REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

CLASSIFICATION OF HAND SIGN LANGUAGE USING DEEP LEARNING ALGORITHM

Israa Adil Mohammed ALYSADEN

Master's Thesis

Supervisor

Prof. Dr. Osman Nuri UÇAN

İstanbul, 2024

CLASSIFICATION OF HAND SIGN LANGUAGE USING DEEP LEARNING ALGORITHM

Israa Adil Mohammed ALYSADEN

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2024

This thesis title CLASSIFICATION OF HAND SIGN LANGUAGE USING DEEP LEARNING ALGORITHM" prepared by ISRAA ADIL MOHAMMED ALYSADEN and submitted on 04/06/2024 has been **accepted unanimously** for the degree of Master of Science in Information Technology.

Prof. Dr. Osman Nuri UÇAN

The Supervisor

Thesis Defense Committee Members:

Prof. Dr. Osman Nuri UÇAN	Department of Electrical and Electronic Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, İstanbul Arel University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's Thesis.

I hereby state that all data and material offered in this capstone project were acquired in complete line with ethical standards and scholastic regulations. I further affirm that, in accordance with the aforementioned standards of academic integrity, all borrowed ideas, arguments, its findings, as well as all the sources indicated in the Reference List, have been correctly cited in the text.

Israa Adil ALYSADEN

Singture

DEDICATION

I want to thank my husband for his efforts along my studying journey. Furthermore, special thanks to my supervisor for his guidelines.

ABSTRACT

CLASSIFICATION OF HAND SIGN LANGUAGE USING DEEP LEARNING

ALYSADEN, Israa Adil Mohammed

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Prof. Dr. Osman Nuri UÇAN

Date: June / 2024

Pages: 60

People with hearing disabilities face many problems, which impede many of their social life issues in all areas of communication, so effective communication is crucial in the development of a nation. It promotes understanding and inclusivity among all members of the community, including those who are deaf. Good communication is key to building and maintaining a strong, cohesive society. I used 29,000 sign language images, each class contained 1,000 images. I built a model from scratch (ASL.model) and compared it with pre-existing models (Xception, Inception, ResNet, VGG16 and MobileNet). The study of intelligent computers that can carry out activities without direct human guidance is known as artificial intelligence (AI), a fast-growing topic within computer science. These tasks may include learning, decision making, and problem solving, and they are often accomplished through the use of algorithms, data, and machine learning techniques. To find the most appropriate classification features to be used for classification, deep learning techniques will be employed in this thesis to create a model for classifying sign language utilizing photographs obtained from the Kaggle depository as a training data set. Deep learning is now widely employed across a variety of industries due to its accuracy and efficiency, particularly for vast yet complicated data, such as photos, sounds, or text, where deep learning algorithms are taught using massive, labeled data sets. In this thesis, we used deep learning to classify a total of twenty-nine sign language-related classes. We put forth a fresh

framework for categorizing sign language. The suggested model was put into practice, trained, verified, and tested. The model passed the test with a 99.97% success rate. To assess the effectiveness of our suggested model (ASL.model) with that of these methods, we also employed five pre-trained models (Xception, Inception, ResNet, VGG16, and MobileNet) of assisting the performance of deep learning algorithms that use Convolutional Neural Networks (CNNs). The five pre-trained models had F1-score levels of 100%, 99.51%, 99.87%, 100%, and 99.68%, respectively. The model Xception and VGG16 outperformed all others in terms of testing accuracy but when the testing time was smaller than 1.45 seconds, the suggested model outperformed all others in terms of time.

Keywords: Artificial Intelligence (AI), Hand Sign Language, Machine Learning, Deep Learning, Data Preprocessing.

ÖZET

DEEP KULLANILARAK EL İŞARET DİLİNİN SINIFLANDIRILMASI ÖĞRENME

ALYSADEN, Israa Adil Mohammed

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi,

Danışman: Prof. Dr. Osman Nuri UÇAN

Tarih: Haziran / 2024

Sayfa: 60

İşitme engelli bireyler, iletişimin her alanında sosyal yaşamdaki pek çok sorunu sekteye uğratan pek çok sorunla karşı karşıyadır; bu nedenle etkili iletişim, bir ulusun gelişmesinde büyük önem taşımaktadır. Sağır olanlar da dahil olmak üzere toplumun tüm üyeleri arasında anlayış ve kapsayıcılığı teşvik eder. İyi iletişim, güçlü ve uyumlu bir toplum inşa etmenin ve sürdürmenin anahtarıdır. 29.000 işaret dili görseli kullandım, her sınıfta 1.000 görsel vardı. Sıfırdan bir model (ASL.model) oluşturdum ve bunu önceden var olan modellerle (Xception, Inception, ResNet, VGG16 ve MobileNet) karşılaştırdım. Doğrudan insan rehberliği olmadan faaliyetleri gerçekleştirebilen akıllı bilgisayarların incelenmesi, bilgisayar biliminde hızla büyüyen bir konu olan yapay zeka (AI) olarak biliniyor. Bu görevler öğrenmeyi, karar vermeyi ve problem çözmeyi içerebilir ve genellikle algoritmalar, veriler ve makine öğrenimi teknikleri kullanılarak gerçekleştirilir. Sınıflandırma için kullanılacak en uygun sınıflandırma özelliklerini bulmak amacıyla, Kaggle deposundan elde edilen fotoğrafları eğitim veri seti olarak kullanarak işaret dilini sınıflandırmaya yönelik bir model oluşturmak amacıyla bu tezde derin öğrenme teknikleri kullanılacaktır. Derin öğrenme, doğruluğu ve verimliliği nedeniyle artık çeşitli endüstrilerde yaygın olarak kullanılıyor; özellikle derin öğrenme algoritmalarının çok büyük, etiketli veri kümeleri kullanılarak öğretildiği fotoğraflar, sesler veya metinler gibi çok büyük ancak karmaşık veriler için. Bu tezde, işaret diliyle ilgili toplam yirmi dokuz dersi sınıflandırmak için derin öğrenmeyi kullandık. İşaret dilini kategorize etmek için yeni bir çerçeve ortaya koyduk.

Önerilen model uygulamaya konuldu, eğitildi, doğrulandı ve test edildi. Model testi %99,97 başarı oranıyla geçti. Önerilen modelimizin (ASL.model) bu yöntemlerle etkinliğini değerlendirmek için ayrıca Evrişimli algoritmaları kullanan derin öğrenme algoritmalarının performansına yardımcı olacak önceden eğitilmiş beş model (Xception, Inception, ResNet, VGG16 ve MobileNet) kullandık. Sinir Ağları (CNN'ler). Önceden eğitilmiş beş modelin F1 puanı seviyeleri sırasıyla %100, %99,51, %99,87, %100 ve %99,68'di. Xception ve VGG16 modeli, test doğruluğu açısından diğerlerinden daha iyi performans gösterdi ancak test süresi 1,45 saniyeden küçük olduğunda önerilen model, zaman açısından diğer tüm modellerden daha iyi performans gösterdi.

Anahtar Kelimeler: Yapay Zeka (AI), El İşaret Dili, Makine Öğrenmesi, Derin Öğrenme, Veri Ön İşleme.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
ASL	:	American Sign Language
CNN	:	Convolutional Neural Network
DL	:	Deep Learning
LSTM	:	Long Short-Term Memory
ML	:	Machine Learning
RNN	:	Recurrent Neural Networks
SVM	:	Support Vector Machine
Tcc	:	Training Accuracy
TL	:	Training Loss
TeT	:	Testing Time
TeAcc	:	Testing Accuracy
TeL	:	Testing Loss
ToT	:	Total Time
TrT	:	Training Time
VAcc	:	Validating Accuracy
VL	:	Validating Loss

INTRODUCTION

1.1 BACKGROUND

Deaf individuals are those who are unable to hear or have significant difficulty hearing. There are many different ways in which deaf people communicate, including sign language, written language, and oral language. Signe language consider is one of the important topics that serve the disabilities people to understand them by the other peoples. This language depends on gestures, facial expressions, body language activities to communicate with others [1]. Every language has its rules, instructions, terminology and grammar so there are many unlimited sign languages, and each language has unique expressions and grammar. Most deaf use this language for communication purposes. So, sign language is also important for the people that work with deaf and hard of hearing like teachers and translators [2].

Sign language is considering the culture for the deaf and it is very important for everyone to know this culture to help deaf to do their needs and understand what they try to do. As well as it is important in education field because those kinds of people need to get their education and find their jobs like the others [3].

Due to modern technology and by using Artificial Intelligence, machine learning and deep learning algorithms we can try to facilitate and understand deaf people and help them.

In last recent years the peoples around the world start learning and understanding deaf language by using American standard sign language to help the deaf and communicate with them. So, this language is become popular to use [1].

1.2 PROBLEM STATEMENT

In this section we will focus the light on the most important problems sign language faced and the deaf needs for communication with the world. However, deaf people face hearing impairments that can significantly impact their lives. They may have difficulty communicating effectively with others, and their safety can be at risk if they are not understood or misunderstood by society. While some cases of hearing impairment can be treated with medicine, severe or profound deafness often cannot be fully treated. This can lead to feelings of hopelessness and difficulty in interacting with society without facing problems. Sign language can help fill the gap between deaf individuals and those who do

not understand sign language. However, many hearing people don't know SL and learning it can be difficult. As a result, there is often a noticeable divide between the deaf community and the hearing mainstream. This can create barriers to communication and understanding, leading to isolation and exclusion for deaf individuals [4]. A Sign Language is trying to identify the gap of communications between deaf people among themselves by using signs language as normal people to ease understanding the language among themselves. Extensive effort has been completed on ASL identification. The goal of sign language identification is to create a tool that can accurately translate sign language into written or spoken language, making communication between deaf (and mute) individuals and the hearing community more efficient. By providing a means of transliteration, sign language identification can help to bridge the gap between these groups and facilitate more convenient communication. Through using a series of photos of deaf patients that are saved in a database of deaf people, we plan to enhance a deep learning model for character detection in this study. There are 29 categories, and from the Network Research Center, 29000 (JPEG) photographs have been chosen for each category. It will be simpler to interact with deaf persons in society if we can comprehend them [5].

1.3 AIMS AND LIMITATIONS IN THIS THESIS

1.3.1 Main Goal

Implementation of software model to classify sign language from images.

1.3.2 Particular Goals

a. Using the deep learning model to recognize 26 more characters (Space, Delete and Nothing).
b. Reduce the cost of classification by using algorithm's deep learning
c. Avoiding mistakes that occur to deaf people during their interaction with society.
d. Enhances the ability of deaf people to interact with society without fear and quickly understand society.
e. Increase detecting skill using deep learning methods.

1.3.3 Limitations We Faced

The system contains many restrictions, only images of letters without words or number, as the number of images used in the system is thousands of images, but it includes only 26 letters in addition to SPACE, DELETE and NOTHING the formation of containers. The form will detect the letter and each letter contains a specific class.

1.4 SIGNIFICATION OF THE THESIS

People often struggle to comprehend the deaf when learning sign language, and not everyone in the deaf community is conversant in sign language. Sign languages have been the subject of extensive study. Additionally, researchers train algorithms to detect letters and words at a level comparable to anyone who is conversant in sign language using deep learning. Deep learning yields encouraging outcomes. Sign language has the potential to significantly improve the lives of individuals by increasing the number of people who can comprehend it. With a greater understanding of sign language, individuals are able to communicate more effectively and live their lives without fear. This, in turn, can facilitate their daily activities and enable them to fully participate in society [6].

1.5 THESIS OUTLINES

This part describes the thesis content. my thesis included Five chapters as follow.

Chapter one includes introduction, research problem, research questions. aims, limitations and the structure of the thesis.

In chapter two included the literature review and the challenges of the previous work that the researcher faced as well as the gaps in their works. Also, in this chapter we classified the literature review according to algorithms and techniques that they used.

Chapter three included the methodology that we proposed to solve the problems and gaps that we defined in the previous work. Also, the results were added in the third chapter.

Chapter four included the discussion about how the research was implemented until got the result.

Lastly was chapter five as the final chapter in my thesis and it was talked about the conclusion and feature works.

