People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of 20 août 1955 of Skikda
Faculty Of Sciences
Department Of Informatics

Master's Dissertation

Arabic Sign Language Recognition System Using Deep Learning

Presented by
Bouakba Oumaima
Krouma Nour El Houda

Specialty
Master2 GLAA

Presented to the Jury:
Dr. Mawloud Mosbah as chairman
Dr. Lahsasana Adel as reviewer
Dr. Bougamouza Fateh as the supervisor

2022 - 2023

ACKNOWLEDGMENT

We would like to thank everyone who has helped make this thesis possible, every teacher or mentor whom work and ideas have helped us veer towards excellence and proficiency, everyone who has helped in the research, the writing ,the implementation of this work and more importantly fixing the errors we faced along the way.

We especially like to thank our colleagues, friends whom without this work would not be what it is today, the ones who helped us push through the obstacles and always strive to better our selves and our work.

And last but certainly not least, we would like to thank our families that provided the utmost support that made achieve the work you see before you today.

ABSTRACT

The aim of this thesis is to create an AI system that is capable if interpreting the user's Arabic sign language input gestures to their matching meaning.

The proposed system leverages the power of deep learning algorithms, specifically neural networks, for robust and high-performance object detection and recognition. The research includes the collection and preprocessing of a large-scale dataset with diverse data.

Experimental evaluations are conducted to assess the performance of the developed AI system, providing insights into the system's accuracy and detection capabilities across different object categories and varying environmental conditions. The thesis also investigates and compares other conventional techniques typically used in the domain of image recognition mainly Convolutional neural networks.

The results demonstrate that the developed AI system achieves competitive object detection and recognition performance, with high accuracy and real-time processing capabilities. The system shows potential for practical deployment in various domains and for other various sign languages.

Overall, this thesis contributes to the field of computer vision by presenting an effective AI system for detecting and translating Arabic sign language alphabet. The developed system paves the way for advancements in the field of education.

Keywords — Arabic sign language, Sign language recognition, Sign language translation, Artificial intelligence, Deep learning, Artificial neural networks, Convolutional neural networks

Contents

List of Figures

List of Tables

ABBREVIATIONS

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

ArSL Arabic Sign Language

SLR Sign Language Recognition

HMM Hidden Markov Model

NN Neural Network

DNN Deep Neural Network

CNN Convolutional Neural Network

RNN Recurrent Neural Network

ANN Artificial Neural Network

CV Computer Vision

ReLU Rectified Linear Unit

GeLU Gaussian Error Linear Unit

General Introduction

The primary objective of this thesis is to develop an advanced AI system that can accurately interpret Arabic sign language gestures and translate them into their corresponding meanings. To achieve this goal, the research relies on the utilization of deep learning algorithms, specifically neural networks, which have proven to be highly effective in object detection tasks.

The thesis encompasses various stages, starting with the collection and preprocessing of a comprehensive and diverse dataset that encompasses a wide range of sign language gestures. This dataset serves as the foundation for training and evaluating the AI system.

Throughout the research, extensive experimental evaluations are conducted to thoroughly assess the performance and capabilities of the developed AI system. These evaluations involve analyzing the accuracy and detection capabilities of the system across different object categories and under varying environmental conditions. Additionally, the thesis delves into the exploration and comparison of other conventional techniques commonly used in the domain of image recognition, with a particular focus on Convolutional Neural Networks (CNNs).

The obtained results from the experiments showcase the effectiveness and competitiveness of the developed AI system in object detection tasks. The system demonstrates high accuracy in recognizing and interpreting Arabic sign language gestures in real-time, indicating its potential for practical deployment in various domains and potentially for other sign languages as well.

The developed system opens up new avenues for advancements in the field of education, enabling improved communication and understanding for individuals with hearing impairments.

Chapter 1

Deep Learning

1.1 Introduction

In recent years, due to quick technological advancements and subsequently the enormous amount of accompanying data, it has become imperative that a new way of finding and managing this data in a quick and resource efficient way is made.

Of course one of the most efficient ways for treating large and redundant tasks and data is through the use of machines, but often due to the degree and difficulty of said tasks various and often complicated methods are used to make it possible for the computer to mimic or surpass human level capability.

In order to do that large and cumbersome scientific techniques have to be used. All these techniques fall under the domain of artificial intelligence, where the end goal is to teach the machine to behave in a desired way.

In recent years, machine learning has become more and more popular in research as well as being incorporated in a varying number of applications, including multimedia concept retrieval, image classification, video recommendation, social network analysis, text mining, and so forth.

Sometimes, the outputs obtained from machine learning are not sufficiently accurate or they do not meet the desired results, so the machines have to undergo a deeper level of learning to ameliorate the results, this process leads to the conveniently named deep

learning. So far we mentioned Artificial intelligence, Machine learning and deep learning, these three concepts are often used interchangeably to mean everything that falls under the umbrella of what artificial intelligence is, but before we go further with this thesis we should take some time to clarify each term.

1.2 Artificial intelligence

Artificial intelligence or AI for short is the field that englobes both Machine Learning (ML) and Deep Learning (DL), it concerns the science of recreating human intelligence in machines and making them able to do tasks that humans can. AI is the root and main domain of all other subsequent fields and domains that aim to train or teach machines how to reason rationally.

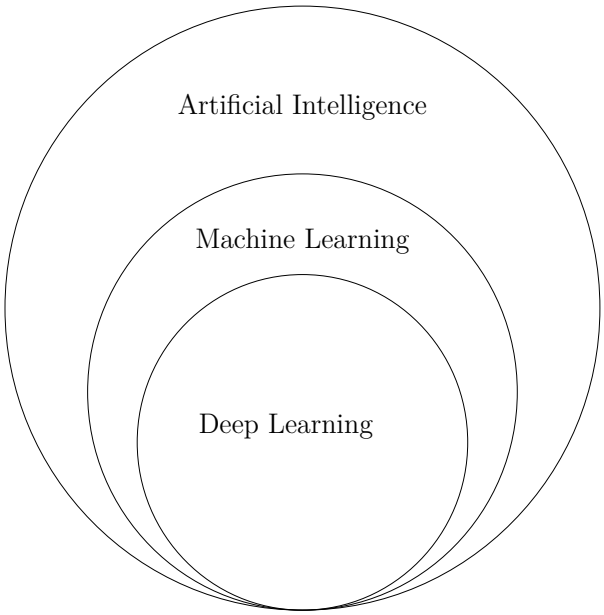

Figure 1.1: Venn diagram of the components of Artificial Intelligence

1.3 Machine learning

If we think of AI as the first layer then machine learning or ML would be the second layer relative to it. Much like the name suggests this field of study refers to building

algorithms and programs that allow the machine to learn and identify patterns without explicitly being programmed to. This works by giving the program a large sum of data referred to as training data and allowing the machine to learn from such data. ML is often used when it comes to complicated problems where it has a large number of inputs and may be difficult (and sometimes impossible) for a human to solve, in these cases a computer intervention would save a lot of time and resources, it is also used in making future predictions based on past data. The learning process differs based on the used approaches, types and algorithms.

Some of ML approaches are :

1.3.1 Supervised learning

Where the training data has with it the correct response that the computer should predict, the program later calculates its performance by referring to the correct responses also known as labels. In supervised learning, the input variables represented as X are mapped to output variables represented as Y by using an algorithm to learn the mapping function f.

$$Y = f(X)$$

The aim of the learning algorithm is to approximate the mapping function to predict the output (Y) for a new input (X). The error from the predictions made during training can be used to correct the output. Learning can be stopped when all the inputs are trained to get the targeted output [1].

1.3.2 Unsupervised learning

Unsupervised learning is when the training data does not have labels with it, and the computer must work by identifying patterns. Clustering and association are an example of problems that use unsupervised learning. K-means algorithm is an unsupervised algorithm typically used in clustering.

1.3.3 Hybrid Learning

As the name suggests, it works by using both supervised and unsupervised approaches, this mix of both approaches can sometimes give good results depending on the problem at hand.

1.3.4 Reinforcement learning

Works by incorporating some kind of dynamic feedback loop where the computer gets a kind of score depending on its performance and wishes to maximize said score.
The goal of every approach is to be able to create a model that can accurately solve problems and identify patterns.

1.4 Deep Learning

We've mentioned above that this field of data science and AI aims at mimicking the human brains reasoning, this mimicking reaches as far as trying to recreate what scientists deem the biological reason that allows humans to learn: neurons, these neurons are later connected and layered together to form neural networks this technique is used in what is known as deep learning; using neural networks to teach machines by feeding them a large number of data.
Deep learning (DL) for short, would be the third layer in relation to AI. As said above, DL concerns using (in most cases) artificial neural networks (ANN) to establish a model that can easily identify patterns.
The word deep refers to the amount of layers that go into making a model, as a one neuron or one neural network layer can not be sufficient (in most cases) to effectively or correctly solve any problems, therefore a numerous number of layers may be used to create an accurately working model.
DL models work by extracting features from given input, if we take an image recognition software for example, an input image which we wish to identify is fed to the program and then the program extracts information from every pixel of the image and runs these values

through multiple layers of the neural network in order to determine the best matching value is from given list of outputs (or in some cases : labels). In our thesis, we wish to create an ANN that is able to distinguish between the signs for the Arabic sign language.

1.4.1 Evolution of Deep Learning

As said previously, many DL applications rely on ANN, of course at first neural networks were not as nearly as developed as they are right now, the first generation of NN are whats known as perceptron; a simple supervised learning algorithm that works by receiving input and calculating output. Perceptons were not efficient in solving complicated problems as it is only good at learning linear solutions.

$$f(x) = 1 \text{ if } W.x + B > 0, \text{ else } 0$$

$$W : \text{vector of weights of input}$$

$$x : \text{vector of inputs}$$

$$B : \text{vector of bias}$$

So in order to improve upon these results, multilayer perceptons were introduced where the feedforward and backpropagation algorithm were used. The feedforward algorithm works by calculating activations of a layer by using the dot product of the weights matrix and inputs vector, and then passing those activations to the next layer. backpropagation works by the same principle but instead of the first layer to the next one, it calculates the activation of the last layer and passes the values to the previous layer. backpropagation allows the algorithm to correct itself by adjusting the randomly set values of weights and biases.

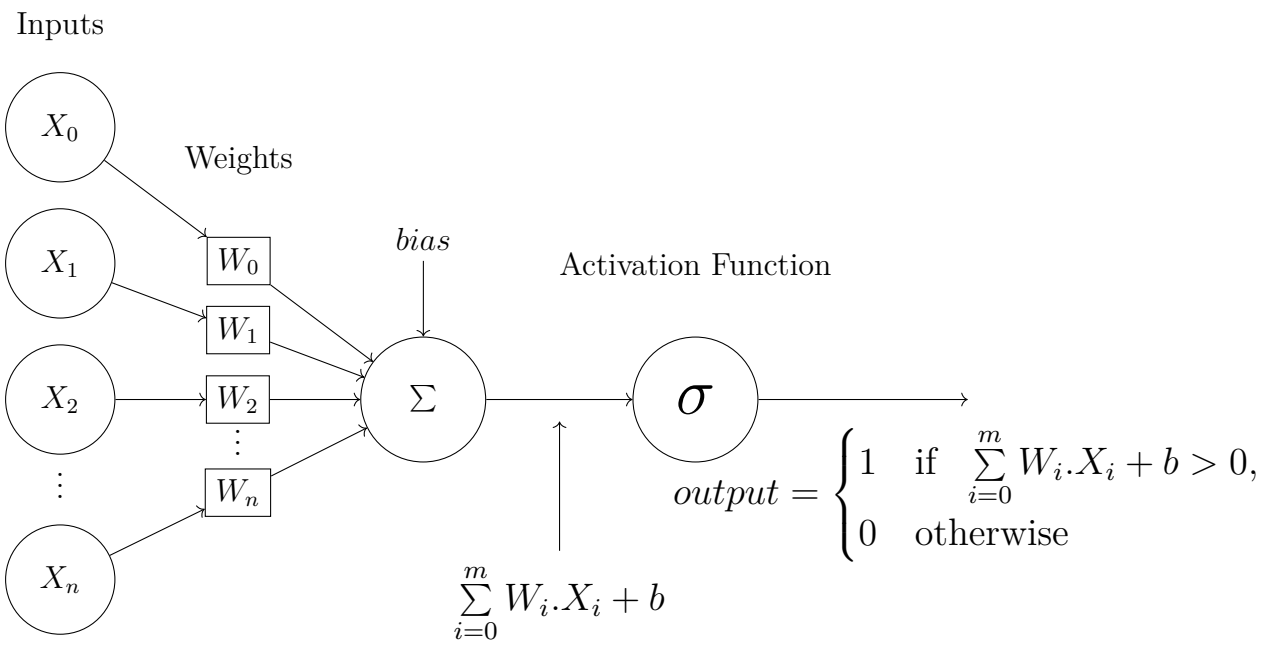

Figure 1.2: An illustration of the general architecture of a Perceptron.

