NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF SOFTWARE ENGINEERING

SIGN LANGUAGE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

M.Sc THESIS

MELTOH M. YOKPE

Nicosia
December 2022

NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF SOFTWARE ENGINEERING

SIGN LANGUAGE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

M.Sc THESIS

MELTOH M. YOKPE

Supervisor
Assoc. Prof. Dr. Kamil Dimililer

Nicosia
December 2022

Approval

We certify that we have read the thesis submitted by Meltoh M. Yokpe titled **"Sign Language Detection Using Convolutional Neural Networks"** and that in our combined opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Educational Sciences.

Examining Committee	Name-Surname	Signature
Head of the Committee:	Assoc. Prof. Dr. Boran Sekeroglu	....
Committee Member:	Assoc. Prof. Dr. Yoney Kirsal-Ever	....
Supervisor:	Assoc. Prof. Dr. Kamil Dimililer	....

Approved by the Head of the Department

30 / 12 /2022

Assoc. Prof. Dr. Boran Sekeroglu
Head of Department

Approved by the Institute of Graduate Studies

..../..../2022

Prof. Dr. Kemal Hüsnü Can Başer
Head of the Institute

Declaration

I hereby declare that all information, documents, analysis, and results in this thesis have been collected and presented according to the academic rules and ethical guidelines of the Institute of Graduate Studies, Near East University. I also declare that as required by these rules and conduct, I have fully cited and referenced information and data that are not original to this study.

Meltoh M. Yokpe

…. /…. /2022

Acknowledgment

Words cannot explain how grateful I am to my supervisor, Assoc. Prof. Dr. Kamil Dimililer, for his tremendous patience and comments. He led me through every step of the thesis documentation process. This thesis would not have been possible without his persistent and insightful instructions.

I also could not have embarked on this path without the assistance of my course advisor, Assoc. Prof. Dr. Boran Şekeroğlu has been extremely helpful throughout my master's degree. I am grateful to my defense committee members who willingly shared their knowledge and experience.

Over everything else, I would like to express my sincere gratitude and love to my family members for their commitment and unwavering belief in me. I owe a great deal to my father, Dr. Mathew Yokpe, who is both my inspiration and the one who guided me to the knowledge's riches. I want to express my gratitude to my late mother for her unwavering love, support, and encouragement throughout my life.

I have a large list of friends I want to thank at this point. I can't name everyone, but I want to express my sincere gratitude to them for their invaluable assistance and support from the time I started my studies until now. Also deserving of praise are the university's librarians, research assistants, and study participants who influenced and motivated me.

Meltoh M. Yokpe

Dedication

This Thesis is dedicated to my late Mom, Patricia Yokpe. You will forever be my strength and motivation.

Abstract

Sign Language Detection Using Convolutional Neural Network

Yokpe, Meltoh
MA, Department of Software Engineering

December 2022, 61 Pages

Convolutional Neural networks (CNN) were used to analyze sign language detection and create a workable system that might be used in the future. The development of sign language identification, the role that machine learning plays in this process, as well as the viability and effectiveness of a python program in this domain, have all been significantly influenced by neural networks. Various ideas and previous literature were examined regarding the study's focus on computational learning. The data gathered for this investigation were from both primary and secondary sources. The convolutional neural network (CNN) model of hand gesture detection was developed using a critical literature assessment, combining qualitative and experimental research goals. Neural networks have greatly improved the field and communication of sign language and its potential for improvement is limitless; given the sheer number of people who could benefit from sign language detection systems, there is little doubt that the current momentum will continue for the foreseeable future. CNNs can automate the feature construction process so that intricate, handcrafted features are not required.

Given that CNN has shown outstanding results in image categorization and pattern recognition tasks, this study presents a deep learning technique for developing a reliable and real-time ASL identification system using CNN. With excellent accuracy, we can identify American sign motions. The prediction model has a cross-validation accuracy of 99% and can be generalized to users and environments that were not present during training.

Keywords: convolutional neural networks (cnn), neural networks, python, deep learning, american sign language (asl).

Özet

İşaret dili tespitini analiz etmek ve gelecekte kullanılabilecek uygulanabilir bir sistem oluşturmak için Evrişimli Sinir ağları (CNN) kullanıldı. Sinir ağları, işaret dili tanımlamasının geliştirilmesinde ve makine öğreniminin bu süreçte oynadığı rolün yanı sıra bu alandaki bir python programının fizibilitesi ve verimliliği üzerinde önemli bir etkiye sahiptir. Çalışmanın hesaplamalı öğrenmeye odaklanmasıyla ilgili çeşitli fikirler ve önceki literatür incelenmiştir. Bu araştırmanın bir parçası olarak, veriler birincil ve ikincil kaynaklardan elde edilmiştir. Niteliksel ve deneysel araştırma amaçları, eleştirel bir literatür taraması kullanılarak işaret dilinin tespiti için bir evrişimsel sinir ağı (CNN) prototipinin geliştirilmesinde birleştirildi. Sinir ağları, işaret dili alanını ve iletişimini büyük ölçüde geliştirmiştir ve gelişme potansiyeli sınırsızdır; İşaret dili tespit sistemlerinden yararlanabilecek çok sayıda insan göz önüne alındığında, mevcut ivmenin öngörülebilir gelecekte devam edeceğine dair çok az şüphe var.

CNN'ler, karmaşık, el yapımı özelliklerin gerekli olmaması için özellik oluşturma sürecini otomatikleştirebilir.

Bir derin öğrenme metodolojisi olarak CNN, görüntü sınıflandırma ve örüntü tanıma görevlerinde olağanüstü performans gösterdiğinden, güvenilir ve gerçek zamanlı bir ASL tanımlama sistemi geliştirmek için bu çalışmada CNN tabanlı bir derin öğrenme yöntemi sunulmaktadır. Amerikan işaret hareketlerini mükemmel bir doğrulukla tanımlayabiliriz. Tahmin modeli, %99 çapraz doğrulama doğruluğuna sahiptir ve eğitim sırasında mevcut olmayan kullanıcılara ve ortamlara genelleme yapabilir.

Anahtar kelimeler: Evrişimli Sinir ağları, Sinir ağları, Amerikan işaret dili, Derin öğrenme, piton programlama.

Table Of Contents

List Of Tables

List Of Figures

List Of Abbreviations

ASL: American Sign Language.

AI: Artificial Intelligence.

API: Application Programming Interface.

ArSL: Arabic Sign Language.

APA: American Psychological Association.

BSL: British Sign Language

BSD: Berkeley Source Distribution

CNN: Convolutional Neural Network

CSL: Chinese Sign Language

FCM: Fuzzy C-Means

GHM: Global Hybrid Model

GPU: Graphics Processing Units

HMM: Hidden Markov Models

HCI: Human-Computer Interaction

IDE: Integrated Development Environment

MDP: Markov Decision Process

MNIST: Modified National Institute of Standards and Technology

MLP: Multilayer Perceptron

SLT: Statistical Learning Theory

SLR: Sign Language Recognition

SML: Symbolic Math Library

CHAPTER I

Introduction

Communication is essential in our daily lives since it allows us to express our thoughts, feelings, ideas, views, and understanding with one another. It's the exchange of information between people that is called communication. Using both verbal and non-verbal methods, this procedure comprises both spoken words and gestures and postures of the communicators as ways of communication. Sign language essentially is a form of nonverbal communication that can be used to express oneself (Tripathi & Nandi, 2015). The deaf and dumb communities throughout the globe rely on this. Protocols for sign language vary from area to region. The most often-used sign language is American Sign Language (ASL). Learning sign language may be beneficial to people with autism, Down syndrome, or speech challenges.

When communicating nonverbally, the body language of the signer involves everything from facial emotions and hand gestures to their physical condition and posture, as well as their gaze. There are between half a million and two million persons in the United States who communicate primarily using sign language (Zell A, 2013). These figures may vary from various sources. However, Trudy Suggs' book mentions that it is quite popular. American sign language is among the top three languages used in the U.S. It seems that 3.67 percent of Americans have a hearing impairment, while 0.3 percent are functionally deaf. In both Canada and the United States, the main language of communication for the deaf and people who find it difficult to hear is American Sign Language (ASL) (Garcia & Alarcon, 2016).

Numerous approaches to translating signals based on gestures and physical characteristics have been developed in the past. Initially, a finger mouse that enables users to manipulate the mouse with their fingertips was developed by Ko & Yang (Suharjito, 2018). The colored glove approach, skin color segmentation, appearance modeling for video sequences, and Hidden Markov Model (HMM) systems are all unique methods. Every aspect of your body has a job to perform. Static and dynamic motions are used in sign language. A signer's hand motions, known as "static gestures," enable the signer to convey his or her thoughts and allow him or her to communicate effectively with others.

Hand gestures are the most common means of human-computer interaction (Neiva, 2018). When a camera detects the motions of a person's body, the information is transferred to a computer, which uses it as input to run various hardware and software (Balakrishna, 2012). As an example, the sound of cymbals crashing together may be generated by a person's hands clasped in front of a camera (Garcia & Alarcon, 2016). Gesture recognition may be used to assist people with physical disabilities to connect with computers, for example, by translating sign language into text. The initial stage in a computer's understanding of human body language is the identification of hand gestures (Sahoo, 2014). Human-computer interaction (HCI) applications including smart TV control, video gaming, telesurgery, and virtual reality all benefit from this technology. Transcribing sign language using hand gesture recognition is a critical use of the technology (Siming, 2019).

The hand movements used in sign language have a complicated structure because they transmit vital information and emotions about humans (Balakrishna, 2012). Global and local finger configurations are two fundamental aspects of these manual expressions that might be used as a starting point. Frame-by-frame analysis of these complementing primitives is necessary for an effective recognition system (Anetha, 2014). It is, however, difficult to compare primitives in Euclidean space because of the time-dependent nature of these frames Most current recognition methods merely consider the hand's local setup (Garcia & Alarcon, 2016). Hand segmentation preprocessing utilizing skin color models or colored gloves is used as input for these systems. This kind of system works well for basic alphabets and numbers, but it doesn't work as well for true sign language movements since it relies on the global setup.

Other approaches merely consider the overall body structure, not the individual finger configurations (Hasanbegovic, 2013). HCI applications using a limited number of basic and well-defined gestures have proven effective, but actual sign language gesture detection has been unsuccessful with these systems. Using gestures like finger-pointing instead of input devices like joysticks, mouse, and keyboards, the technology has the potential to revolutionize the way people interact with computers (Anetha, 2014). It is not necessary for the user to wear any special equipment or to connect any gadgets to their body to utilize gesture recognition. A camera instead of sensors on a gadget like a data glove reads the body's motions. Gesture recognition software can analyze hand and body gestures as well as facial and speech expressions (lip-reading) and eye tracking.

1.1 Why Convolutional Neural Networks?

Given CNN's impressive performance in image categorization and pattern recognition tasks, this study proposes a deep learning technique for developing a reliable and real-time ASL identification system using CNN. Comparisons with other techniques have demonstrated CNNs' superiority and capabilities. CNN requires less preprocessing than other image classification techniques. The network is taught various filters, many of which are hand-invented in other systems. By using a CNN, the images are simplified to a simple structure while retaining qualities that are critical in giving good estimations. The CNN model discussed in this research was tested using the ASL dataset as well as other datasets.

