Artificial Intelligence (AI) Empowered Sign Language Recognition   
using Hybrid Neural Networks

Ambar Saxena

Thesis

Master of Science in Data Science

Liverpool John Moores University

March 2023

**TABLE OF CONTENTS**

[TABLE OF CONTENTS 5](#_Toc129030148)

[TABLE OF CONTENTS 8](#_Toc129030149)

[LIST OF FIGURES 10](#_Toc129030150)

[LIST OF TABLES 11](#_Toc129030151)

[Abstract 12](#_Toc129030152)

[CHAPTER 1 INTRODUCTION 13](#_Toc129030153)

[1.1 Background 13](#_Toc129030154)

[1.2 Problem Statement 15](#_Toc129030155)

[1.3 Aim and Objectives 18](#_Toc129030156)

[1.4 Significance of the Study 19](#_Toc129030157)

[1.5 Scope of the Study 19](#_Toc129030158)

[1.6 Structure of the study 19](#_Toc129030159)

[CHAPTER 2 LITERATURE REVIEW 21](#_Toc129030160)

[2.1 Introduction 21](#_Toc129030161)

[2.2 Word Cloud 22](#_Toc129030162)

[2.3 General framework of Sign Language recognition 23](#_Toc129030163)

[2.4 Sign language categories and datasets 24](#_Toc129030164)

[2.5 Image acquisition 26](#_Toc129030165)

[2.6 Pre-processing and segmentation 28](#_Toc129030166)

[2.7 Feature extraction 31](#_Toc129030167)

[2.8 Review of intelligent classification architectures in sign language recognitions 34](#_Toc129030168)

[2.9 Ensemble modelling or Hybrid neural network in Image classification 39](#_Toc129030169)

[2.10 Limitations and Future work 42](#_Toc129030170)

[2.11 Summary 43](#_Toc129030171)

[CHAPTER 3 45](#_Toc129030172)

[RESEARCH METHODOLOGY 45](#_Toc129030173)

[3.1 Introduction 45](#_Toc129030174)

[3.2 Methodology workflow 45](#_Toc129030175)

[3.3 Dataset Description: 47](#_Toc129030176)

[3.4 Data Pre-Processing 49](#_Toc129030177)

[3.4.1 Image cropping or resizing 50](#_Toc129030178)

[3.4.2 Image normalization 50](#_Toc129030179)

[3.4.3 Dimensional reduction: 51](#_Toc129030180)

[3.4.4 Data augmentation 52](#_Toc129030181)

[3.5 Model Architecture 55](#_Toc129030182)

[3.5.1 Convolutional Neural Network (CNN) 55](#_Toc129030183)

[3.5.2 LSTM 58](#_Toc129030184)

[3.6 Hybrid Neural Network (Proposed Model) 60](#_Toc129030185)

[3.6.1 Introduction 60](#_Toc129030186)

[3.6.2 Design 60](#_Toc129030187)

[3.6.3 Assumptions 61](#_Toc129030188)

[3.6.4 Results 61](#_Toc129030189)

[3.6.5 Limitations: 61](#_Toc129030190)

[3.6.6 Advantage 62](#_Toc129030191)

[3.7 Model Evaluation 62](#_Toc129030192)

[CHAPTER 4 ANALYSIS AND IMPLEMENTATION 65](#_Toc129030193)

[4.1 Introduction 65](#_Toc129030194)

[4.2 Tools 65](#_Toc129030195)

[4.3 Dataset Preparation 66](#_Toc129030196)

[4.4 CNN Model – 1 70](#_Toc129030197)

[4.5 CNN Model – 2 73](#_Toc129030198)

[4.6 CNN Model – 3 75](#_Toc129030199)

[4.7 Hybrid Neural Network (Proposed model) 78](#_Toc129030200)

[CHAPTER 5 83](#_Toc129030201)

[RESULTS AND EVALUATION 83](#_Toc129030202)

[5.1 Introduction 83](#_Toc129030203)

[5.2 Results 83](#_Toc129030204)

[5.3 Evaluation matrix 84](#_Toc129030205)

[5.4 Prediction with Hybrid Model 86](#_Toc129030206)

[5.4.1 Confusion matrix 86](#_Toc129030207)

[5.4.2 Prediction 87](#_Toc129030208)

[CHAPTER 6 88](#_Toc129030209)

[Conclusion and Future Scope 88](#_Toc129030210)

[6.1 Conclusions and Discussion 88](#_Toc129030211)

[6.2 Contribution to Knowledge 88](#_Toc129030212)

[6.3 Future Scope 89](#_Toc129030213)

[References 90](#_Toc129030214)

[APPENDIX A: RESEARCH PROPOSAL 100](#_Toc129030215)

[Abstract 101](#_Toc129030216)

[1. Background 102](#_Toc129030217)

[2. Problem Statement OR Related Research OR Related Work 102](#_Toc129030218)

[4. Aim and Objectives 107](#_Toc129030219)

[5. Significance of the Study 107](#_Toc129030220)

[6. Scope of the Study 108](#_Toc129030221)

[7. Research Methodology 108](#_Toc129030222)

[8. Requirements Resources 113](#_Toc129030223)

[9. Research Plan 114](#_Toc129030224)

[10. Risk & Contingency Plans 114](#_Toc129030225)

[References 114](#_Toc129030226)

# TABLE OF CONTENTS

[TABLE OF CONTENTS 5](#_Toc129030227)

[LIST OF FIGURES 10](#_Toc129030228)

[LIST OF TABLES 11](#_Toc129030229)

[Abstract 12](#_Toc129030230)

[CHAPTER 1 INTRODUCTION 13](#_Toc129030231)

[1.1 Background 13](#_Toc129030232)

[1.2 Problem Statement 15](#_Toc129030233)

[1.3 Aim and Objectives 18](#_Toc129030234)

[1.4 Significance of the Study 19](#_Toc129030235)

[1.5 Scope of the Study 19](#_Toc129030236)

[1.6 Structure of the study 19](#_Toc129030237)

[CHAPTER 2 LITERATURE REVIEW 21](#_Toc129030238)

[2.1 Introduction 21](#_Toc129030240)

[2.2 Word Cloud 22](#_Toc129030241)

[2.3 General framework of Sign Language recognition 23](#_Toc129030242)

[2.4 Sign language categories and datasets 24](#_Toc129030243)

[2.5 Image acquisition 26](#_Toc129030244)

[2.6 Pre-processing and segmentation 28](#_Toc129030245)

[2.7 Feature extraction 31](#_Toc129030246)

[2.8 Review of intelligent classification architectures in sign language recognitions 34](#_Toc129030247)

[2.9 Ensemble modelling or Hybrid neural network in Image classification 39](#_Toc129030248)

[2.10 Limitations and Future work 42](#_Toc129030249)

[2.11 Summary 43](#_Toc129030250)

[CHAPTER 3 45](#_Toc129030251)

[RESEARCH METHODOLOGY 45](#_Toc129030252)

[3.1 Introduction 45](#_Toc129030254)

[3.2 Methodology workflow 45](#_Toc129030255)

[3.3 Dataset Description: 47](#_Toc129030256)

[3.4 Data Pre-Processing 49](#_Toc129030257)

[3.4.1 Image cropping or resizing 50](#_Toc129030258)

[3.4.2 Image normalization 50](#_Toc129030259)

[3.4.3 Dimensional reduction: 51](#_Toc129030260)

[3.4.4 Data augmentation 52](#_Toc129030261)

[3.5 Model Architecture 55](#_Toc129030262)

[3.5.1 Convolutional Neural Network (CNN) 55](#_Toc129030263)

[3.5.2 LSTM 58](#_Toc129030264)

[3.6 Hybrid Neural Network (Proposed Model) 60](#_Toc129030265)

[3.6.1 Introduction 60](#_Toc129030266)

[3.6.2 Design 60](#_Toc129030267)

[3.6.3 Assumptions 61](#_Toc129030268)

[3.6.4 Results 61](#_Toc129030269)

[3.6.5 Limitations: 61](#_Toc129030270)

[3.6.6 Advantage 62](#_Toc129030271)

[3.7 Model Evaluation 62](#_Toc129030272)

[CHAPTER 4 ANALYSIS AND IMPLEMENTATION 65](#_Toc129030273)

[4.1 Introduction 65](#_Toc129030275)

[4.2 Tools 65](#_Toc129030276)

[4.3 Dataset Preparation 66](#_Toc129030277)

[4.4 CNN Model – 1 70](#_Toc129030278)

[4.5 CNN Model – 2 73](#_Toc129030279)

[4.6 CNN Model – 3 75](#_Toc129030280)

[4.7 Hybrid Neural Network (Proposed model) 78](#_Toc129030281)

[CHAPTER 5 83](#_Toc129030282)

[RESULTS AND EVALUATION 83](#_Toc129030283)

[5.1 Introduction 83](#_Toc129030285)

[5.2 Results 83](#_Toc129030286)

[5.3 Evaluation matrix 84](#_Toc129030287)

[5.4 Prediction with Hybrid Model 86](#_Toc129030288)

[5.4.1 Confusion matrix 86](#_Toc129030289)

[5.4.2 Prediction 87](#_Toc129030290)

[CHAPTER 6 88](#_Toc129030291)

[Conclusion and Future Scope 88](#_Toc129030292)

[6.1 Conclusions and Discussion 88](#_Toc129030294)

[6.2 Contribution to Knowledge 88](#_Toc129030295)

[6.3 Future Scope 89](#_Toc129030296)

[References 90](#_Toc129030297)

[APPENDIX A: RESEARCH PROPOSAL 100](#_Toc129030298)

[Abstract 101](#_Toc129030299)

[1. Background 102](#_Toc129030300)

[2. Problem Statement OR Related Research OR Related Work 102](#_Toc129030301)

[4. Aim and Objectives 107](#_Toc129030302)

[5. Significance of the Study 107](#_Toc129030303)

[6. Scope of the Study 108](#_Toc129030304)

[7. Research Methodology 108](#_Toc129030305)

[8. Requirements Resources 113](#_Toc129030306)

[9. Research Plan 114](#_Toc129030307)

[10. Risk & Contingency Plans 114](#_Toc129030308)

[References 114](#_Toc129030309)

TABLE OF CONTENTS

[TABLE OF CONTENTS 5](#_Toc129030310)

[TABLE OF CONTENTS 8](#_Toc129030311)

[LIST OF FIGURES 10](#_Toc129030312)

[LIST OF TABLES 11](#_Toc129030313)

[Abstract 12](#_Toc129030314)

[CHAPTER 1 INTRODUCTION 13](#_Toc129030315)

[1.1 Background 13](#_Toc129030316)

[1.2 Problem Statement 15](#_Toc129030317)

[1.3 Aim and Objectives 18](#_Toc129030318)

[1.4 Significance of the Study 19](#_Toc129030319)

[1.5 Scope of the Study 19](#_Toc129030320)

[1.6 Structure of the study 19](#_Toc129030321)

[CHAPTER 2 LITERATURE REVIEW 21](#_Toc129030322)

[2.1 Introduction 21](#_Toc129030323)

[2.2 Word Cloud 22](#_Toc129030324)

[2.3 General framework of Sign Language recognition 23](#_Toc129030325)

[2.4 Sign language categories and datasets 24](#_Toc129030326)

[2.5 Image acquisition 26](#_Toc129030327)

[2.6 Pre-processing and segmentation 28](#_Toc129030328)

[2.7 Feature extraction 31](#_Toc129030329)

[2.8 Review of intelligent classification architectures in sign language recognitions 34](#_Toc129030330)

[2.9 Ensemble modelling or Hybrid neural network in Image classification 39](#_Toc129030331)

[2.10 Limitations and Future work 42](#_Toc129030332)

[2.11 Summary 43](#_Toc129030333)

[CHAPTER 3 45](#_Toc129030334)

[RESEARCH METHODOLOGY 45](#_Toc129030335)

[3.1 Introduction 45](#_Toc129030336)

[3.2 Methodology workflow 45](#_Toc129030337)

[3.3 Dataset Description: 47](#_Toc129030338)

[3.4 Data Pre-Processing 49](#_Toc129030339)

[3.4.1 Image cropping or resizing 50](#_Toc129030340)

[3.4.2 Image normalization 50](#_Toc129030341)

[3.4.3 Dimensional reduction: 51](#_Toc129030342)

[3.4.4 Data augmentation 52](#_Toc129030343)

[3.5 Model Architecture 55](#_Toc129030344)

[3.5.1 Convolutional Neural Network (CNN) 55](#_Toc129030345)

[3.5.2 LSTM 58](#_Toc129030346)

[3.6 Hybrid Neural Network (Proposed Model) 60](#_Toc129030347)

[3.6.1 Introduction 60](#_Toc129030348)

[3.6.2 Design 60](#_Toc129030349)

[3.6.3 Assumptions 61](#_Toc129030350)

[3.6.4 Results 61](#_Toc129030351)

[3.6.5 Limitations: 61](#_Toc129030352)

[3.6.6 Advantage 62](#_Toc129030353)

[3.7 Model Evaluation 62](#_Toc129030354)

[CHAPTER 4 ANALYSIS AND IMPLEMENTATION 65](#_Toc129030355)

[4.1 Introduction 65](#_Toc129030356)

[4.2 Tools 65](#_Toc129030357)

[4.3 Dataset Preparation 66](#_Toc129030358)

[4.4 CNN Model – 1 70](#_Toc129030359)

[4.5 CNN Model – 2 73](#_Toc129030360)

[4.6 CNN Model – 3 75](#_Toc129030361)

[4.7 Hybrid Neural Network (Proposed model) 78](#_Toc129030362)

[CHAPTER 5 83](#_Toc129030363)

[RESULTS AND EVALUATION 83](#_Toc129030364)

[5.1 Introduction 83](#_Toc129030365)

[5.2 Results 83](#_Toc129030366)

[5.3 Evaluation matrix 84](#_Toc129030367)

[5.4 Prediction with Hybrid Model 86](#_Toc129030368)

[5.4.1 Confusion matrix 86](#_Toc129030369)

[5.4.2 Prediction 87](#_Toc129030370)

[CHAPTER 6 88](#_Toc129030371)

[Conclusion and Future Scope 88](#_Toc129030372)

[6.1 Conclusions and Discussion 88](#_Toc129030373)

[6.2 Contribution to Knowledge 88](#_Toc129030374)

[6.3 Future Scope 89](#_Toc129030375)

[References 90](#_Toc129030376)

[APPENDIX A: RESEARCH PROPOSAL 100](#_Toc129030377)

[Abstract 101](#_Toc129030378)

[1. Background 102](#_Toc129030379)

[2. Problem Statement OR Related Research OR Related Work 102](#_Toc129030380)

[4. Aim and Objectives 107](#_Toc129030381)

[5. Significance of the Study 107](#_Toc129030382)

[6. Scope of the Study 108](#_Toc129030383)

[7. Research Methodology 108](#_Toc129030384)

[8. Requirements Resources 113](#_Toc129030385)

[9. Research Plan 114](#_Toc129030386)

[10. Risk & Contingency Plans 114](#_Toc129030387)

[References 114](#_Toc129030388)

Figure 2.2‑1 Word cloud of 100 most used keyword in SLR 23

Figure 2.2‑2 No. of Research papers published for SL in last decade 24

Figure 2.3‑1 : Sign Language recognition workflow using deep learning 25

*Figure 3.2‑1: Methodology workflow for AI based Sign Language recognition* 47

*Figure 3.3‑1: American Sign Language alphabets* 49

*Figure 3.3‑2: Sign Language Digits Dataset* 50

Figure 3.5‑1 Graphical description of CNN architecture 57

Figure 3.5‑2 Convolutional layer is broken down into its elementary components. 58

Figure 3.6‑1 Graphical description of proposed Hybrid Model 62

Figure 3.7‑1 Confusion matrix 64

Figure 6.3‑1 : Sign Language recognition workflow using deep learning 104

Figure 6.3‑1: American Sign Language alphabets 110

Figure 6.3‑2: Sign Language Digits Dataset 111

Figure 6.3‑3: Methodology workflow for AI based Sign Language recognition 113

Figure 6.3‑1: Gantt Chart 115

# LIST OF FIGURES

[Figure 2.2‑1 Word cloud of 100 most used keyword in SLR 23](#_Toc129030404)

[Figure 2.2‑2 No. of Research papers published for SL in last decade 24](#_Toc129030405)

[Figure 2.3‑1 : Sign Language recognition workflow using deep learning 25](#_Toc129030406)

[*Figure 3.2‑1: Methodology workflow for AI based Sign Language recognition* 47](#_Toc129030407)

[*Figure 3.3‑1: American Sign Language alphabets* 49](#_Toc129030408)

[*Figure 3.3‑2: Sign Language Digits Dataset* 50](#_Toc129030409)

[Figure 3.5‑1 Graphical description of CNN architecture 57](#_Toc129030410)

[Figure 3.5‑2 Convolutional layer is broken down into its elementary components. 58](#_Toc129030411)

[Figure 3.6‑1 Graphical description of proposed Hybrid Model 62](#_Toc129030412)

[Figure 3.7‑1 Confusion matrix 64](#_Toc129030413)

[Figure 6.3‑1 : Sign Language recognition workflow using deep learning 104](#_Toc129030414)

[Figure 6.3‑1: American Sign Language alphabets 110](#_Toc129030415)

[Figure 6.3‑2: Sign Language Digits Dataset 111](#_Toc129030416)

[Figure 6.3‑3: Methodology workflow for AI based Sign Language recognition 113](#_Toc129030417)

[Figure 6.3‑1: Gantt Chart 115](file:///C:\Users\Ambar.Saxena\Desktop\MS\Final_Thesis\Final_Thesis_Ambar_Saxena_v02.docx#_Toc129030418)

# LIST OF TABLES

[Table 2‑0‑1 Summarized benchmark datasets from different countries used by various researchers 27](#_Toc129030419)

[Table 0‑2 Advantage and disadvantages of some acquisition devices 29](#_Toc129030420)

[Table 0‑3 Advantage and disadvantage of image enhancement techniques 31](#_Toc129030421)

[Table 0‑4 Advantage and disadvantage of filtering techniques 32](#_Toc129030422)

[Table 0‑5 . Summary of various advantages and disadvantages of feature extraction techniques 34](#_Toc129030423)

[Table 0‑1 : List of Hardware and Software required 115](#_Toc129030424)

[Table 0‑1: Associated risk and mitigation plan 115](#_Toc129030425)

# Abstract

Hand gestures serve as the primary means of communication in sign languages, which are composed of visual gestures made up of hands, faces, and other bodily motions. Automatic sign language identification is a challenging system due to the diversity of the about 7000 modern sign languages as well as variances in motion position, hand shape, and body part positioning. Although sign language has become more common in recent years, communicating with sign language speakers or signers remains difficult for non-sign language speakers. There has been promising progress in the disciplines of motion and gesture detection utilizing Artificial Intelligent techniques as a result of recent advances in deep learning and computer vision.

The deep learning network makes full use of the advantages of time series classification provided by the recurrent neural network model as well as the feature extraction capabilities of convolutional neural networks in order to achieve more accurate recognition. High precision, scalability, and robustness, on the other hand, remain significant issues in sign language recognition research. The purpose of this research is to examine hybrid neural networks to improve the accuracy and robustness of sign language recognition. with satisfactory results in terms of several performance measures such as accuracy, precision, recall, F1-Score, etc.

# CHAPTER 1 INTRODUCTION

There are over 300 sign languages used by 70 million deaf people worldwide, according to the World Federation of the Deaf (Murray, 2018). However, most communication technologies have been designed for spoken or written language, creating barriers for sign language users in society. While tools such as Imo (Pagebites, 2018) and WhatsApp (Acton & Koum, 2009) have become ubiquitous in daily life, they are not always accessible or effective for deaf people. This can create difficulties in communication between the deaf community and the hearing majority. The development of sign language recognition technology has the potential to bridge this communication gap and facilitate more seamless and inclusive communication for the deaf community.

## Background

A speech impediment is a disorder that affects a person's capacity to communicate through speech and hearing. People who are afflicted by this employ alternative communication methods such as sign language. Although sign language has become more common in recent years, communicating with sign language speakers or signers remains difficult for non-sign language speakers.

Like spoken languages, sign languages have their own grammar and syntax, and can vary significantly depending on the region or culture in which they are used. Sign languages vary from country to country, just like spoken languages. For example, American Sign Language (ASL) is different from British Sign Language (BSL), which is different from Japanese Sign Language (JSL). Sign language recognition and translation systems powered by artificial intelligence (AI) have the potential to significantly improve communication between individuals who use sign language and those who do not.

Artificial Intelligence (AI) has made significant advancements in the field of computer vision in recent years. Computer vision is the ability of machines to interpret and understand images and video. With the help of AI techniques such as deep learning, computer vision algorithms can now recognize objects, faces, and other important features in images and videos with remarkable accuracy.

AI has greatly improved the ability of computer vision systems to perform tasks such as object detection, classification, and segmentation. Object detection involves locating objects within an image or video and drawing bounding boxes around them. Classification involves identifying what the objects are, while segmentation involves separating the objects from their background. AI-based computer vision systems can also be used for facial recognition, which has applications in security, surveillance, and biometric authentication. In addition, AI-based computer vision is being used in the automotive industry for self-driving cars, in medical imaging for disease detection and diagnosis, and in retail for tracking customer behavior and improving store layouts.

Deep learning techniques such as Convolutional Neural Networks (CNNs) have been particularly effective in computer vision. CNNs can automatically learn and identify features within images, making them highly effective in object recognition and classification. Other AI techniques such as Recurrent Neural Networks (RNNs) and Generative Adversarial Networks (GANs) have also been applied in computer vision with promising results.

Sign language recognition is a specific application of computer vision, which is the field of artificial intelligence that enables machines to interpret and understand visual data such as images and videos. Sign language recognition involves using computer vision algorithms to analyze and interpret sign language gestures and movements, with the goal of accurately translating those gestures into natural language.

The development of AI-powered sign language recognition systems has been driven by the increasing availability of large datasets of sign language videos, as well as advances in deep learning techniques and computer vision. These systems use computer algorithms to analyze sign language videos and identify the specific signs being used. This information can then be translated into spoken or written language, or even used to generate sign language animations or avatars that can be used in virtual or augmented reality environments.

One of the key challenges in developing sign language recognition systems is the variability of sign language across different regions and cultures, as well as the variability in the way that different individuals use sign language. To address this challenge, researchers have developed machine learning algorithms that can be trained on large datasets of sign language videos to learn to recognize and translate specific signs, as well as to identify patterns and features that are common across different sign languages and signing styles.

In recent years, hybrid neural networks, which combine multiple neural network architectures to improve the accuracy and performance of image classification systems, have shown significant promise. These hybrid models can combine the strengths of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to recognize the temporal and spatial features of sign language videos more accurately.

Another area of research in AI-powered sign language recognition is the development of gesture recognition systems that can identify and classify nonverbal cues, such as facial expressions and body posture, that are an important part of sign language communication. This can help to improve the accuracy and expressiveness of sign language translations, and can also be useful in other applications, such as emotion recognition and human-computer interaction.

In addition to improving communication between individuals who use sign language and those who do not, AI-powered sign language recognition systems can also be used to improve accessibility in a variety of settings, such as education, healthcare, and public services. For example, sign language recognition systems can be used to provide real-time translations of lectures, doctor-patient consultations, or emergency services communications.

the role of artificial intelligence in sign language recognition has the potential to revolutionize communication between individuals who use sign language and those who do not. With the help of deep learning techniques, computer vision, and hybrid neural networks, AI-powered sign language recognition systems are becoming more accurate and versatile and are opening up new opportunities for improved accessibility and inclusion in a variety of domains.

## Problem Statement

The problem of communication barriers between individuals who use sign language and those who do not has long been recognized, and it can have serious implications in education, healthcare, and other areas of life. Although sign language interpretation and translation services are available, they are often expensive and not readily available in all settings, and there is a shortage of qualified interpreters in many parts of the world. This can lead to a lack of access to important information and services for individuals who use sign language.

The development of AI-powered sign language recognition systems has the potential to address this problem by providing real-time translation and interpretation services that are accurate, efficient, and widely available. However, there are still significant challenges to be overcome, such as the variability of sign language across different regions and cultures, the complexity of sign language grammar and syntax, and the need to accurately capture nonverbal cues such as facial expressions and body posture.

There are several approaches to sign language recognition, including computer vision-based approaches and data-driven approaches. Computer vision-based approaches involve the use of cameras and sensors to capture sign language gestures, while data-driven approaches involve the use of machine learning algorithms to analyze pre-recorded sign language videos or images. Additionally, the development of these systems requires large datasets of sign language videos, which can be difficult and expensive to obtain, especially for less commonly used sign languages or for specific signing styles. There is also a need to ensure that the development and implementation of these systems is culturally sensitive and inclusive, and that they do not reinforce existing biases or stereotypes.