Calculating the activation often results in very large values so a method known as normalizing is used to put all the values in a range between 0 and 1. Normalizing is used through mathematical functions most infamously the Sigmoid function (noted by σ):

$$\sigma(x) = \frac{1}{(1 + e^{-}x)}$$

Other activation functions are used such as ReLU or GeLU among many others. A lot of work and parameters go into configuring a neural network for achieving a good result accuracy, parameters like the learning rate that determine how much of a step the network should make in every iteration (epoch). Additional improvement may look at the most efficient way to initialize the weights to achieve better results quickly, rather than initializing them randomly. But the most important part lies at determining the efficient way of evaluating the systems performance and making adjustments based on said evaluation, The use of a cost function is what allows for the system to be able to make said adjustments. The cost function calculates the error which is the difference between

the correct answer (label in case of supervised learning) and the prediction made by the neural network

$$\text{error} = (\text{label - prediction})$$

A good neural network is one where the errors are very low, which can be accomplished by backpropagating the values to the previous layer to adjust the weights and bias. After calculating the error, the system must determine how to adjust the weights and biases, as some weights need to be bigger and others need to be smaller, this determination is done most notably by using gradient descent ; a mathematical technique used to determine the minimum or a near minimum (local minimum) of the cost function, and then providing back to the program how should the values be changed, either by increasing or decreasing them.

Of course all of this is just the bare bone structure of ANN, in order to achieve better and faster results other types of mathematical calculations must be implemented as well as different types of ANN.

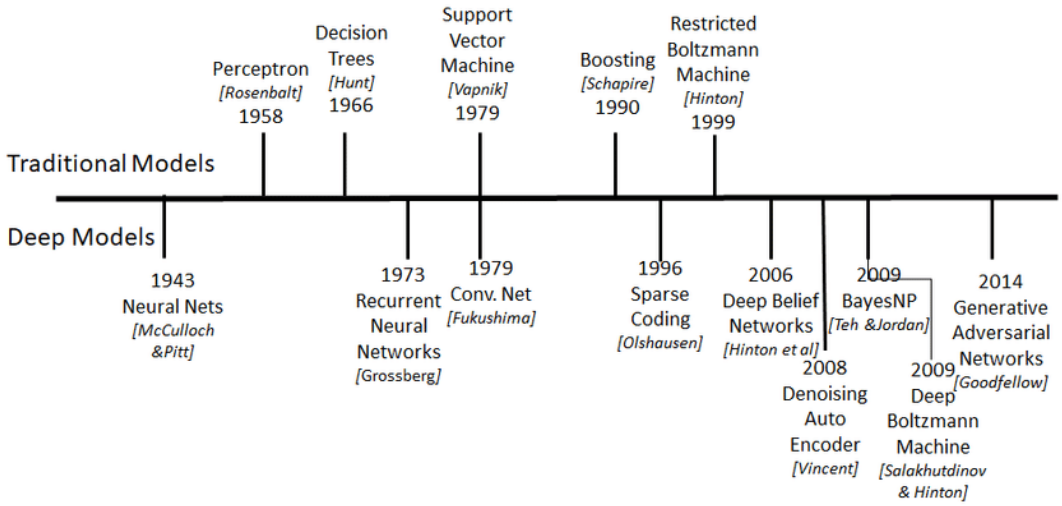

Figure 1.3: Evolution of Deep Models [1]

The performance of neural network classifiers using deep learning improves on a large scale with an increased quantity of data when compared to traditional learning methods.

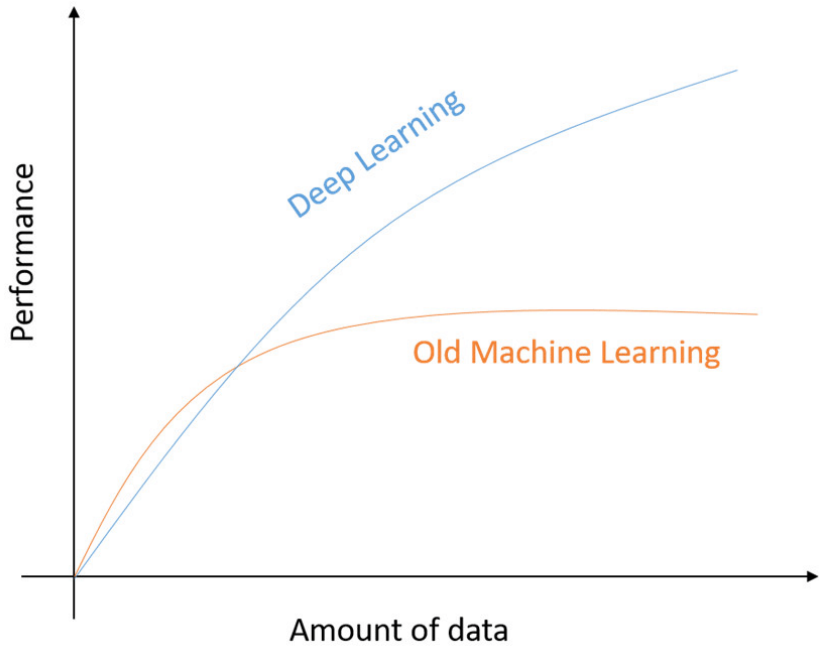

Figure 1.4: The performance of traditional machine learning algorithms and deep learning algorithms [2]

1.5 Computer Vision

One of the fields of image processing research that has attracted a lot of attention recently is computer vision (CV). It is an academic field that studies the realization of vision using computers and is also an artificial intelligence field that allows computers and systems to extract useful information from digital images, videos and other visual data, and act and make recommendations based on that information. If AI gives machines the ability to think, computer vision gives them the ability to see, observe, and recognize. The goal of CV is to give computers abilities comparable to those of the human eye. The objective is to create computer software that uses still image or video data to perform similarly to or better than human vision [3].

The CV journey began in 1960 when Larry Roberts, the father of computer vision, proposed 3D geometrical information extraction from 2D perspective polyhedral concepts in his Ph.D. thesis at MIT. At the initial stage, CV shows how computational models can be used to extract important information from digital images. Elaborating dual goals, vision is used as an autonomous system for the engineering point of view just like a

human can perform a visual task, while computational models are applied in the human biological system for detecting symptoms of diseases in the body [4].

1.6 Deep learning techniques

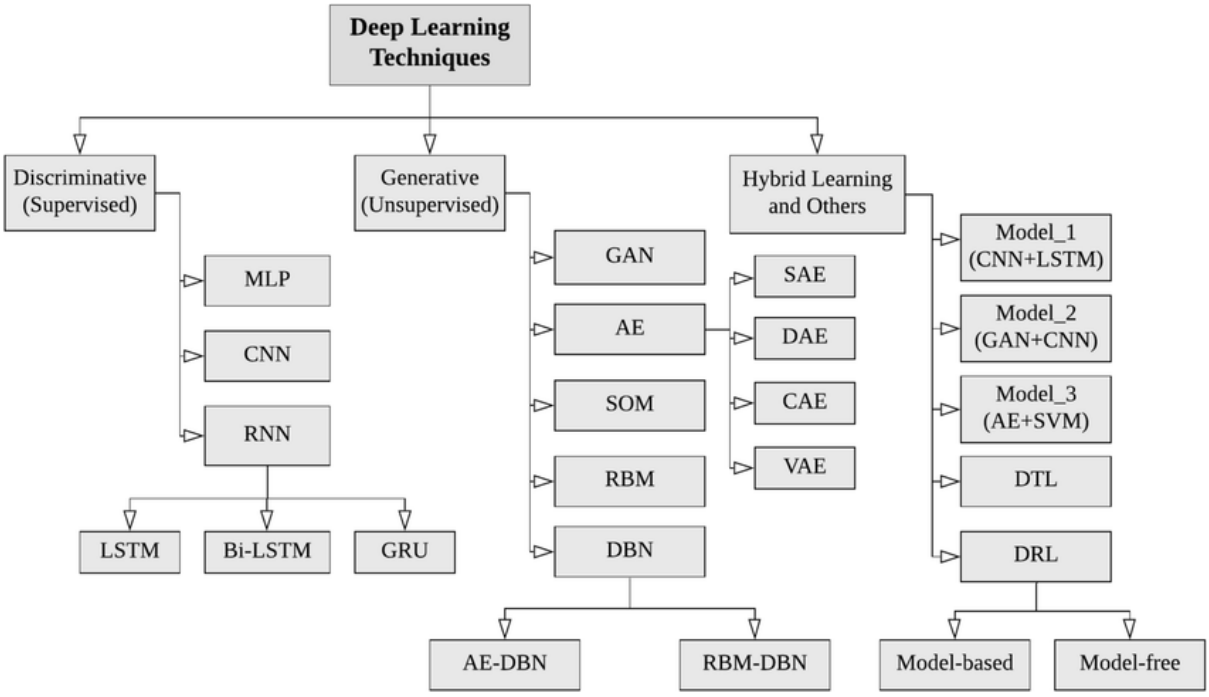

Figure 1.5: Classification of DL techniques [5]

1.6.1 Deep networks for supervised learning

This type of DL techniques is utilized to provide a discriminative function in supervised or classification applications. Multi-layer Perceptron (MLP), Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) are the three basic types of discriminative architectures.

1.6.2 Deep networks for unsupervised learning

This type of DL techniques is commonly used to identify high-order correlation qualities or features for pattern analysis or synthesis, as well as the joint statistical distributions of visible data and their associated classes. The key notion of generative deep architectures is

that specific supervisory information, such as target class labels, is unimportant throughout the learning process [5].
Deep Belief Network (DBN), Self-Organizing Map (SOM), Restricted Boltzmann Machine (RBM), Generative Adversarial Network (GAN) and Autoencoder (AE) are the basic types of generative architectures.

1.6.3 Deep networks for hybrid learning

Hybrid deep learning models are typically made up of multiple deep basic learning models, where the basic model is a supervised or unsupervised deep learning model. There are three types of hybrid deep learning models :

- Hybrid Model 1 : Combining different supervised or unsupervised models.
- Hybrid Model 2 : A supervised model combined with an unsupervised model.
- Hybrid Model 3 : A supervised or an unsupervised model combined with a non-deep learning classifier.

1.6.4 Other approaches

1.6.4.1 Deep Transfer Learning

Deep Transfer Learning (DTL) is a technique that utilizes pre-trained deep neural networks as the starting point for training on a new task.

1.6.4.2 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) combines ANN with a framework of reinforcement learning that helps software agents learn how to reach their goals.

1.7 Convolutional Neural Networks

Before the rise of CNN, the problem with computer recognition was to extract features from the data supplied, which were not sufficiently effective or provided a high degree of

accuracy. In recent times, CNN has attempted to provide a higher level of efficiency and accuracy. CNN is one of the most popular deep neural networks, it takes this name from mathematical linear operation between matrices called convolution. It can be divided into two parts : feature extraction and classification. Feature extraction consists of an input layer, a convolution layer, and a pooling layer, while classification consists of a fully-connected layer and an output layer.

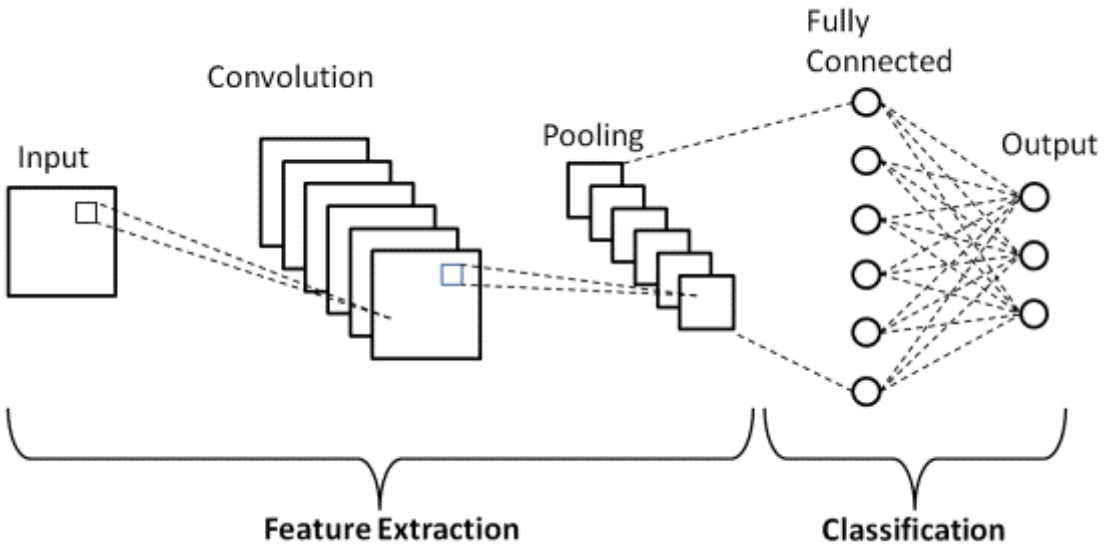

Figure 1.6: A simple schematic representation of a basic CNN [6]

1.7.1 Layers in Convolutional Neural Network

1.7.1.1 Convolution layer

Convolution Layer is the most basic but at the same time most important layer in the CNN architecture that performs feature extraction, which typically combines linear and nonlinear operations (convolution operation and activation function) [7].