1.2 Statement of the Research Problem

Hand signals and gestures are used by those who are unable to speak. The language they speak is difficult for the average person to grasp. As a result, a system is required that can identify various signals and gestures and relay relevant information to the general public (Anetha, 2014). It serves as a link between persons with disabilities and the general public (Balakrishna, 2012). Knowing how to translate the symbolic gestures used by those who are hard of hearing or blind (Text). Several academics have been trying to recognize different sign languages to help deaf people communicate better, and one of these researchers has developed a translator that can translate sign language into normal spoken language and vice versa (Garcia & Alarcon, 2016).

When put into use, some systems designed to carry out these tasks stand out for their efficacy, practicality, and performance. Data gloves are used in glove-based systems to save the precise location of hand motions that are directly measured. Such a system is costly and onerous for the end-users, making it a poor investment. In contrast, a camera-based vision system has a challenge since these systems must be backdrop invariant, light insensitive, and human and camera independent to achieve real-time performance. However, because they just have a few prerequisites, they are seen as more convenient, natural, and less expensive than glove-based approaches (Sahoo, 2014).

Corneliu Lungociu developed a real-time sign language recognition system based on webcam data for the 80 letter signs from the English alphabet, and his system proved to be 80 percent accurate. The photos were preprocessed by calculating the Fourier Descriptor. Under laboratory circumstances, Neiva, F (2018) used an appearance-based global hybrid model (GHM) system developed from an MLP-based feature. A system for identifying static gestures with an average recognition accuracy of 80 percent was given by Balakrishna (2012). As a classifier, they employed Gaussian filtering to remove noise from grayscale. It was found that Suharjito (2018) used a vision-based approach to extract feature vectors from video frames that correctly identified 262 signs. The pictures recorded by a single video camera utilizing a modular frame grabber device were classified based on rules.

Using different degrees of preprocessing to enhance recognition, Zell (2013) trained a picture library of Indian sign language. The Gabor filter is used to extract texture features, while Chan-Vese active contour models are used to extract shape characteristics, forming a feature matrix. On a typical data set, the average recognition rate was roughly 98.61 percent.

1.3 Motive of this Research

The motive behind this paper is to give an outline of Convolutional Neural Networks (CNN)-based sign language identification and a functional system that can be used as a foundation for its implementation.

1.4 Research Questions

i. How has sign language detection been influenced by neural networks?
ii. What is the role of machine learning in sign language detection using neural networks?
iii. How feasible and efficient can a python program be in the detection of sign language?
iv. How can sign language be improved using neural networks?

1.5 Significance of the Study

Two opposing yet complementary motives drive this investigation. The first is that a system of sign language might help bridge the gap between the hearing and the Deaf communities. Second, the use of neural networks in the creation of such a system provides the potential for further research and development of the networks themselves. There is a first motivating element in the reduction of the communication barrier between the deaf and hearing populations. The difficulties that deaf persons have while interacting with the general population have been extensively documented. There are many similarities between the Deaf community and an ethnic group in that they are a distinct cultural and linguistic group within society (in this case sign language).

This study is also being done because the challenge of reading signs is seen as a promising direction for neural network development. Most of the research on neural networks has been done on fictitious or 'toy' issues that have nothing to do with real-world difficulties. It is believed that applying these strategies to a real-world situation would provide fresh insights into neural network methodology, even if this study has been beneficial in creating the core procedures employed in neural networks.

1.6 Limitations of the study

The study shall be limited by the following constraints, which include.

i. Financial constraint
ii. Limitation in literature
iii. Time constraint

1.7 Definition of Terms

American Sign language: Deaf populations in the United States and much of Anglophone Canada use American Sign Language as their primary means of communication. Visual language ASL has both manual and non-manual characteristics, making it a full and well-organized visual language

Neural Network: These are sets of algorithms that use a method that imitates the ability of the human brain to recognize patterns in massive volumes of data. In this context, neural networks refer to natural or artificial systems of neurons.

Sign language: Sign languages are linguistics that conveys meaning using a visual-manual medium. In sign languages, meaning is expressed by combining manual articulations with non-manual components. Natural sign languages have their vocabulary and syntax.

CHAPTER II

Literature Review

2.1 Concept of Sign Language and Sign language Detection

As a form of communication for the deaf, sign language is a frequent method. People who are deaf in various nations use distinct sign languages, which is why sign language is not universal. Sign language's motions and symbols follow a logical structure (Muhammad, 2021). The term "sign" refers to a single gesture. In each sign, there are three separate parts: the form of your hands, where your hands are positioned, and how you move them. Visual-manual modality is used to transmit meaning in sign languages (sometimes referred to as signed languages) (Büyüksaraç, 2015).

Manual articulations and non-manual components are used to convey sign language. Sign languages have their syntax and vocabulary, just like any other natural language. Although there are some commonalities across various sign languages, they are not all universal or mutually understandable (Tripathi & Nandi, 2015). In 1620, Juan Pablo de Bonet published the first instructional book for the deaf. Italian physician Girolamo Cardano argued that it was not required to hear words to grasp concepts. The book was based on his study. To be clear, ASL is a completely distinct language from signed English. Those who are proficient in ASL communicate through their eyes, hands, faces, and other parts of their bodies (Zell A, 2013).

ASL uses a distinct set of words and syntax from English. It will be more difficult to learn to speak ASL as a language than to learn how to communicate using signs and fingerspelling. Both spoken and signed language are regarded by linguists as varieties of natural language, which means that they developed through time without rigorous preparation via an abstract and prolonged aging process (Suharjito, 2018). To avoid confusion, Body language, which is a type of nonverbal communication, must not be linked to Sign language. Because of their widespread usage, sign languages among deaf people have grown to be an essential part of their cultures (Neiva, 2018). Sign language is primarily used by the deaf and hard of hearing but can also be used by those who are unable to speak due to physical or medical reasons (additional and alternative communication) or by those who seem to have deaf relatives, such as children of deaf adults who have difficulty communicating (Balakrishna, 2012).

2.1.1 Types of Sign language

142 sign languages are included in the Ethnologue Languages of the World; however, this number is difficult to pin down owing to the regular creation of new sign languages at schools in rural populations with a high incidence of congenital deafness. Hand gestures, body language, and facial emotions all play a role in sign language, which is utilized by deaf people to express themselves more effectively (Sahoo, 2014). Although many people believe that sign language is universal, there are considerable differences in manual languages across regions. There are several factors at play when it comes to the creation of sign languages, which are similar to how spoken languages are formed. People who don't sign the same language can't comprehend each other since most sign languages aren't mutually intelligible (Siming, 2019).

Every school in certain countries, like Sri Lanka, has its unique sign language that is only known by the pupils who attend that school. Many other nations' sign languages are the same despite their various names; for example, the sign languages of Croatia and Serbia and India, and Pakistan are the same (Garcia & Alarcon, 2016).

PIDGIN SIGNED ENGLISH (PSE) OR SIGNED ENGLISH

Among deaf people in the United States, PSE is the most widely utilized form of sign language. The vocabulary is derived from ASL. However, the word order is in English. As a rule, unnecessary words like "to," "the," and "ed" are often omitted. For many instructors, PSE is easier to learn than either ASL or SEE.

SIGNING EXACT ENGLISH (SEE)

To sign the English language word for word, SEE is based on ASL's sign language, but with additional prefixes and tenses that provide the signer with more alternatives. Signers may expand their vocabulary by learning to utilize SEE. SEE seems to work better for English-speaking parents of deaf children since it provides a visual depiction of the English language itself (Hasanbegovic, 2013).

AUSLAN (AUSTRALIAN SIGN LANGUAGE)

While there are distinct signs for things like animals and colors in each dialect, the grammatical structure is the same in all Auslan dialects. Even though both the United States and Australia use the English language, the ASL and Auslan sign languages are very distinct (Anetha, 2014). It is a two-handed alphabet for Auslan and a one-handed alphabet for ASL.

BRITISH SIGN LANGUAGE (BSL)

Many varieties of BSL are spoken across the country, and they employ a two-hand alphabet. BSL was the primary mode of communication for 15,000 persons in England and Wales in 2011.

CHINESE SIGN LANGUAGE (CSL)

Using a one-handed alphabet, CSL's signs are visual representations of written Chinese characters. The Shanghai dialect of CSL is the most widely spoken among the several dialects of CSL (Anetha, 2014). As of the late 1950s, the language's development has continued, with the help of the Chinese National Association of the Deaf, which is doing all it can to spread the language's usage throughout China.

IRISH SIGN LANGUAGE (ISL)

There are many similarities between ISL and American Sign Language (ASL). However, it is not connected to spoken English or Irish languages. One hand is used to sign the letters. It is estimated that ISL has been in use for more than a thousand years among Ireland's deaf population. Australian, South African, Scottish, and English Catholic missionaries introduced ISL to these countries. Some older Auslan Catholics still utilize ISL variants of BSL today, as well as some older BSL speakers.

JAPANESE SIGN LANGUAGE (JSL)

When comparing JSL to other sign languages, you'll see that it relies on mouthing to tell between signals from letters in the alphabet. In addition, JSL makes stronger use of fingerspelling and air-drawn Japanese characters (Thakur, 2020). Aside from the fact that it draws substantially from the Japanese used in everyday life, JSL is a unique language unto itself.

SPANISH SIGN LANGUAGE (SSL)

More than 90,000 SSL signers are thought to be in use in Spain. SSL is distinct from ASL in the same manner as English and Spanish are distinct languages. Except for Catalonia, where Catalan Sign Language is used, and Valencia, where Valencian Sign Language is used, all of Spain utilizes SSL (Anetha, 2014).

2.1.3 Neural Networks

An artificial neural network uses algorithms to uncover hidden connections between data points by simulating the structure and operation of the brain (Kruti, 2019). Neural networks, in this context, are systems of neurons, either natural or artificial (Thakur, 2020). To get the most outstanding potential outcome, neural networks don't have to change their output criteria since they can adapt to changes in input. Neuronal networks, a type of artificial intelligence, are increasingly being applied in the development of trading systems. Like the brain's neural network, a neural network function similarly (Kruti, 2019). The mathematical function gathers and categorizes information in a neural network, known as a "neuron. "The network is extremely similar to two statistical techniques: curve fitting and regression analysis. A neural network's architecture consists of layers of connected nodes (Kruti, 2019). A perceptron, like multiple linear regression, is used to represent each node. A nonlinear activation function may be fed into the perceptron from a signal created by multiple linear regression. Neuronal networks are a collection of algorithms that use a method that imitates the ability of the human brain to find patterns in enormous amounts of data. Neural networks, in this context, are systems of neurons, either natural or artificial (Anetha, 2014).

