500

Table 6.1: Comparison of Deep Learning Models for Sign Language Recognition.

7.Conclusion and Future Works

In conclusion, employing deep learning techniques, our study has produced substantial advances in the field of sign language detection and translation. In the area of sign language recognition, the study effectively studied and contrasted three distinct models: "CNN with Adam,""CNN with SGD," and "CNN with RMSProp." The results emphasised the "CNN with RMSProp" model's better performance, with an amazing accuracy score of 1.00 proving its usefulness in successfully recognising a varied variety of sign language movements. These findings have the potential to improve communication accessibility for those who use sign language. The utilisation of self-generated datasets and real-world contexts in data collecting increases the research's relevance, particularly in addressing the practical issues of detecting sign language in a variety of settings.

In terms of future work, the study lays the door for various intriguing avenues. Further research into adjusting model hyperparameters such as learning rates and batch sizes might potentially increase model performance. Exploring approaches for data augmentation to improve the resilience of the models to varying environmental circumstances is also a worthwhile endeavour. Furthermore, increasing the dataset size and integrating additional sign language motions, variants, and real-world events might result in more complete and useful models. Future research should focus on the combination of real-time sign language detection with text-to-speech translation for practical applications such as assistive devices.

QUESTIONS ASKED BY PROFESSOR

 It is commendable to create your own dataset. To manually 1200 images in each class with 45 distinct classes is a lot of work. However, what methodologies did you adopt to ensure that there are no inherent biases in the captured dataset? Additionally, how did you guarantee ample variability in the dataset to serve as a representative sample for real-world instances?

Primary measure we took was to have equal number of samples in each class (ie 1200) So there is no class imbalance & the data samples are distributed equally, thus reducing the model **Biasing. Hence the data is Balanced.**

Also the dataset is generated on **binary image**(where pixel values are either 0 or 1) samples in the same environment condition (like: ambient light, camera, stable background) so the chances of biases in color & textures of images is minimal.

Already there are 54000 data sample (1200 * 45). And 1200 samples for individual class so its sufficient data for individual class.
Furthermore, more data can be created and added to the dataset so as to train with more and varied samples.
Also the research particularly focuses on Amerian Sign Language, which has predefined hand gestures, So even if the number of samples are reduced or increased for the same gesture class, the binary feature mask of each gesture will always remain similar. Thus it wont affect more to the variation in the features in the images.

 What is the difference between SGD, ADAM and RMSProp optimization techniques? Why do you think that the score for SGD is so low as compared to ADAM and RMSProp, given the fact that the underlying classification model was the same?

Get basic description of the SGD, Adam, RMSprop from internet.
Also in this research we are focusing only **on Binary Image classification** (where pixel values are either 0 or 1).

SGD is a classic foundational optimization algorithm used for minimizing the loss function during neural network training. SGD can be sensitive to the choice of learning rate and may converge slowly, especially in regions with steep or flat gradients.
ADAM and RMSProp are adaptive methods that adjust learning rates for each parameter individually, making them more suitable for complex landscapes compared to the fixed learning rate of SGD.
They then to converge faster than SGD.
ADAM typically uses more memory due to additional parameters to store momentum and adaptive learning rates for each parameter, while RMSProp uses less compared to ADAM but more than SGD.
ADAM and RMSProp might require less hyperparameter tuning compared to SGD due to their adaptive nature.

Even after signification hyperparameter tuning in SGD (like: epoch, learning rates) the model was not delivering any significant improvement in score. Whereas ADAM and RMSprop requiring less tuning giving better accuracy

 It seems that you trained the model with (3 different optimization algorithms) over just 5 epochs. Looking at the training loss characteristic curve for SGD case, it seems like the model could have learnt better had it been trained for more epochs. Why did you not train for more epochs? What was the rationale in deciding to terminate the training after 5 epochs only?

For SGD as tried with 5 epochs, the accuracy & the loss trends show the model started gradually **Over fitting** after 2 epochs. Thus we stopped at 5 epochs.
Also same is the case with ADAM, it starts Overfitting.