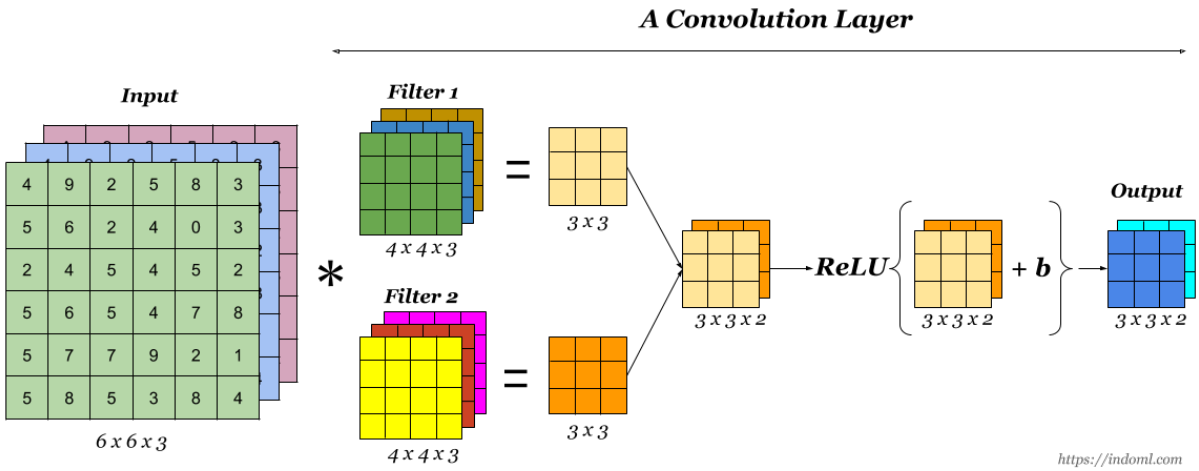

Figure 1.7: Convolution layer [8]

- **Convolution**

Convolution is a linear operation used to extract features, it is applied to the input data to filter the information and produce a feature map. This filter is also called a kernel, or feature detector. To perform convolution, the kernel goes over the input image, doing matrix multiplication element after element. The result for each receptive field (the area where convolution takes place) is written down in the feature map.

- **Nonlinear activation function**

A nonlinear activation function is then applied to the results of a linear operation (convolution). The most common nonlinear activation function used presently is the Rectified Linear Unit (ReLU), which simply computes the function: f(x) = max(0, x)[7].

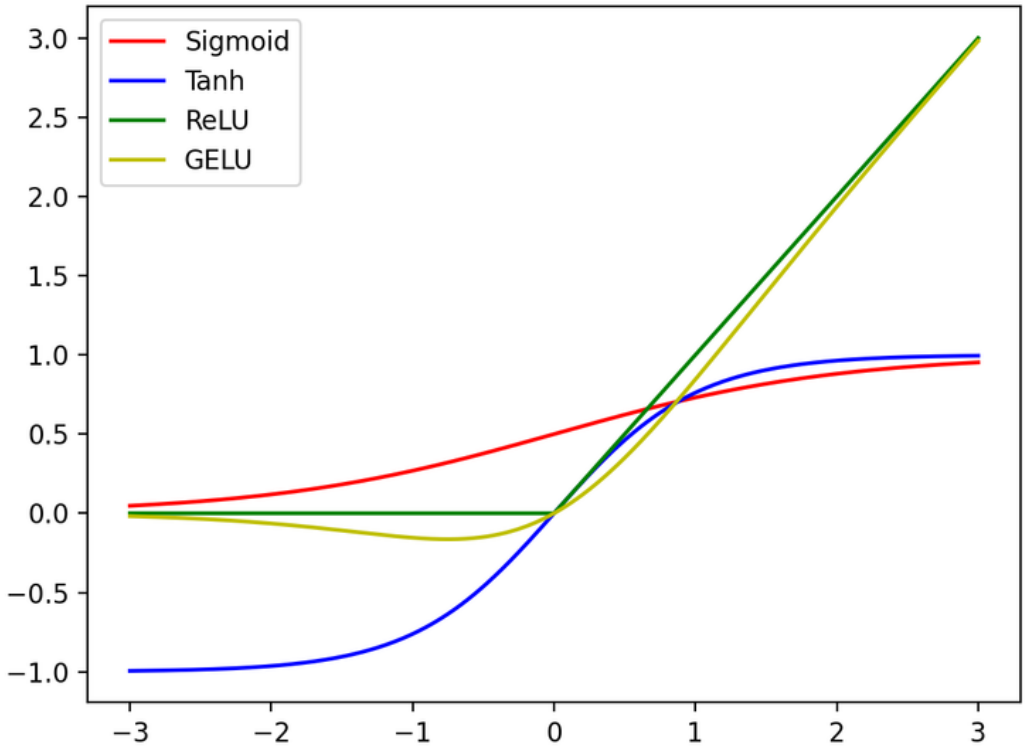

Figure 1.8: Sigmoid, Tanh, ReLU and GELU activation function [9]

1.7.1.2 Pooling layer

The goal of a pooling layer is to reduce the dimensions of the activation map, while also reducing the spatial invariance (hopefully without losing essential information). There are various types of pooling like max pooling, average pooling, stochastic pooling, spatial pyramid pooling. The most popular form of pooling operation is max pooling, it divides the activation map to sub-matrices, and takes the highest value from each sub-matrice [10].

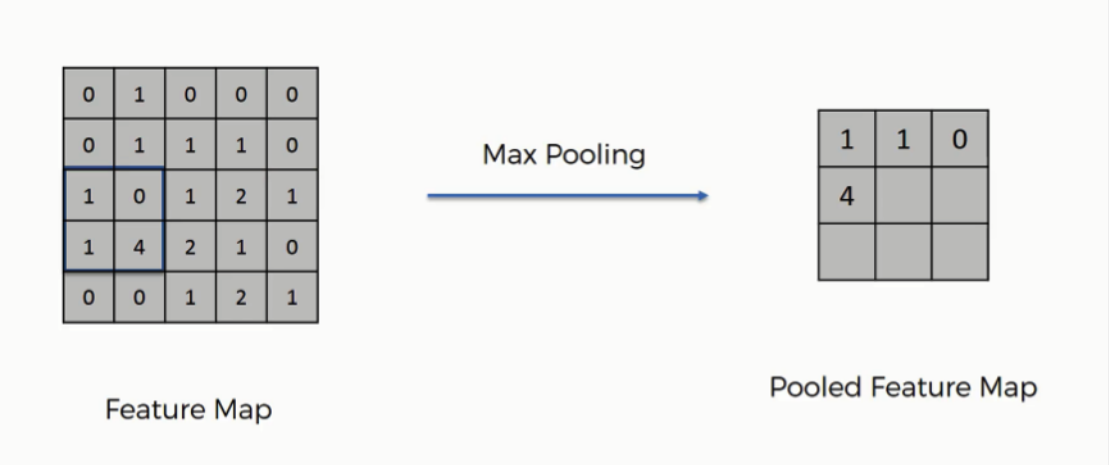

Figure 1.9: Max pooling of a feature map [10]

1.7.1.3 Fully connected layer

Fully-connected layer is often used as the final layer of a CNN. The output feature maps of the final convolution or pooling layer is typically flattened, i.e., transformed into a one-dimensional array of numbers, and connected to one or more fully-connected layers, where each input is connected to each output by a weight [7].
Each fully connected layer is followed by a nonlinear function, such as ReLU.

1.8 CNN architectures

1.8.1 LeNet-5

This architecture was one of the first CNN architectures developed by Yann LeCun in 1998. The LeNet-5 architecture consists of seven layers, two convolution layers, two pooling layers and three fully-connected layers.

1.8.2 AlexNet

This architecture was developed by Alex Krizhevsky in 2012, and in that particular year, it won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The AlexNet architecture consists of eight layers, five convolution layers and three fully-connected layers.

1.8.3 VGG

This architecture was developed by researchers at the University of Oxford in 2014. The VGG architecture consists of multiple layers of convolution and pooling layers and fully-connected layers.

1.8.4 GoogLeNet/Inception

This architecture was developed by Google researchers in 2014, and in that particular year, it won the ILSVRC. The use of "inception" modules, which are made up of multiple layers that perform various types of convolution and pooling layers, is the main innovation of GooLeNet.

1.8.5 ResNet

The Residual Network architecture was introduced by Microsoft researchers in 2015, and in that particular year, it won the ILSVRC. The aim of this architecture is to tackle the issue of vanishing gradients in very deep neural networks, which can cause the network to stop learning and become difficult to train. Different types of ResNet were developed based on the number of layers (from 34 layers to 1202 layers) [11].

1.9 Conclusion

Deep learning has revolutionized the field of AI by pushing the boundaries of what machines can achieve in terms of perception, understanding, and decision-making. Its continuous evolution and advancements hold great promise for addressing complex real-world problems and driving further innovations in various domains.

Chapter 2

Arabic sign language

2.1 Introduction

Disability is a complicated and common concept that even though so many medical and technological advances have been made, people all over the world still endure some type of disability, according to the world health organization 15% of the world population suffers from a type of disability. One of the most common disabilities is hearing impairment ; over 1.5 billion people suffer from it and that number is expected to rise to 2.5 by 2050. A lot of people who are at risk of having or developing hearing problems are young people (between 12-35 years old) [12]. This poses a big challenge regarding their education and communication ; how can communication be established with people who suffer from hearing disability so that they can live their lives and have the same opportunities as people without disability.

2.2 Deafness and hearing impairment

Hearing loss varies in its degree, some having a mild difficulty in their hearing to completely not be able to hear, Deaf is used to refer to people who have profound hearing loss.

26-40 dB	Mild hearing loss
40-70 dB	Moderate hearing loss
70-90 dB	Severe hearing loss
>90 dB	Profound hearing loss (Deafness)

Table 2.2: Types of hearing loss and their severity [13]

This variance in the degree of hearing means that each type has a specific way of communication, some use lip reading also known as oralism and others use sign languages.

- **Oralism**

  Oralism is defined as teaching the person to read the speakers lip, this method is usually used by people hard of hearing, this requires the person to be knowledgeable of the mouth movements necessary to pronounce each letter or word. But this method can be challenging in recognizing silent sounds and words without distinguishable mouth movements.

- **Sign Languages**

  Sign language is defined as an organized collection of hand gestures with specific meanings that deaf people use to communicate in daily life. As a visual language, it uses the movements of hands, face, and body as communication mediums [14]. Sign languages are the most common type of communication of people with hearing disability as it offers more diversity and a wider range of movements. Sign languages are so common that multiple sign languages can exist in the same country or region.

2.3 History of Sign Languages

From the beginning of time and even before the development of language and civilization humans have used gestures both bodily and facially to express and convey their thoughts

and emotions. But it was always seen as a part of cultural gestures and language, so the development of the use of signs and gestures as a separate language would not come to be until recently.

Throughout the years deaf and hearing impaired people have suffered a lot of discrimination and shaming and were deemed less than regular people for their lack of ability to speak, as the fluency of language and speech were regarded as a sign of high intellect the use of sign language was shamed and looked down upon. This went as far as some legislations sought to ban the use of sign language in schools for the deaf [15].

Despite all the negative efforts some progress was established later on, as in the 16th century education for the deaf was first introduced in Paris france 1760, the first public school for the deaf Institut National de Jeunes Sourds de Paris was created by Charles Michel De L'Eppe who also went on to create a french sign language alphabet as well as a dictionary for sign language words. This launched efforts to create other schools for the deaf that utilize and teach sign language as the primary means of communication, as well as efforts to establish sign languages as internationally recognized languages (which is contradictory to the misconception that there is one universal sign language). These efforts continue to this day, as some sign languages were formally recognized as recently as 2022 [16].