Without needing to alter the output criterion, neuronal networks automatically adjust to changes in input and produce optimal results (Deepali, 2016). In the creation of trading systems, neural networks, that have strong foundations in artificial intelligence, are becoming more and more common (Kruti, 2019). Many algorithms replicate the functions of an organism's nervous system to identify patterns in large volumes of data (Kruti, 2019). Neurons and synapses, for example, are modeled on these connections. Forecasting and market research, as well as fraud detection and risk assessment, are just a few of the many financial industries that use these tools. Deep learning techniques make use of neural networks with several layers of processing (Anetha, 2014). In finance, Processes like time-series forecasting, algorithmic trading, categorization of securities, and credit risk modeling are all made possible with the use of neural networks (Chavan, 2020). A neural network performs similarly to the neural network in the brain. The mathematical function gathers and categorizes information in a neural network, known as a "neuron." Two statistical techniques in which the network is quite comparable to layers of linked nodes that form the structure of a neural network are regression analysis and curve fitting (Chavan, 2020). In a perceptron, each node is referred to as such because it resembles a multiple linear regression. A nonlinear activation function may be fed into the perceptron from a signal created by multiple linear regression. In information technology, A hardware or software system known as an artificial neural network (ANN) is based on how neurons in the human brain behave (Deepali, 2016). Artificial neural networks, or ANNs, are a subset of deep learning technology commonly referred to as "neural networks."

Commercial applications of these technologies are primarily focused on complex signal processing or pattern recognition (Chavan, 2020). Numerous significant commercial applications have evolved since the year 2000, such as check processing, speech-to-text conversion, data analysis for oil exploration, weather forecasting, and facial recognition. A lengthy and illustrious history of artificial neural networks goes back to the early days of computing. In 1943, mathematicians Warren McCulloch and Walter Pitts created this method to imitate the human brain's ability to run basic algorithms (Shamrat, 2021). As recently as approximately 2010 a new round of study began. Big data and parallel computing provided the training data and computational resources necessary for data scientists to operate massive artificial neural networks. In the 2012 ImageNet competition, a neural network outperformed humans at photo recognition. Ever since artificial neural networks (ANNs) rapidly gained popularity and their underlying technology has substantially advanced (Chavan, 2020).

2.1.4 Learning process of Neural network

In most cases, a considerable quantity of data is used to train or feed an ANN. Giving data and directing the network on what to do with it is the essence of training. Initial training may use images of actors, non-acting characters, masks, and statues of animals, to create a network that also can recognize actors' faces, for instance (Shailesh, 2020). There are descriptors for each input, including names of the actors or metadata indicating that the input is "not human" or "not an actor."

Many concepts are used by neural networks as they strive to define rules and make decisions based on inputs from previous layers. Examples of these include genetic algorithms, Bayesian methods, and gradient-based training. Inside the data to be represented, simple rules can be utilized to indicate object links (Sharma, 2014). It might be used to train a face recognition system, for example, "Above the eyes, or below the nose, a mustache can be discovered. Mustaches may be seen either above or next to the mouth." The use of preloading rules speeds up the learning process and boosts the model's overall strength.

However, conclusions about the scope of the issue may be useless and harmful, and deciding whether to incorporate rules is extremely important. Furthermore, due to the obvious presumptions, people make when developing algorithms, neural networks reinforce cultural biases (Manar, 2012). Biased data sets are a recurring issue for self-learning systems that discover patterns in data and learn. The method will transmit bias if the underlying data is skewed, which is nearly never the case (Shailesh, 2020).

2.1.5 Advantages of neural networks

Artificial neural networks have several benefits, including:

i. The network is capable of handling numerous jobs at once due to its parallel processing capabilities.
ii. A complete network, not just a database, holds information.
iii. Complex nonlinear interactions may be learned and modeled, which aids in stimulating the real-world relationships between the inputs and outputs.
iv. This implies that even if one or more neurons in the network are corrupted, the output will continue to be generated.
v. Instead of an issue ruining the network immediately, gradual corruption implies that the network will gradually deteriorate over time.
vi. How much of an impact the missing information has on production is determined by how critical it is to your success.
vii. There are no limits on the distribution of the input variables.
viii. The ANN can conclude events and act in accordance with what it has observed. Machine learning is the term used for this. An ANN must be able to understand underlying connections in the data without establishing any predefined associations in the learning process to simulate extremely volatile data with non-constant variance. Predicting the output of unknown data is made possible by ANNs' capacity to generalize and infer invisible correlations in unseen data.

2.1.6 Disadvantages of neural networks

ANNs have several drawbacks, including:

i. A suitable ANN structure is found by trial - and - error or experience since there are no criteria for defining the ideal network architecture.
ii. Hardware-dependent neural networks depend on processors with parallel processing abilities.
iii. This means that all issues must be transformed into numerical values before they are fed into the network.

iv. ANNs suffers from a major shortcoming in that their probing solutions are left unexplained. The network loses credibility when it is unable to explain why or how the solutions work.

2.2 Theoretical Framework

2.2.1 Computational Learning Theory

Mathematical frameworks for quantifying learning tasks and algorithms are known as computational learning theories or statistical learning theories (Akıs, 2018). To get effective results on a broad variety of issues, a machine learning practitioner does not need to master these subfields in detail. This sub-field does, however, give insight into the larger issue of learning from data by gaining a high-level grasp of some of the most prevalent methodologies (Vamplew, 2016). CoLT, or computational learning theory, is a branch of mathematics that examines how formal mathematical approaches might be used to improve learning systems. It aims to quantify learning issues via the application of theoretical computer science methods. This involves describing the degree of difficulty with which a certain skill must be learned (Symeonidis, 2020). An extension or sister of statistical learning theory (SLT), computational learning theory employs formal approaches to characterize learning algorithms. Theories in machine learning are primarily concerned with supervised learning, an inductive learning method (Büyüksaraç, 2015). Learning algorithms are fed data that has been tagged in a meaningful manner. Some examples of samples include descriptions of mushrooms, with labels labeling whether they are edible. These previously tagged samples are used by the algorithm to create a classifier (Tahir, 2021). Using this classifier, the program can now give labels to samples it has never seen before. Unsupervised learning algorithms aim at improving some performance metrics, such as the number of errors produced while training on fresh data (Muhammad, 2021).

2.3 Review of Related works

According to Muhammad (2021), an in-depth study of machine/deep learning methods and techniques for automated sign language identification was undertaken between 2014 and 2021 and found that existing systems need conceptual categorization to interpret all available data appropriately. Consequently, we focused on the commonalities of most sign language recognition techniques. A comprehensive framework for researchers is proposed in this study that addresses the benefits and drawbacks of each of them. These findings show the importance of input modalities in this sector; it seems that recognition based on several data sources, such as vision and sensor-based channels, is better than a unimodal analysis. The ability to interpret continuous sign language communication with a little delay has also been made possible by recent advancements in research, which have taken researchers beyond the basic detection of sign language letters and words.

To understand Bolivian Sign Language, Rodrguez (2021) tested the usage of two neural network techniques: multilayer (MLP) and convolutional (CNN). Using a motion-based technique, we choose the most important frames from a movie and apply a boundary detection algorithm to the chosen frames. We conducted an experiment in which 60 films of four basic BSL phrases were used to test these strategies. Consequently, we discovered that the accuracy of MLP runs from 65 to 88 percent, and that of CNN ranges from 95 to 99 percent, depending on the number of neurons and internal layers employed.

In this paper, Tahir (2021) gave a comparative analysis of all known methodologies and sensors utilized for sign language detection. It was the goal of this research to examine and critique current and developing trends and tactics for the identification of sign language. Using this study, other researchers may learn about all the materials and methods that have been employed for sign language up to now, such as flex resistive sensors, vision sensors, and hybrid systems.

There is difficulty in classifying characters in Indian Sign Language, according to Patil (2021). (ISL). People who are deaf or have a speech handicap can't communicate using sign language alone. For someone who has never studied this language, the gestures produced by persons with disabilities seem jumbled or chaotic. There must be a two-way exchange of information An Indian Sign Language recognition system is introduced in this research. Using a webcam, the user must be able to collect photos of hand motions, and the system must be able to anticipate and display their names. Several computer vision methods, such as gray-scale conversion, dilatation, and mask operation, are used to collect pictures during the processing stage. Our model is trained using a Convolutional Neural Network (CNN). About 95% of our model's predictions have been correct.

We used convolutional neural networks (CNNs) to learn from our dataset and to understand hand signals from input photographs. When we checked their collections, we found that 310 photos (ten sets with 31 separate signs) were included in the total. Using a camera and a computer vision-based technique, the suggested system captures video snapshots. Images are compared to a previously trained dataset and the Bengali numbers are shown. We were able to get a model estimation accuracy of 97.8 % using our dataset.

It was designed specifically for use on a workstation by Symeonidis (2020). An American Sign Language subset of static hand movements will be recognized by it (ASL). When entering data into previous systems, data gloves or markers have been employed. The feature vectors of a training set of motions will be compared with the feature vectors generated by a pattern recognition system. A Perceptron network will be used to create the final system.

Based on an investigation of different Sign Languages, Sulochana (2020) generated a data collection of static photos for 53 odd infant signals, which were then identified using a transfer learning approach based on Deep Learning by using MobileNets, which are convolutional neural networks. It has been possible to further refine this model, which has resulted in lower mistake rates and higher classification accuracy. On the prepared dataset, a classification accuracy of 85.8% was attained.

We see a need for a real-time sign language detection approach like that presented by Moryossef (2020) for videoconferencing. A linear classifier was used to demonstrate that the optical flow characteristics we extracted from the human posture estimate are relevant with an accuracy of 80% on the DGS Corpus. A recurrent model applied directly to the input allows us to enhance accuracy by up to 91% while keeping computation time to no more than 4 milliseconds. A demonstration application to illustrate the use of sign language recognition in browser-based video conferencing apps is described here.

A convolutional neural network-based American Sign Language (ASL) fingerspelling translator was presented by KAUSTUBH (2020). We make use of a pre-trained GoogLeNet framework. For the first time, we developed a robust model that properly classifies letters a through z and another that accurately distinguishes letters a through k. We are convinced that, despite the restrictions of the dataset and the good results we have obtained, we can build a generic ASL letter-to-letter translation.

Based on shape-based parameters including orientation, a center of mass (centroid), state of fingers, and thumb in terms of elevated or folded fingers of the hand and their corresponding locations in the picture, Shailesh (2020) provided a real-time system for hand gesture identification. This paper's technique relies only on the form elements of hand motion. There are no alternative methods of recognizing hand gestures like skin color or texture since these image-based traits are particularly susceptible to light conditions and other effects. A basic webcam, running at 20 frames per second with a resolution of 7 megapixels, has been used to create this method.

Using an image as input, Chavan (2020) demonstrated a Deep Learning-based method for the recognition of signs made in American Sign Language. The technology can forecast the user's 0 to 9-digit signals. It is possible to reduce the size of the Convolutional Neural Network's storage and training time by converting RGB data to grayscale pictures via image processing. Mobile apps and embedded single-board computers will benefit from the experiment's goal of finding a low-complexity combination of image processing and deep learning. The database has been trained from the ground up using deeper networks like Vgg16 and MobileNet v2, as well as smaller networks like LeNet-5 and AlexNet. We compare the recognition accuracies in this article. In the final design, only 10 layers plus a dropout layer were used, improving testing accuracy to 87.50% and training accuracy to 91.37%.

When Thakur (2020) trained a neural network to recognize gestures in sign language, it generated text and voice based on those movements. To facilitate two-way communication without the need for a translator, the approach presented in this study converts text into sign language.

Using datasets of hand gestures alphabets, numbers, words, and phrases in diverse real-time settings, Kruti (2019) conducted a comprehensive evaluation of the work done by researchers throughout the world. This table provides a thorough breakdown of the numerous methods for automating Indian sign language. Finding research holes in the current system and getting ideas for an Indian sign language interpreter may be accomplished via a review.