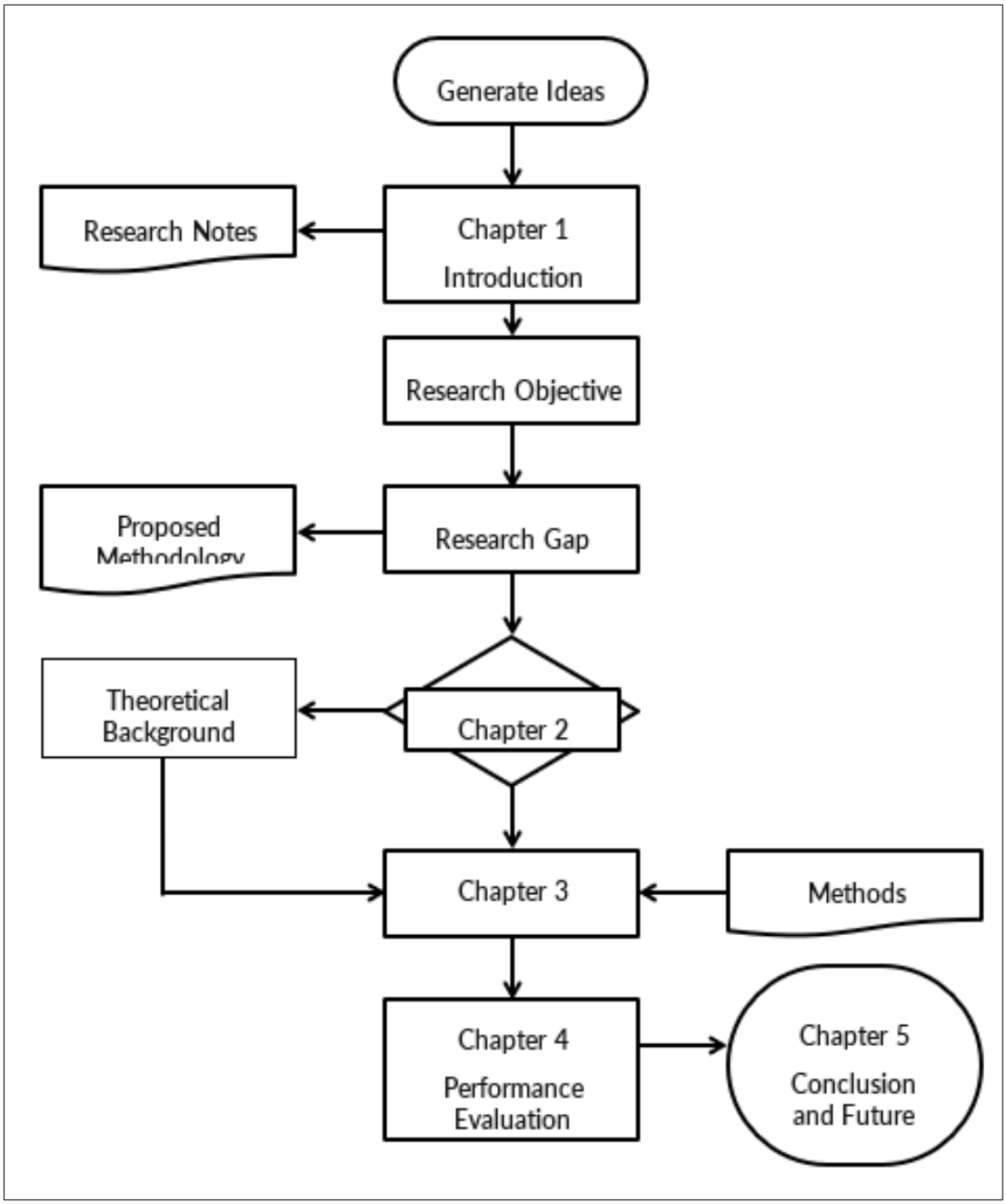

Figure 1.1: Thesis Outline.

2. THERORTICAL BACKGROUND

2.1 BACKGROUND

This section will include the overview of artificial intelligence techniques (AI), such as the Recurrent neural network (RNN) which are used for gathering and collecting large amount of the data for analyzing. This data was gathering from different sensors and those date was not clear so we have to use AI algorithms like RNN to remove the noise in that data and makes ready for use [7].

RNN algothim has a great ability to filter this data at high speed to give good results.

Due developing of the digital world and the wide-spridly of artificial intelligence topics

2.1.1 Overview of Artificial Intelligence

The backbone of the stracture of the Artificial Intelligence is to make the machine can act and thing like the human this is the idea of the AI. The use of AI technology was widly spread in every filed like the robotics that can do what the people can do , natural language processing that can deal with the huam feelings and speech . so it try to make the machine implement the human jobs [8].

a. Reactive machines: AI can directly response on events, meetings and so on.
b. Limited memory: AI systems can learn easly and get the experiences Fastly from the previous experiments.
c. Theory of mind: These AI systems have the capacity to comprehend and make sense of the motives and ideas of others.
d. Self-awareness: These AI programs can think about their own deeds and experiences and have a sense of self.

AI has the ability to completely transform a wide range of markets and has already made a big difference in areas like healthcare, banking, and transportation. The creation and application of AI, however, also raises ethical and societal issues, such as the possible loss of employment to automation and the possibility for prejudices to be reinforced by AI [9].

2.2 MACHINE LEARNING

Machine learning has different kind of classification like:

a. Supervised learning: this type of classifer depend on the labeled data to train and learng in order to classified the date then test it to get the required output. After treaing and testing process we can made our right model depend on the the ruslt of the model implementation.
b. Unsupervised learning: This type of classifier depend on the unlabeled data which the model can train the model and classified the data depend on their feature such as the , color , shapes and so on.
c. Reinforcement learning This type of learning requires to series of actions and adjust the rules in order to get the result depend on specifics conditions.

The use of machine learning algorithms to generate the right model required a big knowledge in mathematical equations. Machine learning algorithms helps to build that model to predict the output and make a decision to solve the problems easily within a short time. The machine learning algorithms became widely used in a different field such as natural Language processing (NLP), robotics, human activity detection, auto-driving cars , drones, agriculture and health care systems[9].

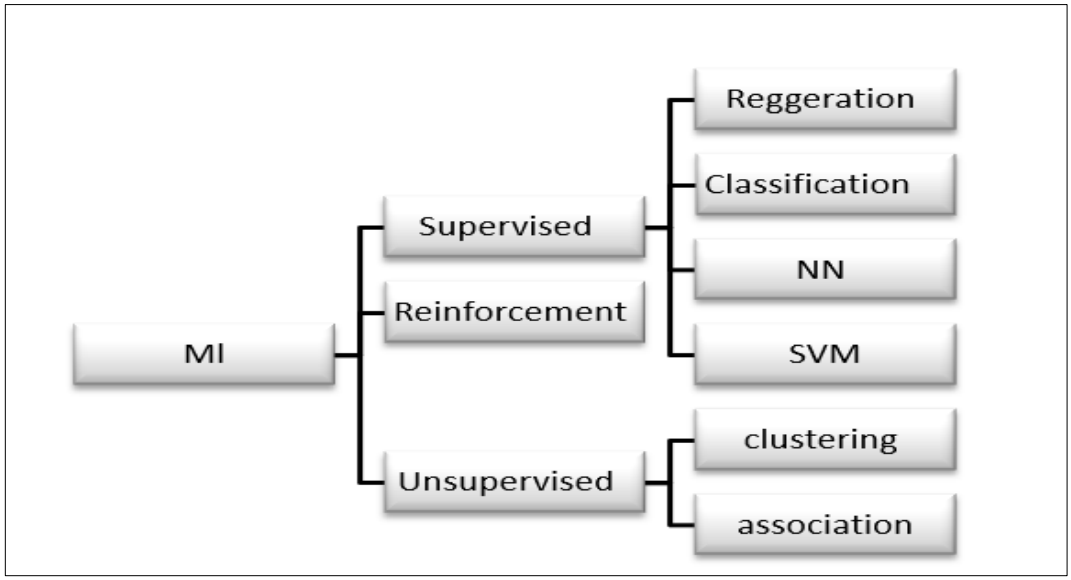

Figure 2.1: Machine Learning Methods [8].

2.2.1 Supervised Learning

If you have data and want to predict the outcome, you can use a supervised machine learning model. A labeled dataset is used to train this type of model, which means that the input data and corresponding output answers are known in advance. The model uses a supervised learning approach to learn how to produce output based on input data. This allows the supervised model to predict the reaction to incoming inputs in uncertain scenarios with high accuracy [9]. It is a process that uses regression and classification techniques to create machine learning models.

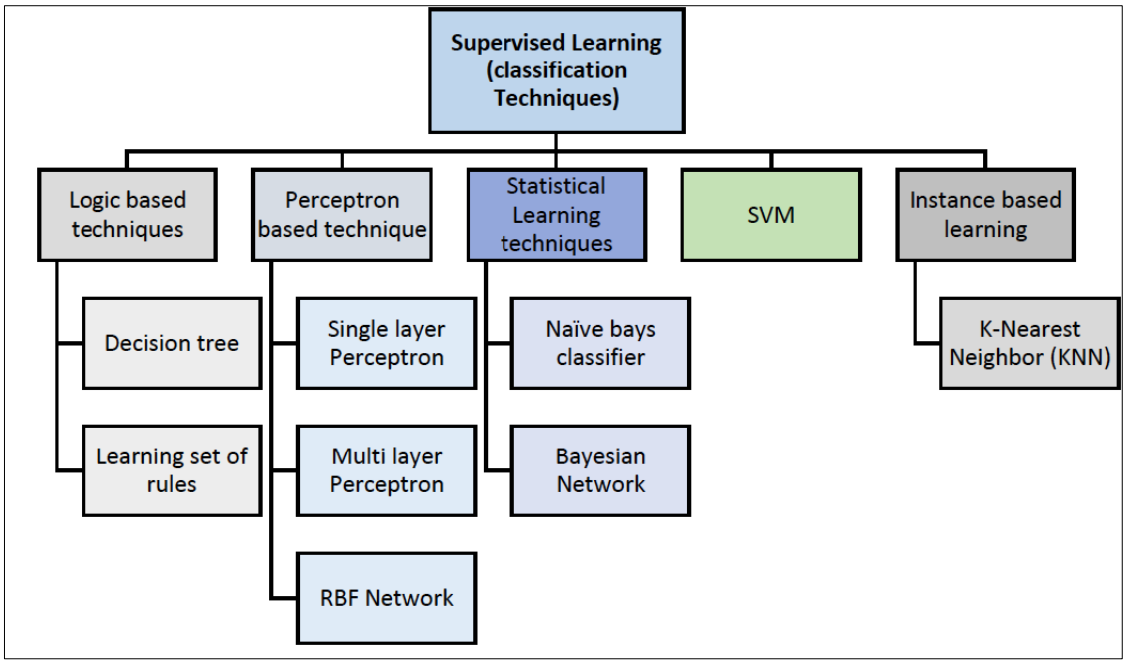

Figure 2.2: Supervised Learning Approaches [9].

a. Classification Techniques: Based on the input data, classification models are used to predict certain outcomes. The classification model can be used, for example, to determine whether a tumor is cancerous or whether an email message is spam. The data provided is classified into pre-defined categories using these forms. Applications of classification models are numerous and include medical imaging, voice recognition, and credit scoring.

Classification algorithms can be used to predict outcomes if the data can be separated into discrete categories. Classification algorithms are frequently used in applications where the goal is letter and number recognition, such as handwriting recognition.

. Unsupervised pattern recognition algorithms, which do not require labeled data, are often used in image processing for tasks such as image segmentation[10].

b. Egression Techniques: If you are trying to predict continuous responses, such as financial asset prices or physical characteristics that are difficult to measure, you can use regression algorithms. They are used to predict a numerical value for a given input. Some common applications for regression algorithms include algorithmic trading, virtual sensing, and electrical load forecasting.

When you work on huge data or if the response you are trying to predict is a real number, such as temperature, you can use regression techniques. Some popular algorithms for doing regression include linear regression, polynomial regression, and support vector regression. These algorithms are commonly used to predict continuous responses in a variety of application.

Regression analysis thereby determines the link between two or more variables. Regression techniques come in a variety of forms and can be used to produce predictions. Three indicators serve as the main motivators for these methods (The number of independent variables refers to number of input variables used in the model, while the type of dependent variable refers to whether the output is continuous or categorical [11].

2.2.2 Unsupervised Learning

Unsupervised learning is a branch of machine learning used to identify underlying data structures or hidden patterns in a dataset. It does not require labeled output data and used to infer conclusions from datasets that have input data but no known responses. Unsupervised learning is often used to find out patterns or relationships in data that may not be immediately apparent [12].

Clustering is popular unsupervised method that is used to discover underlying patterns or groupings in data. It is often used in manage data analysis to identify clusters or groups of data points that are similar to each other. Most common applications that using cluster analysis are gene sequence analysis and object recognition [12].

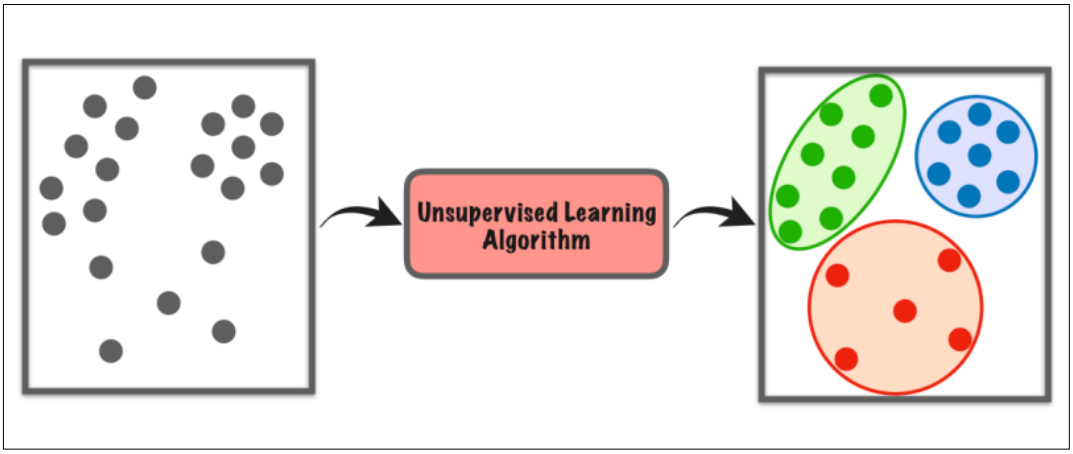

Figure 2.3: Unsupervised Learning Process [12].