These Public schools for the deaf, established in France and eventually across much of Europe and North America, provided an environment in which sign languages could flourish. Following a clear shift toward oralism in the late 19th century, however, as we said earlier sign languages were viewed by many as crude systems of gestures and signing was banned in most schools. Despite that, sign languages continued to thrive outside the classrooms and in deaf communities and clubs, however, and by the mid 20th century oralism began to wane.

But despite all this history and scholarly interest in sign language, research in the field of linguistics only began in 1960. In the years since, the discipline of sign language linguistics has grown considerably, with research on over one hundred sign languages being conducted around the globe. Although the genetic relationships between the world's sign

languages have not been thoroughly researched, we know that historical links have in many cases resulted from migration, world politics, as well as the export of educational systems [17].

All this big variety in sign languages and their culturally linked history as well as the small percentage of non disabled people who know it makes it quite difficult for the people who use it to adequately communicate.

2.4 Sign language in the Arab world

Just like every country and region the Arab world also has its unique and distinct set of sign languages, sometimes multiple within the same region , this is due to the diverse number of dialects. Efforts for creating a unified Arabic Sign Language failed due to the various dialects in the region as well as the culture and tradition that links each sign to its region.

In Algeria for example Langue des Signes Algérienne (LSA) is the mostly used sign language, that originated from the French Sign language, but other than LSA, different sign languages exist such as "Algerian Jewish Sign Language" that originated from the jewish community that lived in Ghardaia as well as "Algerian Sign Language of Laghouat" and many others [13].

Arabic Sign Language (ArSL) is one of the sign languages used in Arab countries. Compared to Arabic, this language is completely distinct and independent. It differs from spoken Arabic in terms of grammar, syntax, and structure. The necessity for machine translation between ArSL and Arabic as well as ArSL recognition grows as a result of these distinctions.

In Arab countries, the community of the hearing impaired makes up about 4 % of the population. This community uses several sign languages. These languages coexist in countries with the same culture, so they may have some signs in common, but most signs are different. In 1999, the League of Arab States (LAS) and the Arab League Educational, Cultural and Scientific Organization (ALECSO) proposed a new sign language

called Arabic Sign Language (ArSL) to standardize these languages into one [18].

2.5 Arabic Sign Language

ArSL alphabets are similar to the forms of Arabic letters, and this similarity can be found in the American and British sign language as the alphabet's letters are represented by one hand.

Figure 2.1: Arabic sign language alphabet [19]

2.6 Sign language communication

Different to oral or written languages, sign language do not form sentences by the putting together the sum of each letter that forms a word or a sentence, instead sentences in sign language the syntax is broken down to categories ; in American sign language for example sentences are broken down to : Time + Subject + Verb + Object.

So translating a sentence like : "I went to France last week" would be in the following form : last week + me + went (go+finish) + France .
Conveying these words can differ from one sign language to another as some of them utilize facial and body gestures as well as hand gestures. Knowing the details and intricacies about sign language sentence structure as a crucial part in creating a system for continuous recognition and translating.

2.7 Sign language recognition

Sign Language Recognition (SLR) is the scientific area responsible for capturing and translating sign speech using artificial intelligence technology known as computer vision. There are two types of SLR systems, isolated SLR and continuous SLR. In isolated SLR, the system is trained to recognize a single gesture (which will be the starting point of our project). Each image is labeled to represent an alphabet, a digit, or some special gesture. In continuous SLR, instead of just one gesture, the system is able to recognize and translate whole sentences [14].

2.8 Sign languages-based approaches

Sign language recognition has two methods : the first one is the hardware method otherwise known as device method where the user has to wear a device that captures the movements through a sensor. The second is the software method by implementing computer vision where a camera captures the movements of the user and recognises them.

2.8.1 Hardware based approaches

by implementing gloves and sensors, it is an electromechanical input device used for human computer interactions, widely used in haptic applications, and works like a regular glove. It is used to capture a lot of physical data such as hand gestures and postures, body movements, and motion tracker, all these movements are interpreted by software that accompanies the glove. In 1983, this approach was used for SLR of signers from recorded

videos and to circumvent many approaches of computer vision, especially, recognition of signs from videos [20]. Certain conditions have to be met for this approach to give accurate results such as accompanying the gloves with a camera which in return requires other conditions such as lightning conditions.

2.8.2 Software based approaches

2.8.2.1 Traditional approaches

In [21], M. Mohandes, S. I. Quadri, and M. Deriche divided the process into three stages: a data collection stage, a feature extraction stage, and a recognition stage using Hidden Markov Model (HMM). A vision-based methodology is used to collect data, and then we need to prepare it to absorb the necessary features to classify it using HMM. The collected data were 4500 signs from 15 samples with 300 signs for the single signer, 11 samples were taken for the training set; the accuracy obtained from the experiments is 88.73%.

In [22], a system was put in place to automatically recognize an ArSL with a Visual Descriptor. In order to create a powerful ArSL alphabet recognizer, the system was designed to generate a large number of visual descriptors. In particular, a one-versus-All SVM was used to classify the generated visual descriptors. According to their findings, the Histograms of the Oriented Gradients (HOG) descriptor significantly outperform the other descriptors. The method that was proposed had a recognition accuracy of 90.55%.

The authors in [23] introduced an automatic Arabic sign language (ArSL) recognition system based on the Hidden Markov Models (HMMs). The main three steps of keeping the tabs on and detecting the hand involved tracking the fingerprints, detecting the edge, and identifying the skin. The system achieves an overall recognition rate reaching up to 82.22%.

The researchers in [24] suggested a method for detecting hands and faces based on skin profiles using input images translated to YCbCr color space. As image transformation, morphological dilation operation was also used. The Prewitt operator was used to detect hand form edges, and the Principal Component Analysis (PCA) was applied to achieve dimensionality reduction and obtain the final feature. The experiment showed a 97% accuracy on KNN based on 150 signs and gestures.

2.8.2.2 Approaches based on deep learning

In [25], a vision-based system for Arabic sign language recognition was suggested. To translate gesture images into Arabic speech in a supervised manner, it uses CNN architecture. This system detects hand sign alphabet and speaks out the corresponding Arabic letter using deep learning model. A dataset with 100 images in the training set and 25 images in the test set for each hand sign is created for 31 letters of Arabic sign language. The accuracy of the reported performance was 90%.

Another Arabic sign language recognition system based on fine-tuning deep CNN was proposed in [26]. As typical deep learning architectures, the system adopted Residual Network (ResNet-152) and Visual Geometry Group (VGG-16) with softmax layer classification after the fully connected layer to improve the prediction performance. The dataset contained 54,049 images distributed around 32 classes of Arabic Signs, the images dimensions are unified on 64 x 64, and many variations of images were presented through the use of different lighting and backgrounds. The accuracy reported was almost 99% and considered to be very high.

In [27], the researchers developed an ArSL recognition system based on Recurrent Neural Networks (RNN). This system is based on four distinct phases: acquisition, processing, feature extraction and image recognition. A color system called Hue, Saturation, and Intensity (HSI) is required for image processing to extract features associated with color layers. As a result, this system recognized 28 ArSL letters with 95% accuracy.

In [28], an Adaptive Neuro-Fuzzy Inference System (ANFIS) was used for ArSL recognition. The system is based on gestures that represent the Arabic alphabet. The dataset used for training and testing the recognition system consists of gray-scale images for all of the 30 gestures, 60 samples for each gesture were taken from 60 different volunteers. For each gesture, 40 out of the 60 samples were used for training purpose, while the remaining 20 samples were used for testing. The result of this system attained 93.55% accuracy in detecting 30 Arabic letters.

2.9 The process of sign languages recognition for software approach

The process of recognition for sign languages works the same as any image recognition system, the image containing the sign we wish to be recognised is fed to the system then processed after that the model returns the best match for that sign.
Just like any other language, sign languages don't have a universal language, each country has its own language. This thesis is concerned with creating a recognition model for Arabic Sign Language.

2.10 Conclusion

Sign languages are an essential and sometimes the only means of communication for those who suffer from hearing disability, but as stated above not many people know it which makes it much more difficult in this day and age.
While a lot of effort has been put toward creating software and technologies that can translate different sign languages, this field of study is still lacking.
Therefore, this thesis attempts to close that technological gap between Arabic sign language and modern implementation as well as help the large demography of people who suffer from hearing disability in the Arab world.

Chapter 3

Conception

3.1 Introcution

Artificial Intelligence (AI) has experienced a remarkable surge in popularity in recent years, captivating the attention of individuals, industries, and governments worldwide. The relentless advancements in AI technology, coupled with its vast potential, have propelled it into the forefront of innovation and transformation across various sectors. From healthcare and finance to transportation and entertainment, AI's impact is pervasive.
Moreover, AI has revolutionized the translation landscape. Natural Language Processing (NLP) algorithms combined with machine learning have paved the way for sophisticated language translation systems. AI-powered translation tools are now capable of processing vast amounts of textual data, learning from patterns, and providing accurate translations across multiple languages. This has proven invaluable in breaking down language barriers, facilitating global communication, and supporting international businesses. In addition to recognition and translation, AI finds application in an array of other fields. Healthcare benefits from AI-based diagnosis systems that analyze medical images and detect anomalies with exceptional precision, aiding in early disease detection.
Among the fields that fall into the category of recognition and translation is the domain of sign language detection more noticeably due to the corresponding popularity of AI, various sums of academic and independent efforts have been made to create deep learning

models that could recognize sign language and eventually translate them. Among them we cite two papers concerned with creating a sign language recognition model for arabic sign language : "A Deep Learning Approach for Arabic Sign Language Recognition" by M. Alsharif, N. AlZaabi, and M. S. Alghamdi (2021) and "Real-time Arabic Sign Language Recognition using Convolutional Neural Networks" by M. A. Al-Badrashiny, M. F. Tolba, and A. El-Sherif (2020).

As we continue to see, deep learning and AI in general is more and more proving its potential in fields that previously were thought impossible to accomplish.

This chapter is dedicated to explain the architecture used for an Arabic sign language recognition system implemented using neural networks and other deep learning techniques.

The chapter begins by presenting our goal and motivations, the systems architecture, and the scientific approaches used to evaluate and measure the end systems performance.

3.2 Motivation and purpose

Even though a surge of effort and interest that have been seen in the domain of sign languages, unfortunately that interest has not yet reached the arab world, where implementing modern techniques and technology to facilitate and solve modern problems has not been a popular domain nor has it been the topic of main focus for neither the public nor the academic institutes. That is why the main goal for this thesis is to create an Arabic sign language recognition system by implementing neural networks and training them on the RGB Arabic Alphabet Sign Language (AASL) dataset that contains the alphabet of the official arabic sign language.

A series of experimentation and adaptability has been put into place in order to create the end result that you will see in this thesis, eventually we compare our system performance with those with other systems of similar nature and dataset.

3.3 Architecture of our recognition system

Before coming to the end results of this thesis we experimented on various techniques and deep learning algorithms in effort to see which one yielded the best results.
We first started with the most appropriate and most used technique for computer vision; Convolutional neural network which was trained on an image dataset that we treated and segmented, with this experiment we determined that the prediction accuracy was not satisfactory even after a lot of changing and tweaking, so we pivoted to an other approach and we switched from training on pixel values of images to training on hand landmarks.
This change of approaches was due to the low performance of the system in early stages of experimentation as well as some observed confusion by the model regarding some similar signs that is why we decided it was necessary to extract deeper level of features and information from the images as to make it easier for the system to distinguish between some similar classes.
One of the main problems that caused low performance was the inability to effectively differentiate the orientation of the hand as well as distinguishing between fingers when the were folded together, that is why implementing a hand tracking library that provided the x, y, z coordinates of each part of the hand was crucial in getting a greater level of tracking accuracy.

The figure shown below is a confusion matrix that shows the performance of the CNN model when predicting each class, as it is demonstrated the model has a harder time with classes with similar gesture than with other classes.