The two methodologies explored by Aks (2018) were feature-based recognition and recognition based on Convolutional Neural Networks (CNN). One method relies on segmentation, feature extraction, and classification, while the other uses segmentation and a convolutional neural network (CNN) to learn the signals in the picture. Tests are carried out utilizing the interface dataset of eight American Sign Language (ASL) signals to determine the systems' recognition rate. Each step of both techniques is thoroughly analyzed. A recognition rate of 95.31 percent was reached using the feature-based SLR system, while a recognition rate of 93.12% was achieved using the CNN-based SLR system. When data augmentation is utilized to extend the training dataset, the CNN-based SLR system attained an accuracy rate of 94.29 percent in experiments.

New methods for recognizing hand gestures during human-computer interaction have been reported by Deepali (2016). Frequently No additional gear, other than a camera, is required to use the hand gesture recognition technology. Users and computers communicate through Human-Computer Interaction (HCI). Our primary goal is to investigate neural network-based hand gesture recognition. For hand gesture recognition, an orientation histogram technique was employed, namely a subset of the ISL (Indian sign language). A picture is transformed into a feature vector, which is compared to a training set of movements, using a pattern recognition algorithm. With this final system in place, a perceptron network is used to implement it.

Both of Vamplew's (2016) streams are addressed in this paper. The first is the development of a viable method for classifying signs that may be used in sign language instruction or other applications that rely on hand gestures for input and output input and output. Using neural networks to create a real-time classification system that can also interpret temporal patterns is the focus of the second study. In the Deaf community, Auslan and other sign languages are the major means of expressing oneself. A communication barrier may emerge between Deaf and hearing persons due to the lack of awareness of these languages outside of these groups. As a result of the work done here, it may be possible to create systems that can help break down this barrier, either by giving computer tools to help people learn sign language or by developing portable systems that translate sign language to speech.

There are three basic phases in the suggested system presented by Büyüksaraç (2015). Fuzzy C-Means Clustering (FCM) and Thresholding are used to segment the face and hand first. FCM is an iterative clustering approach that uses fuzzy partitioning. The feature vectors are retrieved after the face and hands are segmented. When using low-resolution photos, segmentation mistakes are more likely to occur when using low-level features such as bounding ellipses, boxes, and center of mass coordinates. Each hand has a total of 23 characteristics. After extracting the feature vectors, discrete Hidden Markov Models are employed to recognize them (HMM).

Sharma (2014) created a system that can reliably translate Indian sign language numbers so that those who are less wealthy may interact with the outside world in public areas like train stations, banks, and so on without the need for an interpreter. Research presented here provides an automated method for recognizing Indian sign language numeric signals in the form of isolated photographs, which were acquired using merely a standard camera. To make use of the project in the real world, a database of 5000 numerical signs was first built, each with 500 photos. Extracting required characteristics from sign photos is done using direct pixel values and hierarchical centroid approaches. Neural network and kNN classification approaches were used to categorize the indications after extracting features from photos These trials have yielded results with an accuracy of up to 97.10 percent.

Hand motions used in Indian Sign Language (ISL) may be translated into text using a technique provided by Padmavath (2013). A camera is used to record the hand motions that correspond to the ISL English alphabet. Segmenting the hand in the collected frames, the neural network is utilized to identify the alphabet. Finger angle, number of completely open, fully closed, or semi-closed fingers, and finger identification are all inputs to the neural network. Single-handed alphabets have been tested, and the findings have been summarized.

Human hand gesture detection may be improved by using a variety of neural networks for both static photos and dynamic motions. This research focuses on Convolutional neural networks' potential to aid in the detection of hand gestures in Arabic Sign Language (ArSL). Feedforward and recurrent neural networks, together with their various designs (partially and completely recurrent networks), have been discussed in detail. The experiment results show that the proposed system with the fully recurrent design has a static gesture recognition accuracy of 95 percent.

2.5 Gap Analysis

In this study, a wide range of publications related to the topic was examined, from internet websites to research articles published in their numerous web archives. The quantitative results of this research were conflicting since some authors believed in a productive correlation between the topics studied whereas some believed in a negative one. Because of an essential discrepancy in the topic's viewpoints, there has been a significantly high rate of conflict in the research area; hence, this research will fill a gap/discrepancy in the field by upgrading the current system to produce quality results that can aid in reducing the rate of a discrepancy, along with a time gap, because of the duration in which the study is being carried out.

CHAPTER III

Methodology

3.1 Preamble

Data collection and analysis methodologies, participant/sample size (if applicable), CNN, as well as results analysis, are covered in this chapter.

3.2 Research Design

In experimental research, some independent variables were altered and given to other dependent variables to observe how they affect the former. This was done in the investigation. Independent and dependent variables are often tracked for some time to help researchers reach valid conclusions about the connection between the two categories of variables. Sign language identification and convolutional neural networks (CNN) were the factors of interest in the research.

3.3 Participants/Population and Sample

As this was an experimental study with a significant expense associated with it, the participants were confined to the researcher's training and sign-detecting method.

3.4 Data Collection Tools/Materials

Data from both primary and secondary sources were gathered for this investigation mainly from the KERAS MNIST (Modified National Institute of Standards and Technology) database. APA 7th reference style was employed to properly cite and reference the secondary data from a variety of sources including journals, articles, conference papers, etc. The primary data sources for this study were the python code initially written for it and the photos generated for training and evaluating the code, which made up most of the research's content. The study employed a diverse range of packaging and materials, including but not limited to:

Table 3.1. *Packages And Materials*

TENSORFLOW	NUMPY
OPENCV	python 3.8
IMAGE-PROCESSING	pycharm
DIGITAL-IMAGE	Camera
KERAS	

3.5 Data Analysis Procedures

A critical literature assessment and the construction of a prototype for sign language identification using convolutional neural networks (CNN) were used in conjunction with a mix of qualitative and experimental research goals. Using Python, a functional prototype was built to use machine learning techniques to identify sign languages, which enabled us to meet the qualitative goals.

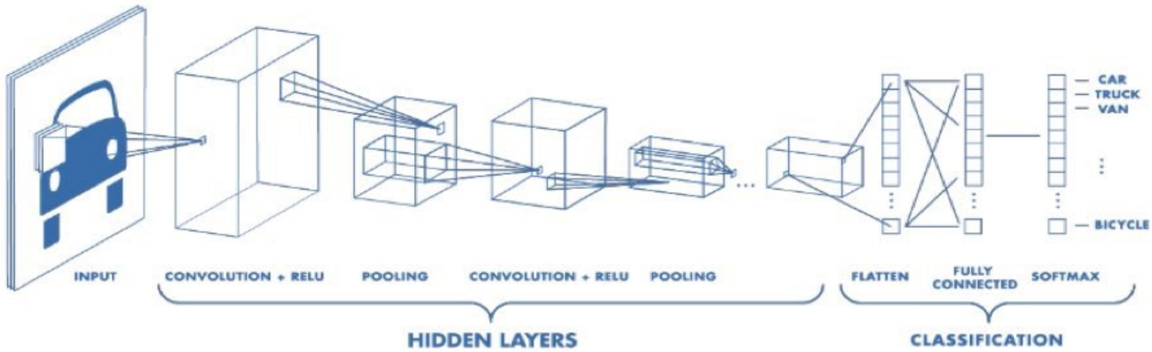
3.6 Convolutional Neural Networks (CNN)

One of the most effective and most implemented types of deep neural networks which have already excelled in several object detection algorithms is the convolutional neural network (CNN), which is gaining popularity in a diverse range of fields, which includes sign language detection. CNN comprises several building layers, including the convolutional layer, the pooling layer, and the fully connected layer. Among many fields in which artificial neural network has affected includes Computer Vision (VS), the main aim of this field is to allow computers/machines to see the world just like humans. Generally, they would be able to view the universe in a similar way humans do and even use that knowledge in terms of implementing tasks like image processing, object recognition, sign language detection, etc.

Deep learning has had a huge significance in recent years. The implementation of deep learning in computer vision tasks has turned out to be highly effective and is gradually becoming dominant with time, especially in convolutional neural network implementation. The main aim of the convolutional neural network is to minimize the images into easily preprocessed formats while keeping the important features for achieving high-level predictions.

CNN can be differentiated from other artificial neural networks mostly in terms of their superiority when it comes to high-performance rate in image, video, and speech/audio classification. They are composed of mainly three preprocessing layers which include the convolution, the pooling, and the fully connected layers.

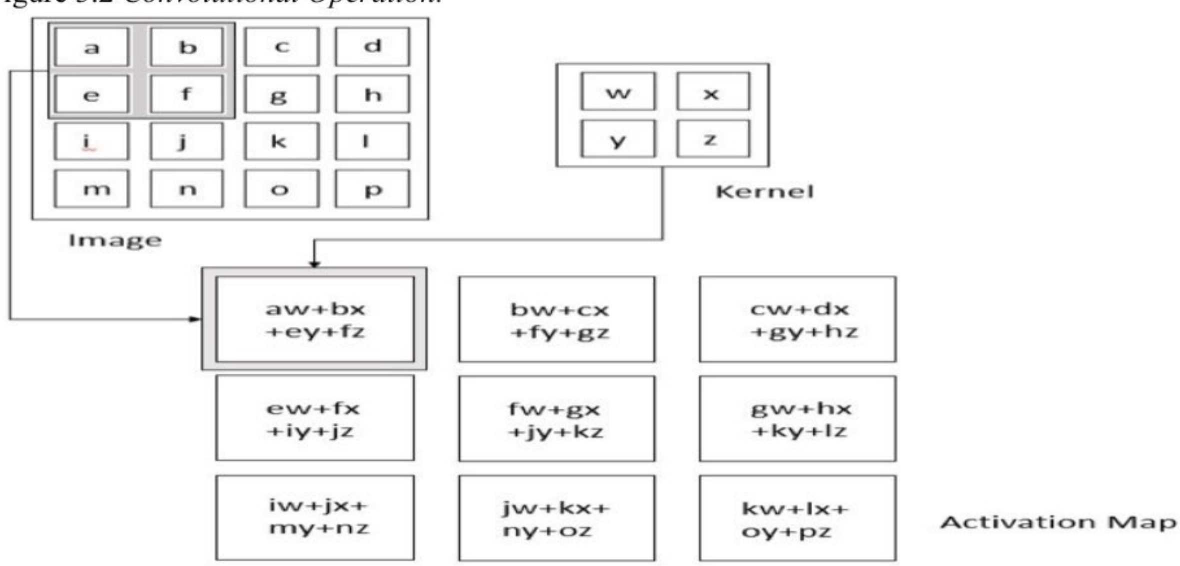
Figure 3.1 *Structure Of A CNN (Source: Mathworks)*



3.6.1 Convolutional Layer/Operation

The first building layer in the network is the convolutional layer. It is the main building layer where most of the computation and classifications happen. The input, data sets, filters, and feature maps are some components required in this first stage. The filter looks for corners or patterns. The image's top left corner is where the filter is positioned and multiplied by numbers with the same indexes. Prior to entering the result into the transformational output matrix, all results must be included. To repeat the process, the filter is moved toward the right position.