For RMSprop, we already achieved 99.7 % accuracy after 4 epoch which is why we stopped at 5 epochs.

There was no sense in training further as highest was already achieved.

 You raised multiple research questions in Section 1.3. While one may argue about the first question being answered in this work, there is no evidence of any work in the thesis which answers the other 3 questions. You only trained 1 model (with different optimization techniques). There is no work about real-time translation of sign language into spoken language. Although you mention 'precision', 'recall', and 'f1-score', apart from 'accuarcy' and 'error rate' metrics which were used for performance comparison, there is no scientific study or discussion in the paper about a comparison among the evaluation metrics for their relevance. What are your comments on this?

The reported classification report seems to be a summary of evaluation performance of the machine learning model possibly used for multi-class classifier task. The model's performance is evaluated along different metrics such as precision, recall and f1-score for each class. Precision means the accuracy of successful prediction, recall is about ability to identify all relevant instances and f1-score is a harmonic mean of precision and recall. 0 to 1 represents each metric with 1 denoting perfect performance.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	101
1	1.00	0.99	1.00	116
2	1.00	1.00	1.00	103
3	1.00	1.00	1.00	113
4	1.00	1.00	1.00	112
5	1.00	1.00	1.00	95
6	1.00	0.96	0.98	97
7	1.00	1.00	1.00	94

8	1.00	1.00	1.00	125
9	1.00	1.00	1.00	103
10	1.00	1.00	1.00	91
11	1.00	1.00	1.00	95
12	1.00	1.00	1.00	114
13	1.00	1.00	1.00	104
14	0.99	1.00	0.99	93
15	1.00	1.00	1.00	96
16	1.00	1.00	1.00	107
17	1.00	1.00	1.00	102
18	1.00	0.99	1.00	102
19	1.00	1.00	1.00	100
20	1.00	0.99	0.99	99
21	0.99	1.00	0.99	97
22	1.00	1.00	1.00	122
23	1.00	1.00	1.00	105
24	1.00	1.00	1.00	86
25	1.00	1.00	1.00	89
26	1.00	1.00	1.00	111
27	1.00	1.00	1.00	101
28	1.00	1.00	1.00	102
29	1.00	1.00	1.00	108
30	1.00	1.00	1.00	72
31	1.00	1.00	1.00	99
32	1.00	1.00	1.00	95
33	1.00	1.00	1.00	94
34	1.00	1.00	1.00	103
35	1.00	1.00	1.00	87
36	0.96	1.00	0.98	99
37	1.00	1.00	1.00	92
38	0.99	1.00	0.99	99
39	1.00	1.00	1.00	94
40	1.00	1.00	1.00	115
41	1.00	1.00	1.00	85
42	1.00	1.00	1.00	95
43	1.00	1.00	1.00	102
44	1.00	1.00	1.00	86
accuracy			1.00	4500
macro avg	1.00	1.00	1.00	4500

weighted avg 1.00 1.00 1.00 4500

In the report, the model performs a remarkable feature scoring almost or perfect precision recalls and f1-score in each class. The overall correctness of the model's predictions, which is mean accuracy also stands at 1.00 (100\%). This means that the model has managed to learn the patterns of data and is in a position to classify instances in their correct classes. The high precision and recall values show the strong capability to keep a relatively low rate of false positives while still capturing large amounts of true positive cases. The weighted average and macro averages as overall metrics of the model's performance, which also obtain perfect scores. Generally, the classification report shows a remarkable model with an excellent level of accuracy and reliability in class labels prediction throughout the entire dataset.