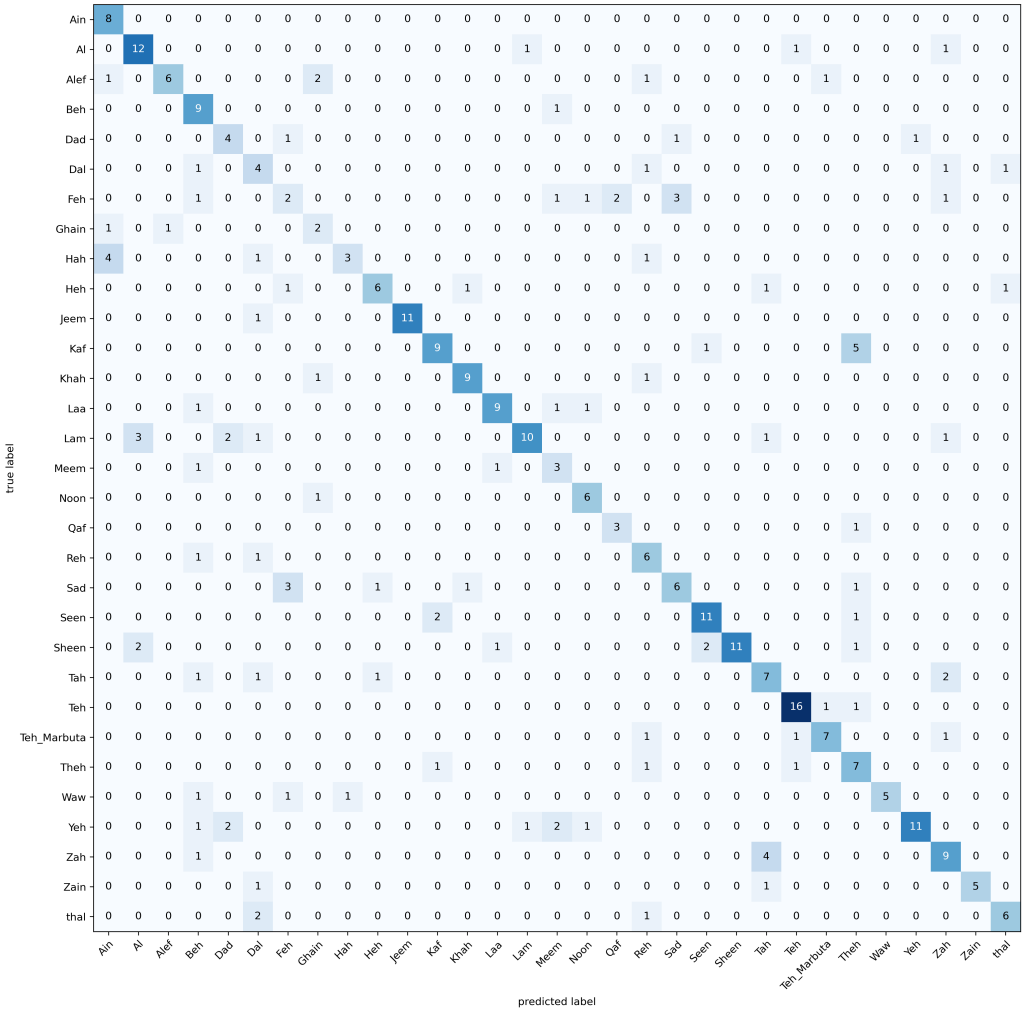

Figure 3.1: Confusion matrix for the CNN model

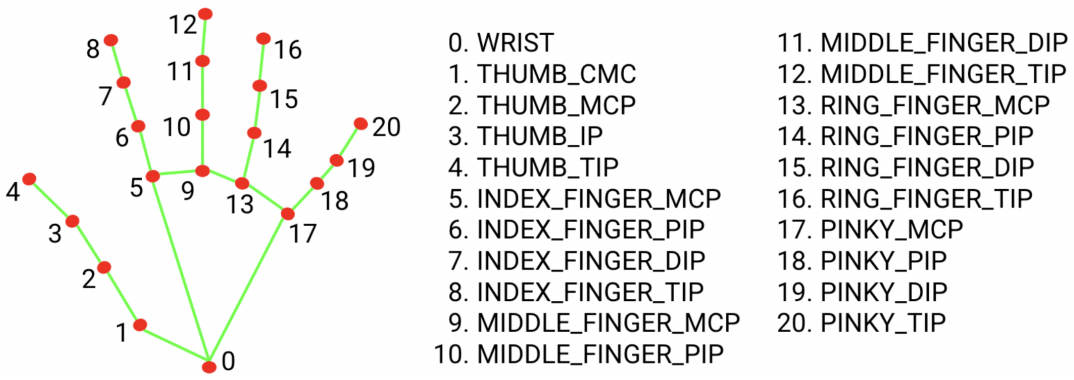

Figure 3.2: Hand landmarks points [29]

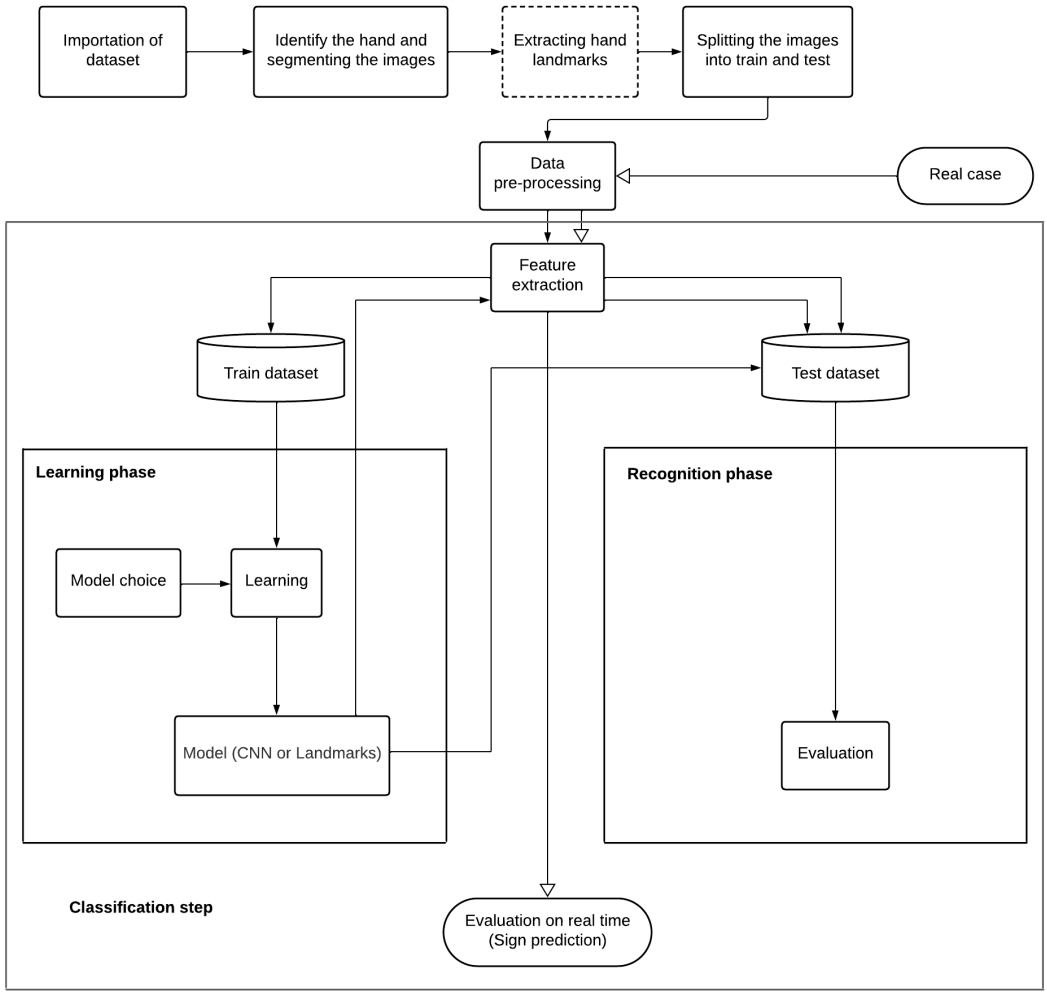

Figure 3.3: The Overall System's Architecture

Regardless of the approach or the technique the process of recognition for sign languages is usually follows the five steps : image acquisition, information extraction, image preprocessing, segmentation, feature extraction, and classification. In the following the sections we will be presented with both approaches we used that gave the top 2 resutls.

3.3.1 Detailed system overview

1. **Data Preperation** In order to build our model we needed to use a dataset. We used the RGB Arabic Alphabet Sign Language (AASL) dataset that introduces 7857 labeled images for the Arabic sign language. We first experimented with various

dataset but eventually settled on this one as it offered the largest and most high quality diverse data. The images were supervised, verified and filtered by a team of experts in Arabic Sign Language to ensure a high-quality dataset. The AASL data was collected from participants and used in a variety of settings, including lighting, backgrounds, image orientation, image resolution, and more [30].

The dataset is organized into 31 folders, each folder represents a single alphabet. Figure 3.4 shows an example of images for various alphabets, while Figure 3.5 shows how many pictures are in each folder.

Figure 3.4: an example of images for various alphabets [30]

#	Letter name in English Script	Letter name in Arabic Script	# of Images
1	ALEF	أ (ألف)	287
2	BEH	ب (باء)	307
3	TEH	ت (تاء)	226
4	THEH	ث (ثاء)	305
5	JEEM	ج (جيم)	210
6	HAH	ح (حاء)	246
7	KHAH	خ (خاء)	250
8	DAL	د (دال)	235
9	THAL	ذ (ذال)	202
10	REH	ر (راء)	227
11	ZAIN	ز (زاي)	201
12	SEEN	س (سين)	266
13	SHEEN	ش (شين)	278
14	SAD	ص (صاد)	270
15	DAD	ض (ضاد)	266
16	TAH	ط (طاء)	227

#	Letter name in English Script	Letter name in Arabic Script	# of Images
17	ZAH	ظ (ظاء)	232
18	AIN	ع (عين)	244
19	GHAIN	غ (غين)	231
20	FEH	ف (فاء)	255
21	QAF	ق (قاف)	219
22	KAF	ك (كاف)	264
23	LAM	ل (لام)	260
24	MEEM	م (ميم)	253
25	NOON	ن (نون)	237
26	HEH	ه (هاء)	253
27	WAW	و (واو)	249
28	YEH	ي (ياء)	272
29	TEH MARBUTA	ة (تاء مربوطة)	257
30	AL	ال	276
31	LAA	لا	268

Figure 3.5: AASL distribution [30]

But because our classification problem requires a specific and thorough detection, the images of the dataset were not adequate to meet the level of precision required to train an accurate model that is why a level of processing was needed to be done to the data. The images originally contained a lot of visual noise ; secondary and background elements that took the majority of the image, which would confuse and lower the performance of the model as it would not understand that only hand signs are to be trained on.

Using a hand detection library we located the hand in each image and kept only the hand while discarding irrelevant details, then the segmented images were prepared for the neural network by resizing and normalizing them.

The data preparation process was almost identical for both the CNN approach and the landmarks approach , the only difference was the type of data that we extracted from the dataset.

For the landmarks model, the hand detection library went through each images of the dataset and extracted the 21 landmarks of each hand and stored them. Each of the extracted information mentioned previously well both go to train their respected models.

2. **Classification**

- **Learning phase**
  - **The CNN model**

    Model Generation : The sign language recognition model built in this study uses CNN as well as another model. As we mentioned earlier in chapter 2, a CNN typically has three layers: a convolution layer, a pooling layer, and a fully connected layer. CNN's architecture will be formed when these layers are stacked. The convolution layer extracts features from the input image, followed by the pooling layer which has the task of reducing the spatial dimensions of the resulting feature maps. The output of these layers is then passed through a fully connected layer that performs classification tasks. In addition to these layers, there are two more important parameters which are the dropout and the activation function. Figure 3.6 shows the architecture of our CNN model. The number of CNN layers is not always the same. This is defined according to the application and its needs.

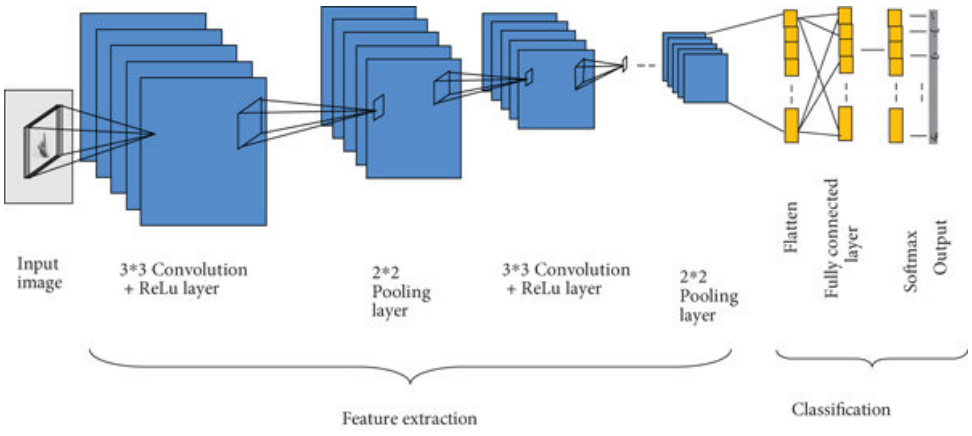

Figure 3.6: The architecture of our CNN model

Our CNN model is constructed as follows:

Input : CNN input data is a 4D array, also called a tensor. The dimensions of the tensor are (B, H, W, C), where B represents the batch size, H represents the height of the input image, W represents the width of the input image, and C represents the number of channels in the input image. The shape of our CNN input data is (32, 50, 50, 1).