Figure 3.2 *Convolutional Operation.*

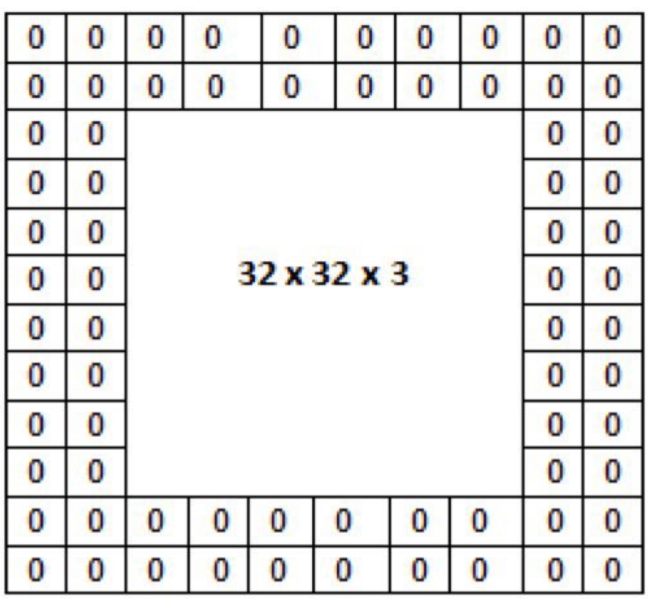


When it comes to computer vision, three key ideas which include scant interactions, parameter sharing, and equivariant representation are used by convolution. In the convolutional layer, for example, when there is an input image of X, the output constitutes a preprocessed fresh image of X, without altering the quality of the output image. There are a few filters, which are mainly the kernels, and their attributes should always be determined during training. The filter's size is often smaller than the size of the actual input image. When a filter is combined with an image, one activation map is created for each filter. The filter slides along each image height and width, as well as every spatial point, producing a dot product among every component of the filter and the input for each convolution.

3.6.2 Padding

In Convolutional operations, the size of the input image is decreased, which is where Padding comes to play. The feature map's size is less than that of the input image. The Stride has an impact on the feature map's size as well. The size becomes even smaller if the stride is increased to 2. When the CNN has many convolutional layers, the feature map's size becomes even smaller towards the end, making it impossible to perform more computations on the feature map. Applying Padding on the input image would help prevent this. Padding generally adds zero to every side of the input image which increases the size of the pixels. Thereby, making the size of the input image and feature map the same.

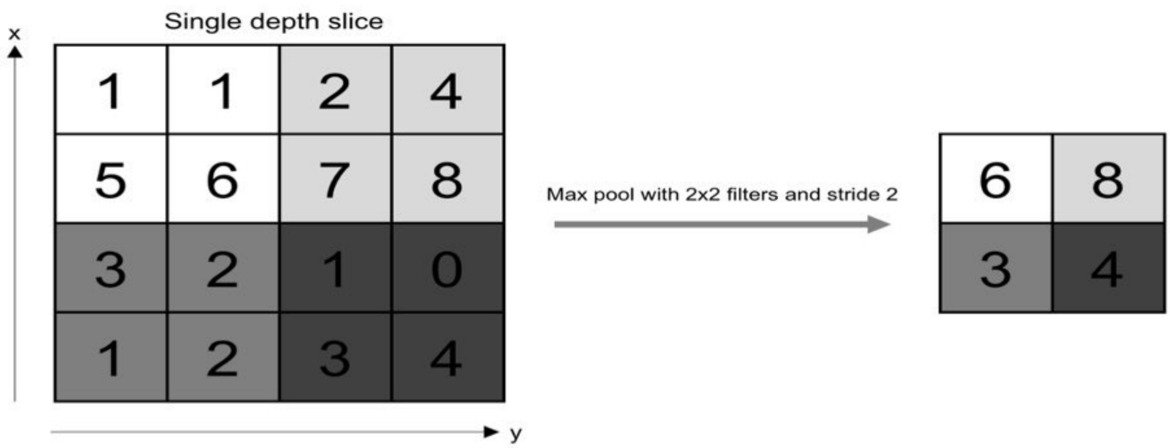
Figure 3.3 *Padding Operation.*



3.6.3 Pooling Layer

This layer makes use of statistical summarization of neighboring outputs and often takes the place of the network output. The Pooling operation is used to reduce and compute parameters. This layer also finds features that are resilient to adjustments in size or position, and it prevents overfitting. There are several procedures for pooling but most of the time, max pooling is used.

Figure 3.4 *Pooling Operation*

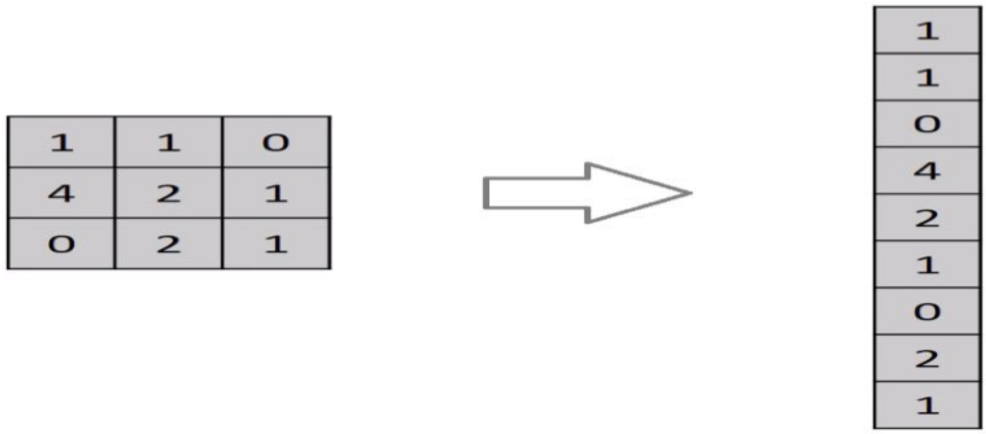


Because of the smaller spatial size of the representation, fewer computations and weights are required. The pooling technique treats every layer of the representation separately.

3.6.4 Flattening Layer

In essence, flattening involves converting the resulting matrix produced in the convolutional and pooling layers to a 1D array. It is crucial since Artificial neural networks use a one-dimensional array as their input.

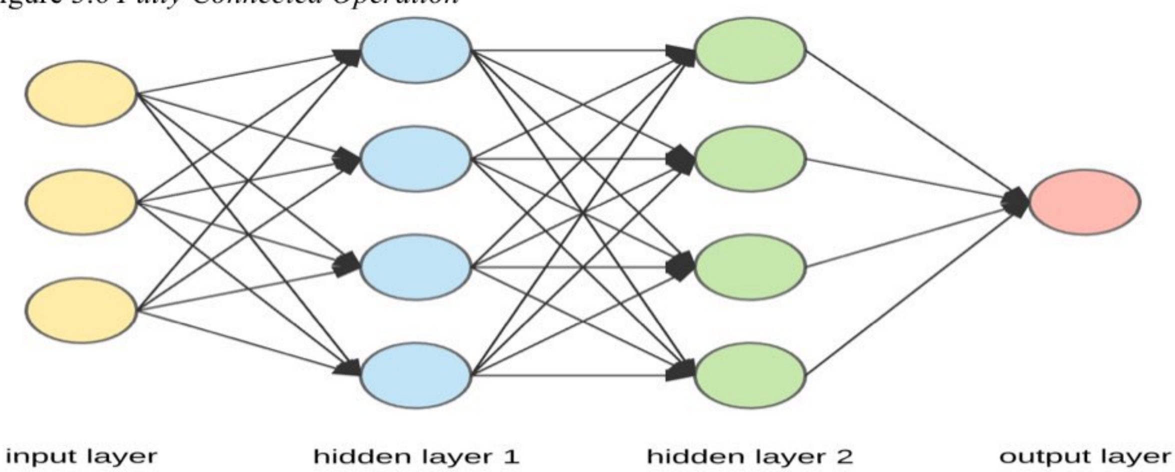
Figure 3.5 *Flattening Operation*



3.6.5 Fully Connected Layer

This layer's learning process uses the information from the one-dimensional array seen previously. The input is the layer that has been flattened before. There may be several fully connected layers. The top layer is responsible for collecting the classification tax. An activation function is always in use in every fully connected layer.

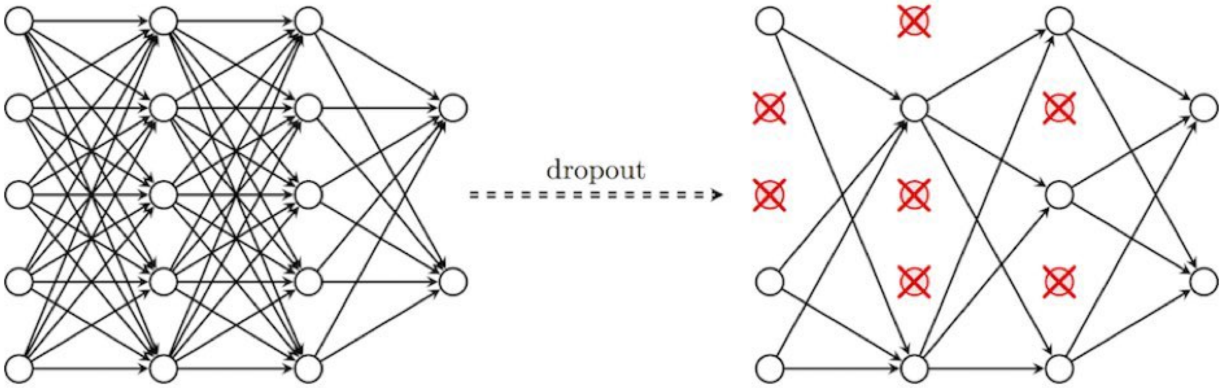
Figure 3.6 *Fully Connected Operation*



3.6.6 Dropout

A normalization method for minimizing overfitting is a dropout. It is referred to as a "Dropout" since it eliminates elements from a neural network that are visible or hidden.

Figure 3.7 *Dropout Operation*



CHAPTER IV

Data Presentation and Results

4.1 Preamble

In this section, the implementation process was talked about, testing, issues that were faced, and the results of the proposed system.

4.2 Implementation:

The language used in this system is python, first, the environment used, and the packages installed to get the project up and running were indicated.

Python 3.8 was used for this project and there are various environments to choose from but in this case, the "python 3.8 venv" pycharm was used as the integrated development environment (IDE) although the visual studio code (VS code) can be used.

4.3 Basic Hardware Requirements

i. A 3MP high-resolution camera
ii. RAM: 8GB or more required
iii. 4GB dedicated GPU Processor: at least an Intel Pentium 4 HDD: at least 10GB 15" or 17" color monitors are available. Scroll, optical, or touch-screen mice Standard-sized 110-key keyboard
iv. Software tools.
v. Import time
vi. from Keras.preprocessing import image
vii. import NumPy as np
viii. from Keras.models import load model
ix. import cv2

4.4 Running This Project

Install python3.8, Keras, TensorFlow, and opencv3. To install python or pycharm you go to their website, then once those are installed you open your IDE and go to the terminal to install the packages, you'll need to get the project running.

To install the packages needed for the project you type, "pip install (name of the package)" e.g.," Pip install Keras".

With the use of media pipe or OpenCV on python, you can use the camera functionality which will record the signs the users will be doing. TensorFlow is usually an open-source library and free for artificial intelligence and machine learning, in this case, it will oversee the machine-learning component of the project. Artificial neural networks have a Python interface thanks to the open-source library known as Keras. The project has five files namely, data.py, model.py, train.py, test.py, and main.py.