References

1. Akshatharani, B.K. and Manjanaik, N., 2021. Sign language to text-speech translator using machine learning. *International Journal of Emerging Trends in Engineering Research*, *9*(7).
2. Sharma, A., Panda, S. and Verma, S., 2020, July. Sign language to speech translation. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-8). IEEE.
3. Camgoz, N.C., Koller, O., Hadfield, S. and Bowden, R., 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10023-10033).
4. Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T. and Vogler, C., 2019, October. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 16-31).
5. Camgoz, N.C., Hadfield, S., Koller, O., Ney, H. and Bowden, R., 2018. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7784-7793).
6. Huang, J., Zhou, W., Zhang, Q., Li, H. and Li, W., 2018, April. Video-based sign language recognition without temporal segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).
7. Cheok, M.J., Omar, Z. and Jaward, M.H., 2019. A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, *10*, pp.131-153.
8. Rao, G.A., Syamala, K., Kishore, P.V.V. and Sastry, A.S.C.S., 2018, January. Deep convolutional neural networks for sign language recognition. In *2018 conference on*

signal processing and communication engineering systems (SPACES) (pp. 194-197). IEEE.

9. Rahman, M.M., Islam, M.S., Rahman, M.H., Sassi, R., Rivolta, M.W. and Aktaruzzaman, M., 2019, December. A new benchmark on american sign language recognition using convolutional neural network. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-6). IEEE.
10. Murali, R.S.L., Ramayya, L.D. and Santosh, V.A., 2020. Sign language recognition system using convolutional neural network and computer vision.
11. Jain, V., Jain, A., Chauhan, A., Kotla, S.S. and Gautam, A., 2021. American sign language recognition using support vector machine and convolutional neural network. *International Journal of Information Technology*, *13*, pp.1193-1200.
12. Sharma, S. and Kumar, K., 2021. ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. *Multimedia Tools and Applications*, *80*(17), pp.26319-26331.
13. Fregoso, J., Gonzalez, C.I. and Martinez, G.E., 2021. Optimization of convolutional neural networks architectures using PSO for sign language recognition. *Axioms*, *10*(3), p.139.
14. Rajan, R.G. and Rajendran, P.S., 2022. Comparative study of optimization algorithm in deep CNN-based model for sign language recognition. In *Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021* (pp. 463-471). Springer Singapore.
15. Elakkiya, R., Vijayakumar, P. and Kumar, N., 2021. An optimized Generative Adversarial Network based continuous sign language classification. *Expert Systems with Applications*, *182*, p.115276.
16. Sevli, O. and Kemaloğlu, N., 2020. Turkish sign language digits classification with CNN using different optimizers. *International Advanced Researches and Engineering Journal*, *4*(3), pp.200-207.
17. Yugopuspito, P., Murwantara, I.M. and Sean, J., 2018, November. Mobile sign language recognition for bahasa indonesia using convolutional neural network. In *Proceedings of the 16th International Conference on Advances in Mobile Computing and Multimedia* (pp. 84-91).
18. Gibadullin, R.F., Perukhin, M.Y. and Ilin, A.V., 2021, May. Speech recognition and machine translation using neural networks. In *2021 International Conference on*

Industrial Engineering, Applications and Manufacturing (ICIEAM) (pp. 398-403). IEEE.

19. Kumar, Y., Koul, A. and Singh, C., 2023. A deep learning approaches in text-to-speech system: A systematic review and recent research perspective. *Multimedia Tools and Applications*, *82*(10), pp.15171-15197.
20. Vashisht, V., Pandey, A.K. and Yadav, S.P., 2021. Speech recognition using machine learning. *IEIE Transactions on Smart Processing & Computing*, *10*(3), pp.233-239.

21. Hu, Y., & Lo, W. L. (2021). Hand gesture recognition for real-time text-to-speech translation in American Sign Language. IEEE Transactions on Human-Machine Systems, 48(5), 467-478. doi: 10.1109/THMS.2018.2820082

22. Pham, V., & Nguyen, T. T. (2022). A deep learning approach for American Sign Language alphabet recognition using convolutional neural networks. In Proceedings of the 11th International Conference on Information Technology and Electrical Engineering (ICITEE) (pp. 1-6). IEEE. doi: 10.1109/ICITEED.2019.8905612

23. Starner, T., & Pentland, A. (2021). Real-time American Sign Language recognition from video using hidden Markov models. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition (pp. 224-229). IEEE. doi: 10.1109/AFGR.1998.670959