الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدةد. مولاي الطاهر

كلية العلوم

قسم: الإعلام الآلي

Mémoire de Master

Spécialité : MICR

Thème

HAND GESTURE RECOGNITION USING DEEP LEARNING

Présenté par :

BENYOUCEF Somia

NEHAL Mohamed Feth Allah

Dirigé par :

Dr. LOKBANI Ahmed Chaouki

Dr. RAHMANI Mohamed Elhadi

Promotion 2023 - 2024

Motivation :

Body language, encompassing both facial expressions and hand gestures, forms a vital part of human communication. However, for individuals who are deaf or hard of hearing, sign language serves as their primary mode of expression. This can create significant barriers when interacting with those who don't understand sign language.

Researchers are actively developing technologies to bridge this gap through the field of Sign Language Recognition (SLR). These systems aim to interpret sign language and translate it into spoken or written language, and vice versa, fostering more inclusive communication.

This research contributes to the advancement of SLR by proposing a deep learning-based system for recognizing both American Sign Language (ASL) and Arabic Sign Language (ArSL).

The proposed system boasts faster letter recognition and a simpler design, with the potential to significantly improve communication accessibility for the deaf and hard of hearing community across the globe.

Acknowledgements

من قال أنا لها ... نالها

و أنا لها و إن أبت رغما عنها أتيت بها

نلتها و عانقت اليوم مجدا عظيما لم يكن الحلم قريبا و لا الطريق سهلا و لكن بفضل الله... وصلت

الحمد لله حبا و شكرا و امتنانا, الحمد لله بفضله أدركت أسمى الغايات

أهدي بكل حب مذكرة تخرجي

إلى **نفسي** العظيمة الفتية التي تحملت كل العثرات و أكملت رغم الصعوبات

إلى أعظم الأشخاص, و أعز الناس على روحي, سندي و ملاذي بعد الله, فخري و اعتزازي... **أمي**

إلى من دامت لي أياديهم وقت ضعفي, إلى ضلعي الثابت و أمان قلبي... أخواتي (**خولة, أسماء, شيماء, ريماس**) و أخي (**وليد**)

إلى من ساندني بكل حب وقت ضعفي ...

إلى كل من أعطاني يد العون من قريب أو بعيد و ساعدني في هذا المشوار و بالأخص أذكر أستاذي **مجدوبي عبد القادر**

بن يوسف سمية

This accomplishment would not have been possible without support of my supervisors, Dr RAHMANI Mohamed Elhadi and Dr LOKBANI Chaouki . I would like to express my sincere appreciation thankfulness to my supervisors for their support, advice and help during the period of my thesis research.

I am literally thankful to my parents who were besides me from beginning of thesis journey till the end.

I would love to thank all my friend TOUMI Karim who was beside me during my research and supporting me when I challenge the difficulties of life

NEHAL Mohamed Feth Allah

Table of Contents:

Liste of Figures:

Liste of tables

List of Acronyms

HCI : human computer interaction
NN : Neural network
CNN : convolutional neural network
ANN : artificial neural network
AI : artificial intelligence
ARSL : Arabic sign language
ASL : American sign language
DCT : discrete cosine transform
HMM : hidden Markov model
KNN : K-nearest Neighbor
MLP : multi-layer perceptron
LMC : leap motion controller
SVM : support vector machine
LSVM : latent support vector machine
DOF :Degree of freedom
OGI : optical gas imaging
PDA : personal digital assistance
VE : virtual environment
MOCAP : motion capture
FACS : facial action coding system
CMOS : complementary metal-oxide-semiconductor
BSL : British sign language
AUSLAN : Australian sign language
LSF : French sign language sign language
JSL : Japanese Sign language
EMG :electromyography
EEG : electroencephalogram
GPU : graphic processing unit
ADAM : adaptive moment esitimation
RMSPROP : root mean squared propagation
ADADELTA : an adaptive learning rate method
RGB : red green bleu

Introduction:

a) Preface :

Human communication is a symphony of spoken words, facial expressions, and physical cues. Among these, gestures play a particularly significant role for Deaf communities.