The project is run in a sequence to allow it to run the data, collect the models and train the data before running the main file.

4.5 Testing

As mentioned above the project is run in a sequence, which is the data.py file which runs the data models which are stored.

Then you go ahead to run the model.py which helps to establish a model though pre-trained models are in a folder in the project model.py helps to establish new models.

Then you must train your data before going ahead to test the data so the sequence will run. Below is a figure of the main.py which runs the main sequence and starts the project.

4.6 Results

As far as the project is run in the sequence mentioned above, a graphic user interface will be displayed on the screen, the webcam in use views the user and the GPU in the computer runs the machine learning process (which is where tensor flow comes in) and the results are displayed on the screen. Note that the project uses machine learning processes to determine landmarks in our hands and ascertain the signs we are using, the packages responsible for this are TensorFlow, Keras, and scikit image.

4.6.1 CNN Simulation

A fully linked network of the CNN structure was used in the first stage to execute ASL translation. The CNN structure used for American Sign Language translation was established using the concept of "keeping the feature space big and shallow in the beginning phases of the network, and then making it smaller and deeper towards the end." Hence, to represent more global, high-level, and representative information, smaller filters were used to gather as much local information as possible, and then I gradually increased the filter width to lower the resulting feature space width. Next, to identify low-level features that are merged to produce numerous complicated shapes (by increasing the number of channels) and aid in class distinction, the number of channels was initially intended to be low. To deepen the feature space and aid in the learning of more levels of global abstract structures, the number of filters is raised. For Max-Pooling parameters, a filter size of 2x2 or 3x3 with a stride = 1 was used. whereas, for the convolutional layer, a 3x3 or 5x5 filter with a stride of one was used. To reduce a large image to a reasonable size and then continue with the convention indicated, larger filter sizes and strides may be employed. The VGG-like model had the highest accuracy score on the validation set of all the evaluated models, so I chose the base architecture depicted below in Table 2. Two convolutional layers, max-pooling, stride, number of channels, padding, kernel/filter size, and fully linked layers make up the CNN's structure.

Table 4.1 *The Applied Structure Of CNN*

Layer(type)	Output Shape	Parameters
conv2d (Conv2D)	(None, 28, 28, 16)	160
max_pooling2d (MaxPooling2D)	(None, 14, 14, 16)	0
conv2d_1 (Conv2D)	(None, 14, 14, 32)	4640
dropout (Dropout)	(None, 14, 14, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 32)	0
dense (Dense)	(None, 7, 7, 128)	4224
flatten (Flatten)	(None, 6272)	0

dense_1 (Dense)	(None, 26)	163098
Total parameters: 172,122 Trainable parameters: 172,122 Non-trainable parameters: 0		

The MNIST data was separated into two portions for CNN training: 70,000 handwritten digit images make up the datasets being used: 60,000 were used for training, and 10,000 were used for testing. The digits were aligned in a stable-size image after being size-normalized (28 by 28 pixels). Images are grayscale, with a number between 0 and 255 representing each pixel (0 represents black, 255 represents white, and the remaining values represent different shades of gray). A common image categorization dataset used to examine different machine-learning methods is the MNIST database. The data that was designated for training is 60% and for validation 40%.

In the simulation process, the magnitude of the accessible training data and the system features used to create the Convolutional Neural Network model were considered. As a result, each input signal was scaled using Z-score normalization throughout the simulation, which increased the generalizability of the model. Additionally, the CNN model was trained using 50 epochs. Conv2d of CNN was used, and the kernel size was 3. As a result, a fully connected network was used for the classification of American Sign Language. As previously stated, 50 epochs were used to train CNN. In each running stage of each epoch, about 60% of the data set was used for training and the remaining 40% for validation. The CNN simulation results are shown in Table 4.2. 0.0405 is the amount of the loss function achieved during training. While the loss function for test data is 0.0291. The accuracy value for test data is 99 percent.

Table 4.2 *Proposed CNN's Simulated Result*

	Loss Function	Accuracy (%)
Training	0.0414	98.76
Testing	0.0300	99.00

Table 4.3 *Simulation results of previous systems using different methods and datasets*

Writer (year)	Methods	Database	Performances based on the accuracy (%)
Abdulwahab et al. (2019)	Deep learning and CNN	Static ASL	99.30
Rasha et al. (2020)	CNN	ASL alphabet	98.53
Atoany et al. (2021)	Siamese CNN	ASL alphabet	98.70
Adithya et al. (2012)	CNN	ASL	94.70
Qing et al. (2018)	2S-CNN	ASL alphabet	92.08
Rahib et al. (2020)	CNN	ASL alphabet	92.22
Seo et al. (2018)	Reflected wave form, CNN	ASL alphabet	90.00
Rashedul et al. (2019)	Hybrid Segmentation and CNN	ASL alphabet	97.28
Gongfa et al. (2019)	Skeletonization algorithm and CNN	ASL alphabet	96.01
Meltoh Yokpe and Kamil Dimililer (2022) (Proposed System)	CNN	ASL alphabet	99.00

Figure 4.1 *Hand landmarks that the system reads*

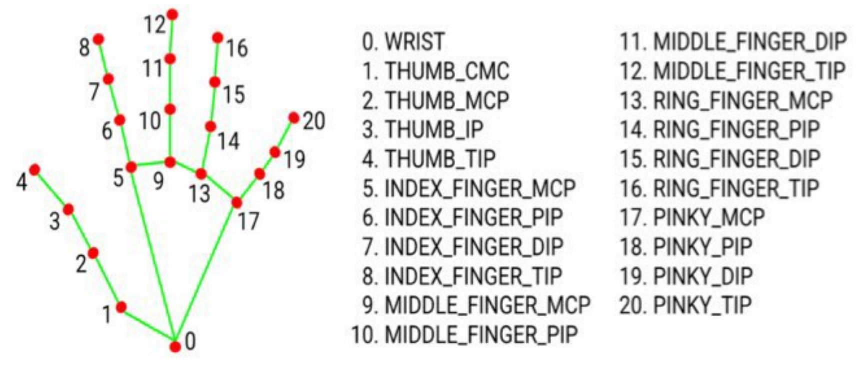


Fig 2. 21 hand landmarks.

Now that all this is out of the way, running the main project which should start the camera.

Figure 4.2 *Detection Module Returns "L"*

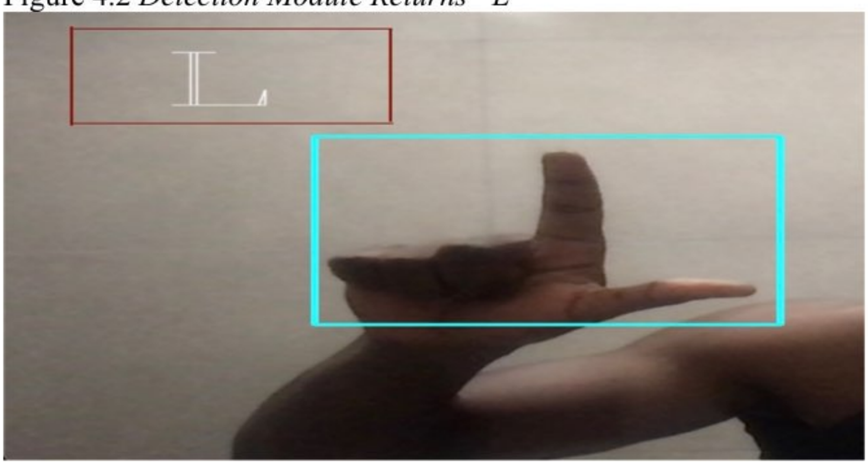


Figure 4.3 *Detection Module Returns "H"*

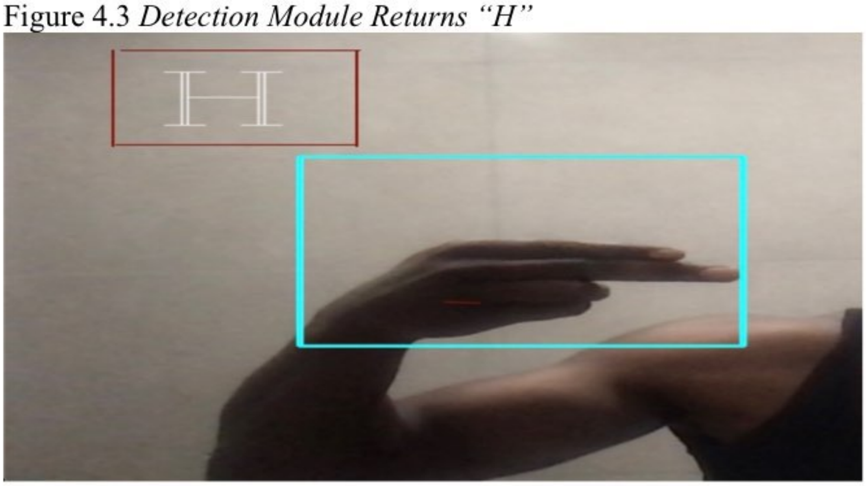


Figure 4.4 *Detection Module Returns "X"*

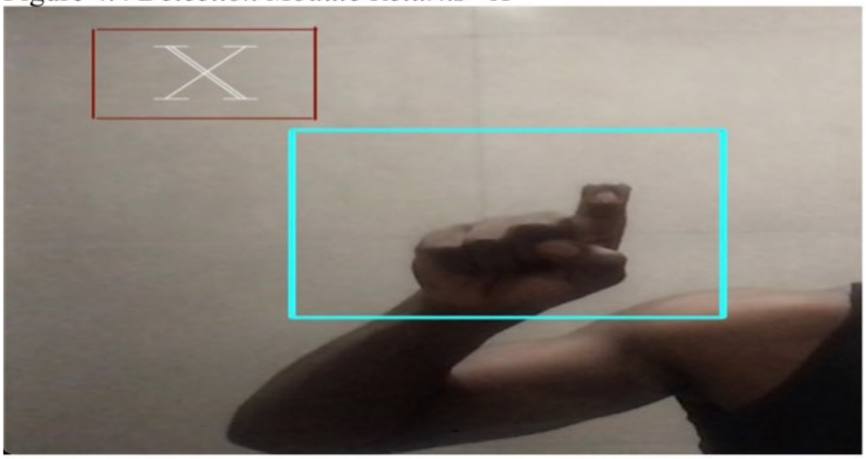
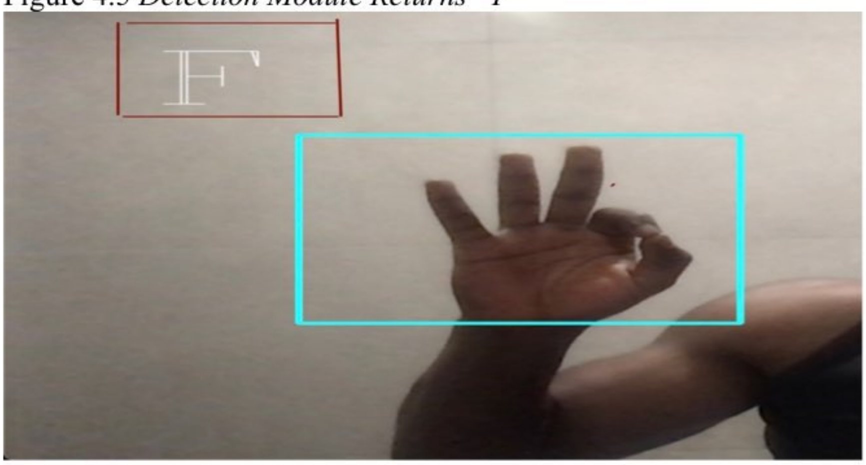


Figure 4.5 *Detection Module Returns "F"*



If you look at the terminal a detection module is running that reads the hand placements and tells if the detection is true or false, and in this case, I have been able to optimize the system to get at least 99% true detection as seen on the images above and the signs it reads are shown. And optimized to detect solely hands and ignore everything else in the background.