Convolution layer : This layer aims to extract features from the input image by applying a set of filters (also known as kernels) to it. In our CNN model the size of the filters used is (3, 3). The input image is convolved with these filters to produce the feature map.

Batch normalization : In CNN, batch normalization is the process of normalizing the input of each layer before passing it through the activation function. The goal of batch normalization is to make the training process faster and more efficient, and to improve the models accuracy.

Pooling Layer : This layer is applied after the convolution layer. This reduces the number of parameters to be learned as well as the computational complexity of the network. In our CNN, we applying max pooling with a pooling size of (2, 2). The input feature map is divided into non-overlapping 2x2 regions, and the maximum value within each region is selected to produce a downsampled output feature map.

Dropout : Dropout is one of the most interesting types of regularization techniques. It works by randomly dropping out (setting to zero) a certain percentage of neurons in the network during training, to avoid overfitting and speed up the learning process. In our CNN, a dropout rate of 0.2 was used, this means that during training 20% of the neurons in the network will be randomly dropped out at each iteration.

Flatten Layer : This layer take the output of a pooling layer (which is 3D array) and converts it into a 1D array, to prepare the data for the fully connected layer that come after it.
After that the outputs get fed into a fully connected layer ,this layer reduces the input vector to a vector of size 64. In this layer, each neuron is connected to every neuron in the previous layer, the output of each neuron is then passed through the ReLU activation function. The function returns 0 when the input is negative, and when the input is positive it returns the input value itself.

Fully Connected layer with 'Softmax' activation function : This is the classification layer. It reduces inputs to a vector equal to the number of predicted classes (31).

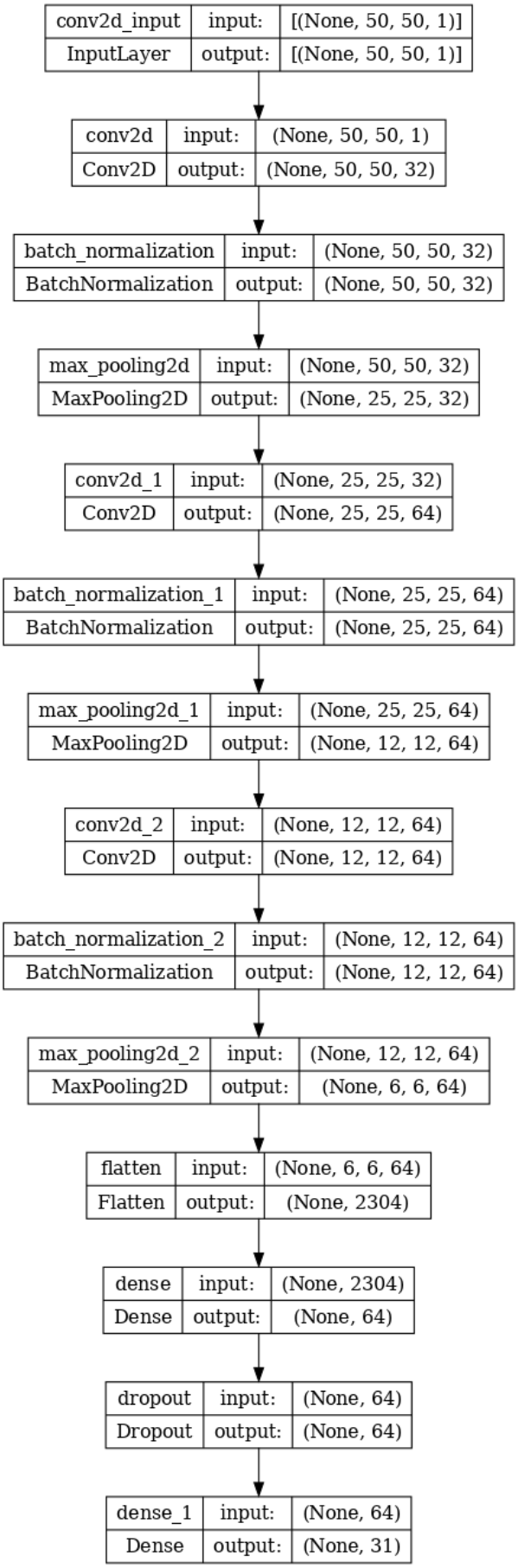

Figure 3.7: The CNN model architecture

The second model architecture consisted of a fully connected neural network, specifically a multi-layer perceptron (MLP). The structure of the network consists of an input layer with a shape of (63), followed by several fully connected (dense) layers. The activation function used for the dense layers is the Rectified Linear Unit (ReLU), which introduces non-linearity to the network. Batch normalization layers are added after some of the dense layers. Batch normalization helps in normalizing the inputs to each layer, improving the stability and performance of the network during training. Dropout layers are also used in the network. Dropout is a regularization technique that randomly sets a fraction of input units to 0 during training, which helps prevent overfitting by reducing the reliance on specific input units. The final layer is a dense layer with 31 units and a softmax activation function. This indicates that the network is designed for a classification task with 31 classes, and the softmax activation function produces a probability distribution over the classes. Overall, this network architecture combines dense layers, batch normalization, and dropout regularization to learn and classify patterns in the input data.

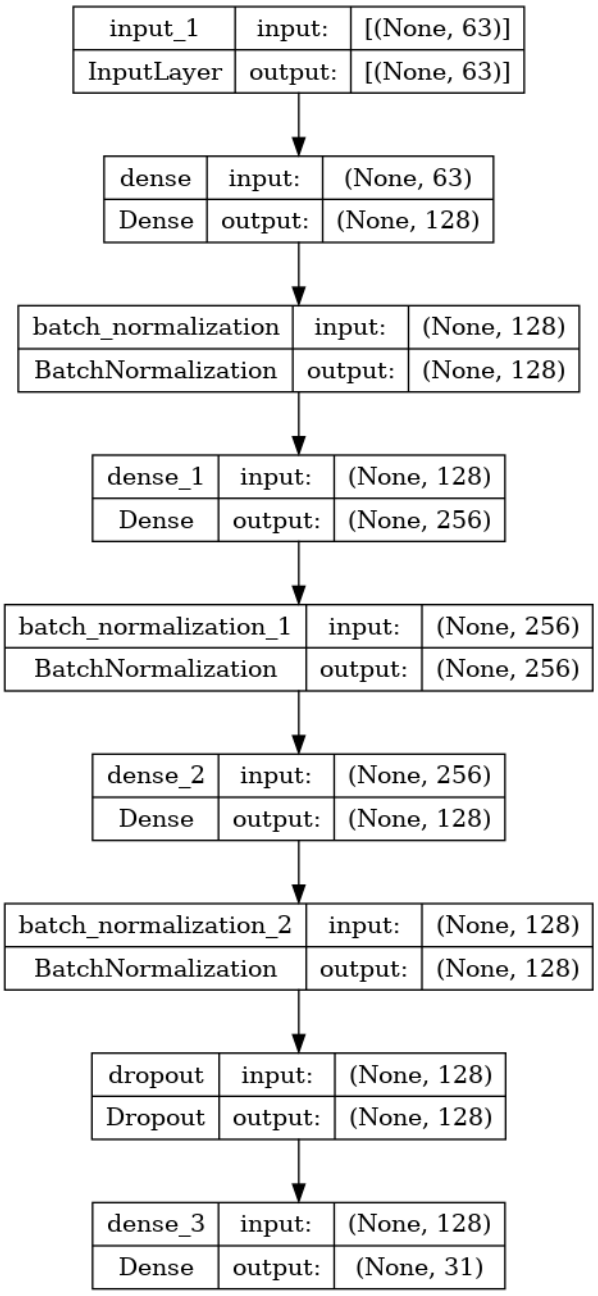

Figure 3.8: The DNN model architecture

– **The Landmarks model**

Model Generation

The neural network architecture we used consists of a sequential model, which is a linear stack of layers. The input layer of the model is defined with a shape of (63) (the 21 landmarks point multiplied by the x, y, z coordinates).

The first hidden layer is a dense layer with 128 units, and the activation

function used is the rectified linear unit (ReLU). ReLU introduces non-linearity to the model and helps in capturing complex patterns within the data. Batch normalization is applied after this layer, which normalizes the activations, making the model more stable during training.

The next hidden layer is another dense layer with 256 units, followed by batch normalization. This layer further learns and extracts relevant features from the input data, contributing to the overall model's capacity to capture intricate patterns.

Another dense layer with 128 units follows, along with batch normalization. Dropout regularization is introduced with a rate of 0.2 after this layer. Dropout randomly sets a fraction of the input units to 0 during training, which helps in preventing overfitting and improving the generalization ability of the model.

The final layer of the neural network is a dense layer with 31 units, representing the output classes of the classification problem. The activation function used for this layer is softmax, which produces a probability distribution over the classes, indicating the likelihood of each class given the input data.

Overall, this neural network architecture consists of multiple dense layers with varying numbers of units, employing ReLU activation, batch normalization, and dropout regularization. These components work together to enable the model to learn complex representations from the input data and make predictions for classes.

Figure 3.9 shows the results of our DNN model on the test base. The confusion matrix shows the predicted labels versus the true labels.

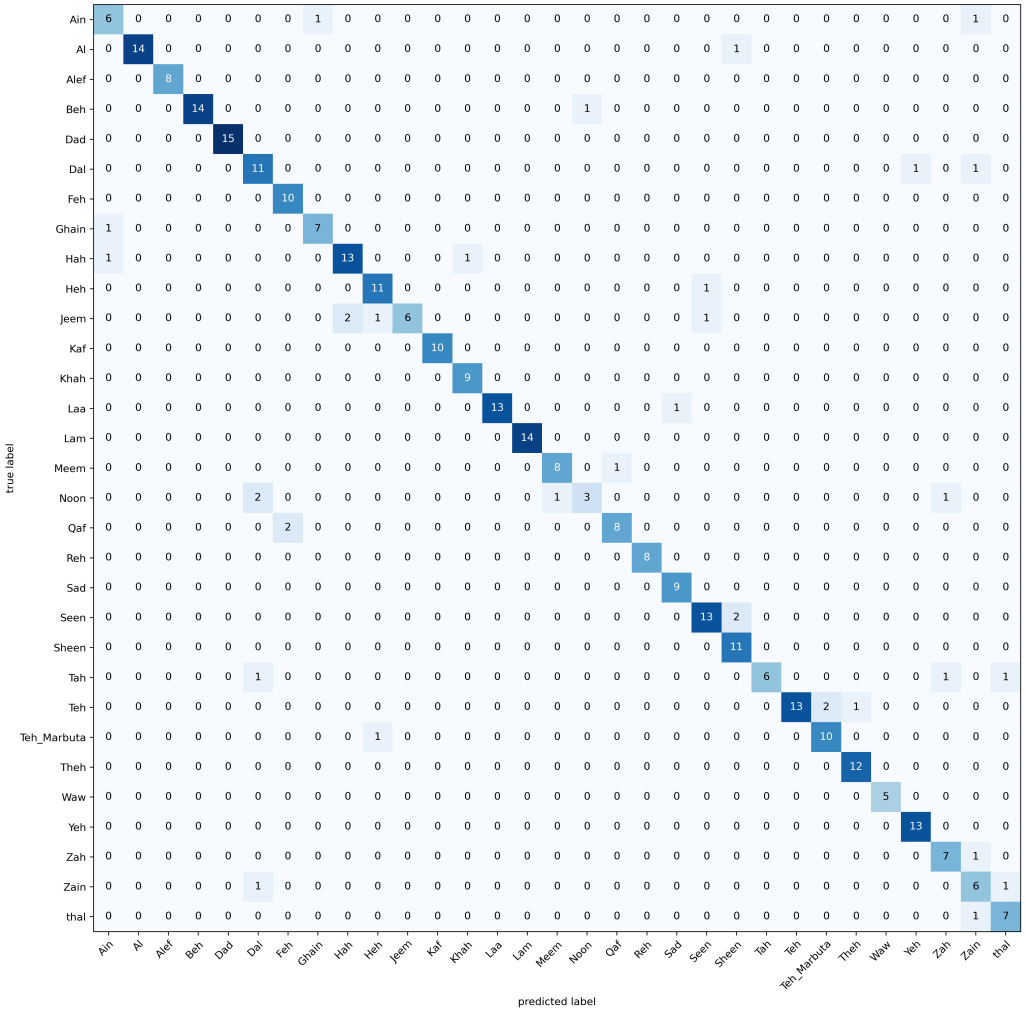

Figure 3.9: Confusion matrix for DNN model

3. **Reconnaissance phase**

- **On the dataset** Testing and evaluating the model on the test dataset involves feeding the test data into the trained model to obtain predictions or classifications. This step assesses the model's performance and generalization ability on unseen data. The predictions or classifications are then compared with the ground truth labels of the test dataset to calculate evaluation metrics as accuracy, precision, recall, and F1 score. These metrics provide insights into how well the model performs on the test dataset and its ability to make ac-

curate predictions. The evaluation results help in understanding the model's strengths and weaknesses, identifying potential areas for improvement, and determining if the model meets the desired performance criteria.