A Gesture is defined as the physical movement of the hands, fingers, arms and other parts of the human body through which the human can convey meaning and information for interaction with each other [1].

Technology is playing a growing role in bridging the communication gap between Deaf and hearing communities. Vision-based approaches similar to those described earlier are being explored to recognize and translate sign language gestures into spoken language or text. This holds immense potential for facilitating real-time communication and promoting greater inclusivity.

For human computer interaction (HCI) interpretation system there are two commonly approaches [2]:

Data Glove Approach: This method utilizes specialized gloves equipped with sensors that track hand and finger movements. By translating these movements into digital signals, the computer can interpret specific gestures and respond accordingly. This approach offers high precision but can be cumbersome and expensive, limiting its widespread use.

Vision-Based Approach: This method leverages camera technology to capture and analyze hand and body movements. Computer vision algorithms then classify these movements as specific gestures. The vision-based approach is more natural and adaptable, as it doesn't require users to wear special equipment. However, factors like lighting and background complexity can affect accuracy.

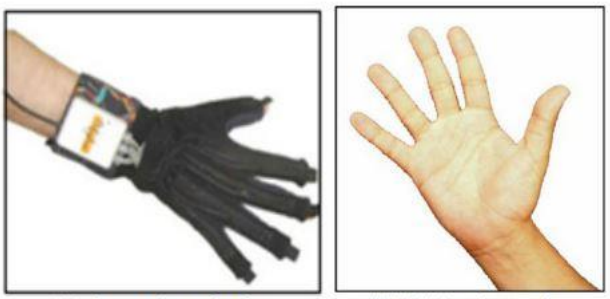

FIGURE 1: Examples of data glove and vision based [2]

As illustrated in Figure 2, hand gestures for human–computer interaction (HCI) started with the invention of the data glove sensor. It offered simple commands for a computer interface [3]. The gloves used different sensor types to capture hand motion and position by detecting the correct coordinates of the location of the palm and fingers [3]. Various sensors using the same technique based on the angle of bending were the curvature sensor , angular displacement sensor, optical fiber transducer , flex sensors and accelerometer sensor [3]. These sensors exploit different physical principles according to their type.

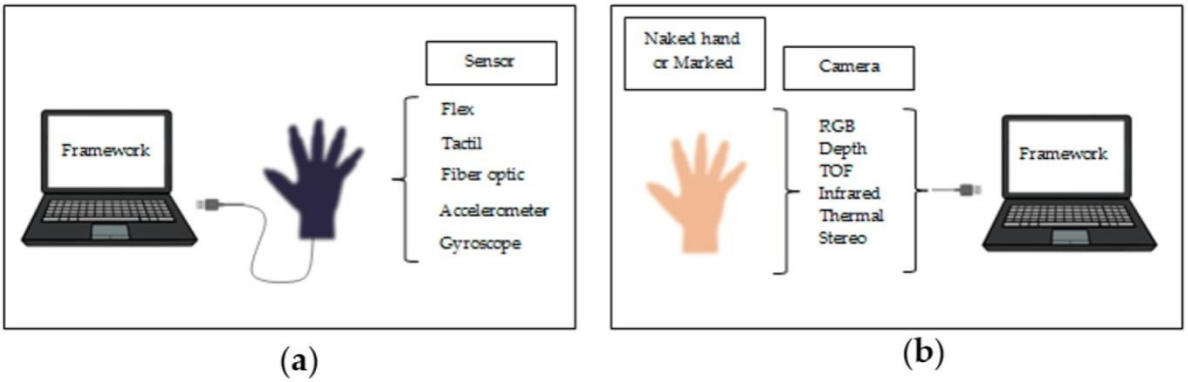

Figure 2. Different techniques for hand gestures. (**a**) Glove-based attached sensor either connected to the computer or portable; (**b**) computer vision–based camera using a marked glove or just a naked hand. [3]

Sign language is one of the common examples for a hand gesture system. It is defined as a linguistic system based on hand motions besides other motions. For instance, most hearing impaired people around the world use universal sign language. Sign language contains three fundamental parts: word level sign vocabulary, non-manual features and finger spelling [8]. One of best methods to communicate with hearing-impaired people is sign language.