While computer-based sign language recognition systems have been developed, their accuracy has been limited by the variability of sign language across different regions and cultures, as well as the variability in the way that different individuals use sign language. This variability can make it challenging for algorithms to accurately recognize and translate sign language, and can limit the accessibility and effectiveness of these systems.

One of the key challenges in developing accurate and robust machine learning models for sign language recognition is the availability of large, diverse, and annotated datasets. To address this challenge, many researchers have focused on the development of new datasets and benchmarks for evaluating sign language recognition systems. These datasets include the RWTH-BOSTON-50 dataset, which contains 50 different sign language gestures performed by five different signers, and the Phoenix-2014-T corpus, which contains more than 10,000 annotated video recordings of German Sign Language.

In recent years, advances in artificial intelligence (AI) and machine learning have led to significant progress in this field, including the development of hybrid neural networks that combine multiple neural network architectures to improve the accuracy and performance of sign language recognition systems. One of the challenges in sign language recognition is the variability in the way that different individuals use sign language, including differences in the speed and style of signing, as well as variations in the use of facial expressions and other nonverbal cues. To address this variability, researchers have developed techniques for adapting hybrid neural networks to the specific characteristics of individual signers, in order to improve the accuracy of the model's translations.

One type of hybrid neural network that has shown promise for sign language recognition is the combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are effective at capturing local features of the sign language gestures, such as the shape of the hands and fingers, while RNNs are good at modeling the temporal dynamics of the sign language, such as the sequence of gestures. By combining the strengths of these two types of neural networks, researchers have been able to develop hybrid models that can accurately recognize sign language gestures and translate them into spoken or written language.

Another technique that has been used to improve the accuracy of sign language recognition systems is the use of attention mechanisms, which allow the model to focus on the most relevant features of the sign language gestures and to ignore irrelevant information. This can improve the model's ability to recognize sign language gestures that are occluded or partially obscured, and to handle variability in the way that different individuals use sign language.

Ensemble learning algorithms are heavily used in novel and high-performance medical image classification pipelines. The goal behind ensemble learning is to combine different models or forecasts to improve prediction performance. To improve generalization performance, ensemble learning mixes numerous separate models. Deep ensemble learning models incorporate the benefits of both deep learning models and ensemble learning. The primary purpose of research is to examine hybrid neural networks while training various 3DCNN and ensemble networks to increase the accuracy and precision of sign language recognition. The aim of this study is to use deep learning networks in artificial intelligence to recognize sign language gestures automatically.

In addition to the technical challenges of sign language recognition, there are also practical considerations, such as the limited availability of training data, especially for less commonly used sign languages or dialects. To address this challenge, researchers have developed methods for synthesizing additional training data using computer graphics or by manipulating existing data to create new examples. Despite the progress that has been made in sign language recognition using neural networks, there is still much work to be done to fully realize the potential of AI for sign language recognition. In the future, researchers will likely continue to explore new techniques and architectures for sign language recognition, and to refine existing approaches to further improve the accuracy and performance of these systems.

AI empowered sign language recognition using hybrid neural networks is a rapidly evolving field with significant potential to improve the lives of individuals who use sign language. By combining the strengths of different neural network architectures, researchers are making progress towards developing accurate and reliable sign language recognition systems that can translate sign language into spoken or written language in real-time.

## Aim and Objectives

The main aim of this research is to analyze hybrid neural networks while training different 3DCNN and ensemble networks to improve the accuracy and robustness of sign language recognition. The aim of the research is to automatically identify the sign language gestures using Artificial Intelligence’s deep learning networks.

The following are the research objectives, which are based on the goal of this study:

* To develop and evaluate the effectiveness of various deep learning models
* To recommend an appropriate deep learning-based network for the recognition of sign language.
* To compare the classification models to find the model that best categorizes sign language gestures.
* To assess the effectiveness of suggested models

## Significance of the Study

A language made up of signals made from hand motions and facial expressions is known as sign language. Those with weak or no hearing use this language to communicate. Hearing-impaired persons may express themselves more freely and more effectively using this kind of communication, which also helps to close the communication gap between them and others.

The transdisciplinary problem of automatic identification of human signs has not yet been entirely resolved. For the recognition of sign language, a variety of methods, including the use of machine learning techniques, have been employed recently.

This study attempts to increase the precision of automatic sign language gesture identification by training a hybrid deep learning network.

## Scope of the Study

The scope of this study is limited to:

* Pre-processing and recognizing static sign language using deep learning networks , CNN, LSTM etc.
* Generating insights on model accuracy with various networks
* The study does not cover Dynamic or continues sign language gestures.

## Structure of the study

Introduction: This section provides an overview of the research problem, context, and significance of the study. It also includes the research questions or hypotheses, objectives, and a brief summary of the methodology used in the study.

Literature Review: This section reviews and synthesizes existing research, theories, and best practices related to the topic of the study. It includes an analysis and critique of previous studies and identifies gaps in the literature that the current study seeks to address.

Methodology: This section explains the research design, data collection methods, and analysis techniques used in the study. It should provide a clear and concise explanation of the methodology that was followed, including any assumptions made and any limitations or potential sources of bias.

Results: This section presents the findings of the study based on the data analysis conducted. This may include tables, figures, and graphs to illustrate the results of the study.

Discussion: This section interprets the results and provides an analysis of their implications. It should also compare and contrast the results with previous research and identify areas where the study contributes new knowledge to the field.

Conclusion: This section summarizes the main findings and conclusions of the study, and discusses their implications and limitations. It should also identify areas for future research and offer suggestions for practical applications of the research.

References: This section lists all the sources cited in the paper and includes any additional sources consulted during the research process.

In addition to these core sections, the study may also include other sections such as an abstract, acknowledgments, appendices, and supplementary materials depending on the scope and nature of the research.

# CHAPTER 2 LITERATURE REVIEW

Several studies have been conducted on the use of AI in sign language recognition. In recent years, researchers have focused on developing deep learning models to improve the accuracy of sign language recognition systems. These models are trained on large datasets of sign language videos to learn the patterns and gestures used in sign language. AI-empowered sign language recognition systems have the potential to revolutionize communication for the deaf and hard-of-hearing communities. With the use of deep learning models and large datasets, these systems can achieve high accuracy in recognizing signs. Future research in this area can focus on developing more robust and versatile sign language recognition systems that can be used in a variety of applications.



## Introduction

Sign language recognition involves interpreting visual signals from the movements and gestures made by a signer's hands, face, and body. These visual signals are captured by a camera and then analyzed and interpreted by a computer system. As such, sign language recognition is considered a computer vision problem because it requires processing visual data to recognize patterns and extract information. Computer vision is a field of study that focuses on enabling machines to interpret and understand visual information from the world around them, making it a natural fit for sign language recognition.

Deep learning is a subset of machine learning that utilizes artificial neural networks with many layers to learn and represent data with multiple levels of abstraction. It has been found to outperform traditional machine learning techniques in various areas, particularly in Computer Vision and Natural Language Processing. Several deep learning models have been used in computer vision, including Convolutional Neural Network (CNN), Deep Boltzmann Machine (RBM), Deep Belief Network (DBN), Auto Encoder (AE), Variational Auto Encoder (VAE), Generative Adversarial Network (GAN), and Recursive Neural Network (RNN), including Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

One of the primary advantages of deep learning is that it eliminates the need for hand-crafted features, as the computational models of multiple processing layers can learn and represent data on their own. This approach is inspired by the structure and function of the human brain and can capture complex structures in large-scale data. The availability of large-scale, high-quality, and publicly available labeled datasets, as well as parallel GPU computing and the development of strong frameworks like TensorFlow, Theano, and MXNET, have contributed significantly to the advancement of deep learning.

## Word Cloud

he figure labeled as "Fig. 3" displays a word cloud that shows the top 100 most commonly used keywords in the research on sign language recognition. The most frequent keywords in the cloud relate to the algorithm used for sign language recognition, such as "CNN", "gestures", "image", and "classification". Additionally, the term "deep learning" has attracted the attention of researchers as a significant area for sign language recognition techniques.

Text, whiteboard

Description automatically generated

Figure 2.2‑1 Word cloud of 100 most used keyword in SLR

To give you an idea of the trend of research papers published in sign language recognition using AI over a longer period of time, I searched for the number of results on Google Scholar using the keywords "sign language recognition deep learning" and filtering by year from 2012-2022. Here are the number of results for each year:

Figure 2.2‑2 No. of Research papers published for SL in last decade

It's important to note that these numbers are not exhaustive and only represent the papers that can be found on Google Scholar using these specific keywords. Additionally, not all of these papers may be directly related to sign language recognition using AI, as the search is not filtered for relevance. Nonetheless, these numbers suggest that the amount of research being done in this area has increased significantly over the past decade.

## General framework of Sign Language recognition

The majority of studies have concentrated to detect the static sign-language motions in pictures or video-clips that were acquired in a testing setting. Image capture, data pre-processing, image segmentation, extraction of features, and classification are the five categories into which the processes of Image-based sign language recognition (SLR) is classified. Gesture acquisition, the first step in sign language recognition, mostly carried out utilizing datasets that were either self-acquired or made publicly available. Preprocessing, the next step, enhances image quality by removing unnecessary noise. Following preprocessing, the area of interest is segmented and extracted from the complete picture. The input image area is converted into feature-vectors for detection in the fourth stage. To identify the target sign, classification, the final stage in vision-based SLR includes comparing the attributes of the input image to those already located in the database.

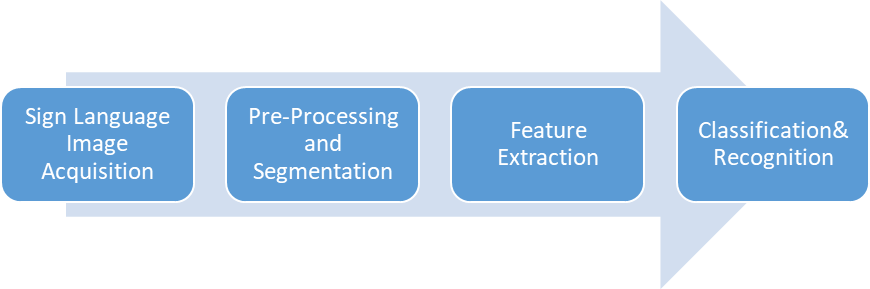


Figure 2.3‑1 : Sign Language recognition workflow using deep learning

## Sign language categories and datasets

Sign languages serve the same purpose as spoken languages but have a distinct structure with their own phonology, syntax, morphology, vocabulary, and grammar. Signs are made with one or both hands in a specific configuration close to the signer's body, using five basic parameters that influence the sign's meaning: handshape, location, palm orientation, movement, and facial expression. The meaning changes when any of these parameters change. Sign languages vary across countries due to social, geographical, and linguistic factors, with different sign languages evolving in deaf communities worldwide. While most countries sharing a spoken language do not share the same sign language, challenges in developing a sign language recognition system include differences in signing speed, image segmentation, and gesture tracking, among others. (Agrawal et al., 2016; Cheok et al., 2019; Kadhim & Khamees, 2020; Lemaster & Monaghan, 2007; Sahoo et al., 2014; Wadhawan & Kumar, 2021).

.A number of benchmark datasets are available to evaluate the performance of sign language recognition systems, including those that focus on static, isolated, and continuous signing. One such dataset is the Purdue RVL-SLLL dataset, which comprises 2576 videos of 14 signers performing various gestures, movements, words, and sentences in American Sign Language (ASL). The dataset includes 39 isolated motion primitives, 62 hand shapes, and sentences. Another dataset for ASL, the American Sign Language Lexicon Video Dataset (ASLLVD), features high-quality video sequences of around 3800 ASL signs corresponding to roughly 3000 signals signed by four native signers. The RWTHBOSTON-104 and RWTHBOSTON-400 datasets were created to develop isolated and continuous ASL recognition, respectively. The RWTHBOSTON-104 dataset includes isolated sign language with a vocabulary of 104 signs and 201 sentences signed by three signers, while the RWTHBOSTON-400 dataset comprises 843 sentences with a vocabulary size of 406 words signed by four signers. The Massey University dataset consists of 36 classes of alphabets (A-Z) and numbers (0-9), totaling 2160 images.

For Arabic Sign Language (ArSL), the Sign Language Database contains 40 sentences repeated 19 times, acquired from 80 signers. The Signs World Atlas dataset includes around 500 static gestures, such as finger spelling and hand motions, as well as dynamic gestures involving body language, lip reading, and facial expressions. The LIBRAS-HCRGBDS dataset focuses on Brazilian Sign Language and consists of 61 hand configurations from five signers, recorded using a Kinect sensor. The British Sign Language (BSL) Corpus created the British Sign Language dataset, which includes videos of 249 people conversing in BSL with annotations of 6330 gestures. Two datasets were created for German sign language recognition: the RWTH-PHOENIX-Weather 2014 dataset, which features continuous sign language of 6861 sentences and 1558 vocabularies, and the SIGNUM Database, which includes a vocabulary of 450 basic gestures and 780 sentences signed by 25 signers.

Three high-quality datasets are available for studies of Polish Sign Language (PSL) variation, including the PSL Kinect 30, PSL ToF 84, and PSL 101 datasets, which contain isolated words totaling between 30 and 101 signs and are performed by only one person. Indian researchers can use the Sign corpus IITA-ROBITA ISL, which was developed collaboratively between 2010 and 2017 and contains only 23 signs performed by a single signer. Of all the available datasets, two stand out for their universal usability: ASLLVD and RWTH-PHOENIX-Weather. These publicly available sign language sets are suitable for interpreting sign language in real-world conditions and are often used as benchmarks in SLR studies to determine the effectiveness of proposed computing techniques.

Table 2‑0‑1 Summarized benchmark datasets from different countries used by various researchers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dataset Name** | **Country** | **No. of Classes** | **No. of Samples** | **Recording Devices** | **Reference** |
| RWTH-BOSTON-50 | Germany/USA | 50 | 20,000 | Webcam | [1] |
| SIGNUM | India | 44 | 2,500 | Data Glove | [2] |
| SSL | Spain | 35 | 3,500 | Leap Motion | [3] |
| BosphorusSL | Turkey | 29 | 10,591 | Kinect | [4] |
| JSL | Japan | 31 | 6,764 | Video Camera | [5] |
| CSL | China | 44 | 12,080 | Video Camera | [6] |
| MOCAP-SLR | USA | 13 | 5,200 | Motion Capture System | [7] |
| MSASL | USA | 2000+ | 30,000+ | Video Camera | [8] |
| LIBRAS | Brazil | 64 | 1,020 | Video Camera | [9] |

## Image acquisition

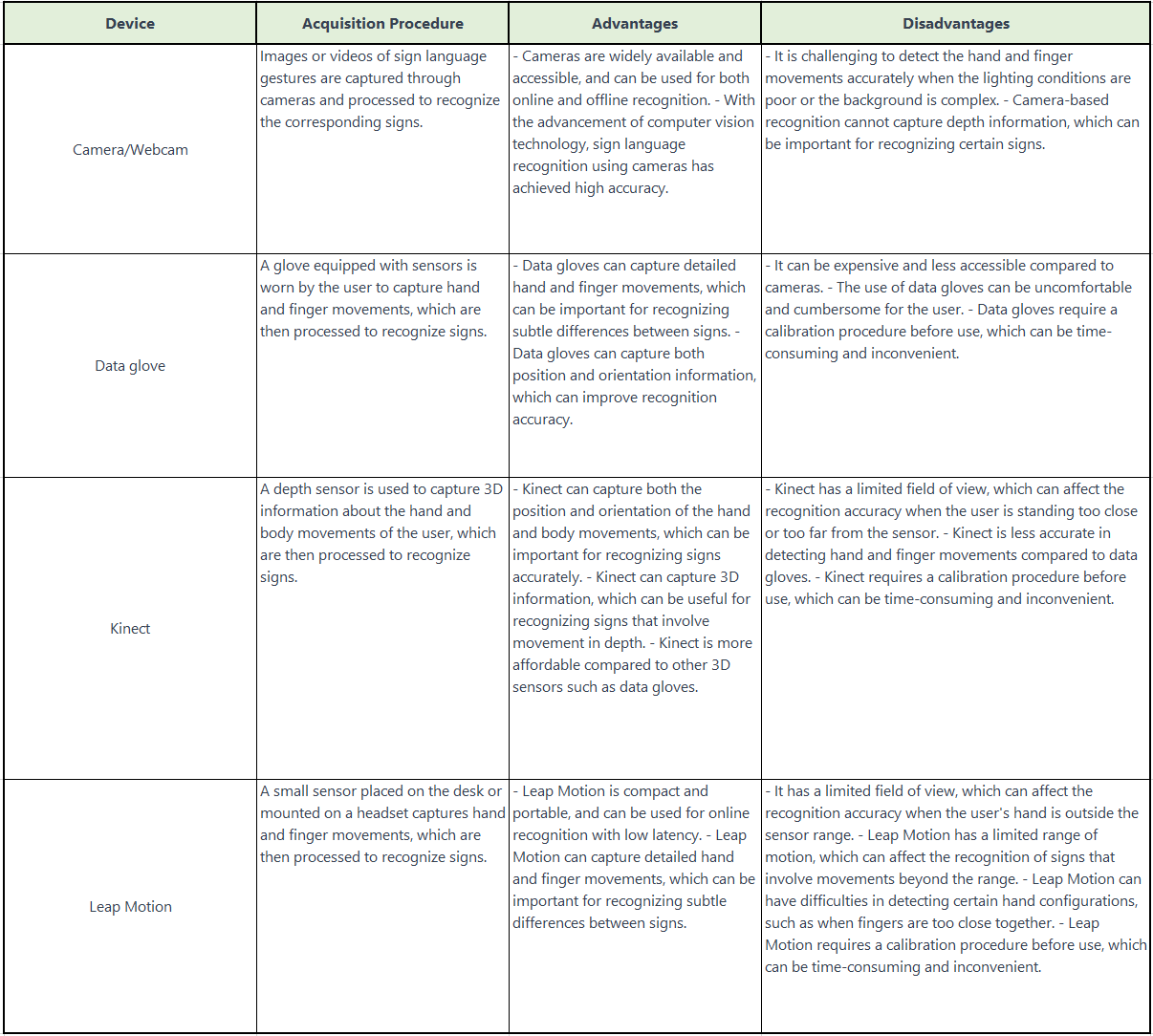
Many researchers use previously defined datasets like ASL Image Dataset (ASLID) , ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) -2010 , ASL Gesture Dataset 2012, RWTH-Phoenix-Weather dataset, ArSL databases and the SIGNUM (Subburaj and Murugavalli, 2022) . Few researchers develop their own datasets for data training due to scarcity of Sign-Language datasets for specific countries. To establish a dataset, researchers create the data from recording the signer. Image capturing techniques of various types were employed by the researchers. Devices used in image capturing are webcams or camera, data-gloves, Kinect, and Leap motion controller. Kinect is a useful and commonly used device. It offers a depth video feed as well as a color video stream. It easily recovers 3D hand motion trajectories and distinguishes the genuine sign image from the background (Kamal et al., 2019). Although the leap motion controller has a smaller operational range than Kinect, it is less expensive and more accurate (Suharjito et al., 2017).

researchers collected sign images for classification using several image collecting devices. The camera or webcam, data glove, Kinect, and leap motion controller are examples of these gadgets (Kamal et al., 2019). Among these devices, a camera or webcam is the most commonly employed by many researchers since it allows better and more natural human-computer connection without the need for extra devices, as opposed to data glove-based interaction. Data gloves have proven to be more accurate in data gathering, but they are also quite expensive and uncomfortable for consumers. Kinect is popular and effective. It gives both a colour video and a depth video stream at the same time. It quickly distinguishes the background from the real sign image and extracts 3D hand motion trajectories (Kamal et al., 2019). The disadvantage of Kinect is its high price.

The leap motion controller has a restricted operating range, however it is a low-cost device with more accuracy than Kinect (Suharjito et al., 2017). The use of a camera for the acquisition of sign images can be found in the literature (Wang et al., 2010; Mekala et al., 2011; Kumarage et al., 2011; Pansare & Ingle, 2016; Athira et al., 2019; Sharma et al., 2021). In hand gesture recognition research proposed by Mehdi and Khan (2002), Gao et al. (2004), Phi et al. (2015), and Pan et al. (2015), data gloves were employed to capture sign images (2020). Jiang et al. (2015), Wang et al. (2015a, 2015b, 2016), Raheja et al. (2016), Carneiro et al. (2017), and Escobedo and Camara (2017) all employed Kinect to capture sign images.

For sign image acquisition, similar experiments used a jump motion controller (Kiselev et al., 2019; Alnahhas et al., 2020; Enikeev & Mustafina, 2021). Table 3 lists the benefits and drawbacks of various data collecting devices for classifications. These tests demonstrated that the images acquired were either static or dynamic, taken as frames of photographs in various postures, backgrounds, and lighting conditions.

Table 0‑2 Advantage and disadvantages of some acquisition devices



## Pre-processing and segmentation

The image pre-processing phase optimizes the input with image and video modification. Preprocessing techniques are used to remove non desirable noise from an input image while simultaneously improving its quality. This can be performed through scaling, color conversion, noise removal, or a mix of these approaches from the original image. With a solid selection of preprocessing procedures, the output of this process can have a significant impact on accuracy. Image enhancement and picture restoration are the two primary categories of image preprocessing techniques.

Gaussian and Median filters are often used techniques for minimizing noise in images or videos. The removal of unwanted information from the input is frequently accomplished using morphological operations (Akmeliawati et al., 2007) and median filtering, which is only employed for picture pre-processing in research (Raghuveera et al., 2020). As an illustration, Badhe et al (2015) used K-means clustering in conjunction with morphological techniques to remove noise after thresholding the input image into binary.

Kishore and kumar, 2012 applied a series of preprocessing steps to the video frames, including noise removal, thresholding, and contour detection.Firstly, noise removal was performed using a Gaussian filter to remove high-frequency noise from the video frames. Then, the frames were thresholded to segment the hand region from the background. This was achieved by setting a threshold value, which separated the foreground pixels from the background pixels.

K.M. Lim. Et al 2016 used a set of preprocessing techniques to enhance the contrast of the hand gesture images and remove unwanted background noise.The preprocessing steps included:Converting the images to grayscale. Applying histogram equalization to improve contrast and brightness. Removing background noise by thresholding the image. Detecting the hand region using skin color segmentation and morphological operations. Removing small image components to reduce the impact of noise.Resizing the image to a fixed size.

P.C. Badhe, V. Kulkarni, 2015 and R. Akmeliawati, M.P. Ooi, Y.C. Kuang 2007, used morphological operations to refine the segmentation process during the image processing pipeline.After the hand region is detected using skin color segmentation, the authors perform morphological operations to remove noise and fill any gaps in the segmented hand region. They use two morphological operations: erosion and dilation. Erosion is used to remove any small isolated regions that are not part of the hand. This is achieved by convolving the binary image with a small structuring element that removes small details. Dilation is used to fill any gaps in the segmented hand region. This is achieved by convolving the binary image with a larger structuring element that expands the hand region. The authors apply erosion and dilation alternately to the segmented hand region until a satisfactory result is obtained. This improves the accuracy of the hand segmentation and helps to remove any noise that may interfere with the feature extraction process.

The technique of breaking an image into discernible pieces is called image segmentation (Egmont-Petersen et al., 2002). It's possible for the segmentation method to be contextual or not. There are two basic approaches used for segmentation; contextual and non-contextual segmentation ([Enikeev & Mustafina, 2020](https://www.sciencedirect.com/science/article/pii/S2667305321000454" \l "bib0069); [Jin et al., 2016](https://www.sciencedirect.com/science/article/pii/S2667305321000454" \l "bib0104); [Al-Shamayleh et al., 2020](https://www.sciencedirect.com/science/article/pii/S2667305321000454" \l "bib0023)). While non-contextual segmentation groups pixels based on global properties, contextual segmentation, like edge identification techniques, considers the spatial connection between highlights. Edge detection, thresholding, region, clustering, and artificial neural network dependent image segmentation techniques are some of the categories.