2.3 DEEP LEARNING

Deep learning (DL) is an approach of ML that involves the use of computational models with many layers of processing in order to representations data at various levels of abstraction. These techniques have been very successful in improving state of art performance in many fields, including genomics, object identification, visual object and speech recognition. Deep learning algorithms use the backpropagation method to learn and identify complex structures in large datasets, allowing them to adapt new data in an automated way [13].

2.4 NETWORK ARCHITECTURES

There are many different types of network architectures that used to build a neural network. Some common types of network architectures include:

2.4.1 VGG 16

The VGG16 model uses a small 3x3 receptive field (also known as a filter) with a stride of 1 pixel throughout the entire network. This is unique because it uniformly applies 3x3 filters. By using two consecutive 3x3 filters, a 5x5 effective receptive field is created. Similarly, three 3x3 filters create a 7x7 receptive field. This means that multiple 3x3 filters used in place of larger receptive area [13]**.** Advantage of using three layers of 3x3 instead of a single one of 7x7 is that there are three non-linear activation layers in addition to the three convolution layers, compared to just one in the 7x7 matrix. This allows the network to make

more discriminatory decisions and converge more quickly. Additionally, using 3x3 layers reduces number of parameters in the model compared to using a 7x3 layer. As figure 2.4 shows:

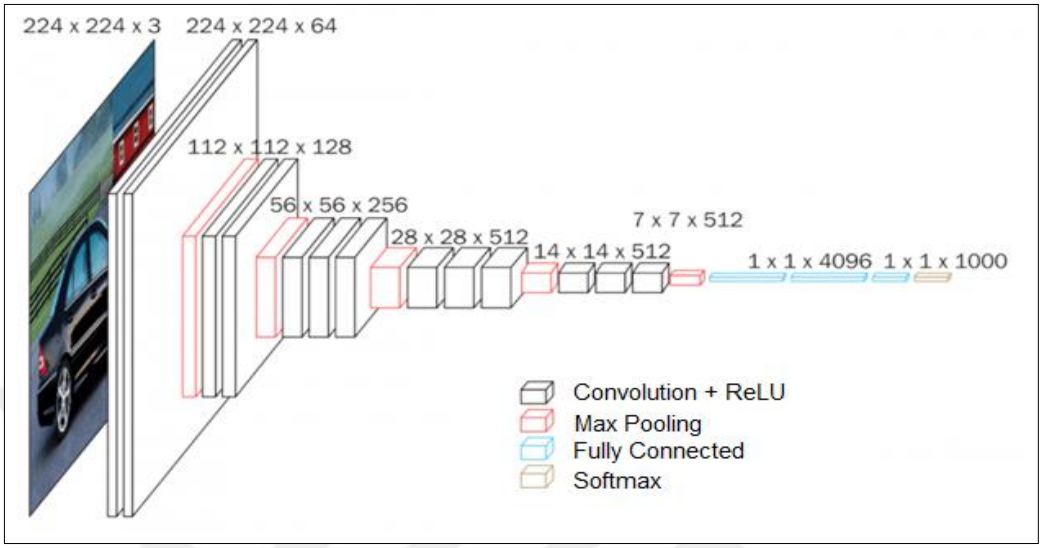

Figure 2.4: VGG-16 Architecture [13].

2.4.2 ResNet

CNN architectute was built by Microsoft research team and was called ResNet. This architecture was widely used in several applications such as image processing because it has the capability to classify the huge amounts of the images in a short time and with high quality performance.

One of the aspects that sets ResNet apart is the way it uses skip connections to allow the network to learn residual functions instead of being tempted to learn the input-output function explicitly. This facilitates the network's learning process and enhances its effectiveness in picture categorization tasks.

Since then, The ResNet architecture has undergone various iterations that have been created and utilized in a range of applications [14]. Figure 2.5 shows ResNet Architecture.

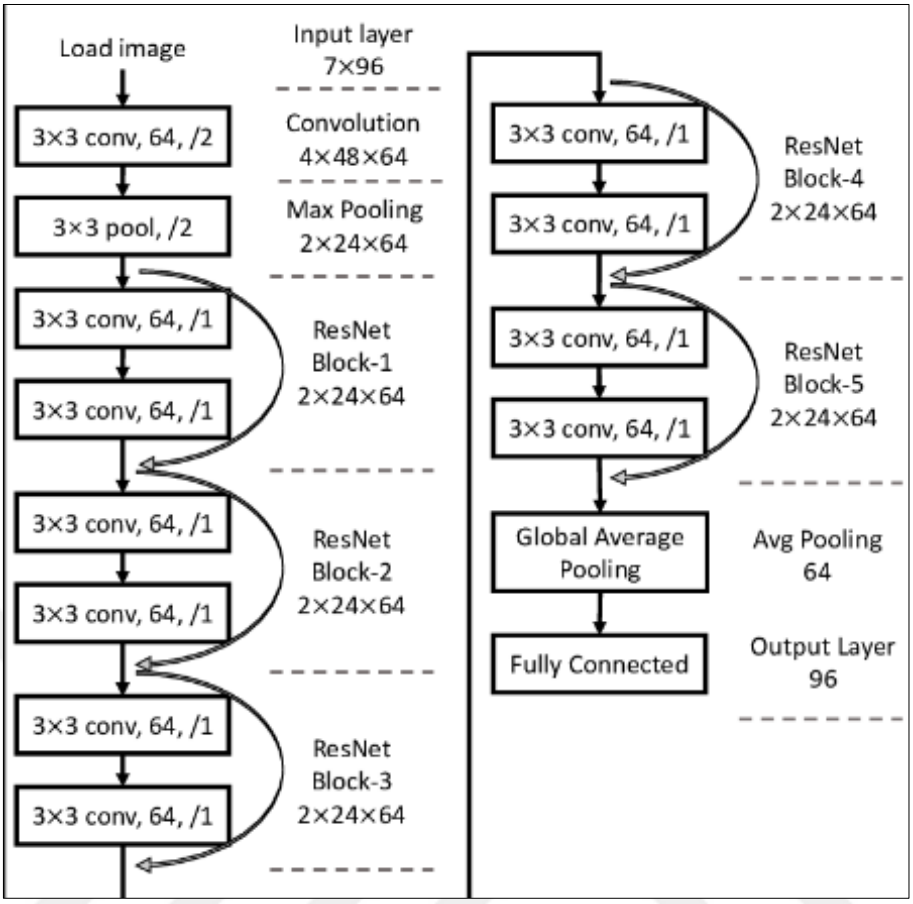

Figure 2.5: ResNet Architecture [14].

The advantage of using skip connections in a residual network is that they allow the network to bypass layers that may degrade its performance during training. This helps to regularize the network and prevent issues such as vanishing or exploding gradients.

2.4.3 MobileNet

In contrast to a traditional convolutional layer, which uses an array of filters to extract features from input data, deep separable convolution begins with a deep convolutional stage that applies a single filter to each input channel. The next step is pointwise convolution of the generated feature maps, which produces a set of features generated from the deep convolution. The increased efficiency of MobileNet and decreased computing requirements are made possible by dividing the convolutional layers into depth and point operations [15]. This is especially useful for mobile devices with constrained processing power. As shown in Figure 2.6, deep convolution follows pointwise convolution to depict a deep separable convolutional sequence.

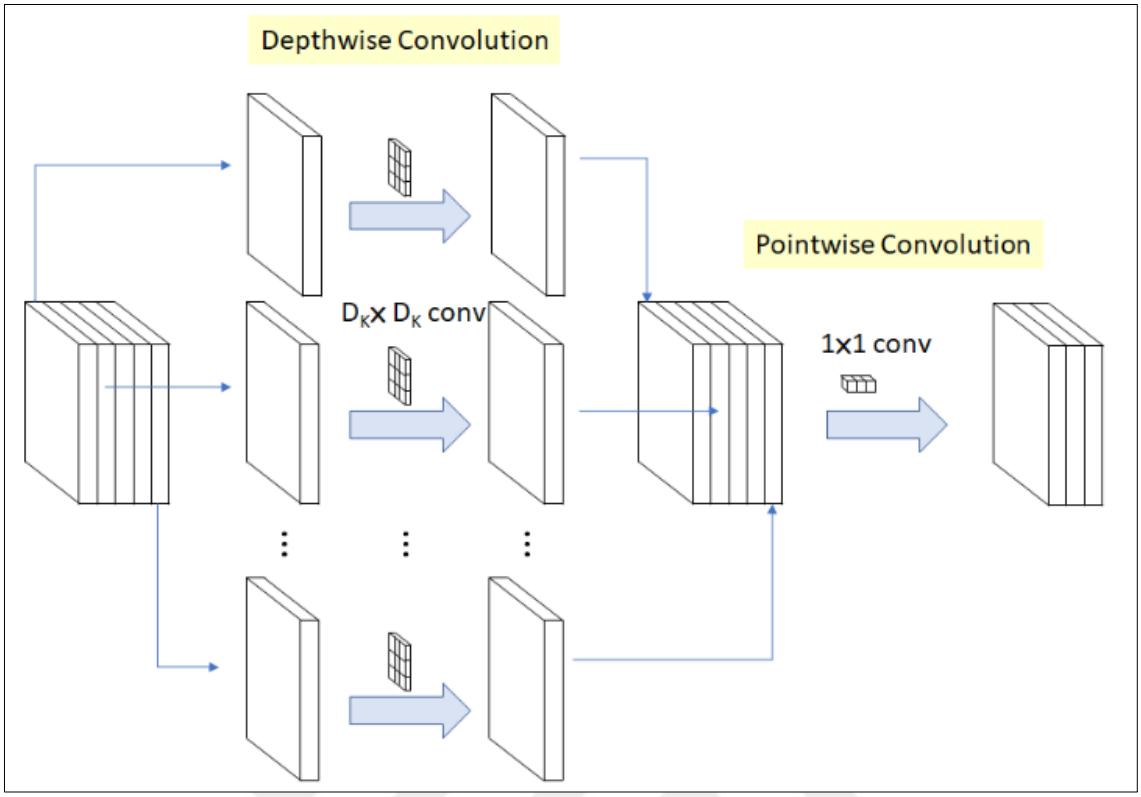

Figure 2.6: Depth Wise Separable Convolution [15].

2.5 DATA PROCESSING

It is the process of analyzing raw data and convert it into structured information. It is a common technique used in data mining and ML to extract useful insights and patterns from huge amounts of data. This process involves a variety of techniques and tools, such as data cleaning, data transformation, and data visualization, to help make sense of the data and extract meaningful information from it [16].

Data are the fuel that powers technology. The majority of data in the real world is noisy. It contains numerous errors, resulting in unstructured data. Data preprocessing steps will be used to convert unstructured data into structured data. As a result, when working with raw data, you will almost certainly use data preprocessing machine learning steps. the most effective data preprocessing steps in the following section [16].

There are some steps for data preprocessing which below:

a. Import the Libraries:

The important libraries listed below should be imported now for data preprocessing. I used the "import" keyword every time I imported a library. The fundamental libraries are those:

a. NumPy: When carrying out mathematical computations, it is frequently utilized in machine learning. Calculations using, for instance, linear algebra, the Fourier transform, and N-dimensional arrays.
b. Matplotlib: Bar charts, pie charts, and line graphs are just a few of the figures and graphs that may be made with this library.
c. Pandas: primarily employed for data manipulation. It is an open-source, free library. Both all data structure functionalities and all data analysis tools are included.
d. Seaborn: It is a library for displaying data. It might be considered an enhanced version of Matplotlib. It mainly serves to increase the informational value of graphs and charts.[17]

b. Importing the datasets:

The data sets import uses the Pandas library. You can easily import data sets from CSV files for small preprocessing. The datasets file can also be in HTML or Xlsx format. However, as you are aware, CSV files are small in size and thus import faster than other formats. and frequently import data from the Kaggle website

c. Fill up the Missing Values in the Data Sets:

There are several techniques used to fill missing data in dataset. The particulars of the data and the cause of the missing numbers will determine which strategy is most suited. The following are a few typical methods for adding missing values:

a. Imputation: This involves using the characteristics of the remaining data to replace the missing values. This might include employing a more sophisticated imputation technique like multiple imputation or predictive modeling, or it can involve substituting missing values with the data's mean or median value.
b. Interpolation: This method determines the value of the missing data by estimating its surrounding data values. This may be done using more sophisticated interpolation techniques like cubic spline interpolation or polynomial interpolation, or with linear interpolation, which estimates the missing value using a straight line.
c. Deletion: If a significant amount of the data set is missing, it may be advisable in some circumstances to remove only the rows or columns that contain the missing values.
d. Convert text or category values to numeric values: Nominal data, sometimes referred to as categorical or textual data, is information that can be grouped or classified but cannot be ordered or ordered.

This type of data often must be converted into a digital form for use in machine learning methods [18]. Hot encoding is a popular method for converting category data into a digital representation.

It is important to carefully consider which method is most appropriate for filling up missing values in a dataset, as the choice of method can significantly impact the accuracy and reliability of the resulting data.

d. Modification of categorical or text values to numerical values:

Categorical or text data, also known as nominal data is data that is placed in one of the category. but it concerns each individual or groups and cannot be put in order of priority. To utilize this type of data in a machine, there is need to understand following points: as with any learning algorithms, it is sometimes crucial to transform it into a numerical format [18].