Recently, sign language may be achieved by some types of robotics using some appropriate sensors used on the body of a patient [4].

The phases used by most studies to carry out their experiments are similar. Pre-processing, or just getting the image prepare to go to the next phase, is the first stage that most research use. After that, image processing becomes ready to receive the entire image so that image processing tools can be used to track it. Many classifiers, including Neural Networks (NN) and Convolutional Neural Networks (CNN), are released by artificial intelligence. These classifiers may classify data based on their configuration and capabilities.

b) Problem Definition:

The communication barrier between Deaf and hearing individuals is a challenge many of us strive to overcome. While traditional methods of learning sign language can be demanding, technological advancements offer exciting possibilities for fostering inclusivity

Our goal is to develop a model capable of recognizing hand gestures representing the alphabets in both Arabic Sign Language (ArSL) and American Sign Language (ASL). This model would then translate these gestures into text, effectively bridging the gap between sign and spoken languages.

This project acknowledges that learning sign language can be a complex endeavor, and we believe technology should play a supporting role. By creating a tool that recognizes and interprets basic signs, we can empower both Deaf communities and those seeking to connect with them.

c) Aims and Objectives:

Our project aims to empower Deaf individuals by fostering more straightforward communication. We envision a future where a seamless bridge exists between Deaf and hearing communities. This bridge will be built through an accurate, automated model using deep learning to translate sign language alphabets into text.

We are committed to building upon the successes of previous research in this field. By carefully analyzing existing models, we aim to optimize performance and create a robust prototype for real-world testing. This iterative process will involve ongoing feedback analysis to identify and rectify any errors in the model's interpretation.

d) Thesis Outline and Chapters' Summary:

The thesis covers six main chapters; the introduction of the thesis, A Survey of Recent Gesture Recognition Research (literature review) , the introduction of gesture recognition, Artificial intelligence , results and discussion and conclusion and future work

Introduction :

Introduction is an initial chapter, which presents the background of the research, the aim and objectives, the research original contributions, and the thesis outlines and chapters' summary.

Chapter 1 - Literature Review :

Chapter one shows the literature review of the Arabic sign language and American sign language , It explains every method applied in every study, including the application. Lastly, it also discusses Dataset acquisition and the Similarity of sign gestures in Arabic sign language and American sign language.

Chapter 2 - Gesture Recognition :

Chapter two presents the definition of gesture recognition. It explains the different recognition approach , The hand gesture recognition and its types also the sign language and Types of Sign Language Detection are discussed in this chapter.

Chapter 3 - : Model Architecture, Training And Testing:

Chapter three includes related work and Design Specification : Artificial Intelligence ,deep learning and It presents each models used in the study .

also shows the results of training and testing the data was trained using; CNN, AlexNet , Lenet , efficientnetb0 and Resnet will be compared. Also the Comparison of the results obtained by the proposed approach and other previous study

Conclusion - : Conclusion and Future Work:

An summary of all the research projects proposed in this thesis is given in chapter three

Chapter 1:

1. Literature Review:

1.1 Introduction:

The field of sign language recognition has witnessed a global surge in research, encompassing both established regions like those using American Sign Language (ASL) and emerging regions like the Arab world. Vision-based systems and sensor gloves are common approaches, with this study focusing on vision-based systems for Arabic Sign Language (ArSL) alphabet identification. We will conduct a focused literature review, examining the most relevant research published in the past decade. Specifically, our analysis will delve into ArSL alphabet recognition systems, exploring the dataset sizes employed and the algorithms and techniques utilized and also the ASL.

Figure 3 illustrates the various comparison parameters employed in this systematic literature review. These parameters include: acquisition mode (e.g., camera, glove), sign type (static or dynamic), signing mode (isolated signs, continuous signing), handedness (single or double-handed), classification techniques used, and average recognition accuracy. By analyzing these parameters, we conducted a comprehensive review of sign languages, including ASL and ArSL.