Table 0‑3 Advantage and disadvantage of image enhancement techniques

|  |  |  |  |
| --- | --- | --- | --- |
| **Enhancement Technique** | **Description** | **Advantages** | **Disadvantages** |
| Histogram Equalization | Adjusts pixel intensity distribution to better utilize the full range of values | Simple and fast, works well with gray-scale images | Can cause over-enhancement of certain regions and loss of image detail |
| Contrast Stretching | Adjusts pixel intensity range to increase contrast | Simple and fast, preserves detail in regions with low contrast | Can result in loss of detail in high contrast regions |
| Gamma Correction | Adjusts the gamma value to change image brightness and contrast | Can be used to compensate for non-linear response of cameras or display devices, allows for fine control over brightness and contrast | May introduce color distortion if not used carefully |
| Unsharp Masking | Enhances edges by subtracting a blurred version of the image from the original | Preserves fine image detail while enhancing edges | Can result in noise amplification and halo artifacts around edges |
| Frequency Domain Filtering | Applies filters in the frequency domain to enhance or suppress certain features | Can be used to remove noise or sharpen edges, allows for fine control over the filtering process | Requires knowledge of Fourier transforms and filter design |
| Wavelet Transform | Decomposes the image into different frequency bands using wavelets | Can be used for multi-resolution image analysis and denoising | Can result in ringing artifacts and introduce edge discontinuities |

Table 0‑4 Advantage and disadvantage of filtering techniques

|  |  |  |
| --- | --- | --- |
| **Filtering Technique** | **Advantages** | **Disadvantages** |
| Gaussian filter | Smoothes out noise and blurring of edges | May over-smooth or blur important details |
| Median filter | Removes salt-and-pepper noise without blurring edges | May not be effective for other types of noise |
| Bilateral filter | Smooths out noise without blurring edges | May be slower than other filters and requires more computation |
| Wiener filter | Removes noise while preserving important details | May not be effective for certain types of noise |
| Homomorphic filter | Adjusts image contrast to enhance details | May require extensive parameter tuning |
| Laplacian filter | Enhances edges and details | May produce unwanted artifacts or noise |
| Sobel filter | Enhances edges and gradients | May be sensitive to noise or produce unwanted artifacts |

## Feature extraction

The feature extraction method is employed to extract the most important properties from an input image. Picture backdrop, image translation, scale, form, rotation, angle, and coordinates are among the properties. It tries to identify the aspects of the acquired image that stand out the most (Patil and Sinha, 2017). It's a type of dimensional reduction in which the most interesting aspects of a photograph are successfully represented. In order to improve learning accuracy and precision, the unneeded component is removed to create the compact feature vector (Khalid et al., 2014). In order to achieve high recognition accuracy, the feature extraction procedure seeks for features that may successfully distinguish across classes. This contributes to the categorization step. PCA, histogram of oriented gradient (HOG), Fourier descriptor (FD), and shift-invariant analysis are the some of the primary feature extraction approaches utilized in SLR that have produced positive results.

S.S. Shivashankara, S. Srinath, 2018 extract features from the preprocessed image using the Histogram of Oriented Gradients (HOG) method. HOG computes the gradient magnitude and orientation of the image at each pixel, and then bins the orientations into histogram cells. The histogram cells are then normalized and concatenated into a feature vector.The authors also use Principal Component Analysis (PCA) to reduce the dimensionality of the feature vector, which helps to speed up the classification process and improve accuracy. The reduced feature vector is then fed into a k-nearest neighbor (k-NN) classifier for classification.

Principal Component Analysis (PCA) is a widely used technique in image processing for feature extraction (Aliyu et al., 2020). It involves a statistical procedure that transforms a series of potentially correlated variables into a set of values for non-correlated variables using an orthogonal transformation (Kumar & Bhatia, 2014). The process of PCA includes calculating new variables, which are referred to as principal components, and using these new variables to create a linear combination of the initial variables. In the PCA process, the principal components are calculated in a way that produces the highest potential variance, and then the larger portion of it is extracted. The eigenvectors and corresponding eigenvalues are then computed. The computed eigenvectors are stored by decreasing the order of the eigenvalues, which form the dimensionality reduction of data in PCA. The algorithm used to calculate PCA is given in studies by Cheok et al. (2019) and Karamizadeh et al. (2013).

The Histogram of Oriented Gradient (HOG) is a method of extracting features that is utilized in image processing for object identification. The HOG-derived features provide a condensed and effective representation of an image that is useful for image classification purposes. It has become one of the most reliable techniques for feature extraction, as it can identify shapes or structures present in an image with great accuracy. Joshi et al 2020 combined the Taguchi method and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) to select the optimal parameters for HOG feature extraction. HOG features were extracted from the preprocessed image using various combinations of HOG parameters such as cell size, block size, and number of orientations. The proposed method was evaluated on a dataset of 50 sign language gestures performed in complex background environments, and achieved an accuracy of 94.2% in sign language recognition.

A. Dudhal, H. Mathkar, A. Jain, O. Kadam, M. Shirole 2019 A hybrid approach is proposed for feature extraction, which combines Scale-Invariant Feature Transform (SIFT) and Convolutional Neural Networks (CNN). First, SIFT features are extracted from the preprocessed image. Then, a pre-trained CNN is used to extract deep features from the SIFT features. The proposed approach was evaluated on a dataset of 800 ISL gesture images, and achieved an overall accuracy of 96.75%. The results suggest that the proposed hybrid feature extraction approach can accurately recognize and classify ISL gestures.

Speed Up Robust Feature (SURF) is an effective technique for feature points extraction . Narayan et al 2018 used the Speeded-Up Robust Features (SURF) feature descriptor and Laplacian eigenmaps, SURF features were extracted from the preprocessed image and Laplacian eigenmaps algorithm was used to reduce the dimensionality of the SURF features and to project them onto a lower-dimensional space. The proposed method was evaluated on a dataset of 30 ISL word signs, and achieved an overall recognition rate of 90%.

The type of features extracted determines the performance of the recognition. Therefore, it is necessary to choose a good feature extraction technique to achieve good performance. [Table 8](https://www.sciencedirect.com/science/article/pii/S2667305321000454" \l "tbl0008) summarise the advantages and disadvantages of various feature extraction techniques used in vision-based sign language recognition systems.

Table 0‑5 . Summary of various advantages and disadvantages of feature extraction techniques

|  |  |  |
| --- | --- | --- |
| **Feature Extraction Technique** | **Advantages** | **Disadvantages** |
| Discrete Cosine Transform (DCT) | Provides a high degree of data compression. Relatively insensitive to image variations. | Lossy compression technique. Information is lost during the compression process. |
| Fourier descriptors | Good at capturing shape information, rotation and scale invariant | Sensitive to noise, may not capture texture information |
| Principal Component Analysis (PCA) | Can reduce the dimensionality of the data. Provides a lower-dimensional representation of the data. | Information is lost during the dimensionality reduction process. PCA is sensitive to outliers in the data. |
| Scale Invariant Feature Transform (SIFT) | Robust to changes in image scale, rotation, and illumination. Provides distinctive features that are invariant to affine distortion. | High computational cost. SIFT is sensitive to occlusion and clutter in the image. |
| Speeded Up Robust Feature (SURF) | Robust to changes in image scale, rotation, and illumination. Faster than SIFT. | High computational cost. 2. SURF is sensitive to occlusion and clutter in the image. |
| Histogram of Oriented Gradients (HOG) | 1. Robust to changes in image scale, rotation, and illumination. 2. Provides a simple and efficient feature representation. | 1. HOG is sensitive to occlusion and clutter in the image. 2. HOG may not be suitable for highly textured images. |
| Local Binary Patterns (LBP) | 1. Provides a simple and efficient feature representation. 2. Robust to changes in image illumination. | 1. LBP is sensitive to image noise. 2. LBP may not be suitable for images with complex textures. |

## Review of intelligent classification architectures in sign language recognitions

After the images have undergone pre-processing, segmentation, and feature extraction, a prediction method must be used to help give the extracted features useful meaning. The last step and most crucial one for the acceptance of gestures is classification. Machine learning enhances performance because it enables machines to learn much like humans do by performing jobs repeatedly. Importing and using models from published research is common practice because creating such models requires significant computational resources.

The use of artificial intelligent techniques in sign language recognition can be categorized into supervised and unsupervised learning. Supervised machine learning involves a process where a set of known training data is used to infer a function from labelled training data, whereas unsupervised machine learning draws inferences from datasets with input data that has no labelled response. After conducting an extensive literature review, various intelligent predictors have been commonly utilized for sign language recognition, such as k-nearest neighbour (KNN), artificial neural network (ANN), support vector machine (SVM), hidden Markov Model (HMM), Convolutional Neural Network (CNN), fuzzy logic and ensemble learning. This section briefs overview of the machine learning techniques used for sign language recognition. Numerous studies have employed machine learning methods for the recognition or classification of sign language, which are reviewed and presented in subsequent sections.

**K-NN** is a type of machine learning algorithm that utilizes distance metrics as its fundamental characteristics. Several sign language recognition techniques employ classification methods based on distance measurement. T.N.T. Huong et al 2016 trained a KNN classifier to recognize the hand gestures based on the PCA features. The proposed method was evaluated on a dataset of 700 hand gesture images, and achieved an overall recognition rate of 96.29%. However, the main limitation of the proposed method is that it can only recognize static hand gestures and not dynamic ones. The authors suggested extending the proposed method to recognize dynamic hand gestures by incorporating a temporal component. They also suggested exploring other feature extraction techniques, such as wavelet transform or local binary patterns, to improve the recognition accuracy. Sharma et al 2020 used K-Nearest Neighbor (K-NN) classifier to classify the hand gesture images based on the extracted features. The authors chose the K-NN classifier due to its simplicity and ability to handle non-linear data. The classifier was trained on a dataset of 560 hand gesture images and tested on a separate dataset of 240 images, achieving an overall recognition rate of 94.25%. The limitation of the K-NN classifier is that it requires a large amount of training data to achieve high accuracy. Future work may involve exploring the use of other classifiers such as support vector machines (SVMs) or artificial neural networks (ANNs) to improve the recognition accuracy.

**Artificial Neural Network (ANN)** is a type of computational algorithm that is inspired by the human brain's biological nervous system. The ANN's highly interconnected networks can compute input values and perform parallel computations for data processing and knowledge representation. It is a subfield of artificial intelligence (AI) that aids in constructing predictive models from extensive databases. ANN is a versatile and adaptive tool that has been utilized to carry out computations such as pattern recognition, matching, and classification. The ANN is typically characterized by three parameters: the interconnection pattern that exists between various layers of neurons, the weight of these interconnections, and the activation function. Raj and Jasuja (2018) developed a system based on Artificial Neural Network (ANN) to identify British sign language alphabets. The dataset used for testing had 780 sign images, and the system achieved an accuracy of 99.01%, which outperformed similar research by Liwicki and Everingham (2009), which had a recognition accuracy of 98.9%. Islam et al. (2017), on the other hand, presented a real-time system for recognizing hand gestures using American Sign Language (ASL). The proposed system used Artificial Neural Network (ANN) with feed forward, back propagation algorithm for training a network using 30 feature vectors to recognize 37 signs of American alphabets and numbers properly. The total gesture recognition rate of this system was 94.32% in real time environment.

**Support Vector Machine (SVM**) is a supervised learning model with associated learning algorithms that are non-probabilistic. It is a popular pattern recognition learning technique for classification regression analysis. SVM can be used to solve both pattern-classification and nonlinear-regression problems, but it is most useful in solving difficult pattern-classification problems. Classification is performed in SVM by differentiating between two or more data classes**.** Athira et al 2017 proposed a system for signer-independent sign language recognition from live videos in the Indian context. The system employs a combination of computer vision and machine learning techniques to recognize signs in real-time. An appropriate feature vector is extracted from the gesture sequence after co-articulation elimination phase. The obtained features are then used for classification using Support Vector Machine(SVM). The system successfully recognized finger spelling alphabets with 91% accuracy and single-handed dynamic words with 89% accuracy. Barbhuiya, et al 2021 in their research used a pre-trained CNN to extract features from the sign language videos. The features are then fed into a separate classifier to recognize the sign. The authors experimented with several pre-trained CNN architectures and found that the VGG16 model performed the best. In the classification stage, the authors used several classifiers, including support vector machines (SVMs) to recognize the sign based on the extracted features. They found that the SVM classifier performed the best, achieving an accuracy of 98.5% .Overall, the paper demonstrates the effectiveness of CNN-based feature extraction and SVM classification for sign language recognition.

[Suharjito et al. (2019)](https://www.sciencedirect.com/science/article/pii/S2667305321000454" \l "bib0254) compared the performance of different types of **hidden Markov models (HMMs)** for sign language recognition. The authors discussed the importance of HMMs in sign language recognition, as they are commonly used to model the temporal dynamics of sign language gestures. They introduced models for predicting ten signs of Argentine sign language by utilizing sign Gaussian Hidden Markov Model (HMM) and Multinomial HMM. They found that when edge detection was employed, Gaussian HMM outperformed Multinomial HMM with a recognition accuracy of 83%.

**Convolutional neural networks (CNNs)** are a type of deep neural network that are commonly used in computer vision applications. They are particularly well-suited for tasks such as image classification, object detection, and image segmentation. At a high level, a CNN consists of multiple layers, each of which performs a different type of computation. The first layer is typically a convolutional layer, which applies a set of learnable filters to the input image to extract local features. The output of this layer is then passed through a non-linear activation function, such as a rectified linear unit (ReLU), to introduce non-linearity into the model. The output of the activation function is then passed through a pooling layer, which down samples the output of the previous layer to reduce its dimensionality and improve computational efficiency. This process of applying a series of convolutional, activation, and pooling layers is repeated multiple times to learn increasingly complex features.

After the convolutional layers, the output is typically flattened into a one-dimensional vector and passed through one or more fully connected layers, which perform a global aggregation of the features learned by the convolutional layers. The final layer of the network is typically a softmax layer, which produces a probability distribution over the possible classes. The weights of a CNN are typically learned through backpropagation and stochastic gradient descent, where the objective is to minimize the difference between the predicted output of the network and the ground-truth labels. CNNs have achieved state-of-the-art performance in many computer vision tasks and are widely used in both academia and industry.

The CNN architecture used in the study by Tao et al 2018 consists of five convolutional layers and two fully connected layers. The authors experimented with different hyperparameters, such as the number of filters in each layer and the dropout rate, and used grid search to find the optimal set of hyperparameters. To improve the recognition accuracy, the authors used an inference fusion technique that combines the predictions from multiple CNNs trained on different subsets of the training data. Jain et al 2012 presented a study on American Sign Language (ASL) recognition using both Support Vector Machines (SVM) and Convolutional Neural Networks (CNN). The authors proposed a methodology for collecting and preprocessing ASL image data, which involves capturing images of hand gestures using a webcam and converting them to grayscale. They then extract features from the images, such as the edges and contours of the hands, and use these as input to the SVM and CNN models for classification. The performance of the SVM and CNN models are compared, and the results show that the CNN outperforms the SVM in terms of accuracy, precision, recall, and F1 score.

Himda et al 2022 presented solution of static alphabet sign recognition by deep learning based on the Alexnet model of categorical cross-entropy as a loss function. They compared its performance with other models namely ResNet50 and Lenet5. Experimental evaluations showed that this methos can recognize Arabic alphabets in most constraining situations. They demonstrated its performance through a qualitative and quantitative evaluation based on famous ArSL2018 images. Their method achieved an average F1-score rate of 97.38% of F-measure against a rate of 97.2% with Resnet50 and 95.28% with Lenet5. Wadhawan and Kumar (2020) concentrate on very widely used scalable networks that are transferred in transfer learning. AlexNet, created by LeNet, is a commonly used deep learning architecture. They evaluated more than 50 CNN models. The data were further evaluated using multiple optimizers, and it was determined that the suggested technique attained the maximum training efficiency of 99.72% on colorful pictures and 99.90% on monochrome images.

Samir Aliyev et al 2022 presented a study on using machine learning techniques for recognizing Azerbaijani Sign Language (AzSL) gestures for mobile platform. They collected a dataset of AzSL gestures and used various feature extraction techniques to represent the gestures. They adapted MobileNet v2 CNN architecture for training and classification. They achived model accuracy of 85 %. Rathi D. (2018) used the GoogLeNet neural network, which has depth in both the directions. Very deep network with high accuracy.

Sharma and kumar 2021 employed 3DCNN which was trained for classification of 100 words on Boston ASL (Lexicon Video Dataset) LVD dataset with more than 3300 English words signed by 6 different signers. 70% of the dataset is used for Training while the remaining 30% dataset is used for testing the model. The proposed work outperforms the existing state-of-art models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%) with accuracy of 96% . Masood Sarfaraz and Thuwal,( 2018) classified 2624 ASL gestures using a pre-trained VGG16 model. Simonyan and Zisserman (2014) presented the VGG network architecture. Although the ResNet architecture can be successfully trained at deep depths, VGG-16 networks are regarded as being exceptionally deep. Sadly, VGG has two significant drawbacks: It is quite slow during training, and the network weights are fairly enormous.

Mannan et al. (2022) presented the performance of a DeepCNN architecture which improves with size of the dataset given, they used data augmentation to expand the input data artificially. They proposed a new deep convolutional neural network (CNN) architecture for sign language recognition that is optimized using a hyperparameter tuning method. The authors evaluated the proposed CNN architecture on the American Sign Language (ASL) dataset and achieved a classification accuracy of 97.06%. They compared their results with several state-of-the-art models, and the proposed CNN outperformed all other models. Katoch et al. (2022) suggested a technique uses the sign langue gestures in a live video stream and predict labels in the form of texts. Segmentation was based on skin color as well as background too . SVM and CNN are used for classification. Bheda and Radpour, (2017) suggested a letter and digit recognition system based on American Sign Language. The proposed CNN-based architecture produced an accuracy of 82.5% . Rao et al., (2018) created a dataset that shows signs at different angles and against different backdrops. They classified them using CNN with different pooling algorithms and the stochastic pooling approach beat the other pooling strategies.

M. E. Morocho Cayamcela and W. Lim 2019 proposed a real-time American Sign Language (ASL) translation system that uses a pre-trained convolutional neural network (CNN) model as a feature extractor, and fine-tunes the model on a smaller dataset of ASL signs. The proposed system achieves high accuracy in recognizing ASL signs, and is capable of translating them into English text in real-time. The study demonstrates the effectiveness of fine-tuning a pre-trained CNN model for sign language recognition. Yirtici and Yurtkan, (2022) employed AlexNet as a pre-trained network, transfer learning method was used to classify TSL with average precision of 99.7% . Gupta, (2022) investigated different sign language motions and created a comparative evaluation of various neural network architectures. For the sign language recognition system, CNN models (GoogLeNet, AlexNet, VGGNet, and EfficientNet) were employed and investigated. The benchmarked ASL dataset was used for testing.

## Ensemble modelling or Hybrid neural network in Image classification

Ensemble modeling and hybrid neural networks are approaches that can be used in computer vision tasks. Ensemble modeling involves combining the predictions of multiple models to improve performance. In computer vision, this can be done by training multiple models on the same dataset and then combining their outputs using techniques such as averaging or weighted voting. Ensemble modeling has been applied to various image classification tasks, such as object recognition, facial recognition, and medical image analysis. It has been shown to improve the accuracy of image classification, especially when individual models have high variance or are prone to overfitting. On the other hand, a hybrid neural network is a type of neural network that combines different types of layers or architectures in order to address specific problems. In computer vision, a hybrid neural network might combine convolutional layers, which are commonly used in image processing, with other types of layers such as recurrent layers for sequential data processing or attention layers for focusing on specific regions of an image. By combining these different layer types, a hybrid neural network can be more effective at handling complex data and producing accurate predictions. Both ensemble modeling and hybrid neural networks can be effective approaches in computer vision, and the choice between them will depend on the specific task and available resources.

Koller et al., (2018) suggested a hybrid technique for continuous sign recognition that combines CNN's strong discriminative features with the Hidden Markov Model's sequence modeling capability (HMM). The obtained data was preprocessed using a dynamic programming method. According to the results, the hybrid CNN-HMM technique outperforms the other known approaches. Deep ensemble models have been used to classify X-ray and CT-scan images for disease diagnosis in medical applications. For instance, a stacked ensemble of CNNs was utilized by Kandel et al., (2021) to detect fractures in X-ray pictures. In order to provide input to eight different conceptual models, the authors combined the stochastic outputs of different CNN models. The ensemble model was said to be 10% more accurate than individual CNN models. Nakata et al 2023 proposed a method for multiclass classification of ultrasound images of hepatic masses using ensemble learning of multiple deep learning models. The method involves creating a base model using a convolutional neural network (CNN), then training multiple instances of the base model with different hyperparameters and data augmentation techniques. The outputs of these models are then combined using a weighted voting scheme to produce the final classification result. The proposed method was evaluated using a dataset of ultrasound images of hepatic masses and achieved higher accuracy and F1-score compared to other state-of-the-art methods.

Hybrid neural networks for image classification typically refer to models that combine different types of neural network architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and/or feedforward neural networks, to achieve higher accuracy and more efficient training. One example of a hybrid neural network for image classification is the Inception architecture, which was introduced in 2014 by Sergedy et al . Inception networks consist of several parallel convolutional pathways with different filter sizes and pooling operations, which are then concatenated to form the final output. This allows the network to learn features at different scales and resolutions, improving accuracy and reducing computational cost. Another example of a hybrid neural network is the residual network (ResNet) architecture, which was introduced in 2015. ResNets use skip connections that allow information to bypass certain layers and flow directly to deeper layers, addressing the problem of vanishing gradients and enabling the network to learn more complex features. Other hybrid neural networks for image classification may combine CNNs with RNNs to incorporate temporal information into the image classification process, or use pre-trained CNNs as feature extractors for other types of classifiers such as support vector machines or decision trees. Overall, hybrid neural networks for image classification offer a powerful approach to improve accuracy and efficiency by combining the strengths of different types of neural network architectures.

Winston et al 2022 proposed a hybrid deep convolutional neural network (CNN) model for iris image recognition. The model consists of three parts: a CNN feature extraction network, a hybrid CNN and deep belief network (DBN) classifier, and a shallow classifier. The CNN feature extraction network is used to extract features from the iris images. The hybrid CNN-DBN classifier is used to classify the extracted features. The shallow classifier is used to improve the classification performance of the hybrid model. The proposed model was evaluated on three iris image datasets and compared to several state-of-the-art models. The results show that the proposed hybrid model outperforms other models in terms of accuracy, sensitivity, and specificity. Alotaibi et al 2020 proposes a hybrid deep neural network model for hyperspectral image classification, which combines the strengths of residual network (ResNet) and Inception architectures. The model consists of a feature extraction network based on a modified ResNet and an Inception module for multi-scale feature learning, followed by a fully connected layer for classification. The proposed model was evaluated on three hyperspectral datasets and compared to several state-of-the-art models. The results show that the proposed hybrid model achieves higher classification accuracy than other models, demonstrating the effectiveness of combining ResNet and Inception architectures. Dhiman et al 2022 The paper proposes a novel machine-learning-based hybrid convolutional neural network (CNN) model for tumor identification in medical image processing. The model consists of a base CNN with several convolutional and pooling layers for feature extraction, followed by a hybrid classifier that combines a multilayer perceptron (MLP) and a support vector machine (SVM) to improve classification accuracy.

## Limitations and Future work

Sign language recognition using machine learning involves teaching computers to recognize and translate sign language gestures into text or speech. While machine learning approaches have shown promising results in this area, they also have limitations and areas for future improvement.

One of the primary limitations is the need for large and diverse datasets for training machine learning models. Sign language has many different dialects and variations, and it can be difficult to collect enough data to cover all of them. This can result in models that are biased towards certain types of signs or signing styles, leading to reduced accuracy and limited generalizability. Another limitation is the challenge of dealing with variations in hand position, movement, and speed. Sign language gestures can be performed in many ways, and machine learning models need to be able to recognize these variations in order to accurately classify signs. Furthermore, machine learning models may struggle to recognize signs that are similar in appearance or motion. This can lead to confusion between different signs and reduce the accuracy of the recognition system. Finally, machine learning models may not be able to handle continuous sign language, where signs are performed in a continuous sequence without pauses between them. This can make it difficult to accurately recognize signs and can limit the usability of sign language recognition systems. Addressing these limitations will require continued research and development in the field of sign language recognition using machine learning. This could involve collecting more diverse datasets, developing new algorithms and architectures to handle variations in sign language, and exploring new input modalities to capture sign language data in different ways.

There is still a lot of work to be done in sign language recognition using machine learning, and several potential areas for future research and development. One area of future work is the development of models that can handle continuous sign language, where signs are performed in a continuous sequence without pauses between them. This could involve developing new algorithms and architectures that can accurately recognize and classify signs even in a continuous stream of gestures. Another area for future work is the development of models that can adapt to individual users, accounting for differences in signing style and preference. This could involve using personalized data to train machine learning models or developing models that can be customized or fine-tuned by individual users. Additionally, researchers may explore new input modalities for capturing sign language data, such as wearable sensors or other types of cameras or motion capture systems. These new input modalities could potentially improve the accuracy and efficiency of sign language recognition. Finally, researchers may focus on developing more accurate and efficient models for sign language recognition, potentially through the use of new architectures, algorithms, or pre-training techniques. These advancements could help to further improve the accessibility and usability of sign language recognition systems, benefiting people with hearing impairments and promoting more inclusive communication. Overall, while machine learning approaches have made progress in sign language recognition, there is still much work to be done to improve accuracy, efficiency, and accessibility for people with hearing impairments.