One of the most used methods of transforming categorical data into numerical form is one-hot. encoding. This includes creating a new binary column for each of the categories in data. For for instance, if there is a column with the names ‘red’, ‘yellow’ and ‘blue’, three new columns would be created: ”red,” “yellow,” and “blue. “ Each row in the original column would be containing a 1 in the new column that corresponds to the value of the original attribute, and 0s in all the other new columns.

Label encoding is another method of transforming categorical data to numerical data type. This implies the process of applying a different whole number or an integer to every category in the data. For example, using the same categories as above, “red” might be assigned the value 0, “yellow” might be assigned the value 1 while the value 2 might be assigned to the attribute “blue”. The choice of the technique that is appropriate for converting categorical or nominal data must be done very carefully.

text documents to numerical vectors because the chosen approach can significantly affect the outcome of ML algorithm. The Label Encoder () class is used for transforming categorical or string like features into vectors.as a numeric values [19].

2.6 MODEL EVALUATION

Model evaluation is the process of evaluating how well a machine learning model performs on a given data set. This is an important stage in developing a machine learning model because it allows you to evaluate the model's ability to generalize to new data and identify

any flaws. Depending on the type of business a machine learning model is used for, its effectiveness can be evaluated using a range of metrics. For example, standardized assessment.while common evaluation metrics for regression tasks include mean absolute error and mean squared error [20].

2.7 RELATED WORK

There are several of previous studies that have investigated using deep learning and machine learning methods for learning and improving hand sign language recognition. Some examples include:

This study [21], discussed the creation of an accurate and useful video-based Egyptian sign language recognition system to help the deaf people in Egypt. The researchers developed a computer vision system that utilised two algorithms and several neural network combinations. The first approach extracted spatial characteristics using a Convolutional Neural Network that was retrained using the inception model. To extract temporal information, the second technique combined a CNN with a Long Short-Term Memory (LSTM). In conclusion, the two models correctly identified 9 words among Egyptian deaf people with an accuracy of 90% and 72%, respectively.

The purpose of the study in [22], was to create a real-time hand gesture identification system. The system consisted of three phases: feature extraction, image acquisition, and recognition. In the first phase, input images of hand gestures were captured using a digital camera at a high frame rate. In the second phase, a scaling, rotation, translation, and position-invariant feature extraction technique was used to extract features of the input image based on the moment feature extraction technique. The system used a neural network for hand gesture identification. The performance of the system was tested with real data, and it was found to have acceptable performance in hand gesture identification, with an accuracy of 88.7% for recognized hand gestures after training the neural network. The system focused on three areas in the feature extraction: the ability to detect and extract human hands from complex images containing human bodies, the ability to work in any lighting condition, and the ability to recognize hand tracking.

A study in [23], aimed to use GPU acceleration and (CNNs) to recognize Italian gestures. CNNs are able to automatically construct features, which reduces the need for complex manual feature construction. The CNN used in the study achieved a high level of accuracy

in recognizing 20 Italian gestures, with an across-validation accuracy of 91.7%. This shows that the prediction model was successful in generalizing to users and circumstances that were not present during training.

A study published in 2016 that used a deep convolutional neural network (CNN) to recognize American Sign Language (ASL) signs from video sequences. The CNN was trained on a dataset of over 10,000 annotated video frames and gives high accuracy of 85.23% on a test set [24].

A study published in 2018 that used a combination of a deep neural network and a support vector machine (SVM) to recognize hand gestures in American Sign Language (ASL). The model was trained on a dataset of over 5,000 annotated video frames gives high accuracy of 93.75% on a test set [25].

Each of these research studies highlights the significance of employing deep learning techniques in sign language classification, as it can save both time and effort while enhancing the model's accuracy. Since some previous studies failed to achieve high accuracy, our proposed model aims to surpass their outcomes. While the studies employed an image dataset for comparison, we utilized a dataset specifically for classification purposes.

In our work, we considered the time factor in the classification process, that wasn’t included in some studies. Specifically, we calculated time that spent on training and testing. This is an important point to consider, as it can impact the accuracy and efficiency of the classification process.

3. RESEARCH PURPOSE

Another potentially widespread area of computer vision that can be used in many industries is hand gesture recognition. It is noteworthy that correct recognition of hand gestures can improve the relation between a human and a machine. A hand gesture is defined as an orientation of the hand with reference to the position of the fingers and the wrist in gesture-based interfaces and sign language. In the third chapter of this thesis, the use of methods based on deep learning and the Python programming language in the process of recognizing hand signals is described. The process involves a number of critical factors necessary for the development of a good hand gesture recognition system. Pre-processing is the first step in our strategy. Before proceeding further, the input images are now in a form that can be used in the deep learning model. To eliminate variability, the pictures are normalized using image standardization. Second stage is to choose the model. Several models such as CNNs were used to identify the most suitable deep learning model for hand gesture recognition. CNNs are often employed in image classification problems, and the encouraging results of using them in applications related to hand gesture recognition are often observed. The third process is typical training. With preprocessed data, the selected deep learning model is then trained. During the process of training, the model is able to learn and distinguish certain features within an input image that relates to particular hand gestures. In order to try and reduce the amount of loss function and tune the weights we have in our model; we apply gradient descent and backpropagation. The next and the last step of the model is model evaluation. For the assessment of the model that has been trained, a different test dataset is utilized. The evaluation metric includes F1 score, recall, precision, and precision. Further, to know which hand signs the model often gets it wrong, we also looked at the confusion matrix. This paper outlines a detailed explanation of the Python programming language and its libraries that were used in developing the model as well as the method applied in recognizing the hand signal. Python is the most preferred language for deep learning due to its simplicity and availability of scientific computing and machine learning libraries. The approach described in this chapter can prove to be a useful tool for practitioners and researchers who wish to develop similar applications with Python and Deep Learning.
The results indicate that the outlined strategy is effective and reveal the areas where the industry can profit most. Altogether, this chapter provides a detailed and systematic guide to hand gesture recognition using Python and deep learning.

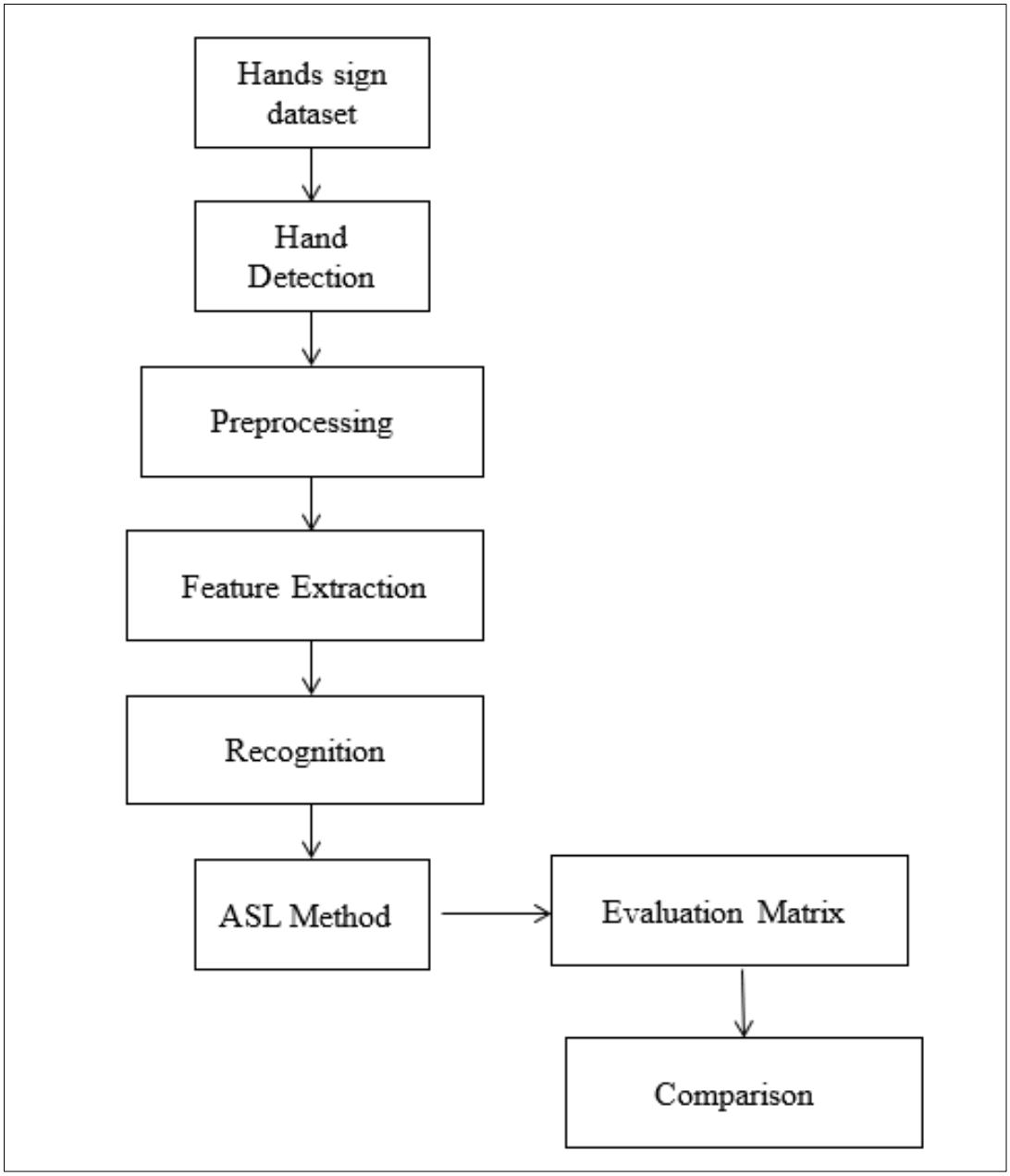

Figure 3.1: The Steps of Proposed Work.

Here is a full description of the stages involved in building a dataset for deep learning using American Sign Language (ASL), as seen in the above graphic that depicts the steps of the proposed study. The procedure is summarized as follows in general:

a. Gather information: Gather video evidence of individuals demonstrating the hand signals for every letter in the ASL alphabet. You may look for already released footage online or film someone making the signals using a camera. Ensure that the videos are crisp and clearly see the hands.

a. Label data: Assign labels to each hand mark in each video frame using an application like Label Image. Marking entails placing a mark (e.g., "A", "B", "C", etc.) on the hand mark after drawing a bounding box around it. Deep learning models require labeling, which can be a tedious procedure.
b. Data Partition: Divide your labeled data into two categories: the training part and the test part.
c. Data preprocessing: Resize images, convert them to grayscale, and normalize pixel values for data preprocessing. This phase ensures that the training set input data is reliable and consistent.
d. Data Optimization: Optimize data by performing transformations such as translation, flipping, and rotating. Your dataset will be larger thanks to data augmentation, which also helps avoid overfitting.
e. Training model: Using an appropriate architecture, such as a convolutional neural network (CNN), train a deep learning model using the training data. To increase performance, choose a suitable optimization method, such as stochastic gradient descent (SGD), and adjust the hyperparameters.
f. Model validation: Using several categories, such as the confusion matrix, evaluate the effectiveness of the methods used in this study.
g. Deploy model: Deploy the trained model by integrating it into a web application, mobile app, or other platform. Make sure the model's performance is satisfactory in the real-world scenario.

3.1 DATASET

Data set using in training and testing of algorithms for classifying sign language was collected form Kaggle website[26]. Out of 87,000 images, only 29,000 were used due to the large amount of data. These images were divided into 29 classes, including letters A through Z, spaces, delete, and nothing, each class contained images**.**

The dataset used for this project was divided into three parts: training, validation and testing. The training stage of dataset included 70% of the images, which totaled 20300 images whereas testing and validation datasets each included 15% of the images, totaling 4350 images each. The images were divided into 29 categories, including letters A to Z, space,

delete, and nothing. The shape of the training data was (29000, 64, 64, 3) and the shape of the training labels was (299000, 29). Figure 3.1 shows the letters used in the dataset.

Figure 3.2: Letters Used in Dataset.

Table 1 shows the number of files in each class.

Table 3.1: Distribution of Images Over the Classes.

Class (Category)	Number of Images
A to Z	Every latter=1000
Delete	1000
Space	1000
Nothing	1000
Total	29000

3.2 EQUIPMENT AND TECHNOLOGY USED

In the programming language, Python is famous and is in high demand. This one is preferred for machine learning jobs because of the numerous features it comes with, among other factors. adaptability. Explaining why Python is so important is particularly well summed up by one of its key strengths.

machine learning is the existence of robust libraries such as Pandas, Keras, Matplotlib, and TensorFlow. These libraries also support data manipulation, data analysis, and data drilling. learning, and thus helps the developers to solve large data problems and create sophisticated machine learning models.

a. There are many libraries available for machine learning tasks in Python, including:There are many libraries available for machine learning tasks in Python, including:
b. NumPy: It is also a versatile library that is used in the manipulation of numerical data and it is a resource that is crucial. for most of the machine learning and data science use cases.
c. Pandas: Pandas is quite popular among the Python data science community and is considered as one of the most important. library for many of the machine learning and data science related tasks.
d. Scikit-learn: A clear and easy to use machine learning library that has a good documentation. algorithms for regression analysis, classification, clustering and dimensionality reduction.
e. TensorFlow: is a package of deep learning and machine learning that has several tools to facilitate training. assess, and enhance models' performance. Neural networks and other machine learning It is used in the creation and training of the models that are used in the various industries.
f. Keras: is a library that is used to develop and train deep learning neural networks. It is based on TensorFlow; and gives a better user interface for building and training the neural networks and other deep learning models.
g. PyTorch: is library for machine learning and deep learning that has tools for model construction. and training of neural nets and other types of machine learning algorithms. It may be recalled that flexibility and easiness of use has been identified as the most important one [27].

h. Matplotlib: Python is another programming language which has a tool for graphics called Matplotlib. This method can be used to create three types of views which include interactive views, animated views and static views.