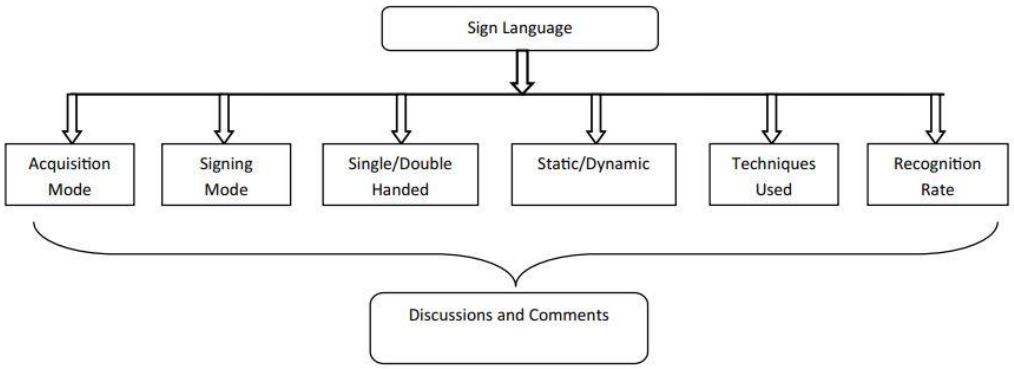

Figure 3 : Comparison parameters [5]

1.2 Arabic Sign Language:

Mohandes et al. [09] proposed an Arabic Sign Language (ArSL) recognition system that focused on identifying dynamic hand signs. They collected a dataset of 4500 samples and employed a region growing technique to track hand movements. To represent the signs, they extracted features such as centroid location, eccentricity of the bounding ellipse, angle of the first principal component, and hand area. The system relied on a 5-state Hidden Markov Model (HMM) for classification and achieved a peak accuracy of 97.3% when using an equal number of samples for training and testing.

Maraqa and Abu-Zaiter [10] presented a vision-based Arabic Sign Language (ArSL) recognition system designed to identify single-handed static signs. Their system utilized a camera to capture images and relied on a recurrent neural network for classification. The dataset they employed comprised a total of 1200 images, and their system achieved an accuracy of 95.11%

Al-Rousan et al. [11] introduced a camera-based sign language recognition system for isolated signs. Their system focused on recognizing dynamic double-handed signs from a set of 30 words. To represent the signs, they extracted features related to location, movement, and orientation using Discrete Cosine Transform (DCT) and zonal coding. A Hidden Markov Model (HMM) was employed for classification, achieving an accuracy of 93.8% in signer-dependent mode and 90.6% in signer-independent mode.

Shanableh and Assaleh [12] proposed a user-independent Arabic Sign Language (ArSL) recognition system. Their system leveraged a combination of camera and colored gloves to capture video recordings of 3450 isolated signs. The collected data underwent preprocessing using median filtering to reduce noise. Subsequently, features based on bounding boxes were extracted to represent the signs. Finally, a K-Nearest Neighbors (KNN) classifier was employed for sign recognition, achieving an accuracy of 87%.

Elons et al. [13] proposed an ArSL recognition system utilizing a Leap Motion sensor. This system focused on identifying dynamic, double-handed signs for isolated words. They extracted features based on finger position and the distances between fingers to represent the signs. For classification, they employed multilayer perceptron neural networks.

Mohandes et al. [14] investigated an Arabic Sign Language (ArSL) recognition system using a Leap Motion Controller (LMC) for data capture. They collected a dataset of 6400 single-handed static signs. The system employed two classification algorithms: Multi-Layer Perceptron (MLP) neural networks and Naïve Bayes classifiers. Interestingly, the study found that Naïve Bayes achieved superior performance compared to the neural network approach.