## Summary

The literature review on sign language recognition using deep learning discussed various approaches for recognizing sign language gestures through deep learning techniques. The review highlights the challenges in sign language recognition, such as variations in hand position, orientation, and movement speed. The review covers different types of deep learning architectures used in sign language recognition, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models. The review also covers the different input modalities used in sign language recognition, such as RGB videos, depth sensors, and skeletal data. The authors compare the accuracy and efficiency of various deep learning models and input modalities used in sign language recognition. The review also discusses recent advancements in sign language recognition, such as the use of transfer learning and Hybrid neural networks to improve recognition accuracy. Additionally, the review highlights the importance of collecting large and diverse sign language datasets for training deep learning models.

Many academics are creating their own tiny datasets to use in the development of their SLR. There are still certain nations and languages for which large databases are unavailable. The type of sign language used in the majority of nations depends entirely on their grammar and how each phrase is presented, such as by utilizing words or phrases. The categorization method used to distinguish sign language varies between scholars as well. Comparing these method to each other in the Sign Language Recognition System remains arbitrary when using their notions and limitations. Based on deep learning approaches like CNN, LSTM, and Bi- LSTM Models exhibit high classification efficiency in a stream of images and videos. Overall, the literature review suggests that deep learning techniques have shown promising results in sign language recognition and have the potential to improve accessibility for people with hearing impairments. However, more research is needed to address the challenges and improve the accuracy and efficiency of sign language recognition systems.

# CHAPTER 3

# RESEARCH METHODOLOGY

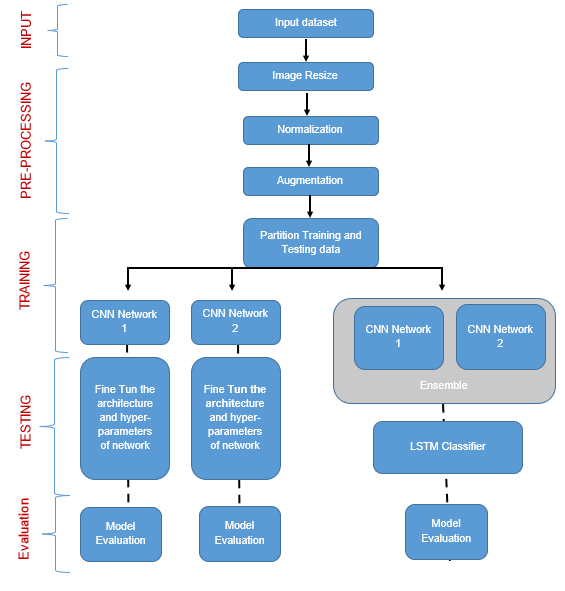


## Introduction

This chapter encompasses the research methodology and the supporting theoretical framework explaining each step in the methodology. It is dedicated to expounding on the technical jargons which will be used throughout the study. The research methodology will explain each step of the study in detail. These sections will include essential sign language recognition system operations like input data selection, pre-processing of the testing data, transformation into a structured and understandable data format, dataset-balancing, application of supervised deep learning techniques, and evaluation of machine learning performance using evaluation matrix. This section will include detailed introduction of the data used in the study. Different preprocessing techniques like image cropping, scaling, normalization, dimension reduction and data augmentation will be briefed. Detailed technical description of the classification methods used in the study will be explained, the confusion matrix and performance evaluation measures will be discussed in technical terms within analysis/evaluation section.

## Methodology workflow

The aim of the research experiment is to create a more efficient deep learning model that can accurately detect, decode, and translate American sign language alphabets and digits. The proposed model utilizes a custom-designed computer vision algorithm that can detect sign language gestures. The proposed methodology is to train few CNN networks individually to recognise sign languages and proposed model is to create an ensemble of these CNN networks and feed into LSTM classifier to achieve better model evaluation as compared to individual networks. Fig 3.1 describes a simple flowchart of the proposed methodology which includes data analysis, preprocessing, training-test split and different model training and evaluation.



*Figure 3.2‑1: Methodology workflow for AI based Sign Language recognition*

## Dataset Description:

American Sign Language (ASL) is a complete, natural language used by the Deaf and hard-of-hearing communities in the United States and some parts of Canada. According to a report from the National Institute on Deafness and Other Communication Disorders (NIDCD) in 2020, there are approximately 11 million people in the United States with hearing loss, and about 2 million of them are classified as deaf. ASL is the primary means of communication for many Deaf individuals in the United States. It is a visual language that uses hand gestures, facial expressions, and body movements to convey meaning. ASL is not based on spoken English, and it has its own grammar, syntax, and vocabulary. ASL is a rich and complex language, and it can be used to convey abstract concepts and ideas. It is also a highly expressive language that allows signers to convey emotions and attitudes. ASL is used in a variety of settings, including education, social interactions, and professional contexts. Like other languages, ASL has regional variations and dialects.

There are several types of American Sign Language (ASL) datasets that are commonly used for research and development of sign language recognition systems. These include:

1. Image-based ASL datasets: These datasets contain images of individual signs in ASL, and are commonly used for hand gesture recognition research. Examples of image-based ASL datasets include the American Sign Language Lexicon Video Dataset (ASLLVD) and the RWTH-BOSTON-50 dataset.
2. Video-based ASL datasets: These datasets contain video recordings of individuals signing complete sentences or phrases in ASL, and are commonly used for sign language recognition research. Examples of video-based ASL datasets include the Phoenix-14 and Phoenix-15 datasets.
3. Continuous signing ASL datasets: These datasets contain continuous video recordings of individuals signing for extended periods of time, and are commonly used for sign language recognition research in real-world scenarios. Examples of continuous signing ASL datasets include the American Sign Language Lexicon Video Corpus (ASLLVC) and the ASL Continuous Signing Corpus (ASL-CSC).
4. Multi-modal ASL datasets: These datasets contain both video and audio recordings of individuals signing in ASL, and are commonly used for research in sign language recognition and translation. Examples of multi-modal ASL datasets include the Multi-Modal Corpus of Sentences in Sign (MMCSS) and the ASL Speech Dataset.
5. Synthetic ASL datasets: These datasets contain synthetic or computer-generated ASL data, and are commonly used for training and testing of sign language recognition systems. Examples of synthetic ASL datasets include the SynthASL dataset and the ASL FingerSpelling dataset.

Overall, the availability of diverse and representative ASL datasets is crucial for the development of accurate and robust sign language recognition systems. This study used two different set of static American Sign language (ASL) datasets which consists of images of alphabets from the American Sign Language and sign language digits images respectively.

* **ASL Alphabet :** The data set contains images of American Sign Language alphabets grouped into 29 folders representing the various classes. Total 87,000 images with 200x200 pixel size sit in the training data set. There are 29 classes, 26 letters from A to Z overall, and three letters each for "SPACE," "DELETE," and "NOTHING." Real-time classification and applications benefit greatly from these three types. Only 29 photos make up the test data collection, which encourages the usage of test images taken in actual environments.



*Figure 3.3‑1: American Sign Language alphabets*

* **Sign Language Digits Dataset:** This Dataset is prepared by students of “Turkey Ankara Ayrancı Anadolu High School”. In this data there are 2062 sign language digits images. As you know digits are from 0 to 9. Therefore, there are 10 unique sign. The dataset consists of images with one-handed display of digits 0 to 9 in sign language. The images are 100x100 in size and RGB in color. It was obtained by 218 students making 10 different signs once. There should be a total of 2180 samples, while there are 2062 samples in the data set. This is probably because some unfavorable images have been removed by creator of the dataset. (Mavi, 2020)



*Figure 3.3‑2: Sign Language Digits Dataset*

## Data Pre-Processing

It is difficult to determine how to optimally prepare visual data while training a convolutional neural network. Data preparation is the use of several morphological methods to minimize noise in data. Before using this in any Deep Learning project, we may want to perform a few pre-processing tasks. Some of the most common are listed in the paragraphs below.

### Image cropping or resizing

Verifying the photos are the same size and aspect ratio is one of the first duties. The majority of neural network models require square input images, so each image must be examined to determine its squareness and then cropped appropriately. Image cropping and resizing are important preprocessing techniques in deep learning that are commonly used to prepare image data for training and testing deep learning models. Image cropping involves selecting a region of interest (ROI) from an image and discarding the remaining pixels. This technique is useful when the ROI is small compared to the size of the original image, and when there are irrelevant or distracting features outside of the ROI. By cropping the image to focus on the ROI, the computational cost of training deep learning models can be reduced, and the accuracy of the models can be improved.

Image resizing involves changing the size of images to a common resolution. This technique is useful when the size of images in the dataset is variable, and when the deep learning models require input images of a fixed size. By resizing the images to a common resolution, the models can be trained and tested more efficiently, and the accuracy of the models can be improved. There are several different methods that can be used for image cropping and resizing in deep learning. One common method is to crop or resize the images to a fixed size, such as 224x224 or 299x299 pixels, which is a common size for many deep learning models. Another method is to crop or resize the images to a size that preserves the aspect ratio of the original image, such as by setting the width or height to a fixed value and scaling the other dimension accordingly.

Saponara, S., Elhanashi, A. (2022) show that resizing the images can have a significant impact on the training time and model performance. The authors find that increasing the image size can lead to longer training times, but it also results in higher detection accuracy for all three models. They also find that reducing the image size can improve the training time, but it can also negatively impact the model's performance.

### Image normalization

Image normalization is a common preprocessing technique in deep learning applications that involves scaling the pixel intensity values of images to a common range. The goal of image normalization is to improve the quality and consistency of image data, making it easier for deep learning models to extract meaningful features and classify images accurately. In image normalization, the pixel intensity values of images are typically scaled to a range between 0 and 1 or -1 and 1. This is typically achieved by dividing each pixel by the maximum pixel intensity value in the image. For example, if an image has pixel intensity values between 0 and 255, each pixel value would be divided by 255 to scale the values to a range between 0 and 1. Alternatively, the values could be scaled to a range between -1 and 1 by subtracting the mean pixel intensity value and dividing by the standard deviation.

There are several benefits of image normalization in deep learning. One of the main benefits is that it can help to reduce the effects of lighting and shadowing in image data, making it easier to classify images accurately. By scaling the pixel intensity values of images to a common range, the effects of variations in lighting and shadowing can be reduced, making it easier for deep learning models to learn consistent features. Image normalization can also improve the stability and convergence of deep learning models during training. This is because image normalization can help to reduce the variability of pixel intensity values across different images, making it easier for the model to learn consistent features. By reducing the variability of pixel intensity values, image normalization can help to prevent overfitting and improve the generalization performance of deep learning models.

Kyung-Mo Koo and Eui-Young Cha (2017) explored the use of image normalization techniques for enhancing image recognition performance. The authors discuss the importance of image normalization in addressing variations in illumination, contrast, and color in images, which can negatively impact image recognition algorithms.

### Dimensional reduction:

Dimension reduction in image classification can involve transforming the original image data from a high-dimensional space to a lower-dimensional space while preserving the most important information. One example of dimension reduction in image classification is converting a color image in RGB format to a black and white (grayscale) image. RGB stands for Red, Green, and Blue, which are the three primary colors that are combined in various proportions to create a wide range of colors in digital images. Each pixel in an RGB image is represented by three values, one for each primary color, resulting in a high-dimensional dataset.

Converting an RGB image to grayscale involves reducing the dimensionality of the image data by collapsing the color channels into a single channel representing the brightness or intensity of each pixel. This is typically done by computing a weighted sum of the three color channels to create a single grayscale value for each pixel.

The formula for converting an RGB image to grayscale is:

Grayscale Value = 0.2989 x Red + 0.5870 x Green + 0.1140 x Blue

where the coefficients 0.2989, 0.5870, and 0.1140 represent the relative luminance of the red, green, and blue color channels, respectively.

By converting an RGB image to grayscale, the dimensionality of the image data is reduced from three channels (RGB) to one channel (grayscale), which can simplify the dataset and make it easier to process and analyze. This can be useful in situations where color information is not relevant to the task, or where the computational complexity of working with RGB images is too high. However, it is important to note that converting an RGB image to grayscale can result in some loss of information, particularly in situations where color information is important for the task at hand. Therefore, it is important to carefully consider the trade-offs between the reduction in dimensionality and the potential loss of information when choosing to convert an RGB image to grayscale for image classification purposes.

C. Saravanan (2010) proposed method which uses a weighted average approach to compute the grayscale value for each pixel in the image, where the weights are based on the luminance values of the red, green, and blue color channels. The authors evaluated the performance of the proposed method using several standard test images and compare it to other existing methods for color-to-grayscale conversion. The results show that the proposed method outperforms the other methods in terms of both visual quality and objective measures of image quality.

### Data augmentation

Data augmentation is a technique used in image classification that involves generating new images from existing ones to increase the size and diversity of a training dataset. This technique is used to enhance the generalization performance of a machine learning model by exposing it to a greater variety of examples. There are many ways to perform data augmentation in image classification, and the choice of augmentation technique depends on the specific application and the characteristics of the images being analyzed. Here are some common techniques used in data augmentation for image classification:

1. Flipping: Flipping an image horizontally or vertically can create a new image that is similar to the original but with a different orientation.
2. Rotating: Rotating an image by a small angle can create a new image that is similar to the original but with a different perspective.
3. Scaling: Scaling an image up or down can create a new image that is similar to the original but with a different size.
4. Cropping: Cropping an image to a different size or aspect ratio can create a new image that is similar to the original but with a different composition.
5. Adding noise: Adding random noise to an image can create a new image that is similar to the original but with a different texture.
6. Changing brightness and contrast: Adjusting the brightness and contrast of an image can create a new image that is similar to the original but with a different overall appearance.
7. Color shifting: Changing the color balance or hue of an image can create a new image that is similar to the original but with a different color palette.
8. Elastic deformation: Applying elastic deformations to an image can create a new image that is similar to the original but with a different shape and texture.

Data augmentation can be performed in real-time during training, where the original training images are transformed on the fly to create new images for each batch. Alternatively, data augmentation can be performed offline by pre-generating a set of augmented images and using them as part of the training dataset.

Agnieszka Mikołajczyk and Michał Grochowski (2018 ) discusses the use of data augmentation techniques to improve the performance of deep learning models in image classification problems. The authors explain that data augmentation can help to address the problem of overfitting that often occurs when training deep neural networks on small datasets. The paper presents a comprehensive review of various data augmentation techniques that can be applied to image datasets, including flipping, rotation, scaling, cropping, and adding noise. The authors also discuss the advantages and disadvantages of different augmentation techniques, and provide guidelines for selecting the most appropriate techniques based on the characteristics of the dataset and the application requirements. The paper then presents an experimental study that evaluates the effectiveness of data augmentation on several deep learning models for image classification tasks. The authors show that data augmentation can significantly improve the performance of these models, especially when the size of the training dataset is small. They also demonstrate that the choice of augmentation technique and the degree of augmentation can have a significant impact on the performance of the model.

While data augmentation is a powerful technique for improving the accuracy and robustness of image classification models, there are situations where it may not be appropriate or effective. Here are some cases where data augmentation may not be suitable:

1. When the dataset is already large and diverse: If the training dataset is already large and diverse, adding more augmented images may not provide significant benefits and may even increase the risk of overfitting.
2. When the dataset has specific characteristics: If the dataset has specific characteristics that are important for the application, such as color or texture, applying certain types of data augmentation may change these characteristics and reduce the performance of the model.
3. When the augmentation techniques are not relevant to the application: Some augmentation techniques, such as flipping or rotating, may not be relevant to certain applications, such as medical imaging or satellite imagery analysis, where the orientation of the objects being analyzed is fixed.
4. When the computational resources are limited: Data augmentation can significantly increase the computational requirements of training deep neural networks, and if the available resources are limited, it may not be feasible to perform extensive data augmentation.
5. When the model is already overfitting: If the model is already overfitting the training data, adding more augmented images may exacerbate the problem and decrease the performance of the model on the validation and test datasets.

In general, it is important to carefully consider the characteristics of the dataset and the application requirements before deciding whether to use data augmentation for image classification, and to carefully evaluate the performance of the model with and without data augmentation to determine the optimal strategy.

## Model Architecture

Model building for image classification involves creating a deep learning model that can accurately classify images into different categories. The goal of this process is to build a model that can learn the underlying patterns and features in the images, and use that knowledge to make accurate predictions on new, unseen images. Convolutional neural network (CNN) architecture are commonly used for image classification because they can learn hierarchical representations of the image features. There are several pre-trained CNN architectures available, such as VGG, ResNet, and Inception, which can be fine-tuned for specific classification task. Alternatively, one can design their own CNN architecture from scratch.

### Convolutional Neural Network (CNN)

Deep learning has proven to be a powerful tool for image classification in recent decades, as it can effectively process large amounts of data. Compared to traditional techniques, deep neural networks, especially Convolutional Neural Networks (CNN), have gained popularity in pattern recognition due to their ability to handle hidden layers. Convolutional Neural Networks (CNNs) have a rich history in the field of computer vision and have been instrumental in many breakthroughs in image classification, object detection, and other tasks

The concept of computer vision dates back to the 1950s, when researchers began attempting to develop systems that could understand visual data. Kunihiko Fukushima and Yann LeCun (1980) developed early versions of the architecture for character recognition tasks. However, it wasn't until the 2010s that CNNs began to achieve significant improvements in image classification accuracy. In 2012, a group of researchers at the University of Toronto made a significant breakthrough in computer vision when they created an AI model called AlexNet. Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton achieved a dramatic improvement in accuracy on the ImageNet dataset, which contains millions of images spanning thousands of categories. AlexNet used a deep architecture with multiple layers of convolutional and pooling operations, as well as dropout regularization, to achieve its high accuracy. This breakthrough sparked a renewed interest in deep learning and CNNs, and led to a wave of research on improving the architecture and training techniques of CNNs.

At the heart of AlexNet was a special type of neural network called Convolutional Neural Networks, which roughly mimics human vision. Since then, numerous advancements have been made in CNN architecture and training methods, including the development of deeper architectures such as VGGNet, ResNet, and InceptionNet, and the use of techniques such as batch normalization, residual connections, and attention mechanisms to improve performance. These advancements have enabled CNNs to achieve state-of-the-art results on a wide range of computer vision tasks, including image classification, object detection, semantic segmentation, and more.

The design of Convolutional Neural Networks (CNNs) was influenced by the structure of neurons in human and animal brains. The visual cortex in the brain of cats, for example, is formed by a complex sequence of cells, which is emulated by the CNN. Goodfellow et al. (2016) identified three crucial advantages of CNNs: equivalent representations, sparse interactions, and parameter sharing. Unlike conventional fully connected (FC) networks, CNNs utilize shared weights and local connections, allowing them to fully exploit the 2D input-data structures like image signals. Fig () describes the graphical description of CNN model. This approach uses very few parameters, which simplifies the training process and accelerates the network, similar to the way the visual cortex cells function. It is worth noting that these cells in the visual cortex extract only small regions of a scene rather than the entire scene, and spatially extract the local correlation present in the input, like local filters over the input.

Diagram

Description automatically generated

Figure 3.5‑1 Graphical description of CNN architecture

Convolutional Neural Networks (CNNs) work by using multiple layers to extract features from images and then using those features to classify the image.

The CNN architecture typically consists of three main types of layers: convolutional layers, pooling layers, and fully connected layers.

1. Convolutional layers: The convolutional layers are the building blocks of CNNs. These layers contain a set of filters or kernels that are applied to the input image. Each filter slides over the image and performs a dot product between the filter weights and the pixel values of the image. The output of the convolutional layer is a feature map that captures the presence of a particular feature in the input image. The filters in the convolutional layer learn to identify different features such as edges, corners, and other patterns. Fig() describes the convolution mechanism .
2. Pooling layers: After each convolutional layer, a pooling layer is added to the network. These layers down sample the output of the convolutional layer by taking the maximum or average value of a group of pixels. This reduces the size of the feature map and helps to make the network more robust to variations in the input image.
3. Fully connected layers: The final layer in a CNN is typically a fully connected layer. This layer takes the output from the previous layers and uses it to classify the image. The fully connected layer is similar to a traditional neural network layer, where each neuron is connected to every neuron in the previous layer. The fully connected layer uses the features learned by the previous layers to predict the output class.

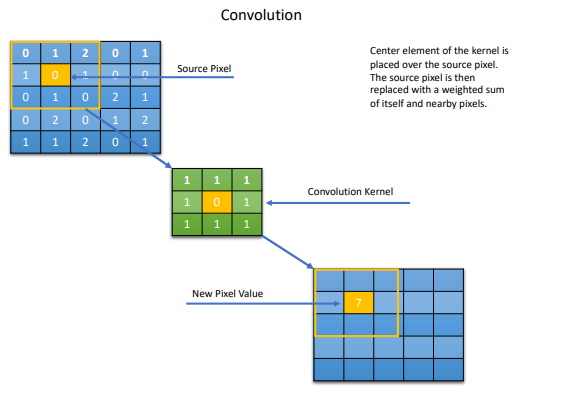


Figure 3.5‑2 Convolutional layer is broken down into its elementary components.

The CNN architecture is trained using a large dataset of labeled images. During training, the weights of the filters in the convolutional layers are updated to minimize the error between the predicted class and the true class label. This is typically done using backpropagation, where the error is propagated back through the network and used to update the weights of the filters.

Once the CNN is trained, it can be used to classify new images. The image is passed through the network, and the output from the fully connected layer is used to predict the class of the input image.In summary, CNNs work by using multiple layers to extract features from images and then using those features to classify the image. The convolutional layers identify important features in the image, the pooling layers down sample the feature maps, and the fully connected layers use the extracted features to predict the class of the image.

Study by Alzubaidi (2021) provides a comprehensive review of CNN architectures, challenges, applications, and future directions. It begins by introducing the basics of deep learning, followed by a detailed discussion of CNN architectures. The challenges of deep learning are also discussed, including the problem of overfitting and the need for large amounts of training data. The paper then provides an overview of the various applications of deep learning in different fields, such as image and speech recognition, natural language processing, and robotics.

### LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is designed to handle the vanishing gradient problem in traditional RNNs, which occurs when backpropagation attempts to update weights over long sequences, causing the gradient to become too small to make meaningful updates. Hochreiter and Schmidhuber came up with LSTM in 1997 as a solution to the RNN gradient vanishing issue. Gers later extended LSTM in 2001. LSTMs are particularly well-suited for sequential data where context over time is important, such as language and speech recognition, handwriting recognition, and video analysis. The key difference between LSTM and traditional RNNs is the inclusion of a memory cell and three gates: the input gate, output gate, and forget gate. The memory cell acts as a storage unit that can hold information over a long period of time, while the gates allow the network to control the flow of information into and out of the cell.

The forget gate determines which information to discard from the memory cell, the input gate decides which new information to add to the memory cell, and the output gate decides what information to output from the memory cell. Each gate is composed of a sigmoid function that outputs values between 0 and 1, representing the strength of the gate. The values of the gates are determined by a combination of the current input and the previous output of the network. LSTMs have been successfully applied to a wide range of tasks, including speech recognition, machine translation, text generation, and sentiment analysis. They have also been combined with other neural network architectures, such as convolutional neural networks (CNNs), to create hybrid models for tasks such as image captioning and video analysis.

There are several advantages of using Long Short-Term Memory (LSTM) with Convolutional Neural Networks (CNN) in image classification tasks:

1. Memory: LSTMs are designed to capture long-term dependencies in sequential data, which can be useful in recognizing patterns in images that span across multiple regions of the image. This memory property can help improve the accuracy of image classification.
2. Sequential data processing: Images can be thought of as sequential data, with pixels arranged in a 2D grid. LSTMs can process sequential data and extract relevant features, which can be combined with the features extracted by the CNN to improve classification performance.
3. Reduced overfitting: LSTMs can be used to regularize CNN models and prevent overfitting, as the LSTM layers can help smooth out noisy or inconsistent features extracted by the CNN.
4. Variable input size: LSTMs can handle variable-length input sequences, which can be useful in image classification tasks where the input images may have different sizes or aspect ratios.

Overall, the combination of CNNs and LSTMs can provide a powerful framework for image classification, especially for tasks where temporal information or long-term dependencies play a crucial role.

Patel & Dhruv (2020) reviewed the state-of-the-art methods for image classification using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). The authors discussed the basic structure of CNN and RNN, their advantages and limitations, and how to use them together for image classification. They also reviewed the current research trends and applications of CNN-RNN models, including object detection, facial recognition, and medical image analysis.