CHAPTER V

Discussion

Given the enormous amount of people who could benefit from sign language recognition systems, there's little doubt that the current trend will continue indefinitely. The implementation of deep learning models, which are still being developed and will only gain greater acceptance in the upcoming years, has spurred recent developments in this subject. SLR technologies have been created during the previous decade by collecting features from sensor information and streaming media and putting these into neural processors… In this study, deep learning based SLR initiatives that have been created over the previous few years have been categorized into clusters based on their key traits.

As this field of study has gained a lot of interest recently, there are a variety of possibilities available. Generally, CNN networks are used for this purpose since they have the finest attributes for this job. Some of the architectures that may be employed when data is gathered in a multimedia format are LSM, RNN, and GRU. Deep learning and machine learning are the most often utilized methods for SLR in literature. We chose to employ classical machine learning algorithms in our system because of their excellent accuracy and cheap processing cost when comparing them to deep learning techniques. Using typical machine learning, the first SLR system involves hand segmentation, feature extraction, and classification as its three primary steps.

A variety of networks were often integrated to provide the best possible results. A variety of data types, still images are included, detailed information, thermal imaging scans, skeletal information, and sequential data may be processed by these models. They have been tested and shown to work. Some of the suggested models were shown to be quite successful, even if they were used for restricted purposes. To ensure worldwide coverage, researchers from all over the globe conduct studies on a wide range of sign language variants. Before real-time, two-way translation systems can be used in the real world, there are still several obstacles that must be cleared, even though science is making major advancements in this field.

Prior to this, it is crucial to improve consistency and get rid of some of the most common misconceptions in the industry (where most algorithms interpret a sign incorrectly). The most direct path forward in terms of complex SLR applications may be provided by hybrid models that combine the best aspects of several different neural network types. In the upcoming years, the subject of sign language recognition is likely to advance quickly, and many research papers may contain crucial elements that will eventually be included in the solution. Even at this early level, many SLR techniques may be employed realistically and give instant help to impaired persons as well as suggest future research areas. Image processing, medicine, and other fields have benefited greatly from the advent of machine learning and deep learning in recent years.

Our current model takes less than a minute to recognize ASL from the input photos. Sign languages are a type of visual language that communicates via the use of the body, hand, and facial expressions. Especially abled persons rely on sign language to communicate with one another. They may use it to express and share their thoughts and emotions with other people. Because not everyone is fluent in sign language, communication is limited. Automated Sign Language Recognition systems can quickly convert sign English motions into frequently spoken language, thereby overcoming this barrier. The TensorFlow object detection API has been used in this study. The American Sign Language alphabet has been used to teach the system. Pictures were taken with a camera and processed with Python and OpenCV to save expenses on data collecting.

The designed method obtained an overall confidence level of over 99 percent. Notwithstanding the software's relatively high confidence rate. The algorithm's fast processing speed allows for results to be generated in real-time. The Convolutional neural network (CNN) design increases processing speed. It has the added benefit that even if the signals are damaged by up to 48% by noise, they can still be recovered. Processing of words and phrase gestures may be included in future updates. In this scenario, the structure's efficiency and speed are heavily influenced by the language and syntax.

CHAPTER VI

Conclusion And Implications for The Future

It still lacks a dictionary in numerous regional languages for American Sign Language (ASL), as well as a computerized translator and other educational and learning resources. In today's world, sign language interpretation is the most researched field. For researchers, the creation of an effective method for locating ROI (Region of Interest – hand) in dynamic environments should be a top priority. Automated identification of numerous right signs relies heavily on feature extraction. More and more researchers are using feature fusion to improve accuracy rates by combining two or more approaches. It's been proven in the research gap that there are still unfinished efforts on dynamic signs with varied backgrounds. A wide range of pictures kinds is now required as data sources for explanation and analysis in today's applications.

Various tasks need the extraction of a wide range of features. When a picture is digitized, scanned, transmitted, or stored, it loses quality as the format is changed. A procedure called image enhancement is needed to improve the visual appeal of the final output picture since it contains many different techniques for boosting an image's visual appeal. Humans can better understand and comprehend pictures that have been enhanced, and these enhanced images may then be used by other automated image processing systems. The picture is subsequently treated with several techniques of feature extraction to improve its computer-readable quality. Recognition software for sign languages is a useful tool for developing expert knowledge, identifying edges, and merging erroneous information from several sources.

Sign language recognition software Classification is the goal of a convolutional neural network (CNN). Especially abled persons rely on sign languages to communicate with one another. They may use it to communicate and share their thoughts and emotions with others. Because not everyone is fluent in sign language, communication is limited. Automated Sign Language Recognition systems can quickly convert sign English motions into frequently spoken language, thereby overcoming this barrier. The TensorFlow object detection API has been used in this study. The American Sign Language alphabet has been used to teach the system.

In real-time, Sign language is recognized by the system. To reduce the cost of data collection, photos were gotten from the ASL database and Python and OpenCV, respectively. An average confidence level of over 99 percent has been achieved by the designed method. The dataset the system was trained on is small and limited, despite the system having a high average rate of confidence. Gestural recognition might be improved by expanding the dataset. The TensorFlow model used may be replaced with another model. Changing the dataset allows the system to be implemented in multiple sign languages.

6.2 Recommendations

The study provides the following recommendations:

i. An improved system's performance can be achieved by incorporating advanced deep learning algorithms into our system's hand detection and segmentation.

ii. With recent breakthroughs in Natural Language Processing, many tailored estimation techniques for auto-completion of words and phrases are now being developed (NLP). Our recognition system will no longer need the user to do each letter individually to construct a phrase after we have included such a technology.

iii. Reinforcement learning is a relatively new approach to deep learning that relies on extracting conclusions from prior failures to guide its present actions. It uses the Markov Decision Process (MDP) to learn. Our feedback mechanism can help a model like this learn from its errors.

iv. The generalization performance offered by single neural networks may be improved by committees of neural networks. They produced generalizations that were better than the mean of their component networks, as was shown.

v. The outputs of a neural network could well be subjected to a post-processing threshold to reduce the number of inaccurate classifications generated by the network. The network's classification accuracy may be improved by labeling certain input patterns as "unclassified" based on this thresholding process.

vi. To solve situations when one or more input values are missing, neural networks may be used It is recommended to use the 'network estimate substitution' approach, which includes training networks to guess missing input values based on known ones.

vii. Compass bearings and the time of day are two examples of data values that should be represented using trigonometric or sawtooth functions rather than a single linear input value. Both the accuracy of the network and the amount of time it takes to train may benefit from these representations.

viii. The arctangent function may be used to decode network outputs corresponding to cyclical data values.

ix. Intelligent initialization of the network's weights may shorten training time for recurrent networks. Neural Transplant Surgery, a novel initialization approach, has been developed and tested on two basic temporal classification tasks with encouraging results. Future studies should focus on the development of more difficult challenges.

REFERENCES

Atwood, J., Eicholtz, M., & Farrell, J. (2012). American sign language recognition system. *Artificial Intelligence and Machine Learning for Engineering Design. Dept. of Mechanical Engineering, Carnegie Mellon University*.

Akış, A. H. (2018). *A low-cost learning-based sign language recognition system* (Master's thesis, Middle East Technical University).

Anetha, K., & Rejina Parvin, J. (2014). Hand talk-a sign language recognition based on accelerometer and SEMG data. *International Journal of Innovative Research in Computer and Communication Engineering*, *2*(3), 206-215.

Al-Qurishi, M., Khalid, T., & Souissi, R. (2021). Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. *IEEE Access*.

Ba Wazir, A. S., Karim, H. A., Abdullah, M. H. L., AlDahoul, N., Mansor, S., Fauzi, M. F. A., ... & Naim, A. S. (2021). Design and implementation of fast spoken foul language recognition with different end-to-end deep neural network architectures. *Sensors*, *21*(3), 710.

Bheda, V., & Radpour, D. (2017). Using deep convolutional networks for gesture recognition in american sign language. *arXiv preprint arXiv:1710.06836*.

Büyüksaraç, B. (2015). *Sign language recognition by image analysis* (Master's thesis, Middle East Technical University).

Chavan, S., Yu, X., & Saniie, J. (2021, May). Convolutional Neural Network Hand Gesture Recognition for American Sign Language. In *2021 IEEE International Conference on Electro Information Technology (EIT)* (pp. 188-192). IEEE.

Dong, C., Leu, M. C., & Yin, Z. (2015). American sign language alphabet recognition using microsoft kinect. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 44-52). Doi:10.1109/CVPRW.2015.7301347

Dangarwala, K. J., & Hiran, D. (2020). Alphabet Classification of Indian Sign Language with Deep Learning. In *International Conference on Innovative Data Communication Technologies and Application* (pp. 569-576). Springer, Cham.

Garcia, B., & Viesca, S. A. (2016). Real-time American sign language recognition with convolutional neural networks. *Convolutional Neural Networks for Visual Recognition*, *2*, 225-232.

Georgiev, M., Eberz, S., & Martinovic, I. (2022). Techniques for Continuous Touch-Based Authentication Modeling. *arXiv preprint arXiv:2207.12140*.

Hasanbegović, H., Mahmutović, E. H., & Beha, A. (2013). Forms of hand in sign language in Bosnia and Herzegovina. *HUMAN*, 25.

He, S. (2019, October). Research of a sign language translation system based on deep learning. In *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)* (pp. 392-396). IEEE.

Jalal, M. A., Chen, R., Moore, R. K., & Mihaylova, L. (2018, July). American sign language posture understanding with deep neural networks. In *2018 21st International Conference on Information Fusion (FUSION)* (pp. 573-579). IEEE.

Kaustubh, J (2020). Sign Language Recognition Using Neural Network. *International Journal of Artificial Intelligence & Applications, 2(2), (22-32).*

Kania, K., & Markowska-Kaczmar, U. (2018, June). American sign language fingerspelling recognition using wide residual networks. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 97-107). Springer, Cham. doi:10.1007/978-3-319-91253-0_10

Kumar, S., Srivastava, R., Pathak, S., & Kumar, B. (2020). Cloud-based computer-assisted diagnostic solutions for e-health. In *Intelligent Data Security Solutions for e-Health Applications* (pp. 219-235). Academic Press.

Lahamy, H., & Lichti, D. D. (2012). Towards real-time and rotation-invariant American Sign Language alphabet recognition using a range camera. *Sensors*, *12*(11), 14416-14441.

Miehle, J., Minker, W., & Ultes, S. (2022). When to Say What and How: Adapting the Elaborateness and Indirectness of Spoken Dialogue Systems. *Dialogue & Discourse*, *13*(1), 1-40.