Python. Thus, Matplotlib enables almost any type of static and dynamic graphics, from basic line, scatter, bar, error, and pie charts, and histograms.. It is known for its adaptability and powerful capabilities and is frequently used in scientific and data visualization sectors. These are just a few of the many Python libraries available for machine learning applications. The specific needs of your machine learning project will determine which library you should choose [28].

3.3 IMAGE FORMAT

Using Python libraries, images were loaded into memory in RGB (red, green, blue) format. JPEG format is used for images. Using these libraries, JPEG files were read during the image loading phase and then converted to RGB format - a common representation of digital images. Each pixel in an image is stored in RGB format as a set of red, green, and blue values, which can be combined to create a wide range of colors. Many image processing and machine learning techniques can be used to edit and modify images once they are stored in memory as RGB files.

3.4 NETWORK ARCHITECTURE

The phrase “network architecture” in machine learning describes the structure of a neural network, which consists of layers of connected “neurons.” The structure of a neural network plays a crucial role in determining how well it performs because it controls how the input data is changed and processed by the network in order to generate output.

a. Typical forms of network architectures include the following: The most basic kind of neural networks are called feedforward neural networks. In these networks, data moves from the input layer to the output layer alone and never back again.
b. Convolutional neural networks (CNNs): The main purpose of these neural networks is to assess data with a grid-like structure, such an image.

In machine learning, a wide range of network topologies can be used; The choice of structure can be influenced by the specific problem being addressed as well as the characteristics

of the data used. They are particularly useful for tasks such as image classification and object detection.

c. Recurrent neural networks (RNNs): These are neural networks that have connections between neurons that allow data to flow in loops, allowing them to process data with a temporal dimension, such as text or time series data.
d. Autoencoders: These are neural networks that are trained to reconstruct their input data at the output layer and are often used for tasks such as dimensionality reduction and data denoising.

Each of these network topologies has advantages and disadvantages, and which one to choose will depend on the specific requirements of the problem being addressed.

Every model we're going to utilize has its own network architecture, we talked about all models as pre-trained models. However, the VGG16 family's concept has been included in our suggested model since it is simple to comprehend, simple to utilize, and particularly effective at extracting characteristics from images. A pooling layer that employs the biggest value max pooling approach is employed after a sequence of convolution layers that employ small 3 x 3 pixel folding filters (image components). After a 2 x 2 stride filter step, a 3 x 3 pleat filter, and zero padding of type "same," where the required padding is computed and applied such that the output size is identical to the input size for each pleating filter, several convolution layers were used. In addition, the corrected linear activation function Relu was used. The grouping layer with a size of 2 x 2 that is then applied after these layers is based on the layer with the largest value [29].

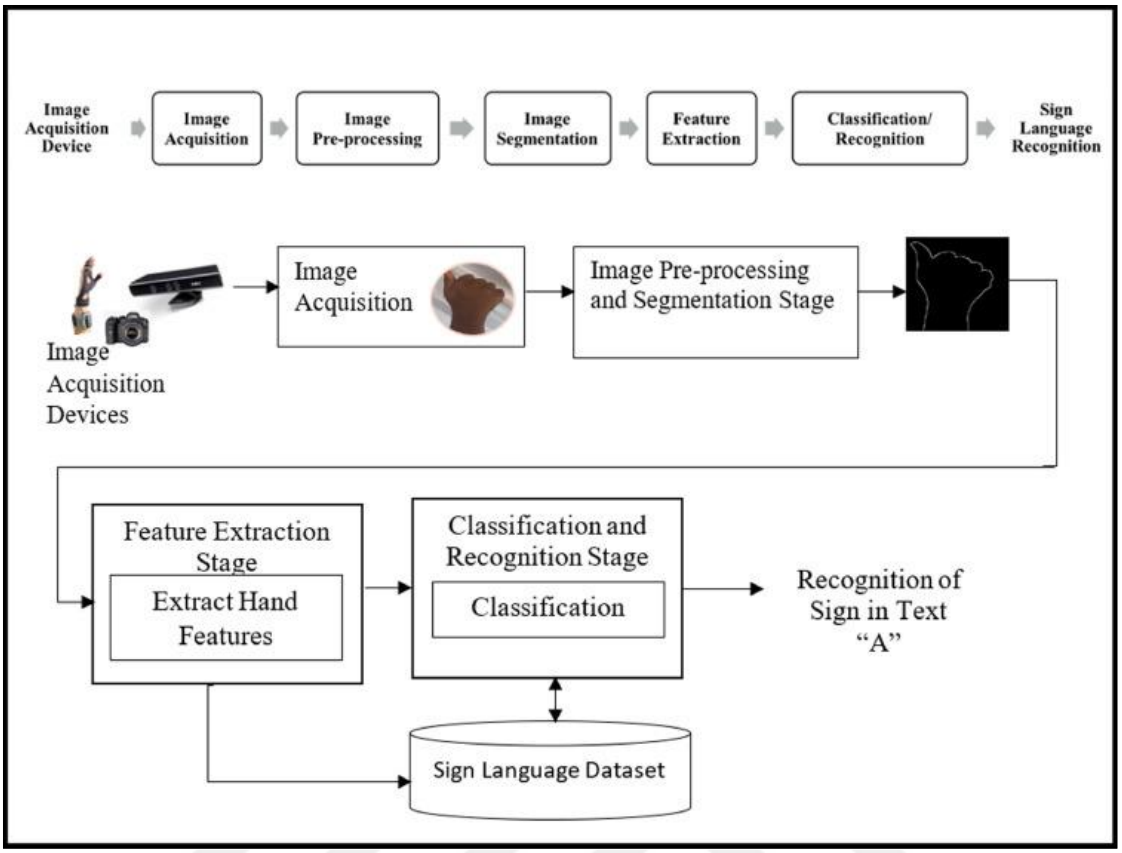

Figure 3.3: Architecture of Proposed ASL Approach.

3.5 PROPOSED MODELS (ASL. MODEL)

In figure 3.4 the first layer of the convolution layer is added, which is conv2D, 2D is that the image is two dimension, this has three parameters, the first is the number of filters, the second (3 x 3) is the size of the filter, and the third is the activation function, input shape=(64,64, 3))) 64 Image length and width. Note each layer has an activation rule.

Now we will add another layer, the one parameter, here we will divide the feature map into 2x 2 folds of the conversion, and then we will extract the largest value from each square with two values according to what they divided, and thus we made a resize of the feature

Now we will do the process of filtering what you mean I got it from the pooling layer by entering the one vector filter.

Output layer here we took six because we have 29 categories of class, I mean it will be shown according to the 29 categories.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 conv2d_1 (Conv2D)           (None, 60, 60, 64)        18496

 max_pooling2d (MaxPooling2D  (None, 30, 30, 64)       0
 )

 conv2d_2 (Conv2D)           (None, 28, 28, 128)       73856

 conv2d_3 (Conv2D)           (None, 26, 26, 128)       147584

 max_pooling2d_1 (MaxPooling  (None, 13, 13, 128)      0
 2D)

 conv2d_4 (Conv2D)           (None, 11, 11, 256)       295168

 conv2d_5 (Conv2D)           (None, 9, 9, 384)         885120

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 384)        0
 2D)

 flatten (Flatten)           (None, 6144)              0

 dense (Dense)               (None, 512)               3146240

 dense_1 (Dense)             (None, 29)                14877

=================================================================
Total params: 4,582,237
Trainable params: 4,582,237
Non-trainable params: 0
```

Figure 3.4: Proposal Model Layers.

3.6 DEEP LEARNING WITH ASL APPROACH

Deep learning for hand sign recognition involves the use of neural networks to classify hand signs based on input images. The process can be broken down into several steps, including data preprocessing, model training, and prediction. Here's a brief overview of the math involved:

a. Data preprocessing: The input images are first preprocessed to ensure that they are of a consistent size and that they have similar lighting and background conditions. This is typically done using techniques such as resizing, normalization, and augmentation.

b. Model training: A deep learning model is trained using a dataset of labeled hand sign images. The model is typically a convolutional neural network (CNN), which is designed to extract meaningful features from the input images.

c. Prediction: The model processes the input image and generates a probability distribution across the available hand sign categories as output. The class with the highest probability is selected as the predicted output. Here are some mathematical equations involved in the deep learning process:

d. Convolution operation: The convolution operation is used to extract features from the input image. It involves sliding a small filter (kernel) over the image and computing the dot product between the filter and the pixels it overlaps. The output of the convolution operation is a feature map.

$$f(i,j) = (w * x)(i,j) = \Sigma m \Sigma n \; w(m,n) \; x(i+m,j+n) \tag{3.1}$$

where f(i,j) is the value of the feature map at position (i,j), w is the filter, x is the input image, and m and n are the indices of the filter.

a. Activation function: The activation function introduces nonlinearity into the model, allowing it to capture complex patterns in the input data. The most commonly used activation function in CNNs is the rectified linear unit (ReLU), which is defined as:

$$ReLU(x) = max(0,x) \tag{3.2}$$

b. Loss function: The loss function measures the difference between the predicted output and the true label. The most commonly used loss function for classification problems is cross-entropy loss, which is defined as:

$$L(y, \hat{y}) = - \Sigma i \; y(i) \log(\hat{y}(i)) \tag{3.3}$$

where y is the true label (one-hot encoded), $\hat{y}$ is the predicted probability distribution, and i is the index of the class.

c. Backpropagation: Backpropagation is used to update the weights of the neural network during training. It involves computing the gradient of the loss function with respect to the weights and using this gradient to update the weights using an optimization algorithm such as stochastic gradient descent (SGD).

These are just a few of the mathematical concepts involved in deep learning for hand sign recognition. The field is constantly evolving, and new techniques and models are being developed all the time.

4. RESULTS AND DISCUSSION

4.1 EXPERIMENTAL RESULT OF EACH MODEL

As I mentioned before dataset was divided into three sets or stages: training, testing, and validation. The training set consisted of 70% of the total number of samples, while the validation and testing stages consisted 15% for each. Out of the 87,000 images in the dataset, only 43,500 were used for the analysis. The dataset included 29 classes of images, ranging from A to Z, as well as images representing spaces, delete, and nothing. The dataset was split in this way in order to provide a representative sample for training, testing, and validating machine learning models.

4.2 TRAINING AND VALIDATION OF EACH MODEL

By finding patterns and connections in data, machine learning algorithms can learn from the data. They make judgments, gain confidence, and function better with this understanding. It is important to remember that the performance of a machine learning model is greatly affected by the level of training data. Model performance will increase with the accuracy and relevance of the training data. As such, selecting and preparing training data appropriately is essential before building and optimizing a machine learning model.

Available data is often divided into three machine learning (ML) categories: training set, validation set, and test set. The machine learning model is trained on the training set; Its performance is evaluated and its hyperparameters are modified in the validation set; Its final performance is evaluated on the test set.

a. The training set, which is used to fit the machine learning model to the data, is usually the largest of the three. After being trained on the training set, the model gains the ability to predict outcomes by identifying patterns and correlations in the data.
b. The performance of the model is evaluated and its hyperparameters are modified using a validation set. The results of the validation set are used to adjust the hyperparameters of the model to improve efficiency, and they are also used to evaluate the ability of the model to generalize to new data.

c. The final performance of the model is evaluated using the test set. It is the final stage of the model development process and is used to evaluate how effectively the model generalizes to new inputs.

   Typically, the available data is divided into three sets: a test set (15%), a validation set (15%), and a training set (70%). This leaves enough data to evaluate the final performance of the model and allows an appropriate amount of data to be used for training and validation.

4.2.1 The Proposed Model (ASL. Model)

As figure 4. 1 provides the training and validation performance of the proposed model using the last. 20 epochs of training. The figure shows the loss and accuracy of the model which is named ASL model. The model achieved a training loss of 0. 01 percent and an accuracy of approximately 99 percent. 06%. Based on these results, there is evidence that the model is doing well for the training data set as evidenced by the low loss figures. and high accuracy