## Hybrid Neural Network (Proposed Model)

A hybrid neural network is a type of neural network that combines two or more different neural network architectures into a single, unified model. The goal is to leverage the strengths of each individual architecture and create a more powerful model that can better solve complex problems. There are many ways to create a hybrid neural network, but one common approach is to combine a feedforward neural network with a recurrent neural network. The feedforward network is used to extract features from the input data, while the recurrent network is used to capture the temporal dependencies in the data. The output of the feedforward network is fed into the recurrent network, which then produces the final output.

Model building, training and evaluation is performed for the following networks:

* Few different CNN networks (least layers to deep layers) individually.
* Ensemble of above CNNs of varying resolutions followed by classification using LSTM

### Introduction

The proposed model is composed of three different CNN models, which are trained individually, later ensembled and then fed into an LSTM layer for the purpose of image classification. The goal is to leverage the strengths of multiple CNN models and LSTM to improve the accuracy of image classification.

### Design

The model consists of three different CNN models, each of which is designed to extract different features from the input images. These three models are then combined by concatenating their outputs and fed into an LSTM layer, which captures the temporal dependencies in the feature maps. The output of the LSTM layer is passed through fully connected layers to produce the final classification.

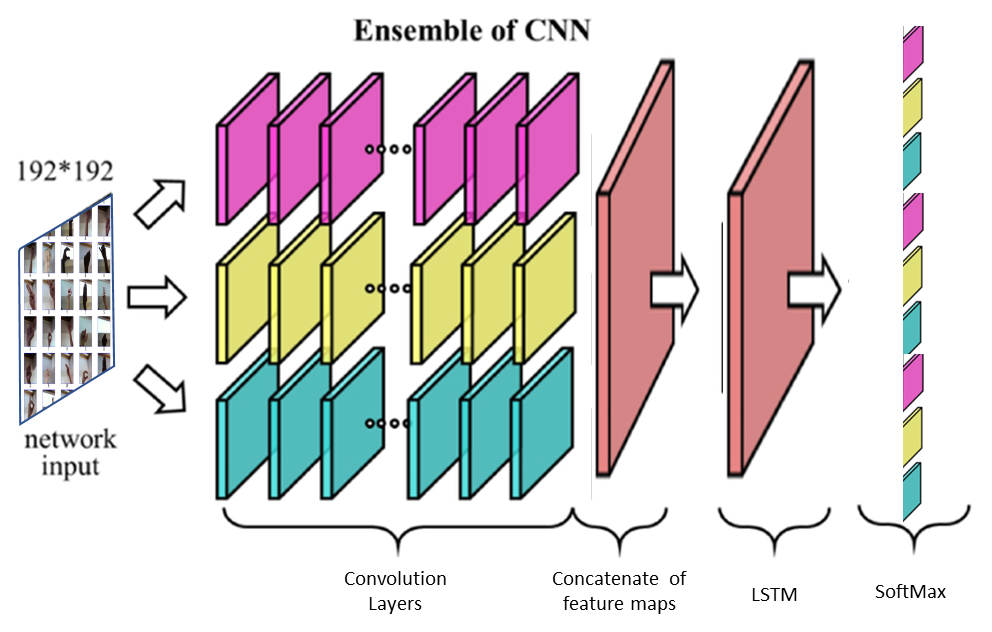


Figure 3.6‑1 Graphical description of proposed Hybrid Model

### Assumptions

The proposed model assumes that the ensemble of multiple CNN models will lead to improved accuracy in image classification. It also assumes that the use of LSTM will capture the temporal dependencies in the feature maps, leading to further improvements in accuracy.

### Results

The expected results of the proposed model are improved accuracy in image classification compared to using a single CNN model. The accuracy of the model will be measured using standard metrics such as accuracy, precision, and recall.

### Limitations:

One potential limitation of the proposed model is the potential for overfitting due to the large number of parameters involved. To mitigate this, regularization techniques such as dropout and weight decay will be used. Another potential limitation is the computational complexity of the model, which may require significant computational resources.

### Advantage

Hybrid neural networks are typically more robust to noise and variations in the data than CNN-based models. This is because LSTM is able to smooth out noise and capture the underlying patterns in the data, even when there is significant variation or distortion in the input images. Hybrid neural networks are often better at generalizing to new or unseen data than CNN-based models. This is because the LSTM component can learn to recognize underlying patterns in the data that are common across different examples, allowing the model to more accurately classify new data even if it is different from the training examples.

## Model Evaluation

The effectiveness of the network obtained in this study for American Sign Language recognition will be evaluated in relation to the model training parameters, which include the number of layers, filters, and optimizers. The study evaluated the performance of the approach on the test dataset, using four distinct measures to analyze its effectiveness. This process typically involves calculating various metrics, such as accuracy, precision, recall, and F1 score, as well as visualizing the results using confusion matrices and training and loss graphs.

In this study models are compared with following evaluations

**Training loss** refers to the error that the model makes on the training set during training. The goal of training a model is to minimize the training loss, which means that the model is learning to make better predictions on the training data. During the training process, the model is presented with a batch of training data, makes predictions, and then adjusts its weights and biases to reduce the difference between the predicted output and the true output. This process is repeated for multiple epochs until the training loss reaches a minimum or a plateau.

**Validation loss,** on the other hand, refers to the error that the model makes on a separate validation set, which is a subset of the data that is not used for training. The purpose of the validation set is to evaluate the performance of the model on new, unseen data, and to prevent overfitting. Overfitting occurs when the model becomes too complex and starts to memorize the training data rather than learning the underlying patterns in the data. Validation loss is used to monitor whether the model is overfitting or not. If the validation loss starts to increase while the training loss is decreasing, this indicates that the model is overfitting and it may be necessary to adjust the model architecture, regularization, or other hyperparameters.

The training and validation loss can be visualized as a function of the number of epochs during training. Typically, the training loss decreases over time as the model learns, while the validation loss may initially decrease but eventually starts to level off or increase if the model is overfitting. The goal is to find the point where the validation loss is minimized, which usually corresponds to a good trade-off between model complexity and generalization performance.

A **confusion matrix** is a table that summarizes the performance of a classification model on a set of test data for which the true values are known. It is a table with four different combinations of actual and predicted values for a binary classification problem. The four combinations are true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Here's an example confusion matrix for a binary classification problem:

In the matrix, the rows represent the actual class labels of the data, while the columns represent the predicted class labels. Fig () describes the confusion matrix

The four possible outcomes are:

True positives (TP): the number of correct positive predictions.

False positives (FP): the number of incorrect positive predictions.

True negatives (TN): the number of correct negative predictions.

False negatives (FN): the number of incorrect negative predictions.

Once the predicted labels have been obtained, various metrics can be calculated to evaluate the performance of the model. Some common metrics for image classification include accuracy, precision, recall, and F1 score.

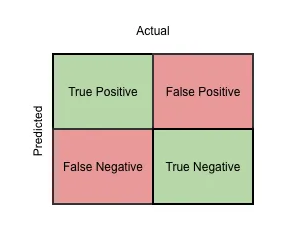


Figure 3.7‑1 Confusion matrix

**Accuracy:** This measures the overall percentage of images that were correctly classified by the model.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

**Precision:** This measures the proportion of true positive predictions among all positive predictions. It indicates how often the model correctly predicted a specific class.

Precision = TP / (TP + FP)

where TP is the number of true positives, and FP is the number of false positives.

**Recall (Sensitivity):** This measures the proportion of true positive predictions among all actual positive instances. It indicates how well the model was able to detect a specific class.

Recall = TP / (TP + FN)

where TP is the number of true positives, and FN is the number of false negatives.

**F1 score:** This is a weighted average of precision and recall, and provides a single value that summarizes the model's performance.

F1 score = 2 \* (Precision \* Recall) / (Precision + Recall)

where Precision and Recall are the precision and recall of the model, respectively.

# CHAPTER 4 ANALYSIS AND IMPLEMENTATION



## Introduction

Previous chapter of methodology deals with theoretical aspects of image classification and proposed model in general. This chapter details the practical aspect of data analysis and model building, training and experiments. This includes the selection and preparation of the dataset, the choice of model architecture, and the implementation of the model.

## Tools

To perform image classification, there are various tools and technologies available that can be used. Here are some of the essential tools used for this study:

**Kaggle** Notebook is a cloud-based data science environment provided by Kaggle, a popular platform for data science and machine learning competitions. It is an online tool that allows users to create, edit, and run Python code and machine learning models directly from their web browser. Kaggle Notebook provides a complete data science environment with all the necessary tools and libraries pre-installed, making it easy for users to get started with their data science projects quickly. Kaggle provides a powerful cloud-based computing environment that includes CPU, GPU, TPU, RAM, and storage resources, allowing users to access powerful hardware resources . This enables users to perform data analysis, machine learning, and deep learning tasks without the need for expensive local hardware.

**Python:** Python is a widely used programming language for image classification tasks. It has numerous libraries and frameworks such as TensorFlow, PyTorch, and Keras that help in building and training neural networks for image classification.

**Jupyter Notebook**: Jupyter Notebook is a web-based tool that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It is a great tool for data exploration, analysis, and visualization.

TensorFlow: TensorFlow is an open-source machine learning library developed by Google. It is widely used for image classification tasks and provides various pre-trained models and APIs for image classification.

**Keras:** Keras is a high-level neural networks API that runs on top of TensorFlow, developed to simplify the process of building and training deep learning models. It provides a user-friendly interface for creating and training neural networks for image classification tasks.

**Scikit-learn:** Scikit-learn is a popular Python library for machine learning. It provides various algorithms for image classification tasks such as decision trees, random forests, and support vector machines (SVMs).

**OpenCV:** OpenCV is an open-source computer vision library that provides various tools for image processing and analysis. It is widely used for image classification tasks and provides various algorithms for image feature extraction and object detection.

**Matplotlib** is a Python data visualization library used for creating static, interactive, and animated visualizations in Python. It is a widely used library in the data science community and is used for creating various types of plots, including line plots, scatter plots, bar charts, histograms, and more.

## Dataset Preparation

As described in chapter 3 this study used following datasets:

* **ASL Alphabet**
* **Sign Language digit dataset**

Both the datasets were merged together and used, table no () provides the details of the input datasets, link of the datasets is given in Appendix C.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Classes** | | **Total images** | **Size** |
| ASL Alphabets | 29 | A-Z, Space, Nothing, Del | 87000 | 1.1 Gb |
| Sign Language Digit Dataset | 10 | 0-9 | 2062 | 40 Mb |
|  |  |  |  |  |

The steps for data preparation performed are described below:

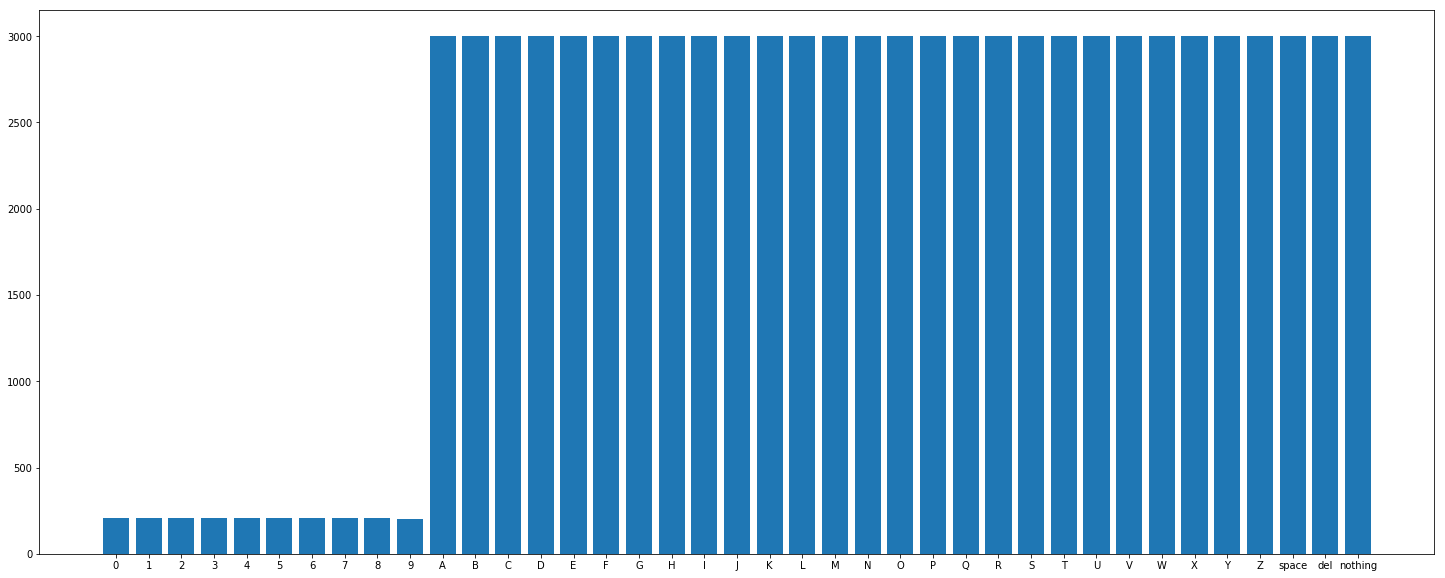
**Download** files from Kaggle website : The image files for the datasets can be downloaded from the Kaggle website (cs.rochester.edu, 2022). The files available are as follows:

ASL Alphabet – validation& test set

Sign Language digit dataset – validation & test set

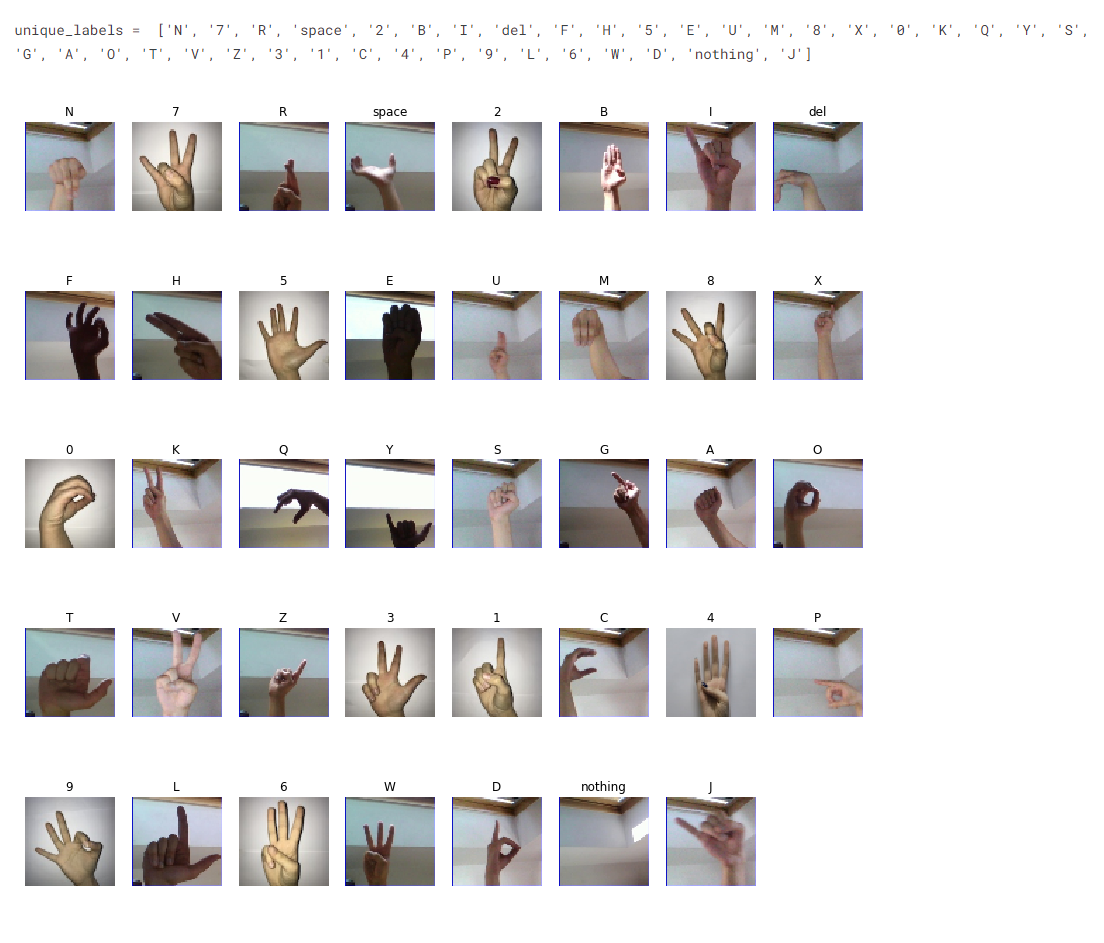
**Merged** datasets: Alphabet dataset has 87000 images and Digit dataset has 2082 images, both training and test datasets were merged together respectively to train model together with alphabets and digits sign language .

**Class Balance**: There are 39 unique class are there in input dataset, Class balance were checked across the dataset and it was found that alphabet classes have 3000 each and digit classes have 200 plus each, Bar chart for different classes is shown in fig



It was found that most of the classes are sufficiently balanced, though digits classes have less images as compare to alphabets.

**Plotting** one image for each class : each class were loaded and qced to analyses it’s visible characteristics and class assignment. Figure () shows the one image from the each 39 classes



**Resizing:** Alphabet dataset and Digit dataset have different size of pixels 200x200 and 100x100 respectively. Input images were resized into 64x64 pixels to create uniform dataset. The size of image can the size of the image can affect the accuracy of the classification model. Larger images can contain more information, but they can also be slower to process. Smaller images can be faster to process, but they may not contain enough information for the model to make accurate predictions. This was done primarily to limit the cost for GPU utilization and OpenAI API usage.

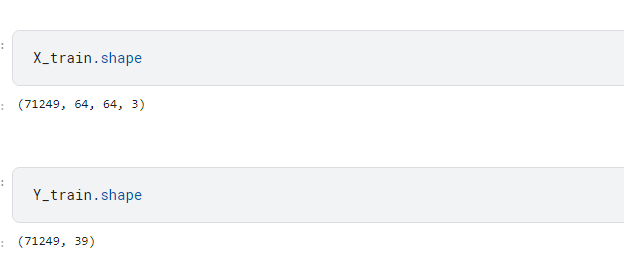
**Normalization:** Normalization converts the pixel values of the images from integers in the range of 0-255 to floating-point values in the range of 0-1. This is done by first casting the array to the *float32* data type using the *astype()* method and then dividing each element of the array by 255.0. The reason for normalizing the pixel values to the range of 0-1 is to ensure that the image data has a consistent scale across all images in the dataset. This makes it easier for machine learning models to learn patterns and relationships in the data.



**One hot coding:** The class labels in the *labels* list are converted to integers using the *labels\_dict* dictionary, and then converted to one-hot encoding format using the *to\_categorical()* function from the *keras.utils* module. One-hot encoding is a process of converting categorical data into a binary vector representation. For example, the class label 'A' is converted to a binary vector of length 39, where the 10th element is 1 and all other elements are 0. This is useful for training deep neural networks because it allows the network to output a probability distribution over all possible classes.

**Augmentation**: Data augmentation was not applied here. Data augmentation is a common technique used in computer vision to increase the size of a dataset by creating modified versions of the original images. By applying transformations such as rotations, flips, and zooms to the images, data augmentation can help to improve the performance of a machine learning model by reducing overfitting and increasing its ability to generalize to new, unseen data. However due to high size of input dataset with quite balance classes across it was decided to no to augment dataset

**Training and validation data split** : The *train\_test\_split* function from the *sklearn.model\_selection* module was used to split the dataset into a training set and a validation set. The function randomly shuffles the data and divides it into two parts based on the *test\_size* argument, which specifies the percentage of the dataset to be used for validation. In this case, the validation set size is 20% of the original dataset, which means that 80% of the dataset is used for training the model. The function returns four arrays: *X\_train, X\_test, Y\_train,* and *Y\_test.* The *X\_train* and *Y\_train* arrays contain the training data and labels, while the *X\_test* and *Y\_test* arrays contain the validation data and labels. By using *train\_test\_split* function, we can evaluate the performance of the model on data that it has not seen during training, which helps to avoid overfitting and ensures that the model generalizes well to new data.



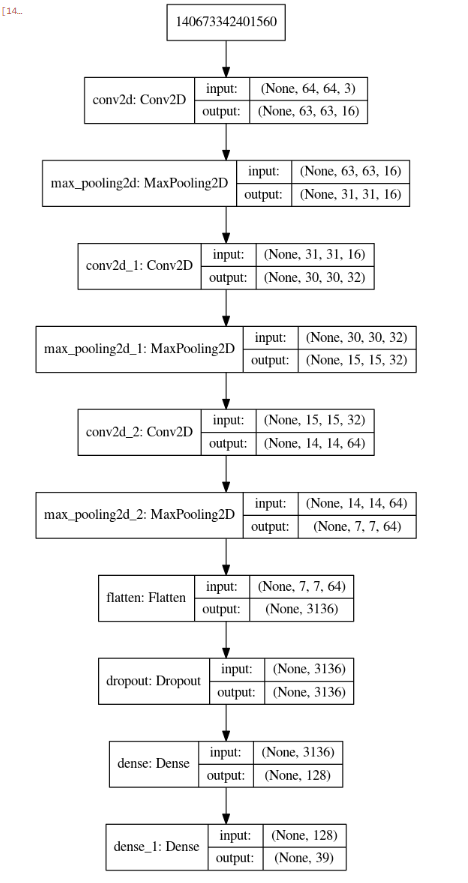
## CNN Model – 1

First model was created using A CNN (Convolutional Neural Network) type of neural network that is often used for image classification tasks. It consists of one or more convolutional layers, followed by pooling layers, and then fully connected layers. Convolutional layers apply filters to the input image to detect specific features, such as edges or corners. The output of the convolutional layer is a feature map that highlights the presence of those features in the input image. Pooling layers down sample the output of the convolutional layers by taking the maximum or average of small regions of the feature map. Fully connected layers are used at the end of the CNN to classify the input image. They take the flattened output of the previous layers and apply weights to each neuron to determine the probability of each class. CNN models are trained using backpropagation to adjust the weights of each layer in order to minimize the loss between the predicted output and the actual output.

The model architecture consists of three 2D convolutional layers (**Conv2D()**) with 16, 32, and 64 filters respectively, each followed by a max-pooling layer (**MaxPooling2D()**) with a 2x2 pool size. The **input\_shape** of the first convolutional layer is set to **(64, 64, 3)**, indicating that the input images are 64x64 pixels with 3 color channels (RGB).

After the final pooling layer, the output is flattened into a 1D array using **Flatten()**. A dropout layer (**Dropout()**) is then applied with a rate of 0.5 to prevent overfitting.

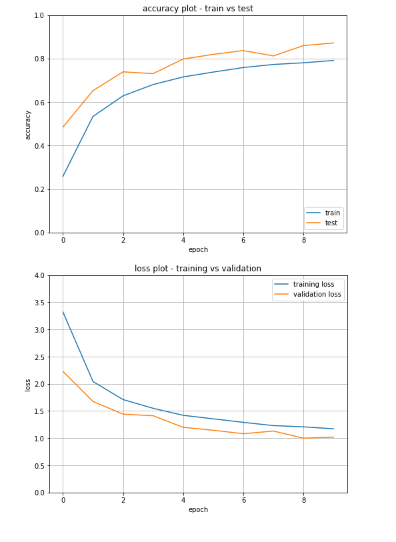
Two dense layers (**Dense()**) are added after the dropout layer. The first dense layer has 128 neurons and uses the ReLU activation function. A regularization term with L2 weight decay of 0.05 is applied to the layer using the **kernel\_regularizer** argument to help prevent overfitting. The second dense layer has 39 neurons (the number of classes in the dataset) and uses the softmax activation function to output class probabilities.



The model was compiled with the **adam** optimizer, **categorical\_crossentropy** loss function, and accuracy metric.

The model was trained using the **fit()** method. This function takes in the training data **X\_train** and **Y\_train**, sets the batch size to 128, and trains the model for 10 epochs. A validation split of 0.2 is used, which means that 20% of the training data is used for validation and the remaining 80% is used for training.

The function returns the training history, which includes the loss and accuracy on both the training and validation sets for each epoch. Training and loss curves are visual representations of the model's performance during the training process. The training curve represents the change in the model's performance (typically measured by accuracy) on the training data as the number of epochs increases. On the other hand, the loss curve represents the change in the model's loss (e.g., cross-entropy loss) on the training data as the number of epochs increases.



Fugure () show the Training and loss curve of model -1 , model-1 achieved 79% training and 87% validation accuracy

## CNN Model – 2

The second model is also a Convolutional Neural Network (CNN) model that uses the Keras Sequential API to create the model. The architecture of this model is similar to the first model, with some differences in the number of layers and their configurations.