Maraqa, M., & Abu-Zaiter, R. (2008, August). Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. In *2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)* (pp. 478-481). IEEE.

Moryossef, A., Tsochantaridis, I., Aharoni, R., Ebling, S., & Narayanan, S. (2020, August). Real-time sign language detection using human pose estimation. In *European Conference on Computer Vision* (pp. 237-248). Springer, Cham.

Neiva, D. H., & Zanchettin, C. (2018). Gesture recognition: A review focusing on sign language in a mobile context. *Expert Systems with Applications*, *103*, 159-183.

Nadgeri, S., & Kumar, D. (2020, June). An Analytical Study of Signs Used in Baby Sign Language Using Mobilenet Framework. In *Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT)*.

Padmavathi, S., Saipreethy, M. S., & Valliammai, V. (2013). Indian sign language character recognition using neural networks. *IJCA Special Issue on Recent Trends in Pattern Recognition and Image Analysis RTPRIA*, *1*, 40-45.

Patil, R (2021). Indian Sign Language Recognition using Convolutional Neural Network. *ITM Web of Conferences, 2, (10-21).*

Rajam, P. S., & Balakrishnan, G. (2012). Recognition of Tamil sign language alphabet using image processing to aid deaf-dumb people. *Procedia Engineering*, *30*, 861-868.

Rodríguez Villarroel, J. P., Ponce de León Espinoza, N., & Arteaga Sabja, W. (2021). Multilayer and convolutional neural networks for Bolivian Sign Language recognition: an empirical evaluation. *Acta Nova*, *10*(1), 22-41.

Rastgoo, R., Kiani, K., & Escalera, S. (2018). Multi-modal deep hand sign language recognition in still images using restricted Boltzmann machine. *Entropy*, *20*(11), 809.

Sahoo, A. K., Mishra, G. S., & Ravulakollu, K. K. (2014). Sign language recognition: State of the art. *ARPN Journal of Engineering and Applied Sciences*, *9*(2), 116-134.

Shamrat, F. J. M., Chakraborty, S., Billah, M. M., Kabir, M., Shadin, N. S., & Sanjana, S. (2021). Bangla numerical sign language recognition using convolutional neural networks. *Indonesian Journal of Electrical Engineering and Computer Science*, *23*(1), 405-413.

Sharma, M., Pal, R., & Sahoo, A. K. (2014). Indian sign language recognition using neural networks and KNN classifiers. *ARPN Journal of Engineering and Applied Sciences*, *9*(8), 1255-1259.

Thakur, A., Budhathoki, P., Upreti, S., Shrestha, S., & Shakya, S. (2020). Real time sign language recognition and speech generation. *Journal of Innovative Image Processing*, *2*(2), 65-76.

Tripathi, K., & Nandi, N. B. G. (2015). Continuous Indian sign language gesture recognition and sentence formation. *Procedia Computer Science*, *54*, 523-531.

Vaitkevičius, A., Taroza, M., Blažauskas, T., Damaševičius, R., Maskeliūnas, R., & Woźniak, M. (2019). Recognition of American sign language gestures in a virtual reality using leap motion. *Applied Sciences*, *9*(3), 445.

Vamplew, P (2016). Recognition of Sign Language Using Neural Networks, *Flinders University of South Australia.*

Wiryana, F., Kusuma, G. P., & Zahra, A. (2018, September). Feature extraction methods in sign language recognition system: a literature review. In *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)* (pp. 11-15). IEEE.

Zell, S (2013). Simulation neuronaler Netze. München. Finding structure in time. *Connectionist psychology, (289–312).*

APPENDICES

Appendix A

User Manual

Sign language detection using CNN is software developed by the author. It aims to allow an easy way to communicate between the hard of hearing and the general public by translating hand gestures into text format.

The language used in this system is python, firstly, indicates the environment used and the packages installed to get the project up and running.

Python 3.8 was used for this project and there are various environments to choose from but in this case, the "python 3.8 venv" pycharm was used as the integrated development environment (IDE) though you can use visual studio code (VS code).

- **Hardware Requirements**

1. The Hardware Interfaces Required are: Camera: Good quality,3MP
2. Ram: Minimum 8GB or higher
3. GPU: 4GB dedicated Processor: Intel Pentium 4 or higher HDD: 10GB or higher Monitor: 15" or 17" color monitor Mouse: Scroll or Optical Mouse or Touch Pad Keyboard: Standard 110 keys keyboard
4. Software tools.
5. HandExtractor.py
6. import cv2
7. import NumPy as np
8. from Keras.models import load model
9. from Keras.preprocessing import image
10. import time

- **Running This Project**

Install python3.8, Keras, TensorFlow, and opencv3. To install python or pycharm you go to their website, then once those are installed you open up your IDE and go to the terminal to install the packages, you'll need to get the project running.

To install the packages needed for the project you type, "pip install (name of the package)" e.g.," Pip install Keras".

With the use of media pipe or OpenCV on python you can use the camera functionality which will record the signs the users will be doing, TensorFlow is a free and open-source library for machine learning and artificial intelligence and in this case, it will be handling the machine learning going in the project, Keras is an open-source library that provides a python interface for artificial neural networks. The project has five files namely, data.py, model.py, train.py, test.py, and main.py.

The project is run in a sequence to allow it to run the data, collect the models and train the data before running the main file.

- **Testing**

I mentioned above that the project is run in a sequence, which is the data.py file which runs the data models which are stored, below are the results of running the data.py file.

Then you go ahead to run the model.py which helps to establish a model though pre-trained models are in a folder in the project model.py helps to establish new models.

Then you have to train your data before going ahead to test the data so the sequence will run. Below is a figure of the main.py which runs the main sequence and starts the project.

Appendix B

Source Code

- **Data.py**

```
import numpy as np
from keras.datasets import mnist
(X_train, Y_train) , (X_test , Y_test) = mnist.load_data()
X_train = np.array(X_train[:,:])
X_train = np.array([np.reshape(i, (28,28)) for i in X_train])
X_test = np.array(X_test[:,:])
X_test = np.array([np.reshape(i, (28,28)) for i in X_test])
num_classes = 26
y_train = np.array(Y_train).reshape(-1)
y_test = np.array(Y_test).reshape(-1)
y_train = np.eye(num_classes)[y_train]
y_test = np.eye(num_classes)[y_test]
X_train = X_train.reshape((60000, 28, 28, 1))
X_test = X_test.reshape((10000, 28, 28, 1))

def prediction(pred):
   return(chr(pred+ 65))

def keras_predict(model, image):
   data = np.asarray( image, dtype="int32" )

   pred_probab = model.predict(data)[0]
   pred_class = list(pred_probab).index(max(pred_probab))
   return max(pred_probab), pred_class

def keras_process_image(img):
```

```
    image_x = 28
    image_y = 28
    img = cv2.resize(img, (1,28,28), interpolation = cv2.INTER_AREA)

    return img

def crop_image(image, x, y, width, height):
    return image[y:y + height, x:x + width]

def main():
    l = []

    while True:

        cam_capture = cv2.VideoCapture(0)
        _, image_frame = cam_capture.read()
    # Select ROI
        im2 = crop_image(image_frame, 200,200,300,300)
        image_grayscale = cv2.cvtColor(im2, cv2.COLOR_BGR2GRAY)
        image_grayscale_blurred = cv2.GaussianBlur(image_grayscale, (15,15), 0)
        im3 = cv2.resize(image_grayscale_blurred, (28,28), interpolation =
cv2.INTER_AREA)

        im4 = np.resize(im3, (28, 28, 1))
        im5 = np.expand_dims(im4, axis=0)

        pred_probab, pred_class = keras_predict(model, im5)

        curr = prediction(pred_class)
```

```
        cv2.putText(image_frame, curr, (100, 150), cv2.FONT_HERSHEY_COMPLEX,
4.0, (255, 255, 255), lineType=cv2.LINE_AA)
    # Display cropped image
        cv2.rectangle(image_frame, (200, 200), (500, 500), (255, 255, 00), 3)
        cv2.imshow("frame",image_frame)

    #cv2.imshow("Image4",resized_img)
        cv2.imshow("Image3",image_grayscale_blurred)

        if cv2.waitKey(25) & 0xFF == ord('q'):
                cv2.destroyAllWindows()
                break
if __name__ == '__main__':
    main()

cam_capture.release()
cv2.destroyAllWindows()
```

- **Model.py**

```
import keras
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from keras.models import Sequential
from keras.layers import Conv2D,MaxPooling2D, Dense,Flatten, Dropout
from keras.datasets import mnist
import matplotlib.pyplot as plt
from keras.utils import np_utils
classifier = keras.models.Sequential()
```

```
classifier.add(Conv2D(filters=16,
kernel_size=(3,3),strides=(1,1),padding='same',input_shape=(28,28,1),activation='relu',
data_format='channels_last'))
classifier.add(keras.layers.MaxPooling2D(pool_size=(2,2)))
classifier.add(Conv2D(filters=32,
kernel_size=(3,3),strides=(1,1),padding='same',activation='relu'))
classifier.add(keras.layers.Dropout(0.6))
classifier.add(keras.layers.MaxPooling2D(pool_size=(2,2)))
classifier.add(keras.layers.Dense(128, activation='sigmoid'))
classifier.add(keras.layers.Flatten())
classifier.add(keras.layers.Dense(26, activation='softmax'))
```

- **Test.py**

```
import model
import data
model.classifier.compile(optimizer='SGD',loss='categorical_crossentropy',
metrics=['accuracy'])
accuracy = model.classifier.evaluate(x=data.X_test,y=data.y_test,batch_size=32)
print("Accuracy: ",accuracy[1])
```

- **Train.py**

```
import model
import data
model.classifier.compile(optimizer='SGD',loss='categorical_crossentropy',
metrics=['accuracy'])
model.classifier.fit(data.X_train, data.y_train, epochs=50, batch_size=100)
```

- **Main.py**

```
import cv2
import numpy as np
from keras.models import load_model
```

```
from skimage.transform import resize, pyramid_reduce
import PIL
from PIL import Image
model = load_model('SignModel.h5')
```

Appendix C

Turnitin Similarity Report

Bryan Yokpe Thesis After Jury

INBOX | NOW VIEWING: NEW PAPERS ▾

Submit File

Online Grading Report | Edit assignment settings | Email non-submitters

AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
Bryan Yokpe Mattoh	ABS 29112022	0%	--	--	▢	1966109764	29-Nov-2022
Bryan Yokpe Mattoh	ALLTH 29112022	1%	--	--	▢	1966109593	29-Nov-2022
Bryan Yokpe Mattoh	CH1 29112022	0%	--	--	▢	1966109563	29-Nov-2022
Bryan Yokpe Mattoh	CH2 29112022	3%	--	--	▢	1966109833	29-Nov-2022
Bryan Yokpe	CH3 12122022	3%	--	--	▢	1973006466	12-Dec-2022
Bryan Yokpe Mattoh	CH4 29112022	5%	--	--	▢	1966109670	29-Nov-2022
Bryan Yokpe Mattoh	CH5 29112022	0%	--	--	▢	1966109629	29-Nov-2022
Bryan Yokpe Mattoh	CNC 29112022	0%	--	--	▢	1966109712	29-Nov-2022

Assoc. Prof. Dr.
Kâmil Dimililer