```
Epoch 1/20
40/40 [==============================] - 19s 121ms/step - loss: 3.5409 - accuracy: 0.1789 - val_loss: 2.5602 - val_accuracy: 0.2809
Epoch 2/20
40/40 [==============================] - 4s 108ms/step - loss: 2.0396 - accuracy: 0.4195 - val_loss: 1.6094 - val_accuracy: 0.5184
Epoch 3/20
40/40 [==============================] - 5s 125ms/step - loss: 1.2037 - accuracy: 0.6467 - val_loss: 0.8971 - val_accuracy: 0.7462
Epoch 4/20
40/40 [==============================] - 4s 105ms/step - loss: 0.7408 - accuracy: 0.7783 - val_loss: 0.5670 - val_accuracy: 0.8347
Epoch 5/20
40/40 [==============================] - 4s 107ms/step - loss: 0.4655 - accuracy: 0.8568 - val_loss: 0.5184 - val_accuracy: 0.8432
Epoch 6/20
40/40 [==============================] - 4s 100ms/step - loss: 0.3448 - accuracy: 0.8976 - val_loss: 0.2542 - val_accuracy: 0.9276
Epoch 7/20
40/40 [==============================] - 4s 104ms/step - loss: 0.2536 - accuracy: 0.9240 - val_loss: 0.2080 - val_accuracy: 0.9487
Epoch 8/20
40/40 [==============================] - 4s 101ms/step - loss: 0.1537 - accuracy: 0.9600 - val_loss: 0.1532 - val_accuracy: 0.9572
Epoch 9/20
40/40 [==============================] - 4s 100ms/step - loss: 0.1025 - accuracy: 0.9754 - val_loss: 0.1264 - val_accuracy: 0.9651
Epoch 10/20
40/40 [==============================] - 4s 100ms/step - loss: 0.0748 - accuracy: 0.9812 - val_loss: 0.1161 - val_accuracy: 0.9706
Epoch 11/20
40/40 [==============================] - 4s 101ms/step - loss: 0.0591 - accuracy: 0.9863 - val_loss: 0.0645 - val_accuracy: 0.9862
Epoch 12/20
40/40 [==============================] - 4s 95ms/step - loss: 0.0488 - accuracy: 0.9887 - val_loss: 0.0718 - val_accuracy: 0.9789
Epoch 13/20
40/40 [==============================] - 4s 101ms/step - loss: 0.0689 - accuracy: 0.9818 - val_loss: 0.1148 - val_accuracy: 0.9713
Epoch 14/20
40/40 [==============================] - 4s 100ms/step - loss: 0.0595 - accuracy: 0.9861 - val_loss: 0.0740 - val_accuracy: 0.9832
Epoch 15/20
40/40 [==============================] - 4s 101ms/step - loss: 0.0560 - accuracy: 0.9879 - val_loss: 0.0678 - val_accuracy: 0.9837
Epoch 16/20
40/40 [==============================] - 4s 106ms/step - loss: 0.0441 - accuracy: 0.9893 - val_loss: 0.0575 - val_accuracy: 0.9906
Epoch 17/20
40/40 [==============================] - 4s 100ms/step - loss: 0.0266 - accuracy: 0.9949 - val_loss: 0.0532 - val_accuracy: 0.9867
Epoch 18/20
40/40 [==============================] - 4s 96ms/step - loss: 0.0234 - accuracy: 0.9951 - val_loss: 0.0548 - val_accuracy: 0.9848
Epoch 19/20
40/40 [==============================] - 4s 111ms/step - loss: 0.0190 - accuracy: 0.9957 - val_loss: 0.0463 - val_accuracy: 0.9899
Epoch 20/20
40/40 [==============================] - 4s 102ms/step - loss: 0.0380 - accuracy: 0.9906 - val_loss: 0.0972 - val_accuracy: 0.9754
Time elapsed in seconds: 107.55182981491089
```

Figure 4.1: Loss and Accuracy to Every Epoch to ASL. Model.

While going through the model training process, the loss and accuracy are noted and then evaluated by a validation dataset. Validation set is used to help the model's performance during training and make sure that it has not memorized the training data and is not fitting it too well. Here are the outcomes of the validation, usually depicted in figures or graphs to show the evolution of the model's performance. Figure 4.2 and Figure 4.3 present the loss and accuracy values obtained during the training of the model, respectively, using the validation dataset.

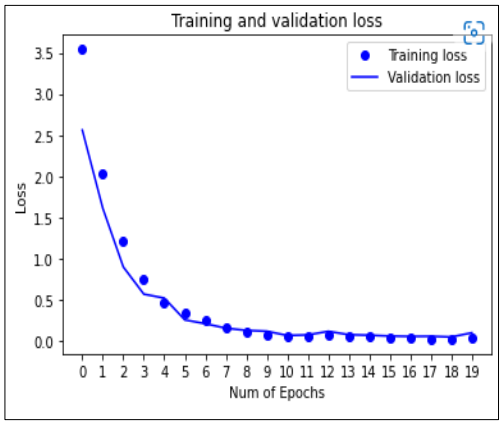

Figure 4.2: ASL. Model Training Loss, Validation Loss, 20 Epochs.

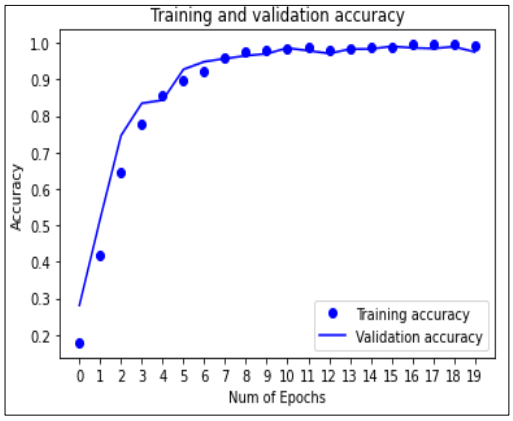

Figure 4.3: ASL.model Training Accuracy, Validation Accuracy, 20 Epochs.

Overfitting is a usual issue in the ML and it happens when the model is trained too much on good on the training data and weak on unseen data, by memorizing the training data. This can happen where the model occurs when the decision function is too complex or when it has been learned from a dataset that is too small or contains too much noise. The major issue that can come from overfitting is that of generalization since it ends up being poor. The model will give low accuracy on new data that it has not encountered before. In the case described, the impression is that the model gets good accuracy on the training set but its The accuracy is relatively lower on the validation data. This disparity between the training set accuracy and overfitting is shown by validation accuracy. In response to this issue, you could It is suggested to try and use techniques like regularization or early stopping to decrease the model's complexity. to stop a specific iteration of the model or to stop the model from training altogether. Matplotlib is a library used in data visualization. In the case described, t was used to come up with a chart that would highlight the monitoring of the results of the training and validation process in terms of loss and accuracy. From the chart it is clearly seen that the loss of the size of the validation and training datasets decreases with each iteration, meaning that the model is learning well. model is meeting the intended objectives. It also shows that the validation accuracy of the model increases to reach 99.86%, which is an excellent percentage. Finally, it seems that the model took 1.45 seconds to test the model with the testing dataset, and that it achieved a testing accuracy of 99.97% and a

testing loss of 0.44%. This indicates that the model was able to classify 29000 pictures belonging to 29 categories correctly. As shown in table 4.1 and table 4.2.

Table 4.1: Model Evaluation to Proposal Model (ASL. Model).

Model Evaluation	Tr.Acc	Tr.L	V.Acc	V.L	Te.L	Te.Acc
Results %	100	00.01	99.86	1.9	00.44	99.97

Table 4.2: Time Elapsed in Seconds Training and Testing to ASL. Model

Time	Tr. T	Te.T	To.T
Time elapsed in seconds	107.5518	1.4525	109.0043

4.2.2 VGG16

Dropout layers are utilized among layers in the model VGG16, which uses 16 convolutional layers before moving on to a fully connected hidden layer. This prevents the network from becoming over-allocated. The loss is measured as $3.0740e^{-4}$, and this is shown in Figure 4.4 and Figure 4.5. The training accuracy was 100%.

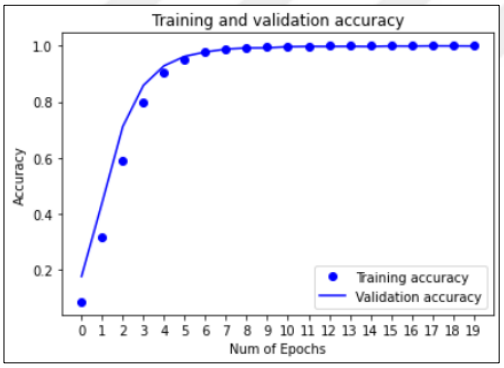

Figure 4.4: VGG16 Training Accuracy Validation Accuracy, 20 Epochs.

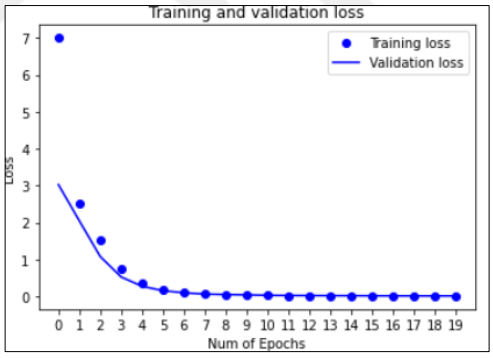

Figure 4.5: VGG16 Training Loss Validation Loss, 20 Epochs.

After validating the VGG16 model using the validation dataset, the validation accuracy is measured, and the model has achieved an accuracy of 100% and a loss of .0081%. This is shown in a table 4.3 and table 4.4.

Table 4.3: Model Evaluation to Model VGG16.

Model Evaluation	Tr.Acc	Tr.L	V.Acc	V.L	Te.L	Te.Acc
Result%	100	00.21	99.88	8.1	1.00	99.82

Table 4.4: Time Elapsed in Seconds Training and Testing to VGG16 Model.

Time	Tr.T	Te.T	To.T
Time elapsed in seconds	360.2306	4.0057	364.2363

4.2.3 ResNet50

The number of layers is increased to 50 hidden layers in this algorithm while applying a number of different techniques to obtain higher accuracy, avoid over-compositing and reduce elapsed time.

We note from Figure 4.6 and Figure 4.7 that after using the dataset to train the ResNet 50 model, we were able to achieve training accuracy of 99.78% and for the loss one of 0.0114%.

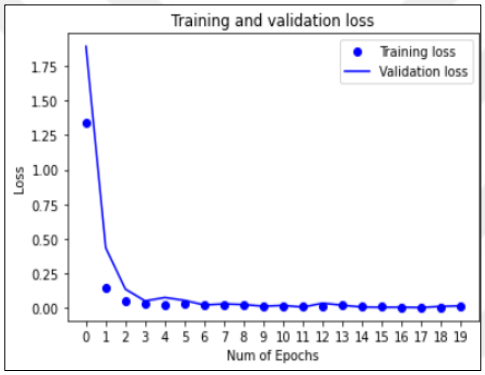

Figure 4.6: ResNet Training Loss Validation Loss, 20 Epochs.

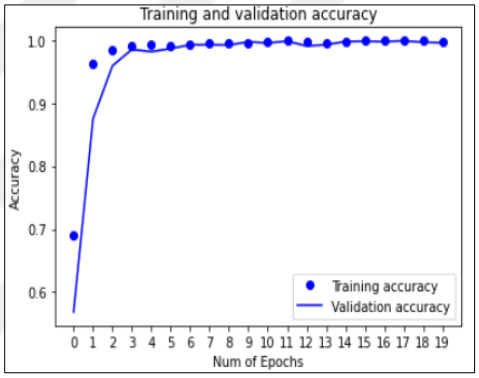

Figure 4.7: ResNet Training Accuracy, Validation Accuracy, 20 Epochs.

After validating the ResNet model using the validation dataset, the validation accuracy is measured, and the model has achieved an accuracy of 99.92 % and a loss of 0.0021. This is shown in a table 4.5 and table 4.6.

Table 4.5: Model Evaluation to ResNet Model.

Model Evaluation	Tr.Acc	Tr.L	V.Acc	V.L	Te.L	Te.Acc
Results%	100	00.21	99.88	8.1	1.00	99.82

Table 4.6: Time Elapsed in Seconds Training and Testing to ResNet Model.

Time	Tr.T	Te.T	To.T
Time elapsed in seconds	354.7519	5.4411	360.193

4.2.4 MobileNet

The model, which has less layers, is developed using MobileNet version 1, a lightweight deep convolutional neural network that is much lower in size and performs much faster than many standard neural network models. The data are trained on the same volume of data as those used in our model to ensure accurate comparison. The data are trained with a batch size of 128 and 20 epochs.

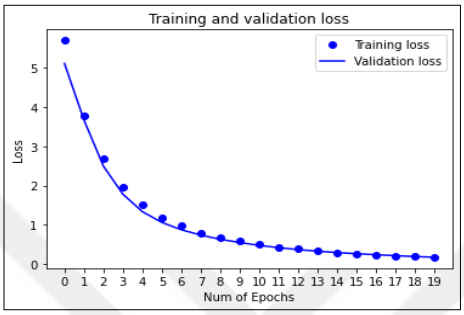

Figure 4.8: MobileNet Training Loss, Validation Loss, 20 Epochs.

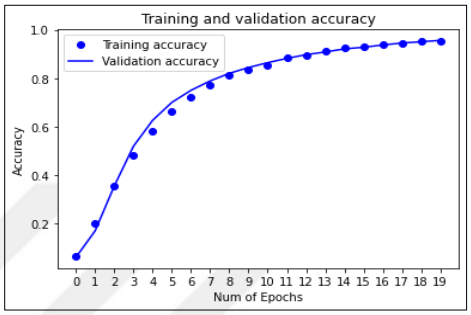

Figure 4.9: MobileNet Training Accuracy Validation Accuracy, 20 Epochs

We notice from Figure 4.8 and Figure 4.9 that following the dataset's training of the MobileNet model, we attained a training accuracy of 100% and a loss accuracy of 0.11 and we prevented over-fitting using data augmentation.

After validating the model using the validation dataset, the validation accuracy is measured, and the model has achieved an accuracy of 97.75 % and a loss of 1.23%. This is shown in a table 4.7 and table 4.8.

Table 4.7: Result the Model Evaluation to MobileNet.

Model Evaluation	TrAcc	TrL	VAcc	V.L	TeL	TeAcc
Results%	100	00.11	99.75	1.23	1.44	99.68

Table 4.8: Time Elapsed in Seconds Training and Testing to MobileNet Model.

Time	TrT	TeT	ToT
Time elapsed in seconds	134.5369	1.6453	136.1822

4.2.5 InceptionV3

The inceptionV3 consists of 22 layers, and as usual, precautions were taken against overfitting and using other techniques or a different working mechanism for speed and access to the highest accuracy.