The architecture of Model 2 consists of 3 sets of convolutional layers with increasing number of filters, followed by max pooling layers to down sample the feature maps. Each set consists of a Conv2D layer with 3x3 kernel size and 'relu' activation function, and a MaxPooling2D layer with 3x3 pool size. The use of larger kernel sizes results in larger receptive fields which can capture more contextual information from the input image.

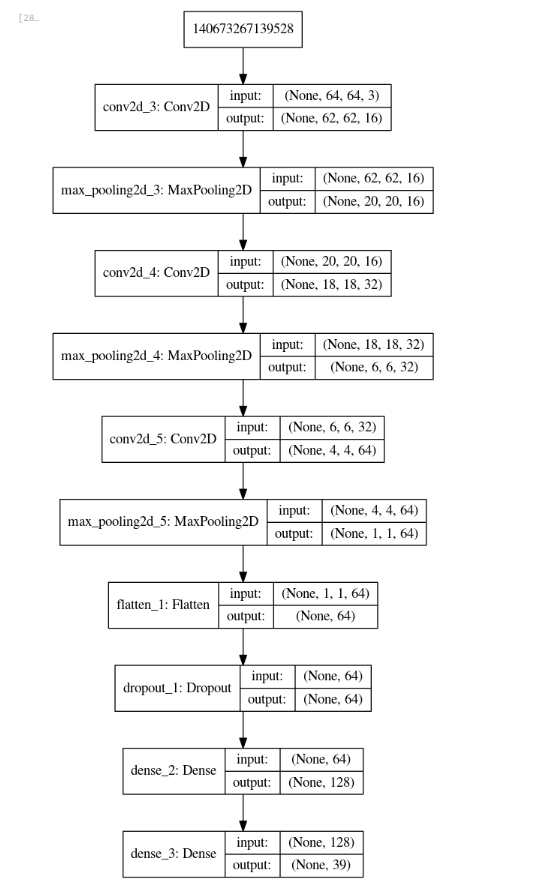
After the convolutional layers, there is a Flatten layer to convert the 3D feature maps into a 1D vector, followed by a Dropout layer with a dropout rate of 0.5 to reduce overfitting. Then, there is a Dense layer with 128 neurons and 'relu' activation function, and a Dense output layer with 39 neurons (equal to the number of classes) and 'softmax' activation function.

The model is compiled using the 'adam' optimizer, 'categorical\_crossentropy' loss function, and 'accuracy' metric.

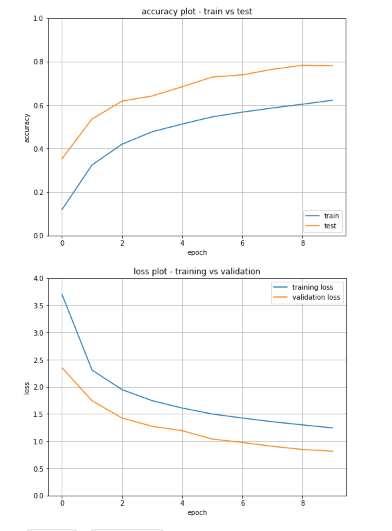
Model 2 has some differences compared to Model 1:

* It uses a 3x3 kernel size instead of 2x2 in the convolutional layers.
* It uses a 3x3 pool size instead of 2x2 in the max pooling layers.
* It does not include BatchNormalization layers.
* It uses a higher dropout rate of 0.5 instead of 0.25.
* It uses a lower L2 regularization rate of 0.05 instead of 0.01.

The main difference between the two models is the kernel and pool size used in the Conv2D and MaxPooling2D layers. The larger kernel and pool size in Model 2 may result in a more coarse-grained feature extraction, which can be beneficial in certain cases where the features to be detected are larger and more complex. However, this may also lead to a loss of finer details in the image, and may make the model less effective at detecting smaller features.

. 

Similar to the first model, the model is compiled with the Adam optimizer and categorical cross-entropy loss function. The model is then trained on the training data using the fit method with a batch size of 128, 10 epochs, and a validation split of 0.2.



## CNN Model – 3

The third model is also a convolutional neural network (CNN) designed for image classification. It has a similar structure to the first two models, but with some additional layers.

First, the model adds a Conv2D layer with 128 filters, each with a kernel size of 2x2 and an activation function of ReLU. This is followed by a MaxPooling2D layer with a pool size of 2x2, which reduces the dimensions of the output of the convolutional layer.

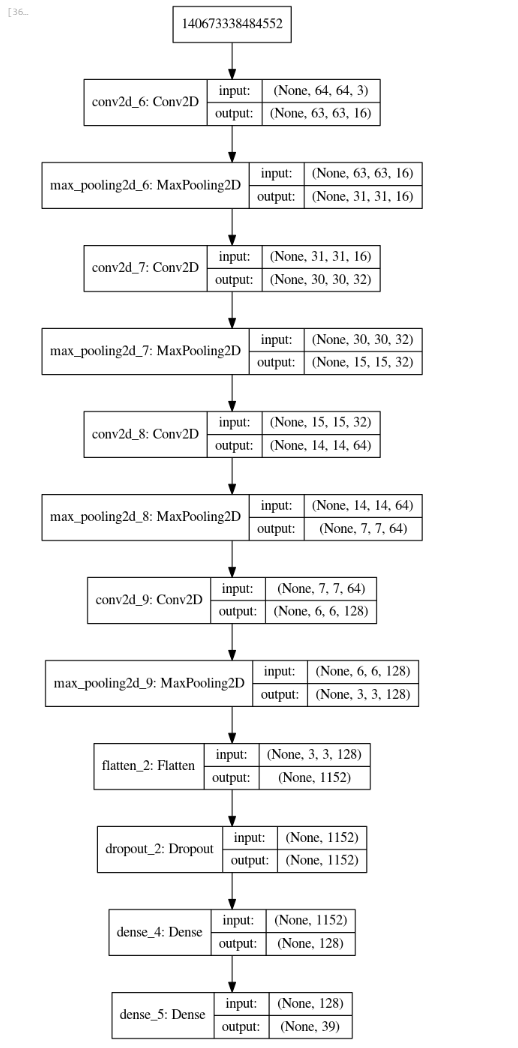
The same pattern of adding a convolutional layer followed by a pooling layer is repeated three times, with 32, 64, and 128 filters in each layer respectively, and with kernel sizes of 2x2 in the first two layers and 3x3 in the last layer.

After the convolutional and pooling layers, the model flattens the output using the Flatten() layer, which converts the multidimensional output into a one-dimensional array.

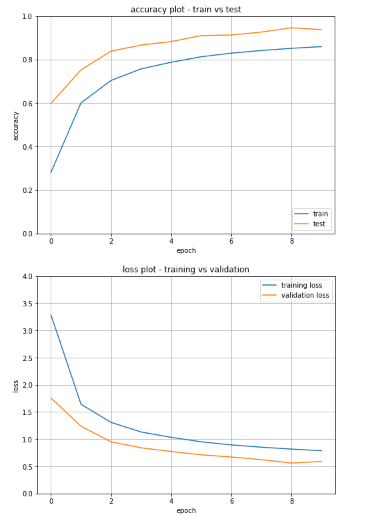
Next, the model adds a Dropout layer with a dropout rate of 0.5 to prevent overfitting. This is followed by a fully connected layer with 128 neurons and a ReLU activation function, which applies a linear transformation to the input data and then applies the ReLU function element-wise to the output.

Finally, the model adds another fully connected layer with 39 neurons and a softmax activation function, which outputs a probability distribution over the 39 classes. This allows the model to make predictions by choosing the class with the highest probability.

Compared to the first two models, this model has more convolutional layers, which allows it to learn more complex features from the input images. It also has more filters in each layer, which increases the number of parameters and allows the model to capture more fine-grained details. The additional pooling layer with a larger kernel size also helps to reduce the dimensions of the output more quickly, which can lead to faster training times.



Like the previous models, this model is compiled with the Adam optimizer and categorical cross-entropy loss function, and it is trained using the fit() method with a batch size of 128 and 10 epochs.



## Hybrid Neural Network (Proposed model)

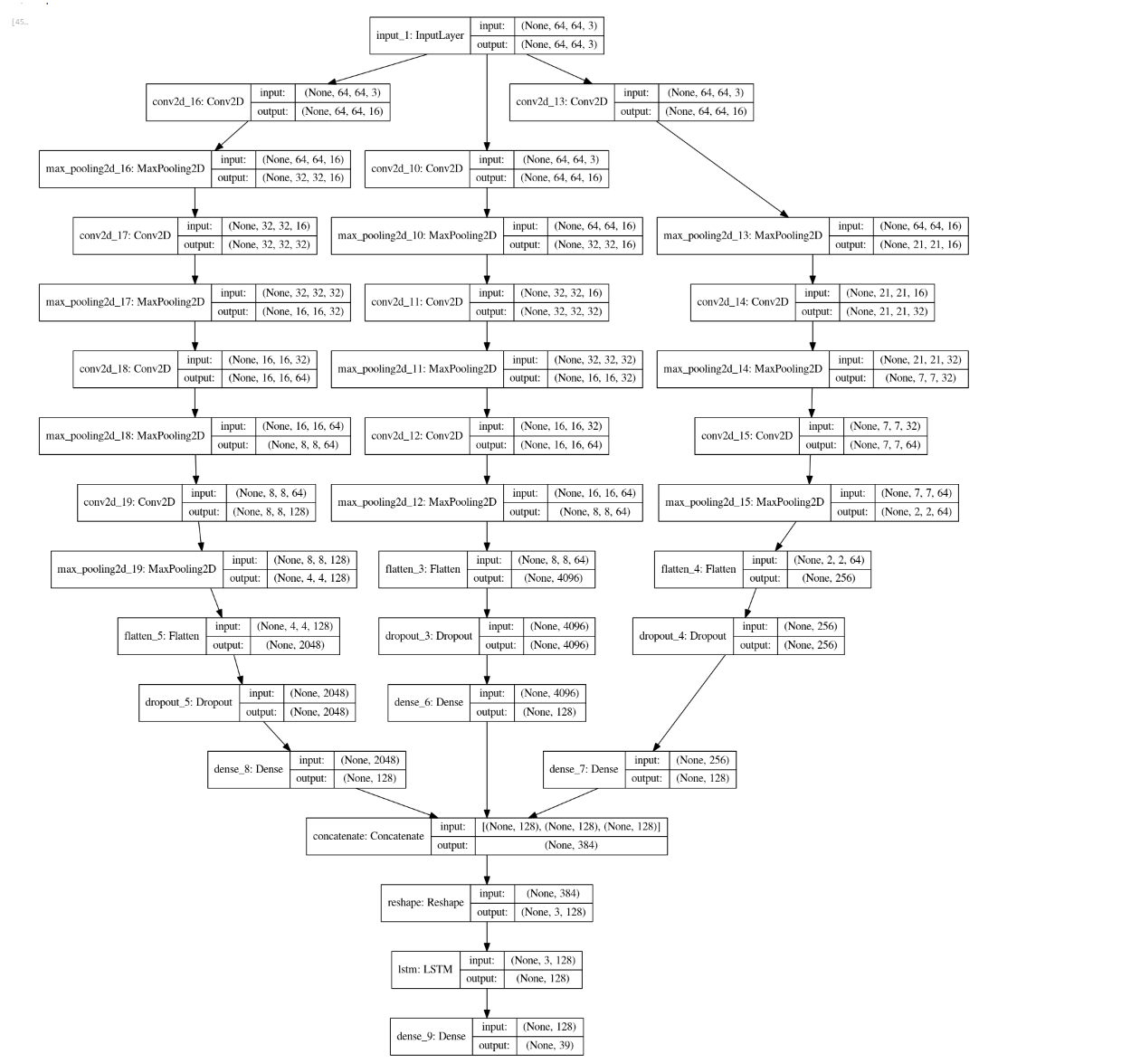
Hybrid neural networks are a combination of two or more types of neural networks that work together to solve a particular problem. These networks are designed to incorporate the best features of each network to improve performance and achieve better accuracy. Hybrid neural networks allow developers to create models that leverage the strengths of each network while minimizing their weaknesses.

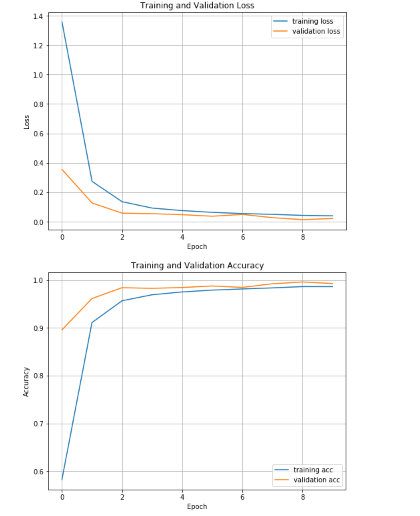
The proposed model is an ensemble of three CNNs (Model 1,2,3) and an LSTM, which work together to process the input images and make predictions. By creating an ensemble of the outputs of three different CNNs and feeding them into an LSTM layer, the proposed model can capture different types of information from the input images and learn to make predictions based on both spatial and temporal features. This can potentially lead to better performance compared to using a single CNN or RNN model alone.

The proposed model is a combination of three Convolutional Neural Networks (CNNs) used above (Model1,2,3) and a Long Short-Term Memory (LSTM) layer for classification. The model takes input images of size 64x64x3. Here is a detailed explanation of the model architecture:

1. The model has three branches, each consisting of a CNN followed by a fully connected (Dense) layer.
2. The first branch of the model consists of a 3-layer CNN followed by a flatten layer and a dropout layer. The first CNN layer has 16 filters of size 2x2, followed by a max-pooling layer of size 2x2. The second CNN layer has 32 filters of size 2x2, followed by another max-pooling layer of size 2x2. The third CNN layer has 64 filters of size 2x2.
3. The second branch of the model consists of another 3-layer CNN followed by a flatten layer and a dropout layer. The first CNN layer has 16 filters of size 3x3, followed by a max-pooling layer of size 3x3. The second CNN layer has 32 filters of size 3x3, followed by another max-pooling layer of size 3x3. The third CNN layer has 64 filters of size 3x3.
4. The third branch of the model consists of a 4-layer CNN followed by a flatten layer and a dropout layer. The first CNN layer has 16 filters of size 2x2, followed by a max-pooling layer of size 2x2. The second CNN layer has 32 filters of size 2x2, followed by another max-pooling layer of size 2x2. The third CNN layer has 64 filters of size 2x2. The fourth CNN layer has 128 filters of size 2x2.
5. The outputs of the three branches are concatenated and fed to a reshape layer to add a timestep dimension of size 3.
6. The output of the reshape layer is fed to an LSTM layer of size 128.
7. The output of the LSTM layer is fed to a fully connected layer with 39 neurons and a softmax activation function for classification.
8. The model is trained using the Adam optimizer, categorical cross-entropy loss function, and accuracy as the evaluation metric.

Overall, the proposed model leverages the power of CNNs for feature extraction and an LSTM layer for sequence modeling to achieve high accuracy in image classification tasks.





# CHAPTER 5

# RESULTS AND EVALUATION



## Introduction

This chapter details the outcome of the model building experiments, it includes the results of each model in previous chapter is trained. The chapter provides an evaluation of the trained model, including metrics such as accuracy, precision, and recall. It compares the models tested with proposed hybrid neural network model.

## Results

Training and loss curves have been shared in chapter 4 for each model . Table () compares the training and loss for all models.



Model -1 has given 79% accuracy with training dataset and 87% accuracy with test dataset , accuracy decreased with Model-2 due increase of filter size and got 62% with training and 78% with test dataset. With one conv layer increased and smaller filter size Model-3 achieved 86% training and 93% test accuracy but overall proposed Hybrid model which consist all of other model as ensemble and followed by LSTM classifier has given an accuracy of 98% with training dataset and 99% with test dataset.

## Evaluation matrix

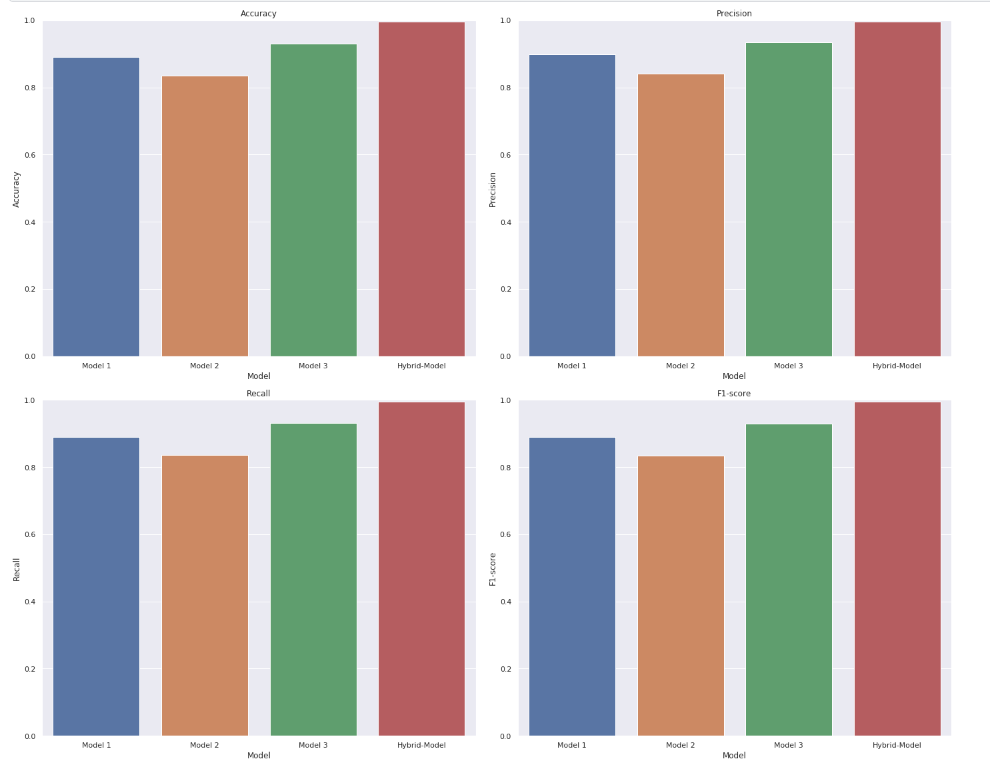
Evaluation metrics are used to measure the performance of a machine learning model. These metrics provide information about how well the model is performing on the given task. Commonly used evaluation metrics in machine learning include accuracy, precision, recall, and F1-score. These metrics are used to evaluate the performance of classification models, where the goal is to predict the class label of an input instance.

Accuracy is the ratio of correctly predicted instances to the total number of instances. Precision is the ratio of correctly predicted positive instances to the total number of predicted positive instances. Recall is the ratio of correctly predicted positive instances to the total number of actual positive instances. F1-score is the harmonic mean of precision and recall.

By using these evaluation metrics, we can get a clear idea of how well a model is performing and identify areas for improvement. It can help in selecting the best model among different models and tuning the model parameters to achieve better performance. Table () shows the evaluation matrix for each model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Model 1 | 0.871947 | 0.882743 | 0.871947 | 0.871564 |
| Model 2 | 0.779375 | 0.800516 | 0.779375 | 0.780278 |
| Model 3 | 0.938865 | 0.942784 | 0.938865 | 0.939006 |
| Hybrid-Model | 0.993432 | 0.993538 | 0.993432 | 0.993396 |

1. Model 1: This model achieved an accuracy of 0.8719, which means that it correctly classified 87.19% of the test data. The precision and recall scores are also high, with values of 0.8827 and 0.8719 respectively. The F1-score, which is a harmonic mean of precision and recall, is 0.8716. Overall, this is a decent model with fairly balanced precision and recall.
2. Model 2: This model achieved an accuracy of 0.7794, which is lower than the other models. The precision score is 0.8005, which means that when the model predicts a positive label, it is correct 80.05% of the time. The recall score is 0.7794, which means that the model correctly identifies 77.94% of all positive instances. The F1-score is 0.7803, which is the harmonic mean of precision and recall. This model has relatively low accuracy, but the precision score is higher than the recall score, which suggests that it tends to make more false negative errors.
3. Model 3: This model achieved an accuracy of 0.9389, which is quite high. The precision and recall scores are both above 0.93, with values of 0.9428 and 0.9389 respectively. The F1-score is 0.9390. This model has the highest precision and recall scores among the four models, indicating that it is better at both minimizing false positives and false negatives.
4. Hybrid-Model: This model achieved the highest accuracy of 0.9934, which is significantly higher than the other models. The precision and recall scores are also very high, with values of 0.9935 and 0.9934 respectively. The F1-score is 0.9934. This model has very high precision and recall, which means it performs very well in minimizing both false positives and false negatives.

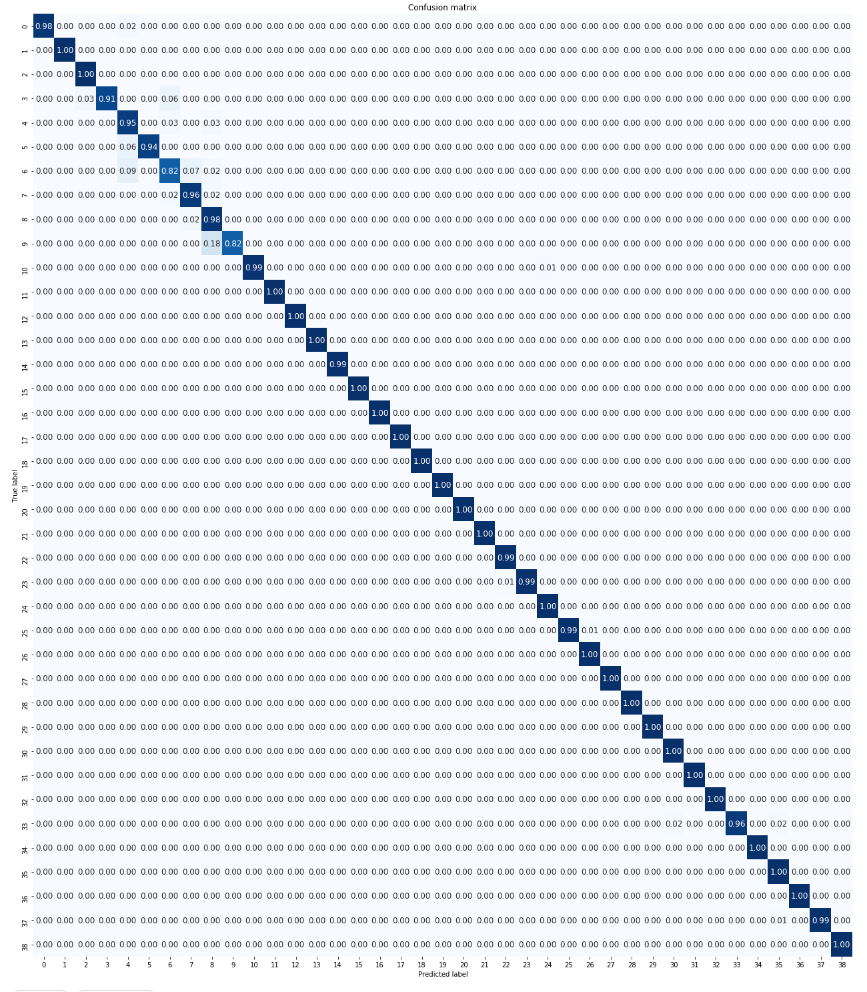


## Prediction with Hybrid Model

After training with train and validation dataset, proposed Hybrid model was used to predict unseen test dataset and evaluated with confusion matrix and plotting test images with predict and input labels.

### Confusion matrix

Confusion matrix is a table that is used to evaluate the performance of a classification model. It displays the number of true positives, true negatives, false positives, and false negatives for each class in the classification problem. The rows represent the actual class and the columns represent the predicted class. The diagonal elements represent the number of correct predictions, while the off-diagonal elements represent incorrect predictions.



### Prediction

Unseen test data was predicted using proposed Hybrid Model and Fig () shows images with actual labels against predicted labels. Model has predicted all labels correctly .



In summary, the hybrid model has the highest accuracy and precision/recall scores among the four models, while Model 2 has the lowest accuracy and a higher precision score compared to the recall score. Model 1 and Model 3 have relatively balanced precision and recall scores.

# CHAPTER 6

# Conclusion and Future Scope



## Conclusions and Discussion

This study explored the Artificial Intelligent based sign language recognition using CNN neural networks and further using Hybrid neural network. This work presented a comprehensive overview of recent research done in sign language recognition field and discussed various aspects around it.

The aim of this study was to train a Hybrid neural network to recognize American Sign language and digits with higher accuracy. Experiments performed in this study proved that proposed Hybrid Model outperformed the other CNN models across the evaluation matrix. The results suggest that the proposed models can achieve high performance in ASL recognition, which can have important applications in improving communication for the hearing-impaired.