- **Real time recognition**

  The system works by following these steps : image acquisition, feature extraction, preprocessing, segmentation, feature extraction and finally classification.

  (a) Acquisition : It is the process of collecting a certain series of images that show different signs in different conditions, e.g., different hands, lighting, environment, etc. In our case the acquisition of the user input occurs through the devices camera where an object detection and tracking system identifies the hand and then captures the sign made and then feeds the frame of the captured sign as well as the corresponding hand landmarks into the models to make a prediction.

  (b) Preprocessing : For the image to be fed into the CNN model, an image's quality is first improved and unwanted noise is removed using preprocessing techniques. This can be achieved by resizing, converting color, removing noise, or using a combination of these techniques. The output of this process can greatly affect the accuracy. Our preprocess is done by resizing the image to 50 * 50 size gray image and then reshaping it to a (2500,) vector.

  For the landmarks model the preprocess is done by identifiy the hand landmarks and extracting each 21 point of them and putting them into a flat array of shape 63.

  (c) Segmentation : Is the process of partitioning an image into meaningful regions known as segments. In the context of this study,hand gestures would be the image segments. The process of image segmentation is utilized to extract the region of interest.

(d) Feature extraction : The process of feature extraction is used to obtain the most relevant features from an input image. The goal is to find the most distinctive features in the obtained image. This process is often accomplished by reducing the raw data (also known as dimensionality reduction), transformation to other color space, or extracting some custom properties that better describe certain aspects of the objects in the image, so that the classification is more robust and accurate.

(e) Classification : Finally, the extracted features whether they are pixels or landamrks are used to categorize the sign language gestures into their corresponding alphabets.

3.4 Experiments and results

While running various experiments on both models we eventually determined that the landmarks model is the best model for the recognition process as it offers better adaptability to various lighting condition and also gives more accurate tracking of the position of each finger especially relative to other fingers.

3.4.1 The number of dense layers

The search for the DNN model architecture to use begins by determining the number of dense layers. Table 3.1 shows the train and the test accuracy obtained on a model running 2 dense layers, 3 dense layers and 4 dense layers.

Number of dense layers	Train accuracy (%)	Test accuracy (%)
2	92	91
3	98	91
4	97	91

Table 3.1: Table of experimentation done on number of dense layers

As can be seen from the results presented in Table 3.1, three dense layers gave us the best results (the train accuracy is 98% and the test accuracy is 91%) and this is the number of dense layers used to build our DNN model.

3.4.2 The size of dense layers

Once the number of dense layers was determined, a second experiment was carried out to determine the size of the dense layers. The size of the dense layer (or units) is one of the most fundamental parameters. The results obtained are shown in Table 3.2.

Dense layer size			Train accuracy (%)	Test accuracy (%)
1st layer	2nd layer	3rd layer		
32	64	32	95	86
64	128	64	95	89
128	256	128	97	91
256	512	256	97	79

Table 3.2: Table of experimentation done on dense layers size

After the experiments presented in Table 3.2, we chose the sizes used in the second experiment (1st layer : 64, 2nd layer : 128 and 3rd layer : 64).

3.4.3 Type of activation function

The next experiment consists of finding the best activation function to use for the model. We selected two activation functions, ReLU which is one of the most popular activation functions used in DL, and GeLU which is a smooth approximation of the ReLU function that aims to address some of its limitations. The results obtained are shown in Table 3.3.

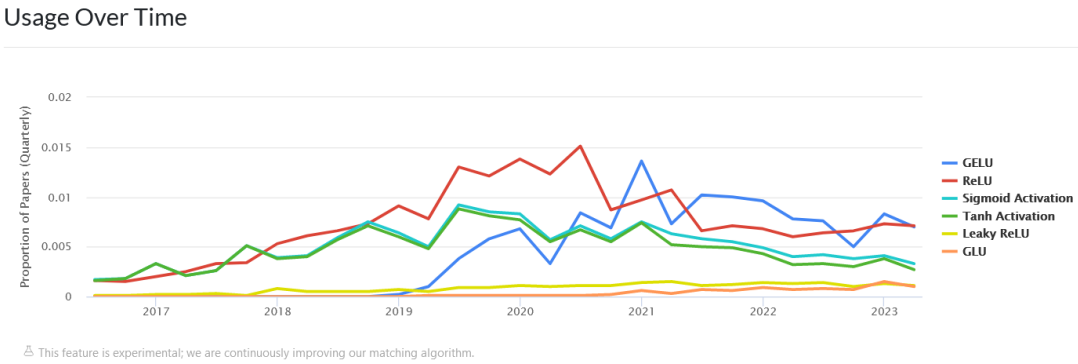

Figure 3.10: Activation funcitons usage over time [31]

Activation function	Train accuracy (%)	Test accuracy (%)
ReLU	97	91
GeLU	97	90

Table 3.3: Table of experimentation done on activation function

As can be seen from the results presented in Table 3.3, ReLU gave us the best test accuracy (91%) and this is the activation function used to build our DNN model.

3.4.4 The dropout rate

The next experiment consists in finding the Dropout rate to use to prevent overfitting and improve the performance of our MLP model. The dropout rate is typically set between 0.2 and 0.5, which means that 20% to 50% of neurons are dropped out during training. The results of this experiment for different Dropout rates are shown in Table 3.4.

Dropout	Train accuracy (%)	Test accuracy (%)
0.2	97	91
0.3	97	88
0.4	98	90
0.5	97	90

Table 3.4: Table of experimentation done on dropout rate

As can be seen from the results presented in Table 3.4, it is clearly observable that the dropout of 0.2 gave us the best results.

3.4.5 Cross-validation technique

In deep learning, cross-validation can be used to evaluate the performance of NN model. It can help avoid overfitting and it also allows the tuning of hyperparameters such as the number of epochs more effectively by providing a more robust estimate of performance. The most common type of cross-validation is k-fold cross-validation. In this technique, the data is divided into k subsets (or folds) of equal size. The model is trained on k-1 folds and tested on the remaining fold. This process is repeated k times, with each fold being used as the test set once. K-fold cross validation with k = 5 is a popular choice because it strikes a balance between computational efficiency and accuracy, providing a reasonable estimate of the model's performance without requiring too much computational resources.

The results obtained when using k-fold cross-validation (with k=5) are shown in Table 3.5.

Cross-validation (k-fold)	Train accuracy (%)	Test accuracy (%)
Without cross-validation technique	97	91
With cross-validation technique	90	93

Table 3.5: Table of experimentation done on activation function

As can be seen from the results presented in Table 3.5, when we used cross-validation technique (k-fold) the test accuracy improved.

3.5 Performance comparison

As of the date of writing this thesis no works were done based on this dataset that can be compared to our work. For this reason, we tried in this part to compare the performance of the two models used in our work, the CNN model and the Landmarks model.

Model	Train accuracy (%)	Test accuracy (%)
CNN	80	73
Landmarks	90	93

Table 3.6: Performance comparison (CNN and Landmarks)

3.6 Conclusion

Using all of the above experimentation and methods has led us to architecture that the system is based on ;a sequential model with multiple layers, an input layer of shape (63) and that includes several dense layers with activation functions like ReLU and batch normalization to capture complex patterns in the data. Dropout regularization is applied to prevent overfitting. The final layer is a dense layer with 31 units and softmax activation, used for multi-class classification tasks.

Chapter 4

Implementation

4.1 Introduction

In this section, we first present the development tools, then describe the various steps involved in implementing the Arabic Sign Language recognition system, as well as the interfaces and system functionality.

4.2 Software tools

In this section, we present the software tools needed to develop our system.

4.2.1 Python

Python is an interpreted, object-oriented, and high-level programming language, it was developed by Guido van Rossum in 1991. The first version of Python, known as Python 1.0, was published in 1994 and it was designed to be a simple language that could be easy to learn. Over the years, the language has developed to become one of the most popular programming languages in the world. Python is widely used in web development, data mining, scientific computing, machine learning, etc.

4.2.1.1 Importance of the Python programming language

Python is a programming language that has gained immense popularity in recent years. Here are some reasons why Python is important:

- **Easy to learn and use**

  The python language has a simple syntax and easy-to-understand code that make it an ideal language for beginners.

- **Libraries and frameworks**

  Python has a large number of libraries and frameworks that make it easier to develop complex applications quickly. Some popular libraries and frameworks are NumPy, Keras, Tensorfow, etc.

- **Large community**

  Python is supported by a large community of developers who contribute to the development of the language and provide assistance to new users.

- **Open source**

  Python is an open source programming language, which means that it is free to use and modify, and there are no licensing restrictions.

4.2.1.2 Libraries used

- **NumPy**

Figure 4.1: NumPy [32]

NumPy (short for Numerical Python) is a popular Python library used for scientific and numerical computing. It offers a powerful way to store and manipulate large

arrays of data, with a variety of advanced mathematical functions for performing complex operations.

- **TensorFlow**

Figure 4.2: Tensorflow [33]

TensorFlow is an open-source library developed by Google researchers for artificial intelligence applications. It offers a wide range of tools and functions to build and train deep neural networks. One of TensorFlows key features is that it supports both Graphical Processing Units (GPUs) and Central Processing Units (CPUs). It also has a compilation time thats much faster than other popular deep learning libraries, such as Keras and Torch.

- **Keras**

Figure 4.3: Keras logo [34]

Keras is a high-level neural networks API written in Python and it is designed to simplify the process of building and training deep learning models. It is an open-source library that can run on top of many deep learning frameworks such as TensorFlow, Theano, and CNTK.

- **OpenCV**

Figure 4.4: OpenCV [35]

OpenCV (Open Source Computer Vision) is a popular open-source computer vision library that offers a wide range of tools for processing images and videos that makes it easier for people to create advanced computer vision programs quickly. It is written in C++ and has bindings for various programming languages such as Python. In Python, OpenCV can be used to read and write images, resize images, detect edges in images, detect faces in images and videos, and perform other operations.

- **Scikit-learn**

Figure 4.5: Scikit-learn [36]

Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python that provides many tools for classification tasks. It offers various algorithms and techniques for classification like logistic regression, decision trees, support vector machines (SVM), etc. Additionally, Scikit-learn offers tools for data preprocessing, feature selection, and model evaluation. It allows users to split the

data into training and testing sets, cross-validate models, and tune hyperparameters to optimize performance. Due to its flexibility and ease of use, Sklearn is widely used in both academia and industry. It has a large community of users who contribute to its development and maintenance.

- **Mediapipe**

Figure 4.6: Mediapipe [37]

MediaPipe is an open-source framework developed by Google that provides a set of tools for building real-time computer vision and machine learning applications. One of the most popular applications of Mediapipe is hand tracking, which allows developers to build applications that can detect and track the landmarks of the hand in real-time.

- **Tkinter**

  Tkinter is a standard GUI (Graphical User Interface) library for Python. It provides a set of tools and widgets for creating desktop applications with graphical interfaces. It has a simple syntax and it comes pre-installed with Python, so there is no need to install any additional software. Many types of widgets such as buttons, labels, text boxes, radio buttons, and many more are supported by Tkinter.

4.2.2 Kaggle Notebook

Kaggle Notebook is a cloud-based development environment provided by Kaggle. It allows users to write and execute Python code in a web browser using a Jupyter Notebook environment. Kaggle Notebook offers pre-installed libraries such as NumPy, Pandas, Scikit-learn, and TensorFlow to make deep learning tasks easier. Users can also upload their own data for analysis or access datasets from Kaggle's repository. Additionally,

Kaggle Notebook offers a community-driven environment where users can share their work with others, ask for help or feedback on their projects, and participate in competitions.

4.3 Implementation steps

Now that the model is generated all that is left is the implementation.

The process for the detection goes as follows the user inputs signs through the device's camera, where the hand is identified and the 21 landmarks are picked up, those landmarks are then saved and reshaped to numpy array of shape (63,) which then gets fed to the model to make a prediction. This process goes for each frame of a video capture .