After training the model with 20 epochs and 128 as batch size, in the training set, the model had an accuracy of 99.79% and a loss of 4.46%. Figures 4.10 and 4.11 illustrate the development of training accuracy and loss.

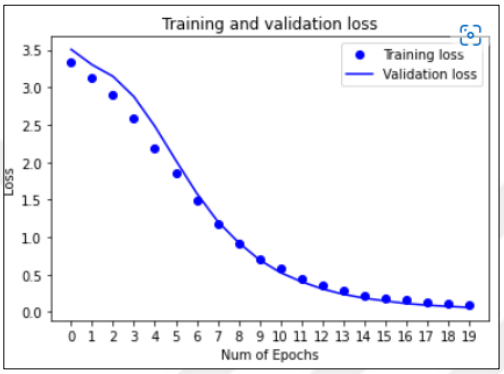

Figure4.10: Inception Training Loss Validation Loss, 20 Epochs.

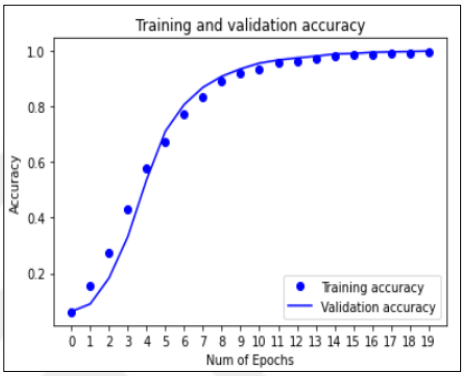

Figure 4.11: Inception Training Accuracy Validation Accuracy, 20 Epochs.

After validating the InceptionV3 model using the validation dataset, the validation accuracy is measured, and the model has achieved an accuracy of 99.75% and a loss of 6.90%. This is shown in a table 4.9 and table 4.10.

Table 4.9: Model Evaluation to Model Inceptionv3.

Model Evaluation	Tr.Acc	Tr.L	V.Acc	V.L	Te.L	Te.Acc
Results%	99.79	4.46	99.75	6.09	6.9	99.51

Table 4.10: Time Elapsed in Seconds Training and Testing to Incpetionv3 Model.

Time	Tr.T	Te.T	To.T
Time elapsed in seconds	231.1987	3.4684	238.1955

4.2.6 Xception

The Xception model is created from several layers After training the model with 20 epochs and 128 as batch size, a 100% accuracy rate and a 0.01% loss in the training set were attained by the model. Figures 4.12 and 4.13 depict the development of training accuracy and loss.

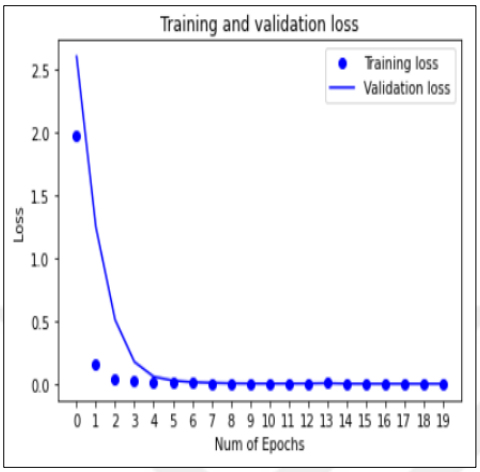

Figure 4.12:Xception Training Loss, Validation Loss, 20 Epochs.

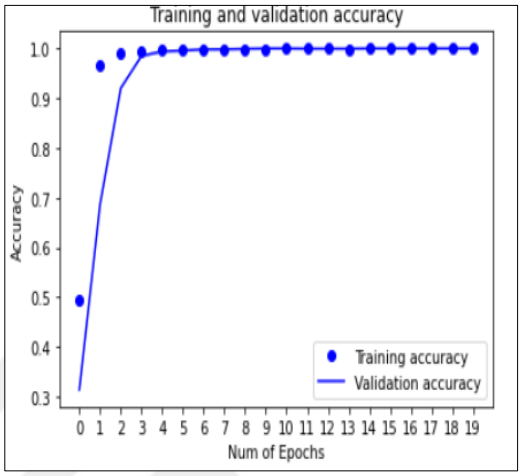

Figure 4.13:Xception Training Accuracy, Validation Accuracy, 20 Epochs.

After validating the Xception model using the validation dataset, the validation accuracy is measured, and the model has achieved an accuracy of 99.18% and a loss of 0.0320. This is shown in a table 4.11 and table 4.12.

Table 4.11: Mosel Evaluation to Xception Model.

Model Evaluation	Tr.Acc	Tr.L	V.Acc	V.L	Te.L	Te.Acc
Results%	100	00.01	99.98	0.13	0.04	100

Table 4.52: Time Elapsed in Seconds Training and Testing to Xcpetion Model.

Time	Tr.T	Te.T	To.T
Time elapsed in seconds	390.9375	3.4486	394.3861

4.3 TESTING EACH MODEL

In this case, the proposed model (ASL.model) was compared to a number of previously trained models, including VGG16, ResNet, MobileNet, Inception, and Xception. The same

volume of data was used in the training, testing, and validation phases of each model in order to create a strong basis for comparison. Data augmentation techniques were used in order to prevent overfitting and further improve the performance of the models. The same data set, the same partitioning, and the same classification categories were used to evaluate the models. In order to maintain high accuracy in a shorter period of time, time-related aspects were also taken into account.

4.4 RESULT AND DISCUSSION

Models were trained, verified and tested on an appropriate dataset; Process accuracy, loss and time performance were recorded. Model testing accuracy rates ranged from 100% to 99.68%, but they were not all the same. The Inception model had the lowest test accuracy, at 99.51%, while the Xception and VGG16 models had the highest test accuracy rates, at 100%. Despite these differences in testing accuracy, all of the models had relatively high testing accuracies, as can be seen in Table 4.13.

Table 4.63: Accuracy of All Models.

Model	Tr.Acc:	V.Acc	Te.Acc:
Xception	100.00%	99.98%	100.00%
Inception	99.79%	99.75%	99.51%
ResNet50	99.97%	99.92%	99.87%
VGG16	100.00%	98.95%	100.00%
MobileNet	100.00%	99.75%	99.68%
ASL.model	100.00%	99.86%	99.97%

In terms of Loss, Xception and ASL.model was ranked first with testing loss equal to 00.01%and the Inception ranked last with testing loss 4.46% (as shown in Table 4.14).

Table 4.74: Loss of All Models.

Model	Tr.L	V.L	Te .L
Xception	00.01%	00.13%	00.04%
Inception	4.46%	6.09%	6.90%
ResNet	0.09%	0.21%	0.39%
VGG16	0.02%	0.045%	0.02%
MobileNet	0.11%	1.23%	1.44%
ASL.model	0.01%	0.19%	0.44%

In term of Time performance, ASL.model was ranked first with testing time is equal to 1.4525 seconds and Resnet was ranked last with testing time 5.4411seconds.

Table 4.15 shows us all time to all model's training and testing times we got the least time model MobileNet was ranked last with testing time 136.18 second.

Table 4.85: Time Performance of All Models.

Model	Tr.T(sec)	Te.T(sec)	To.T (sec)
Xception	390.9375	3.4486	394.3861
Inception	231.1987	3.4684	234.6671
Resnet	354.7519	5.4411	360.193
VGG16	360.2306	4.0057	364.2363
MobileNet	134.53689	1.6453	136.1822

Figure 4.14 displays the inceptionV3 model's accuracy, recall, F1-score, and support for the twenty-nine classes that it was used to classify the data points from.

	precision	recall	f1-score	support
A	0.9923	1.0000	0.9961	129
B	1.0000	1.0000	1.0000	120
C	1.0000	1.0000	1.0000	132
D	1.0000	1.0000	1.0000	142
E	1.0000	0.9856	0.9928	139
F	1.0000	1.0000	1.0000	132
G	1.0000	0.9921	0.9960	127
H	0.9921	1.0000	0.9960	125
I	0.9843	0.9843	0.9843	127
J	0.9915	1.0000	0.9957	117
K	0.9919	1.0000	0.9960	123
L	1.0000	1.0000	1.0000	121
M	0.9862	0.9931	0.9896	144
N	0.9922	1.0000	0.9961	127
O	1.0000	1.0000	1.0000	109
P	1.0000	1.0000	1.0000	104
Q	1.0000	1.0000	1.0000	136
R	0.9920	0.9920	0.9920	125
S	0.9922	0.9922	0.9922	129
T	0.9839	0.9919	0.9879	123
U	0.9926	0.9926	0.9926	136
V	0.9823	1.0000	0.9911	111
W	1.0000	1.0000	1.0000	111
X	0.9928	0.9787	0.9857	141
Y	1.0000	0.9848	0.9924	132
Z	1.0000	0.9859	0.9929	142
del	1.0000	1.0000	1.0000	138
nothing	1.0000	1.0000	1.0000	116
space	0.9929	0.9929	0.9929	140
accuracy			0.9951	3698
macro avg	0.9951	0.9954	0.9953	3698
weighted avg	0.9952	0.9951	0.9951	3698

Figure 4.14: Classification Report to Inception Model.

Figure 4.15 shows the same metrics of MobileNet model the twenty-nine classes that the model is used for classifying them from the dataset.

	precision	recall	f1-score	support
A	1.0000	1.0000	1.0000	129
B	1.0000	1.0000	1.0000	120
C	1.0000	1.0000	1.0000	132
D	1.0000	1.0000	1.0000	142
E	1.0000	1.0000	1.0000	139
F	1.0000	1.0000	1.0000	132
G	1.0000	1.0000	1.0000	127
H	1.0000	1.0000	1.0000	125
I	1.0000	0.9921	0.9960	127
J	1.0000	1.0000	1.0000	117
K	0.9919	1.0000	0.9960	123
L	1.0000	1.0000	1.0000	121
M	0.9931	1.0000	0.9965	144
N	1.0000	0.9921	0.9960	127
O	0.9909	1.0000	0.9954	109
P	1.0000	1.0000	1.0000	104
Q	1.0000	1.0000	1.0000	136
R	1.0000	0.9760	0.9879	125
S	0.9922	0.9845	0.9883	129
T	1.0000	1.0000	1.0000	123
U	0.9574	0.9926	0.9747	136
V	0.9909	0.9820	0.9864	111
W	0.9911	1.0000	0.9955	111
X	1.0000	0.9929	0.9964	141
Y	1.0000	1.0000	1.0000	132
Z	1.0000	1.0000	1.0000	142
del	1.0000	0.9928	0.9964	138
nothing	1.0000	1.0000	1.0000	116
space	1.0000	1.0000	1.0000	140
accuracy			0.9968	3698
macro avg	0.9968	0.9967	0.9967	3698
weighted avg	0.9968	0.9968	0.9968	3698

Figure 4.15: Classification Report to MobiNet Model.

Twenty-nine classes from the dataset that were classified using the Xception model are shown in Figure 4.16 along with their metrics.

	precision	recall	f1-score	support
A	1.0000	1.0000	1.0000	129
B	1.0000	1.0000	1.0000	120
C	1.0000	1.0000	1.0000	132
D	1.0000	1.0000	1.0000	142
E	1.0000	1.0000	1.0000	139
F	1.0000	1.0000	1.0000	132
G	1.0000	1.0000	1.0000	127
H	1.0000	1.0000	1.0000	125
I	1.0000	1.0000	1.0000	127
J	1.0000	1.0000	1.0000	117
K	1.0000	1.0000	1.0000	123
L	1.0000	1.0000	1.0000	121
M	1.0000	1.0000	1.0000	144
N	1.0000	1.0000	1.0000	127
O	1.0000	1.0000	1.0000	109
P	1.0000	1.0000	1.0000	104
Q	1.0000	1.0000	1.0000	136
R	1.0000	1.0000	1.0000	125
S	1.0000	1.0000	1.0000	129
T	1.0000	1.0000	1.0000	123
U	1.0000	1.0000	1.0000	136
V	1.0000	1.0000	1.0000	111
W	1.0000	1.0000	1.0000	111
X	1.0000	1.0000	1.0000	141
Y	1.0000	1.0000	1.0000	132
Z	1.0000	1.0000	1.0000	142
del	1.0000	1.0000	1.0000	138
nothing	1.0000	1.0000	1.0000	116
space	1.0000	1.0000	1.0000	140
accuracy			1.0000	3698
macro avg	1.0000	1.0000	1.0000	3698
weighted avg	1.0000	1.0000	1.0000	3698

Figure 4.16: Classification Report to Xception Model.

The same metrics of the (ASL.model) are displayed in Figure 4.17 with the same number of classes.

	precision	recall	f1-score	support
A	1.0000	1.0000	1.0000	129
B	1.0000	1.0000	1.0000	120
C	1.0000	1.0000	1.0000	132
D	1.0000	1.0000	1.0000	139
E	1.0000	1.0000	1.0000	132
F	1.0000	1.0000	1.0000	127
G	1.0000	1.0000	1.0000	125
H	1.0000	1.0000	1.0000	127
I	1.0000	1.0000	1.0000	117
J	1.0000	1.0000	1.0000	123
K	1.0000	1.0000	1.0000	121
L	1.0000	1.0000	1.0000	144
M	1.0000	1.0000	1.0000	127
N	1.0000	0.9908	0.9954	109
O	1.0000	1.0000	1.0000	136
Q	1.0000	1.0000	1.0000	104
P	1.0000	1.0000	1.0000	129
S	1.0000	1.0000	1.0000	123
T	1.0000	1.0000	1.0000	125
R	1.0000	1.0000	1.0000	136
U	1.0000	1.0000	1.0000	111
V	1.0000	1.0000	1.0000	111
W	1.0000	1.0000	1.0000	142
Z	1.0000	1.0000	1.0000	141
X	1.0000	1.0000	1.0000	142
Y	0.9925	1.0000	0.9962	132
del	1.0000	1.0000	1.0000	138
nothing	1.0000	1.0000	1.0000	116
space	1.0000	1.0000	1.0000	140
accuracy			0.9997	3698
macro avg	0.9997	0.9997	0.9997	3698
weighted avg	0.9997	0.9997	0.9997	3698

Figure 4.17: Classification Report to ASL Model.

With the same number of classes, Figure 4.17 displays the same metrics as (VGG16).

	precision	recall	f1-score	support
A	1.0000	1.0000	1.0000	129
B	1.0000	1.0000	1.0000	120
C	1.0000	1.0000	1.0000	132
D	1.0000	1.0000	1.0000	142
E	1.0000	1.0000	1.0000	139
F	1.0000	1.0000	1.0000	132
G	1.0000	1.0000	1.0000	127
H	1.0000	1.0000	1.0000	125
I	1.0000	1.0000	1.0000	127
J	1.0000	1.0000	1.0000	117
K	1.0000	1.0000	1.0000	123
L	1.0000	1.0000	1.0000	121
M	1.0000	1.0000	1.0000	144
N	1.0000	1.0000	1.0000	127
O	1.0000	1.0000	1.0000	109
P	1.0000	1.0000	1.0000	104
Q	1.0000	1.0000	1.0000	136
R	1.0000	1.0000	1.0000	125
S	1.0000	1.0000	1.0000	129
T	1.0000	1.0000	1.0000	123
U	1.0000	1.0000	1.0000	136
V	1.0000	1.0000	1.0000	111
W	1.0000	1.0000	1.0000	111
X	1.0000	1.0000	1.0000	141
Y	1.0000	1.0000	1.0000	132
Z	1.0000	1.0000	1.0000	142
del	1.0000	1.0000	1.0000	138
nothing	1.0000	1.0000	1.0000	116
space	1.0000	1.0000	1.0000	140