This study proved that ensemble of different models can improve the sign language recognition accuracy over the individual models. There are many different types of neural networks, each with its strengths and weaknesses. Hybrid neural networks allow developers to create models that leverage the strengths of each network while minimizing their weaknesses. Though Hybrid neural network have many advantages but it can be more complex to design and train compared to traditional neural networks, as they involve combining multiple models and architectures. Due to their increased complexity, hybrid neural networks may require longer training times compared to traditional neural networks. The increased complexity of hybrid neural networks can make it difficult to interpret the results and understand how the model arrived at its predictions.

## Contribution to Knowledge

This study showed that Hybrid neural networks can achieve higher accuracy for image classification than traditional neural networks because they can learn both linear and non-linear relationships between the input and output data. Hybrid neural networks can be customized to the specific needs of a particular problem by combining different types of neural networks. Hybrid neural networks can be designed to handle a wide range of data types, including image, text, and time series data.

## Future Scope

There are several areas that could be explored as future work in sign language recognition, including:

* Incorporating more advanced techniques for feature extraction and representation, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to capture more complex spatial and temporal relationships in the sign language gestures.
* Investigating the use of transfer learning and fine-tuning approaches to leverage pre-trained models for related tasks, such as image or video classification, and adapt them to sign language recognition.
* Developing more comprehensive and diverse sign language datasets, including different sign languages and dialects, to enable more accurate and robust recognition across various settings and populations.
* Exploring the use of multimodal data sources, such as depth sensors or haptic devices, to capture additional information about the sign language gestures and improve recognition performance.
* Investigating the use of real-time systems and wearable devices for sign language recognition, which could have practical applications in communication and accessibility for the deaf and hard-of-hearing communities.

# References

Akmeliawati, R., Ooi, M.P.-L. and Kuang, Y.C., (2007) Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network. *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*, pp.1–6.

Badhe, P.C. and Kulkarni, V., (2015) Indian sign language translator using gesture recognition algorithm. *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, pp.195–200.

Bheda, V. and Radpour, D., (2017) Using Deep Convolutional Networks for Gesture Recognition in American Sign Language. *CoRR*, [online] abs/1710.06836. Available at: <http://arxiv.org/abs/1710.06836>.

Egmont-Petersen, M., de Ridder, D. and Handels, H., (2002) Image processing with neural networks—a review. *Pattern Recognition*, [online] 3510, pp.2279–2301. Available at: <https://www.sciencedirect.com/science/article/pii/S0031320301001789>.

Gupta, N., (2022) Sign Language Recognition Using Diverse Deep Learning Models. In: C.M. and A.S. and P.C.M.A. and P.B.R. Sanyal Goutam and Travieso-González, ed., *International Conference on Artificial Intelligence and Sustainable Engineering*. Singapore: Springer Singapore, pp.463–475.

Kamal, S.M., Chen, Y., Li, S., Shi, X. and Zheng, J., (2019) Technical Approaches to Chinese Sign Language Processing: A Review. *IEEE Access*, 7, pp.96926–96935.

Kandel, I., Castelli, M. and Popovič, A., (2021) Comparing Stacking Ensemble Techniques to Improve Musculoskeletal Fracture Image Classification. *Journal of Imaging*, [online] 76. Available at: <https://www.mdpi.com/2313-433X/7/6/100>.

Katoch, S., Singh, V. and Tiwary, U.S., (2022) Indian Sign Language recognition system using SURF with SVM and CNN. *Array*, [online] 14, p.100141. Available at: <https://www.sciencedirect.com/science/article/pii/S2590005622000121>.

Khalid, S., Khalil, T. and Nasreen, S., (2014) A survey of feature selection and feature extraction techniques in machine learning. In: *2014 Science and Information Conference*. pp.372–378.

Koller, O., Zargaran, S., Ney, H. and Bowden, R., (2018) Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs. *International Journal of Computer Vision*, [online] 12612, pp.1311–1325. Available at: <https://doi.org/10.1007/s11263-018-1121-3>.

Mannan, A., Abbasi, A., Javed, A.R., Ahsan, A., Gadekallu, T.R. and Xin, Q., (2022) Hypertuned Deep Convolutional Neural Network for Sign Language Recognition. *Computational Intelligence and Neuroscience*, [online] 2022, p.1450822. Available at: <https://doi.org/10.1155/2022/1450822>.

Masood Sarfaraz and Thuwal, H.C. and S.A., (2018) American Sign Language Character Recognition Using Convolution Neural Network. *Smart Computing and Informatics*. Springer , pp.403–412.

Mavi, A., (2020) A New Dataset and Proposed Convolutional Neural Network Architecture for Classification of American Sign Language Digits. *CoRR*, [online] abs/2011.08927. Available at: <https://arxiv.org/abs/2011.08927>.

Patil, S.B. and Sinha, G.R., (2017) Distinctive Feature Extraction for Indian Sign Language (ISL) Gesture using Scale Invariant Feature Transform (SIFT). *Journal of The Institution of Engineers (India): Series B*, [online] 981, pp.19–26. Available at: <https://doi.org/10.1007/s40031-016-0250-8>.

Raghuveera, T., Deepthi, R., Mangalashri, R. and Akshaya, R., (2020) A depth-based Indian Sign Language recognition using Microsoft Kinect. *Sādhanā*, [online] 451, p.34. Available at: <https://doi.org/10.1007/s12046-019-1250-6>.

Rao, G.A., Syamala, K., Kishore, P.V. v and Sastry, A.S.C.S., (2018) Deep convolutional neural networks for sign language recognition. In: *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*. pp.194–197.

Rathi, D., (2018) Optimization of Transfer Learning for Sign Language Recognition Targeting Mobile Platform. *CoRR*, [online] abs/1805.06618. Available at: <http://arxiv.org/abs/1805.06618>.

Sharma, S. and Kumar, K., (2021) ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. *Multimedia Tools and Applications*, [online] 8017, pp.26319–26331. Available at: <https://doi.org/10.1007/s11042-021-10768-5>.

Simonyan, K. and Zisserman, A., (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. [online] ICLR. Available at: <http://arxiv.org/abs/1409.1556>.

Subburaj, S. and Murugavalli, S., (2022) Survey on sign language recognition in context of vision-based and deep learning. *Measurement: Sensors*, [online] 23, p.100385. Available at: <https://www.sciencedirect.com/science/article/pii/S2665917422000198>.

Suharjito, Anderson, R., Wiryana, F., Ariesta, M.C. and Kusuma, G.P., (2017) Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output. *Procedia Computer Science*, [online] 116, pp.441–448. Available at: <https://www.sciencedirect.com/science/article/pii/S1877050917320720>.

Wadhawan, A. and Kumar, P., (2020) Deep learning-based sign language recognition system for static signs. *Neural Computing and Applications*, [online] 3212, pp.7957–7968. Available at: <https://doi.org/10.1007/s00521-019-04691-y>.

Yirtici, T. and Yurtkan, K., (2022) Regional-CNN-based enhanced Turkish sign language recognition. *Signal, Image and Video Processing*, [online] 165, pp.1305–1311. Available at: <https://doi.org/10.1007/s11760-021-02082-2>.

D. Enikeev, S. Mustafina

Recognition of sign language using leap motion controller data

Proceedings of the 2nd international conference on control systems, mathematical modeling, automation and energy efficiency, SUMMA 2020 (2020), pp. 393-397, [10.1109/SUMMA50634.2020.9280795](https://doi.org/10.1109/SUMMA50634.2020.9280795)

C.M. Jin, Z. Omar, M.H. Jaward

A mobile application of American sign language translation via image processing algorithms

Proceedings of the 2016 IEEE region 10 symposium, TENSYMP 2016 (2016), pp. 104-109, [10.1109/TENCONSpring.2016.7519386](https://doi.org/10.1109/TENCONSpring.2016.7519386)

A.S. Al-Shamayleh, R. Ahmad, N. Jomhari, M.A.M. Abushariah

Automatic Arabic sign language recognition: A review, taxonomy, open challenges, research roadmap and future directions

Malaysian Journal of Computer Science, 33 (4) (2020), pp. 306-343, [10.22452/mjcs.vol33no4.5](https://doi.org/10.22452/mjcs.vol33no4.5)

S.S. Shivashankara, S. Srinath

American sign language recognition system: an optimal approach

Int. J. Image Graph. Signal Process. (2018)

I. Aliyu, K.J. Gana, A.A. Musa, M.A. Adegboye, C.G. Lim

Incorporating recognition in catfish counting algorithm using artificial neural network and geometry

KSII Transactions on Internet and Information Systems, 14 (12) (2020), pp. 4866-4888, [10.3837/tiis.2020.12.014](https://doi.org/10.3837/tiis.2020.12.014)

G. Kumar, P.K. Bhatia

A detailed review of feature extraction in image processing systems

Proceedings of the international conference on advanced computing and communication technologies, ACCT, February (2014), pp. 5-12, [10.1109/ACCT.2014.74](https://doi.org/10.1109/ACCT.2014.74)

M.J. Cheok, Z. Omar, M.H. Jaward

A review of hand gesture and sign language recognition techniques

International Journal of Machine Learning and Cybernetics, 10 (1) (2019), pp. 131-153, [10.1007/s13042-017-0705-5](https://doi.org/10.1007/s13042-017-0705-5)

G. Joshi, S. Singh, R. Vig

Taguchi-TOPSIS based HOG parameter selection for complex background sign language recognition

Journal of Visual Communication and Image Representation, 71 (2020), Article 102834, [10.1016/j.jvcir.2020.102834](https://doi.org/10.1016/j.jvcir.2020.102834)

A. Dudhal, H. Mathkar, A. Jain, O. Kadam, M. Shirole

Hybrid SIFT feature extraction approach for indian sign language recognition system based on CNN

Lecture notes in computational vision and biomechanics, 30 (2019), pp. 727-738, [10.1007/978-3-030-00665-5\_72](https://doi.org/10.1007/978-3-030-00665-5_72)

[View article](javascript:;)

[View in Scopus](https://www.scopus.com/inward/record.url?eid=2-s2.0-85060203809&partnerID=10&rel=R3.0.0)[Google Scholar](https://scholar.google.com/scholar_lookup?title=Hybrid%20SIFT%20feature%20extraction%20approach%20for%20indian%20sign%20language%20recognition%20system%20based%20on%20CNN&publication_year=2019&author=A.%20Dudhal&author=H.%20Mathkar&author=A.%20Jain&author=O.%20Kadam&author=M.%20Shirole)

A. Narayanan, T.S. Sharmila, P. Gajalakshmi

Word sign recognition of invariant images based on SURF with Laplacian eigenmaps

Proceedings of the 2nd international conference on computer, communication, and signal processing: Special focus on technology and innovation for smart environment, ICCCSP 2018 (2018), [10.1109/ICCCSP.2018.8452825](https://doi.org/10.1109/ICCCSP.2018.8452825)

S. Selim, E. Elhinamy, H. Othman, W. Abouelsaadat, M.A.M. Salem

A review of machine learning approaches for epileptic seizure prediction

In E.-K. M. W. B. E.-D. A. M. T. M. A. A. Salem A. Abbas H.M. (Ed.)

Proceedings of the ICCES, 14th international conference on computer engineering and systems, Institute of Electrical and Electronics Engineers Inc (2019), pp. 239-244, [10.1109/ICCES48960.2019.9068190](https://doi.org/10.1109/ICCES48960.2019.9068190)

T.N.T. Huong, T.V. Huu, T.Le Xuan, S.V Van

Static hand gesture recognition for Vietnamese sign language (VSL) using principle components analysis

Proceedings of the international conference on computing, management and telecommunications, ComManTel 2015 (2016), pp. 138-141, [10.1109/ComManTel.2015.7394275](https://doi.org/10.1109/ComManTel.2015.7394275)

A. Sharma, A. Mittal, S. Singh, V Awatramani

Hand gesture recognition using image processing and feature extraction techniques

Procedia Computer Science, 173 (2019) (2020), pp. 181-190, [10.1016/j.procs.2020.06.022](https://doi.org/10.1016/j.procs.2020.06.022)

R.D. Raj, A. Jasuja

British sign language recognition using HOG

Proceedings of the IEEE international students’ conference on electrical, electronics and computer science, SCEECS 2018 (2018), pp. 1-4, [10.1109/SCEECS.2018.8546967](https://doi.org/10.1109/SCEECS.2018.8546967)

S. Liwicki, M. Everingham

Automatic recognition of fingerspelled words in british sign language

Proceedings of the 2009 IEEE conference on computer vision and pattern recognition, CVPR 2009 (2009), pp. 50-57, [10.1109/CVPR.2009.5204291](https://doi.org/10.1109/CVPR.2009.5204291)

[View article](javascript:;)

[View in Scopus](https://www.scopus.com/inward/record.url?eid=2-s2.0-70449588909&partnerID=10&rel=R3.0.0)[Google Scholar](https://scholar.google.com/scholar_lookup?title=Automatic%20recognition%20of%20fingerspelled%20words%20in%20british%20sign%20language&publication_year=2009&author=S.%20Liwicki&author=M.%20Everingham)

Mohammad Mahadi Hasan, M. Khaliluzzaman, S.A. Himel, R.T Chowdhury

Hand sign language recognition for Bangla alphabet based on Freeman chain code and ANN

Proceedings of the 4th international conference on advances in electrical engineering, ICAEE (2017), pp. 749-753, [10.1109/ICAEE.2017.8255454](https://doi.org/10.1109/ICAEE.2017.8255454)

2017, 2018-Janua

P.K. Athira, C.J. Sruthi, A. Lijiya

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 (2019), pp. 0-10, [10.1016/j.jksuci.2019.05.002](https://doi.org/10.1016/j.jksuci.2019.05.002)

A.A. Barbhuiya, R.K. Karsh, R. Jain

CNN based feature extraction and classification for sign language

Multimedia Tools and Applications, 80 (2) (2021), pp. 3051-3069, [10.1007/s11042-020-09829-y](https://doi.org/10.1007/s11042-020-09829-y)

W. Tao, M.C. Leu, Z. Yin

American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion

Eng. Appl. Artif. Intell., 76 (2018), pp. 202-213

V. Jain, A. Jain, A. Chauhan, S.S. Kotla, A. Gautam

American sign language recognition using support vector machine and convolutional neural network

Int. J. Inf. Technol., 13 (2021), pp. 1193-1200, [10.1007/s41870-021-00617-x](https://doi.org/10.1007/s41870-021-00617-x)

I. Hmida and N. B. Romdhane, "Arabic sign language recognition algorithm based on deep learning for smart cities," The 3rd International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2022), Hybrid Conference, Sharjah, United Arab Emirates, 2022, pp. 119-127, doi: 10.1049/icp.2022.2426.

Azerbaijani sign language recognition using machine learning approach

Samir Aliyev1, Ali Abd Almisreb2 and Sherzod Turaev3

Published under licence by IOP Publishing Ltd

Journal of Physics: Conference Series, Volume 2251, Second International Conference on Robotics and Artificial Intelligence (RoAI 2021) 29/11/2021 - 30/11/2021 Online

Citation Samir Aliyev et al 2022 J. Phys.: Conf. Ser. 2251 012007

DOI 10.1088/1742-6596/2251/1/012007

M. E. Morocho Cayamcela and W. Lim, "Fine-tuning a pre-trained Convolutional Neural Network Model to translate American Sign Language in Real-time," 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2019, pp. 100-104, doi: 10.1109/ICCNC.2019.8685536.

Nakata, Norio, and Tsuyoshi Siina. 2023. "Ensemble Learning of Multiple Models Using Deep Learning for Multiclass Classification of Ultrasound Images of Hepatic Masses" Bioengineering 10, no. 1: 69. <https://doi.org/10.3390/bioengineering10010069>

Winston, J.J., Hemanth, D.J., Angelopoulou, A. et al. Hybrid deep convolutional neural models for iris image recognition. Multimed Tools Appl 81, 9481–9503 (2022). <https://doi.org/10.1007/s11042-021-11482-y>

C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

Alotaibi, B., Alotaibi, M. A Hybrid Deep ResNet and Inception Model for Hyperspectral Image Classification. PFG 88, 463–476 (2020). <https://doi.org/10.1007/s41064-020-00124-x>

Dhiman, G.; Juneja, S.; Viriyasitavat, W.; Mohafez, H.; Hadizadeh, M.; Islam, M.A.; El Bayoumy, I.; Gulati, K. A Novel Machine-Learning-Based Hybrid CNN Model for Tumor Identification in Medical Image Processing. Sustainability 2022, 14, 1447. <https://doi.org/10.3390/su14031447>

Saponara, S., Elhanashi, A. (2022). Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance. In: Saponara, S., De Gloria, A. (eds) Applications in Electronics Pervading Industry, Environment and Society. ApplePies 2021. Lecture Notes in Electrical Engineering, vol 866. Springer, Cham. <https://doi.org/10.1007/978-3-030-95498-7_2>

Koo, KM., Cha, EY. Image recognition performance enhancements using image normalization. Hum. Cent. Comput. Inf. Sci. 7, 33 (2017). <https://doi.org/10.1186/s13673-017-0114-5>

C. Saravanan, "Color Image to Grayscale Image Conversion," 2010 Second International Conference on Computer Engineering and Applications, Bali, Indonesia, 2010, pp. 196-199, doi: 10.1109/ICCEA.2010.192.

A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 2018 International Interdisciplinary PhD Workshop (IIPhDW), Świnouście, Poland, 2018, pp. 117-122, doi: 10.1109/IIPHDW.2018.8388338.

Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybernetics* **36**, 193–202 (1980). <https://doi.org/10.1007/BF00344251>

Goodfellow, Ian

Author Profile

; Bengio, Yoshua; Courville, Aaron

Deep learning. (English) Zbl 1373.68009

Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press (ISBN 978-0-262-03561-3/hbk; 978-0-262-33743-4/ebook). xxii, 775 p. (2016).

Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 (2021). <https://doi.org/10.1186/s40537-021-00444-8>

Dhruv, P., Naskar, S. (2020). Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review. In: Swain, D., Pattnaik, P., Gupta, P. (eds) Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1101. Springer, Singapore. <https://doi.org/10.1007/978-981-15-1884-3_34>

S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

Gers, F., 2001. *Long short-term memory in recurrent neural networks* (Doctoral dissertation, Verl

ag nicht ermittelbar).

References:

1. T. Starner, J. Weaver, and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1998.
2. D. A. Patel, K. D. Kothari, and H. P. Sinha, "Sign Language Recognition using Feature Extraction," International Journal of Emerging Technologies and Applications in Engineering, Technology and Sciences, vol. 7, no. 8, pp. 18-22, 2014.
3. E. J. Lleida, R. P. Vail, and D. P. W. Ellis, "A Spanish Sign Language Video Corpus for Sign Language Recognition," in Proceedings of the LREC Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora, 2012.
4. A. C. Kose, M. Y. Yemez, and A. Erçil, "BosphorusSL: A Large-scale Sign Language Recognition Corpus," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.
5. M. Tanaka, T. Kawamura, and Y. Fukui, "Japanese Sign Language Recognition Based on Data Glove and Image Processing," in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2004.
6. J. Zhao, L. Liu, Z. Tian, and L. Shao, "Chinese Sign Language Recognition using Multi-modal Features," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.
7. S. Prabhakar, J. L. Wyatt, A. R. Hanson, and D. P. Miller, "MOCAP-SLR: A Motion Capture Corpus for the Spoken Language and Gesture Interface," in Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 2004.
8. B. Zhang, Y. Zhu, J. Liu, L. Liu, and W. Jia, "American Sign Language Recognition Using Deep Learning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017.
9. L. S. Machado, A. H. F. Laender, and R. A.

# APPENDIX A: RESEARCH PROPOSAL

AI empowered Sign Language recognition using hybrid neutral networks

Ambar Saxena  
M.Sc. Data SCIENCE, LJMU

Research Proposal

NOVEMBER 2022

# Abstract

Hand gestures serve as the primary means of communication in sign languages, which are composed of visual gestures made up of hands, faces, and other bodily motions. Automatic sign language identification is a challenging system due to the diversity of the about 7000 modern sign languages as well as variances in motion position, hand shape, and body part positioning. Although sign language has become more common in recent years, communicating with sign language speakers or signers remains difficult for non-sign language speakers. There has been promising progress in the disciplines of motion and gesture detection utilizing Artificial Intelligent techniques as a result of recent advances in deep learning and computer vision.

The deep learning network makes full use of the advantages of time series classification provided by the recurrent neural network model as well as the feature extraction capabilities of convolutional neural networks in order to achieve more accurate recognition. High precision, scalability, and robustness, on the other hand, remain significant issues in sign language recognition research. The purpose of this research is to examine hybrid neural networks to improve the accuracy and robustness of sign language recognition. with satisfactory results in terms of several performance measures such as accuracy, precision, recall, F1-Score, etc.

# 1. Background

AI is made up of many fields, the most notable of which is Machine Learning. Deep learning, a part of machine learning, has now surpassed all other prediction techniques and algorithms. Deep learning approaches have benefitted several disciplines, including computer vision. Image categorization and recognition is one of the most widely utilized industrial applications of computer vision. Deep learning algorithms have been shown to produce cutting-edge outcomes in image classification applications. The most widely used deep learning algorithms for image categorization are convolutional neural networks (CNN).

A speech impediment is a disorder that affects a person's capacity to communicate through speech and hearing. People who are afflicted by this employ alternative communication methods such as sign language. Although sign language has become more common in recent years, communicating with sign language speakers or signers remains difficult for non-sign language speakers.

Because many sign languages are natural languages with grammar rules and vocabulary items, they can be recognized using artificial intelligence-based translation algorithms. As significant advances in natural language processing and computer vision technologies, notably image and video captioning, are made, related methods for improving automated sign language interpretation can be researched further.

Ensemble learning algorithms are heavily used in novel and high-performance medical image classification pipelines. The goal behind ensemble learning is to combine different models or forecasts to improve prediction performance. To improve generalization performance, ensemble learning mixes numerous separate models. Deep ensemble learning models incorporate the benefits of both deep learning models and ensemble learning. The primary purpose of research is to examine hybrid neural networks while training various 3DCNN and ensemble networks to increase the accuracy and precision of sign language recognition. The aim of this study is to use deep learning networks in artificial intelligence to recognize sign language gestures automatically.

# 2. Problem Statement OR Related Research OR Related Work

The majority of studies have concentrated to detect the static sign-language motions in pictures or video-clips that were acquired in a testing setting. Image capture, data pre-processing, image segmentation, extraction of features, and classification are the five categories into which the processes of Image-based sign language recognition (SLR) is classified. Gesture acquisition, the first step in sign language recognition, mostly carried out utilizing datasets that were either self-acquired or made publicly available. Preprocessing, the next step, enhances image quality by removing unnecessary noise. Following preprocessing, the area of interest is segmented and extracted from the complete picture. The input image area is converted into feature-vectors for detection in the fourth stage. To identify the target sign, classification, the final stage in vision-based SLR includes comparing the attributes of the input image to those already located in the database.

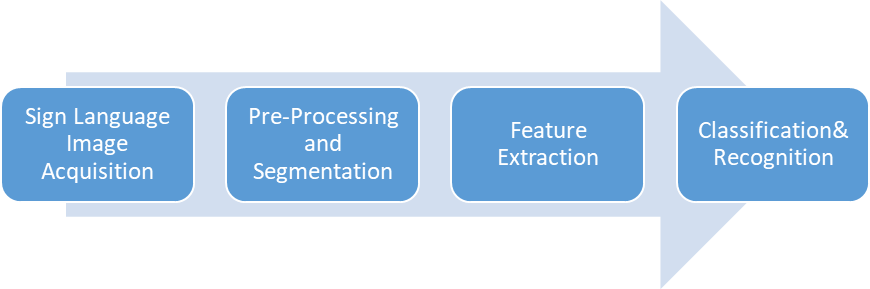


Figure 6.3‑1 : Sign Language recognition workflow using deep learning

**2.1 Image acquisition**

Many researchers use previously defined datasets like ASL Image Dataset (ASLID) , ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) -2010 , ASL Gesture Dataset 2012, RWTH-Phoenix-Weather dataset, ArSL databases and the SIGNUM . Few researchers develop their own datasets for data training due to scarcity of Sign-Language datasets for specific countries. To establish a dataset, researchers create the data from recording the signer. Image capturing techniques of various types were employed by the researchers. Devices used in image capturing are webcams or camera, data-gloves, Kinect, and Leap motion controller. Kinect is a useful and commonly used device. It offers a depth video feed as well as a color video stream. It easily recovers 3D hand motion trajectories and distinguishes the genuine sign image from the background [Kamal et al., 2019]. Although the leap motion controller has a smaller operational range than Kinect, it is less expensive and more accurate [Suharjito et al., 2017].