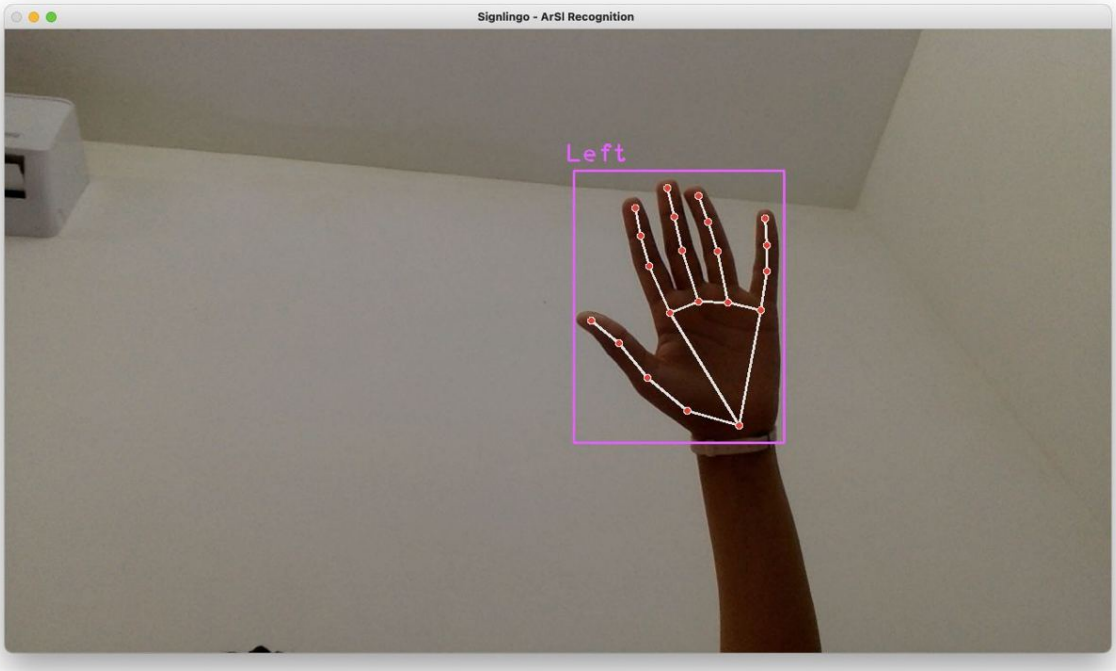

Figure 4.7: Implementation of the landmarks model

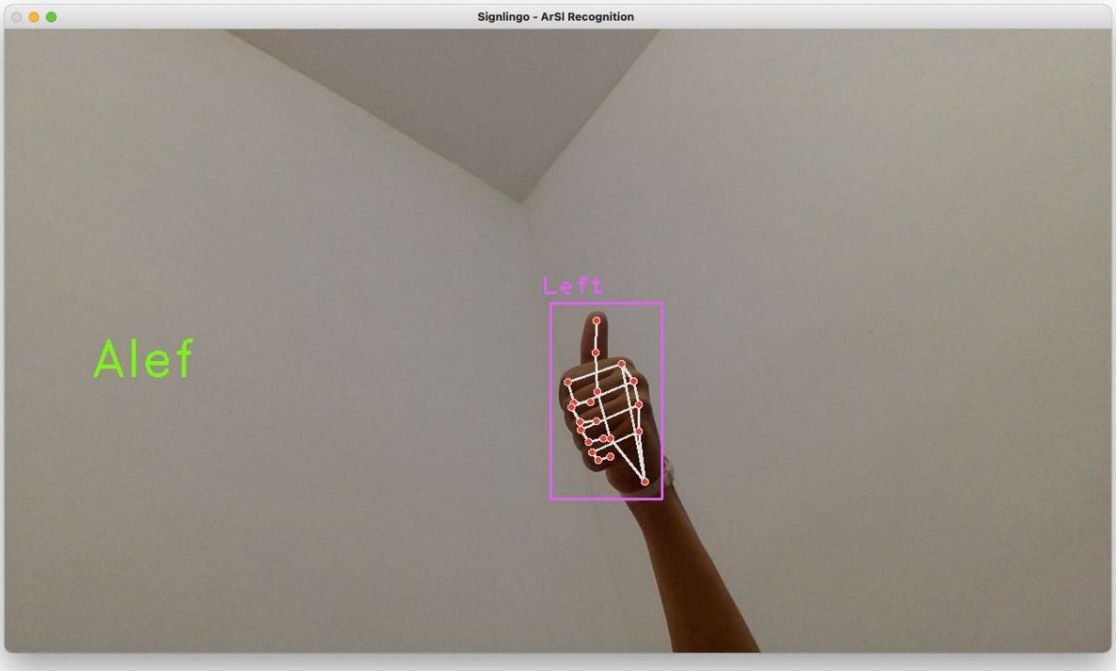

Figure 4.8: Results of prediction on user input

4.4 Conclusion

We presented above all the implementation steps that we took in order to deploy and implement our model. All the above mentioned libraries have made the process of the implementation and the deployment of the model easy .

General Conclusion

In conclusion, the thesis focused on Arabic sign language detection and explored various methodologies and techniques to address the challenges associated with this task.

The thesis highlighted the importance of using AI and computer vision for solving recognition tasks.

Through extensive experimentation and analysis, the thesis demonstrated the effectiveness of modern object detection models, such as CNN and ANN, which leverage deep learning architectures. These models showcased remarkable performance in terms of accuracy, speed, and scalability, making them suitable for real-world applications.

The availability of large-scale annotated datasets, such as AASL dataset, was identified as a crucial factor in the making of this project.

Ultimately, the findings and contributions of this thesis contribute to the broader field of computer vision and have practical implications in various industries. The advancements in object detection techniques have the potential to enhance safety, improve efficiency, and enable new applications that benefit society as a whole.

While the thesis successfully addressed Arabic sign language alphabet recognition, there are still open challenges that require further research. These include the recognition of continuous signs and optimizing the speed of the detection and translation as well as implementing the model for other sign languages.

Overall, the thesis provides valuable insights into sign language recognition methodologies, techniques, and challenges. It serves as a foundation for future research endeavors and lays the groundwork for the continued advancement in this field.

References

[1] A. Mathew, A. Arul, and S. Sivakumari, "Deep learning techniques: An overview," in Jan. 2021, pp. 599–608, ISBN: 978-981-15-3382-2. DOI: 10.1007/978-981-15-3383-9_54.

[2] M. Z. Alom, T. Taha, C. Yakopcic, *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, p. 292, Mar. 2019. DOI: 10.3390/electronics8030292.

[3] Y. Matsuzaka and R. Yashiro, "Ai-based computer vision techniques and expert systems," *AI*, vol. 4, no. 1, pp. 289–302, 2023, ISSN: 2673-2688. DOI: 10.3390/ai4010013. [Online]. Available: https://www.mdpi.com/2673-2688/4/1/13.

[4] A. Ayub Khan, A. Laghari, S. Awan, Lyari, and P. Karachi, "Machine learning in computer vision: A review," *ICST Transactions on Scalable Information Systems*, vol. 8, Apr. 2021. DOI: 10.4108/eai.21-4-2021.169418.

[5] I. Sarker, "Ai-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems," *SN Computer Science*, vol. 3, Feb. 2022. DOI: 10.1007/s42979-022-01043-x.

[6] Phung and Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, p. 4500, Oct. 2019. DOI: 10.3390/app9214500.

[7] R. Yamashita, M. Nishio, R. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights into Imaging*, vol. 9, Jun. 2018. DOI: 10.1007/s13244-018-0639-9.

[8] IndoML. "Student notes: Convolutional neural networks (cnn) introduction." (2018), [Online]. Available: https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/.

[9] X. Qi, J. Wang, Y. Chen, Y. Shi, and L. Zhang, *Lipsformer: Introducing lipschitz continuity to vision transformers*, Apr. 2023.

[10] A. Ajit, K. Acharya, and A. Samanta, "A review of convolutional neural networks," Feb. 2020, pp. 1–5. DOI: 10.1109/ic-ETITE47903.2020.049.

[11] L. Alzubaidi, J. Zhang, A. J. Humaidi, *et al.*, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[12] World Health Organization, *World report on disability*, 2011. [Online]. Available: https://www.who.int/teams/noncommunicable-diseases/sensory-functions-disability-and-rehabilitation/world-report-on-disability.

[13] H. Abdelouafi, “ hearing disability and characteristics of hearing-impaired,” Jan. 2021.

[14] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh, *Sign language recognition system using tensorflow object detection api*, Jan. 2022.

[15] L. R. Nendauni, *The development of sign language: A synopsis overview*, Feb. 2021. DOI: 10.13140/RG.2.2.19207.93609.

[16] World Federation of the Deaf. “The legal recognition of national sign languages.” (Year of publication), [Online]. Available: https://wfdeaf.org/news/the-legal-recognition-of-national-sign-languages/.

[17] R. Pfau, M. Steinbach, and B. Woll, *Sign Language: An International Handbook* (Handbook of Linguistics and Communication Science: Handbucher zur Sprach-und Kommunikationswissenschaft). De Gruyter Mouton, 2012, ISBN: 9783110204216. [Online]. Available: https://books.google.dz/books?id=DTdgMwEACAAJ.

[18] H. Luqman and S. Mahmoud, “A machine translation system from arabic sign language to arabic,” *Universal Access in the Information Society*, vol. 19, Nov. 2020. DOI: 10.1007/s10209-019-00695-6.

[19] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, “Arasl:arabic alphabets sign language dataset,” *Data in Brief*, vol. 23, p. 103 777, Feb. 2019. DOI: 10.1016/j.dib.2019.103777.

[20] A. Sultan, W. Makram, M. Kayed, and A. Ali, “Sign language identification and recognition: A comparative study,” *Open Computer Science*, vol. 12, pp. 191–210, May 2022. DOI: 10.1515/comp-2022-0240.

[21] M. Mohandes, S. Quadri, and M. Deriche, “Arabic sign language recognition an image-based approach,” Jun. 2007, pp. 272–276, ISBN: 0-7695-2847-3. DOI: 10.1109/AINAW.2007.98.

[22] M. Alzaidan, R. Alzohairi, R. Alghonaim, W. Alshehri, and S. Aloqeely, “Image based arabic sign language recognition system,” *International Journal of Advanced Computer Science and Applications*, vol. 9, Jan. 2018. DOI: 10.14569/IJACSA.2018.090327.

[23] A. Youssif, A. Aboutabl, and H. Ali, “Arabic sign language (arsl) recognition system using hmm,” *International Journal of Advanced Computer Science and Applications*, vol. 2, Nov. 2011. DOI: 10.14569/IJACSA.2011.021108.

[24] E. Hemayed and A. Hassanien, “Edge-based recognizer for arabic sign language alphabet (ars2v-arabic sign to voice),” Dec. 2010, pp. 121–127. DOI: 10.1109/ICENCO.2010.5720438.

[25] M. Kamruzzaman, “Arabic sign language recognition and generating arabic speech using convolutional neural network,” *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–9, May 2020. DOI: 10.1155/2020/3685614.

[26] Y. Saleh and G. Issa, “Arabic sign language recognition through deep neural networks fine-tuning,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, p. 71, May 2020. DOI: 10.3991/ijoe.v16i05.13087.

[27] M. Maraqa and R. Zitar, “Recognition of arabic sign language (arsl) using recurrent neural networks,” Sep. 2008, pp. 478–481. DOI: 10.1109/ICADIWT.2008.4664396.

[28] O. Al-Jarrah and A. Halawani, "Recognition of gestures in arabic sign language using neuro-fuzzy systems," *Artif. Intell.*, vol. 133, pp. 117–138, Dec. 2001. DOI: 10.1016/S0004-3702(01)00141-2.

[29] *Hand landmarks detection guide | mediapipe | google for developers*. [Online]. Available: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker.

[30] M. Al-Barham, A. Alsharkawi, M. Al-Yaman, *et al.*, *Rgb arabic alphabets sign language dataset*, 2023. DOI: 10.48550/ARXIV.2301.11932. [Online]. Available: https://arxiv.org/abs/2301.11932.

[31] *GELU*, https://paperswithcode.com/method/gelu, Accessed:26 06 2023.

[32] *Numpy*, https://en.wikipedia.org/wiki/NumPy, Accessed: [26-06-2023].

[33] *Tensorflow*, https://fr.wikipedia.org/wiki/TensorFlow, Accessed: [26-06-2023].

[34] ActuIA, *Keras - actuia*, https://www.actuia.com/keras/, Accessed: [26-06-2023].

[35] *Opencv*, https://fr.m.wikipedia.org/wiki/Fichier:OpenCV_Logo_with_text.png#/search, Accessed: [26-06-2023].

[36] *Scikit-learn*, https://fr.m.wikipedia.org/wiki/Fichier:OpenCV_Logo_with_text.png#/search, Accessed: [26-06-2023].

[37] A. Vidhya. "Pose detection in image using mediapipe library." (2022), [Online]. Available: https://www.analyticsvidhya.com/blog/2022/03/pose-detection-in-image-using-mediapipe-library/.