accuracy			1.0000	3698
macro avg	1.0000	1.0000	1.0000	3698
weighted avg	1.0000	1.0000	1.0000	3698

Figure 4.18: Classification Report to VGG16 Model.

Figure 4.19 shows the precision, recall, F1-score and support of (ResNet) model the twenty-nine classes that the model is used for classifying them from the dataset.

	precision	recall	f1-score	support
A	1.0000	1.0000	1.0000	183
B	1.0000	1.0000	1.0000	192
C	1.0000	1.0000	1.0000	187
D	1.0000	1.0000	1.0000	181
E	1.0000	1.0000	1.0000	182
F	1.0000	1.0000	1.0000	196
G	1.0000	1.0000	1.0000	186
H	1.0000	1.0000	1.0000	195
I	0.9947	1.0000	0.9973	187
J	0.9890	0.9945	0.9917	181
K	1.0000	1.0000	1.0000	189
L	1.0000	1.0000	1.0000	211
M	1.0000	1.0000	1.0000	188
N	1.0000	1.0000	1.0000	180
O	1.0000	1.0000	1.0000	212
P	1.0000	1.0000	1.0000	215
Q	1.0000	1.0000	1.0000	194
R	1.0000	1.0000	1.0000	196
S	1.0000	0.9950	0.9975	202
T	0.9945	1.0000	0.9972	181
U	1.0000	1.0000	1.0000	191
V	1.0000	0.9824	0.9911	170
W	0.9849	1.0000	0.9924	196
X	1.0000	1.0000	1.0000	203
Y	1.0000	0.9895	0.9947	191
Z	1.0000	1.0000	1.0000	181
del	1.0000	1.0000	1.0000	188
nothing	1.0000	1.0000	1.0000	191
space	1.0000	1.0000	1.0000	198
accuracy			0.9987	5547
macro avg	0.9987	0.9987	0.9987	5547
weighted avg	0.9988	0.9987	0.9987	5547

Figure 4.19: Classification Report to ResNet Model.

When evaluating our suggested model's performance in terms of accuracy, loss, and time (ASL.model) with the other five fine-tuned pre-trained models, we found the VGG16 model is considered one of the best due to its high testing accuracy (100%) and testing time performance (1.45 seconds).

5. CONCLUSION AND FUTURE RESEARCH DIRECTION

5.1 CONCLUSION

In this thesis, we focused on the classification of utilizing convolutional neural networks (CNNs) for sign language. We provided an overview of artificial intelligence, machine learning, and the problem of sign language classification. We also discussed the scope and limitations of the thesis, as well as the objectives and significance of the study.

The concept of CNNs is described, and the kind of multiple layers that are often employed in these networks: activation, pooling, full-connected, convolution and batch normalization layers. We also talked about feedforward and backpropagation method of training CNN. Below is the summary of some of the network architectures that have been under analysis in this paper, namely, VGG16, Inception-v3, Xception, ResNet, and MobileNet. As for the dataset, we described it, the methods and tools that were used for data processing and model assessment. We also discussed the previous works that have been carried out on the classification of sign languages and how our work contributes to it.
Last of all, we explain the method of our research, the tools and language used, the dataset applied, the network structure of the proposed model, and the format of the images. We evaluated our proposed model (ASL. model) against five pre-trained ImageNet models: VGG16, ResNet, MobileNet, Inception-v3, Xception. One thousand of 29,000 sign language images were in each of the categories and were used in the dataset. This is done so that time and accuracy can be easily compared and our experimental results are presented in tables. The evaluation exposed that the Xception and VGG16 models recorded the most elevated level of accuracy of 100%, while that of the proposed model got to 99%. 97%. The model that has been tested in less than 1 which is our model. 45 seconds, it also recorded the best timingperformer.
In summary, the results presented in this section prove the efficiency of the proposed model for sign language images' classification and show the advantages of the model in comparison to the others.

5.2 FUTURE WORK

Future research might build on this study's findings in a number of different ways. One possibility is to explore other deep learning architectures or techniques that may improve the performance of the models. For example, we may experiment with various neural network topologies, including RNNs or transformer models, or we could experiment with different techniques for training the models, such as transfer learning or reinforcement learning. Another possibility is to expand the dataset and try using more diverse or larger datasets to see if this affects the performance of the models. Additionally, Investigating the usage of different sorts of data might be fascinating of data, such as video or audio, for hand sign language classification and see how well the models perform in these settings.

REFERENCES

[1] A. Aljabar, "BISINDO (Bahasa Isyarat Indonesia) Sign Language Recognition Using CNN and LSTM," Advances in Science, Technology and Engineering Systems Journal, vol. 5, no. 5, pp. 282–287, 2020.

[2] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," Expert Syst Appl, vol. 182, p. 115657, 2021.

[3] S. Masood, H. C. Thuwal, and A. Srivastava, "American sign language character recognition using convolution neural network," in Smart Computing and Informatics, Springer, 2018, pp. 403–412.

[4] K. Tripathi and N. B. G. C. Nandi, "Continuous Indian sign language gesture recognition and sentence formation," Procedia Comput Sci, vol. 54, pp. 523–531, 2015.

[5] J. Singha and K. Das, "Indian sign language recognition using eigen value weighted euclidean distance-based classification technique," arXiv preprint arXiv:1303.0634, 2013.

[6] F. J. M. Shamrat, S. Chakraborty, M. M. Billah, M. Kabir, N. S. Shadin, and S. Sanjana, "Bangla numerical sign language recognition using convolutional neural networks," Indonesian Journal of Electrical Engineering and Computer Science, vol. 23, no. 1, pp. 405–413, 2021.

[7] P. J. Braspenning, F. Thuijsman, and A. J. M. M. Weijters, Artificial neural networks: an introduction to ANN theory and practice, vol. 931. Springer Science & Business Media, 1995.

[8] H. D. Nguyen, N. D. Dao, and M. Shin, "Machine learning-based prediction for maximum displacement of seismic isolation systems," Journal of Building Engineering, vol. 51, p. 104251, 2022.

[9] Naseri, Raghda Awad Shaban, Ayça Kurnaz, and Hameed Mutlag Farhan. "Optimized face detector-based intelligent face mask detection model in IoT using deep learning approach." Applied Soft Computing 134 (2023): 109933.

[10] R. L. Lawrence and A. Wright, “Rule-based classification systems using classification and regression tree (CART) analysis,” Photogramm Eng Remote Sensing, vol. 67, no. 10, pp. 1137–1142, 2001.

[11] Khalef, Dhirgham Atia, Ayca Kurnaz Turkben, Hameed Mutlag Farhan, and Raghda Awad Shaban Naseri. "Optic disc segmentation in human retina images with meta heuristic optimization." In 2022 International Conference on Artificial Intelligence of Things (ICAIoT), pp. 1-6. IEEE, 2022.

[12] M. Al-Qurishi, T. Khalid, and R. Souissi, “Deep learning for sign language recognition: Current techniques, benchmarks, and open issues,” IEEE Access, 2021.

[13] R. L. Lawrence and A. Wright, “Rule-based classification systems using classification and regression tree (CART) analysis,” Photogramm Eng Remote Sensing, vol. 67, no. 10, pp. 1137–1142, 2001.

[14] M. Usama *et al.*, “Unsupervised machine learning for networking: Techniques, applications and research challenges,” IEEE access, vol. 7, pp. 65579–65615, 2019.

[15] D. A. van Dyk and X.-L. Meng, “The art of data augmentation,” Journal of Computational and Graphical Statistics, vol. 10, no. 1, pp. 1–50, 2001.

[16] Hammoodi, Sabah Abdul Rasool, Kamal Turki Aftan, and Mohammed Rhael Ali. "Management of Hydatid cysts of parotid glands." Journal of stomatology, oral and maxillofacial surgery 124.6 (2023): 101465

[17] T. Jayalakshmi and A. Santhakumaran, “Statistical normalization and back propagation for classification,” International Journal of Computer Theory and Engineering, vol. 3, no. 1, pp. 1793–8201, 2011.

[18] S. Masood, H. C. Thuwal, and A. Srivastava, “American sign language character recognition using convolution neural network,” in Smart Computing and Informatics, Springer, 2018, pp. 403–412.

[19] S. Johnny and S. J. Nirmala, “Sign Language Translator Using Machine Learning,” SN Comput Sci, vol. 3, no. 1, pp. 1–6, 2022.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[21] Ali, Mohammed Rhael, and Elham Hazeim Abdulkareem. "Efficiency of BTX-A in the alleviation of hemifacial pain." Journal of International Dental and Medical Research 13.1 (2020): 321-326.

[22] S. M. Sam, K. Kamardin, N. N. A. Sjarif, and N. Mohamed, “Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3,” Procedia Comput Sci, vol. 161, pp. 475–483, 2019.

[23] C. Szegedy *et al.*, “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[24] H. Pan, Z. Pang, Y. Wang, Y. Wang, and L. Chen, “A new image recognition and classification method combining transfer learning algorithm and mobileNet model for welding defects,” IEEE Access, vol. 8, pp. 119951–119960, 2020.

[25] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, “A novel image classification approach via dense-MobileNet models,” Mobile Information Systems, vol. 2020, 2020.

[26] X. Xia, C. Xu, and B. Nan, “Inception-v3 for flower classification,” in *2017 2nd* international conference on image, vision and computing (ICIVC), 2017, pp. 783–787.

[27] Ali, Mohammed Rhael, et al. "Botulinum Toxin-A For Management of Migraine: An Experience in Iraq." History of Medicine 9.2 (2023): 416-425.

[28] D. I. Swasono, H. Tjandrasa, and C. Fathicah, “Classification of tobacco leaf pests using VGG16 transfer learning,” in 2019 12th International Conference on Information & Communication Technology and System (ICTS), 2019, pp. 176–181.

[29] T. N. Abu-Jamie and S. S. Abu-Naser, “Classification of Sign-language Using VGG16,” 2022.