2.2 Pre-processing and segmentation

The image pre-processing phase optimizes the input with image and video modification. Preprocessing techniques are used to remove non desirable noise from an input image while simultaneously improving its quality. This can be performed through scaling, color conversion, noise removal, or a mix of these approaches from the original image. With a solid selection of preprocessing procedures, the output of this process can have a significant impact on accuracy. Image enhancement and picture restoration are the two primary categories of image preprocessing techniques.

Gaussian and Median filters are often used techniques for minimizing noise in images or videos. The removal of unwanted information from the input is frequently accomplished using morphological operations [Akmeliawati et al 2007] and median filtering, which is only employed for picture pre-processing in research [Raghuveera et al 2020]. As an illustration, Badhe et al. (2015) used K-means clustering in conjunction with morphological techniques to remove noise after thresholding the input image into binary.

The technique of breaking an image into discernible pieces is called image segmentation [Egmont-Petersen et al., 2002]. It's possible for the segmentation method to be contextual or not. While non-contextual segmentation groups pixels based on global properties, contextual segmentation, like edge identification techniques, considers the spatial connection between highlights. Edge detection, thresholding, region, clustering, and artificial neural network dependent image segmentation techniques are some of the categories.

**2.3 Feature extraction**

The feature extraction method is employed to extract the most important properties from an input image. Picture backdrop, image translation, scale, form, rotation, angle, and coordinates are among the properties. It tries to identify the aspects of the acquired image that stand out the most [Patil & Sinha, 2017]. It's a type of dimensional reduction in which the most interesting aspects of a photograph are successfully represented. In order to improve learning accuracy and precision, the unneeded component is removed to create the compact feature vector (Khalid et al., 2014). In order to achieve high recognition accuracy, the feature extraction procedure seeks for features that may successfully distinguish across classes. This contributes to the categorization step. PCA, histogram of oriented gradient (HOG), Fourier descriptor (FD), and shift-invariant analysis are the some of the primary feature extraction approaches utilized in SLR that have produced positive results.

**2.4** **Classification**

After the images have undergone pre-processing, segmentation, and feature extraction, a prediction method must be used to help give the extracted features useful meaning. The last step and most crucial one for the acceptance of gestures is classification. Machine learning enhances performance because it enables machines to learn much like humans do by performing jobs repeatedly. Importing and using models from published research is common practice because creating such models requires significant computational resources.

Wadhawan et al. (2020) concentrate on very widely used scalable networks that are transferred in transfer learning. AlexNet, created by LeNet, is a commonly used deep learning architecture.

D. Rathi (2018) used the GoogLeNet neural network, which has depth in both the directions. Very deep network with high accuracy.

Masood et al. (2018) classified 2624 ASL gestures using a pre-trained VGG16 model. Simonyan and Zisserman (2014) presented the VGG network architecture. Although the ResNet architecture can be successfully trained at deep depths, VGG-16 networks are regarded as being exceptionally deep. Sadly, VGG has two significant drawbacks: It is quite slow during training, and the network weights are fairly enormous.

The most significant deep learning neural network models are convolutional neural networks (CNN) for image recognition and classification, which is widely accepted. By alternating between a great number of layers of convolution and pooling, a CNN architecture can be created.

Mannan et al (2022) presented the performance of a DeepCNN architecture which improves with size of the dataset given, they used data augmentation to expand the input data artificially.

Katoch et al (2022) suggested a technique uses the sign langue gestures in a live video stream and predict labels in the form of texts. Segmentation was based on skin color as well as background too . SVM and CNN are used for classification.

Yirtici et al (2022) employed AlexNet as a pre-trained network, transfer learning method was used to classify TSL with average precision of 99.7%

Neeraj Gupta (2022) investigated different sign language motions and created a comparative evaluation of various neural network architectures . For the sign language recognition system, CNN models (GoogLeNet, AlexNet, VGGNet, and EfficientNet) were employed and investigated. The benchmarked ASL dataset was used for testing.

Sharma et al. (2021) used the Boston ASL LVD dataset to train 3DCNN, The training data comprises 70% of the whole dataset, whereas the testing dataset comprises 30%. The proposed approach beats existing models in terms of precision (3.7%), f-measure (3.9%) and recall (4.3%).

Bheda et al. [12] suggested a letter and digit recognition system based on American Sign Language. The proposed CNN-based architecture produced an accuracy of 82.5% .

Rao et al. [13] created a dataset that shows signs at different angles and against different backdrops. They classified them using CNN with different pooling algorithms and the stochastic pooling approach beat the other pooling strategies.

Koller et al. [14] suggested a hybrid technique for continuous sign recognition that combines CNN's strong discriminative features with the Hidden Markov Model's sequence modeling capability (HMM). The obtained data was preprocessed using a dynamic programming method. According to the results, the hybrid CNN-HMM technique outperforms the other known approaches.

Wadhwan et al. (2020) evaluated more than 50 CNN models. The data were further evaluated using multiple optimizers, and it was determined that the suggested technique attained the maximum training efficiency of 99.72% on colorful pictures and 99.90% on monochrome images.

Deep ensemble models have been used to classify X-ray and CT-scan images for disease diagnosis in medical applications. For instance, a stacked ensemble of CNNs was utilized by Kandel et al. (2021) to detect fractures in X-ray pictures. In order to provide input to eight different conceptual models, the authors combined the stochastic outputs of different CNN models. The ensemble model was said to be 10% more accurate than individual CNN models.

Many academics are creating their own tiny datasets to use in the development of their SLR. There are still certain nations and languages for which large databases are unavailable. The type of sign language used in the majority of nations depends entirely on their grammar and how each phrase is presented, such as by utilizing words or phrases. The categorization method used to distinguish sign language varies between scholars as well. Comparing these method to each other in the Sign Language Recognition System remains arbitrary when using their notions and limitations. Based on deep learning approaches like CNN, LSTM, and Bi- LSTM Models exhibit high classification efficiency in a stream of images and videos.

# 4. Aim and Objectives

The main aim of this research is to analyze hybrid neural networks while training different 3DCNN and ensemble networks to improve the accuracy and robustness of sign language recognition. The aim of the research is to automatically identify the sign language gestures using Artificial Intelligence’s deep learning networks.

The following are the research objectives, which are based on the goal of this study:

* To develop and evaluate the effectiveness of various deep learning models
* To recommend an appropriate deep learning-based network for the recognition of sign language.
* To compare the classification models to find the model that best categorizes sign language gestures.
* To assess the effectiveness of suggested models

# 5. Significance of the Study

A language made up of signals made from hand motions and facial expressions is known as sign language. Those with weak or no hearing use this language to communicate. Hearing-impaired persons may express themselves more freely and more effectively using this kind of communication, which also helps to close the communication gap between them and others.

The transdisciplinary problem of automatic identification of human signs has not yet been entirely resolved. For the recognition of sign language, a variety of methods, including the use of machine learning techniques, have been employed recently. This study attempts to increase the precision of automatic sign language gesture identification by training a hybrid deep learning network.

# 6. Scope of the Study

The scope of this study is limited to:

* Pre-processing and recognizing static sign language using deep learning networks , CNN, LSTM etc.
* Generating insights on model accuracy with various networks
* The study does not cover Dynamic or continues sign language gestures.

# 7. Research Methodology

The methodology used in this study will include essential sign language recognition system operations like input data selection, pre-processing of the testing data, transformation into a structured and understandable data format, dataset-balancing, application of supervised deep learning techniques, and evaluation of machine learning performance using evaluation matrix. These procedures are as follows:

**7.1 Dataset Description:**

In this study I will use two different set of datasets which consists of images of alphabets from the American Sign Language and sign language digits images respectively.

* **ASL Alphabet :** The data set contains images of American Sign Language alphabets grouped into 29 folders representing the various classes. Total 87,000 images with 200x200 pixel size sit in the training data set. There are 29 classes, 26 letters from A to Z overall, and three letters each for "SPACE," "DELETE," and "NOTHING." Real-time classification and applications benefit greatly from these three types. Only 29 photos make up the test data collection, which encourages the usage of test images taken in actual environments.



Figure 6.3‑1: American Sign Language alphabets

* **Sign Language Digits Dataset:** This Dataset is prepared by students of “Turkey Ankara Ayrancı Anadolu High School”. In this data there are 2062 sign language digits images. As you know digits are from 0 to 9. Therefore, there are 10 unique sign. The dataset consists of images with one-handed display of digits 0 to 9 in sign language. The images are 100x100 in size and RGB in color. It was obtained by 218 students making 10 different signs once. There should be a total of 2180 samples, while there are 2062 samples in the data set. This is probably because some unfavorable images have been removed by creator of the dataset.



Figure 6.3‑2: Sign Language Digits Dataset

**7.2 Data Pre-Processing**

It is difficult to determine how to optimally prepare visual data while training a convolutional neural network. Data preparation is the use of several morphological methods to minimize noise in data. Before using this in any Deep Learning project, we may want to perform a few pre-processing tasks. Some of the most common are listed in the paragraphs below.

Verifying sure the photos are the same size and aspect ratio is one of the first duties. The majority of neural network models require square input images, so each image must be examined to determine its squareness and then cropped appropriately.

Image scaling: We can scale each image correctly once we have made sure that all images are square (or have a specific aspect ratio). We selected images that were 100 pixels wide and high. Each image's width and height will need to be scaled.

Each input parameter (in this example, pixel) needs to have an uniform data distribution, thus it is essential to first normalize the data. This facilitates network training while accelerating convergence.

Dimensional reduction: A single gray-scale channel can be created by combining the RGB channels. When the performance of neural networks is permitted to be independent of that dimension, there are usually considerations for lowering other dimensions or making the training problem more tractable.

* Convert RGB to gray scale
* Image enhancement and contrast
* Noise filtering and removing
* Resizing

**7.3 Data augmentation**

The inclusion of the original data set together with modified copies of older photos is another typical pre-processing strategy called augmentation. Common affine transformations include scaling, rotations, and others. To expose the neural network to a number of various factors augmentation is applied. As a result, there is a lower chance that the neural network would identify negative characteristics in the data set.

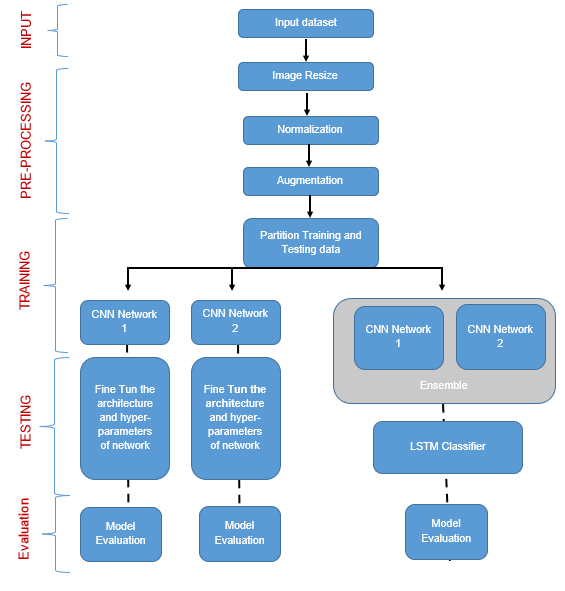


Figure 6.3‑3: Methodology workflow for AI based Sign Language recognition

**7.4 Model training**

Convolutional neural networks will be used for the model training. The suggested model will undergo training. The classifier places the preprocessed sign images in the appropriate category after classification. The dataset of several ASL signs will be used to train the classifier. The training set will comprise 80% of the dataset after it has been separated into training and validation sets. To bring unpredictability into the deep learning model training process, the dataset must be randomly rearranged. This prevents the model from favoring specific parameters. I'll be evaluating and training for the following CNN networks

* Few different CNN networks (least layers to deep layers)
* Ensemble of few CNNs of varying resolutions followed by classification using LSTM

**7.5 Testing & Hyper parameters Tunning**

The constructed sign language recognition system will be tested using various hyperparameters. Some other parameters will be tested in order to fine-tune the network design Based on early findings and after employing a few optimizations to increase model precision and accuracy with best CPU/GPU compute usage.

The network will be tested using a variety of optimizers, including Stochastic Gradient Descent, Adaptive Moment Estimation (Adam), Adagrad, and Adadelta (SGD).

**7.6 Experimentation and Evaluation**

The effectiveness of the network obtained in this study for American Sign Language recognition will be evaluated in relation to the model training parameters, which include the number of layers, filters, and optimizers. The models' average precision, recall, F1-score, and accuracy will be computed and compared.

# 8. Requirements Resources

Table 0‑1 : List of Hardware and Software required

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| Laptop/Computer 4GB RAM / 250GB Hardisk | Microsoft office |
| Google Colab GPU 16GB/ 1.59 GHz | Jupyter/Anaconda/Python |
| Kaggle Kernel 16GB /1.32GHz | Google Colab |

# 9. Research Plan

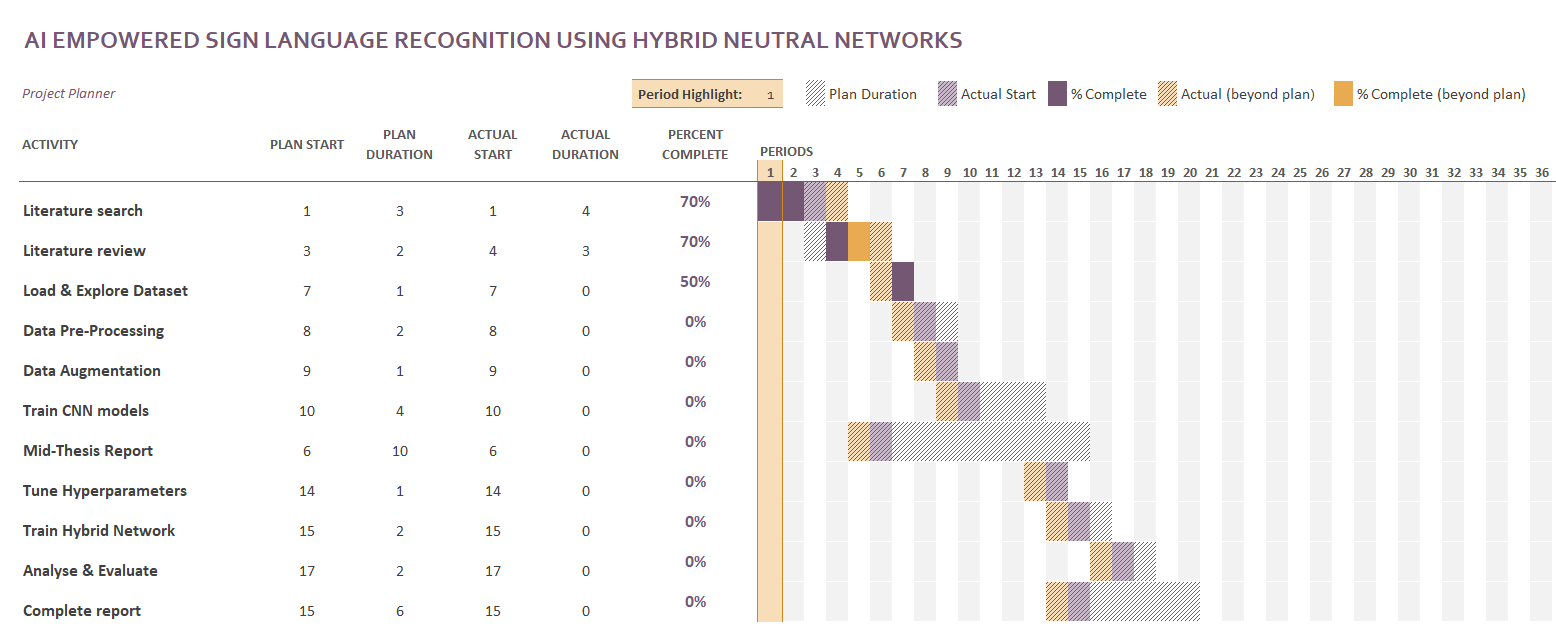


Figure 6.3‑1: Gantt Chart

# 10. Risk & Contingency Plans

Table 0‑1: Associated risk and mitigation plan

|  |  |
| --- | --- |
| **RISK** | **Contingency Plan** |
| Proposed methodology does not work | summaries all the experiments performed |
| High GPU requirement | reduce dataset / network layers to fit into available GPUs |
| non- availability of required software/hardware | discuss with Upgrad student support |

# 

# References

References

Close, R.A., A Reference Grammar for Students of English, Longman, 1975.

Gowers, E.G., (revised by B. Fraser), The Complete Plain Words, Penguin,

England, 1977.

O’Connor, M. and Woodford, F.P., Writing Scientific Papers in English,

Elsevier, Amsterdam, 1975.

Parsons, C.J., Thesis and Project Work, Allen and Unwin, London, 1973.

Trelease, S.F., How to write Scientific and Technical Papers, M.I.T. Press,

Cambridge, Mass., 196

Reichard, R. and Johnson, S.K. (2011) Leader self-development as organizational strategy. The Leadership Quarterly, 22 (1), pp.32-42

Franke, F.R. and Chasim, B. (1981) Peasants, peanuts, profits and pastoralists. Ecologist, v.11 (4), pp.156-168

Reichard, R. and Johnson, S.K. (2011) Leader self-development as organizational strategy. The Leadership Quarterly, 22 (1), pp.32-42

Franke, F.R. and Chasim, B. (1981) Peasants, peanuts, profits and pastoralists. Ecologist, v.11 (4), pp.156-168

Mavi, A., (2020), “A New Dataset and Proposed Convolutional Neural Network Architecture for Classification of American Sign Language Digits”, arXiv:2011.08927

I.A.Adeyanju, O.O.Bello, M.A.Adegboye (2021) Machine learning methods for sign language recognition: A critical review and analysis. Intelligent Systems with Applications v.12, <https://doi.org/10.1016/j.iswa.2021.200056>

S. Subburaj, S. Murugavalli (2022) Survey on sign language recognition in context of vision-based and deep learning . Measurement: Sensors, v.23, <https://doi.org/10.1016/j.measen.2022.100385>

S.M. Kamal, Y. Chen, S. Li, X. Shi, J. Zheng (2019) ,**Technical approaches to Chinese sign language processing: A review,** IEEE Access, 7 (2019), pp. 96926-96935, [10.1109/ACCESS.2019.2929174](https://doi.org/10.1109/ACCESS.2019.2929174)

Suharjito, R. Anderson, F. Wiryana, M.C. Ariesta, G.P Kusuma (2017) **Sign language recognition application systems for deaf-mute people: A review based on input-process-output,** Procedia Computer Science, 116 (2017), pp. 441-448, [10.1016/j.procs.2017.10.028](https://doi.org/10.1016/j.procs.2017.10.028)

R. Akmeliawati, M.P. Ooi, Y.C. Kuang (2007) **Real-time Malaysian sign language translation using colour segmentation and neural network** IEEE Instrumentation & Measurement Technology Conference IMTC 2007, Warsaw (2007), pp. 1-6

T. Raghuveera, R. Deepthi, R. Mangalashri, *et al.(2020)* **A depth-based Indian sign language recognition using Microsoft Kinect** Sādhanā, 45 (2020), p. 34, [10.1007/s12046-019-1250-6](https://doi.org/10.1007/s12046-019-1250-6)

P.C. Badhe, V. Kulkarni (2015) **Indian sign language translator using gesture recognition algorithm**  IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar (2015), pp. 195-200

M. Egmont-Petersen, D. De Ridder, H. Handels (2002) **Image processing with neural networks- A review.** Pattern Recognition, 35 (10) (2002), pp. 2279-2301, [10.1016/S0031-3203(01)00178-9](https://doi.org/10.1016/S0031-3203(01)00178-9)

S.B. Patil, G.R. Sinha (2017) **Distinctive feature extraction for Indian sign language (ISL) gesture using scale invariant feature transform (SIFT)** Journal of The Institution of Engineers (India): Series B, 98 (1) , pp. 19-26, [10.1007/s40031-016-0250-8](https://doi.org/10.1007/s40031-016-0250-8)

S. Khalid, T. Khalil, S. Nasreen (2014) **A survey of feature selection and feature extraction techniques in machine learning.** Proceedings of the 2014 science and information conference, SAI 2014 , pp. 372-378, [10.1109/SAI.2014.6918213](https://doi.org/10.1109/SAI.2014.6918213)

Wadhawan, A., Kumar, P. Deep learning-based sign language recognition system for static signs. *Neural Comput & Applic* **32**, 7957–7968 (2020). <https://doi.org/10.1007/s00521-019-04691-y>

D. Rathi (2018) **Optimization of Transfer Learning for Sign Language Recognition Targeting Mobile Platform** arXiv preprint arXiv:1805.06618

S. Masood, H.C. Thuwal, A. Srivastava, S. Satapathy, V. Bhateja, S. Das (2018) **American sign language character recognition using convolution neural network.** Smart Computing and Informatics. Smart Innovation Systems and Technologies, vol. 78, Springer, Singapore (2018)

K. Simonyan, A. Zisserman (2015) **Very deep convolutional networks for large-scale image recognition .** International Conference on Learning Representations, ICLR (2015)

[Abdul Mannan](https://pubmed.ncbi.nlm.nih.gov/?term=Mannan+A&cauthor_id=35535197), [Ahmed Abbasi](https://pubmed.ncbi.nlm.nih.gov/?term=Abbasi+A&cauthor_id=35535197), [Abdul Rehman Javed](https://pubmed.ncbi.nlm.nih.gov/?term=Javed+AR&cauthor_id=35535197), [Anam Ahsan](https://pubmed.ncbi.nlm.nih.gov/?term=Ahsan+A&cauthor_id=35535197), [Thippa Reddy Gadekallu](https://pubmed.ncbi.nlm.nih.gov/?term=Gadekallu+TR&cauthor_id=35535197), [Qin Xin](https://pubmed.ncbi.nlm.nih.gov/?term=Xin+Q&cauthor_id=35535197) (2022) Hypertuned Deep Convolutional Neural Network for Sign Language Recognition. Computational Intelligence and Neuroscience Apr 30;2022:1450822.  doi: 10.1155/2022/1450822.

Shagun Katocha, VarshaSingh, Uma Shanker Tiwary (2022) Indian Sign Language recognition system using SURF with SVM and CNN. Array v.14 Article 100141. <https://doi.org/10.1016/j.array.2022.100141>

Yirtici, T., Yurtkan, K. (2022) Regional-CNN-based enhanced Turkish sign language recognition. SIViP 16, 1305–1311 <https://doi.org/10.1007/s11760-021-02082-2>

Gupta, N. (2022). Sign Language Recognition Using Diverse Deep Learning Models. In: Sanyal, G., Travieso-González, C.M., Awasthi, S., Pinto, C.M., Purushothama, B.R. (eds) International Conference on Artificial Intelligence and Sustainable Engineering. Lecture Notes in Electrical Engineering, vol 837. Springer, Singapore. <https://doi.org/10.1007/978-981-16-8546-0_38>

Sharma, S., Kumar, K. ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. *Multimed Tools Appl* **80**, 26319–26331 (2021). <https://doi.org/10.1007/s11042-021-10768-5>

Bheda V, Radpour D (2017) Using deep convolutional networks for gesture recognition in American sign language. arXiv preprint [arXiv:1710.06836](http://arxiv.org/abs/1710.06836)

Rao GA, Syamala K, Kishore PVV, Sastry ASCS (2018) Deep convolutional neural networks for sign language recognition. In: IEEE conference on signal processing and communication engineering systems (SPACES), pp 194–197

Koller O, Zargaran S, Ney H, Bowden R (2018) Deep sign: enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. Int J Comput Vis 126(12):1311–1325

Wadhawan, A., Kumar, P. Deep learning-based sign language recognition system for static signs. *Neural Comput & Applic* **32**, 7957–7968 (2020). <https://doi.org/10.1007/s00521-019-04691-y>

I. Kandel, M. Castelli and A. Popovič, "Comparing Stacking Ensemble Techniques to Improve Musculoskeletal Fracture Image Classification", Journal of Imaging, vol. 7, no. 6, pp. 100, 2021.

Pulkit Rathi, Kuwar Gupta, Raj, Soumya Agarwal, Anupam Shukla (2020) **sign language recognition using ResNet50 deep neural network architecture .** <https://doi.org/10.2139/ssrn.3545064>

P. Molchanov, S. Gupta, K. Kim, J. Kautz (2015)

**Hand gesture recognition with 3D convolutional neural networks**

2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, Boston, MA (2015), pp. 1-7, [10.1109/CVPRW.2015.7301342](https://doi.org/10.1109/CVPRW.2015.7301342)

Tushar AK, Ashiquzzaman A, Islam MR (2017) Faster convergence and reduction of overfitting in numerical hand sign recognition using DCNN. In: Humanitarian technology conference (R10-HTC), IEEE Region 10, pp 638–641

Oyedotun OK, Khashman A (2017) Deep learning in vision-based static hand gesture recognition. Neural Comput Appl 28(12):3941–3951