A summary of the reviewed ArSL recognition systems is presented in Table 1:

Author	Acquisition mode	Single/ double handed	Static/dynamic	Signing mode	Technique used	Recognition rate
Mohandes et al. [09]	Camera	Both	Dynamic	Isolated	HMM	97.3%
Maraqa and Abu-Zaiter [10]	Camera	Single	Static	Isolated	Recurrent neural network	95.11%
Assaleh et al. [15]	Camera	Both	Dynamic	Both	HMM	75% (sentence), 94% (Word)
Al-Rousan et al. [11]	Camera	Double	Dynamic	Isolated	HMM	93.8%
Shanableh and Assaleh [12]	Camera	Both	Dynamic	Isolated	KNN	87%
Mohandes et al. [14]	Camera	Double	Static	Isolated	HMM	95.2%
Dahmani and Larabi [16]	Camera	Single	Static	Isolated	KNN and SVM	DB1: 88.87%, DB2: 96.88%
Elons et al. [13]	Leap motion sensor	Double	Dynamic	Isolated	Multilayer perceptron neural networks	88%
Ahmed and Aly [17]	Camera	Both	Static	Isolated	HMM	99.97%
Mohandes et al. [18]	Leap motion sensor	Single	Static	Static	MLP neural networks and Naïve Bayes	MLP: 98%, Naïve Bayes:>99%
Tubaiz et al. [19]	Gloves	Both	Dynamic	Continuous	KNN	98.90%
Sarhan et al. [20]	Kinect	Both	Dynamic	Isolated	HMM	80.47%
Hassan et al. [21]	DB1: gloves, DB2: Polhemus G4	Both	Dynamic	Continuous	HMM and modifed KNN	DB2: 97% (word), 85% (sentence)
Hamed et al. [22]	Kinect	Single	Static	Isolated	SVM	99.2%
Darwish [23]	Camera	Single	Static	Isolated	HMM	92.40%

Table 1: Summarized review of Arabic sign language recognition systems

1.2.1 Discussion and Comments :

The data are analyzed and plotted the results in Figure 4.
Cameras are the dominant tool in ArSL research, used in 54% of studies. Kinect, gloves, and Leap Motion each account for 13%, while Polhemus trackers make up 7%.
Research in ArSL focuses heavily on dynamic signs (57%), with static signs following at 43% and it focuses on signs used with both single and double hands (50%), followed by single-handed signs (29%) and double-handed signs (21%).
Hidden Markov Models (HMMs) dominate ArSL research, used in 47% of the work. Hybrid techniques follow at 20%, while Neural Networks (NNs) and Support Vector Machines (SVMs) see the least use.

An analysis of ArSL recognition systems reveals a promising landscape: 70% of systems achieve accuracy exceeding 90%, with an additional 23% between 80 and 89%. Only a small minority (7%) fall below 80%.

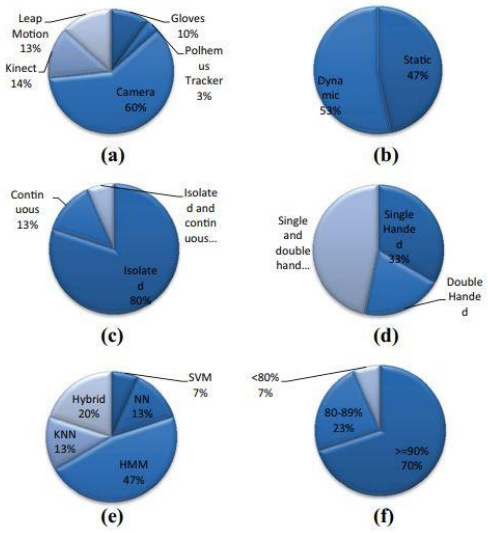

Figure 4: a Usage of different data acquisition techniques used in ArSL systems. b Research work carried out on static/dynamic signs in ArSL. c Percentage of research work carried out on the basis of signing mode in ArSL. d Percentage of research work carried out on the basis of single/double handed signs in ArSL. e Percentage of research work carried out on technique used for recognition of signs. f Accuracy of research for different ArSL systems

1.3 American Sign Language:

Oz and Leu (2011) [24] proposed a sensory glove-based system for translating American Sign Language (ASL) signs into English. Their system focused on recognizing static signs formed with one hand. By collecting data for 50 ASL signs and employing a neural network classifier, they achieved an accuracy of 90%.

Sun et al. tackled American Sign Language (ASL) recognition with two approaches. In one study [25], they built a Latent Support Vector Machine (LSVM) model using a combination of features extracted from Kinect data, including motion (optical flow) and shape information (Histogram of Oriented Gradients), achieving 86% accuracy for continuous signing. In another work [26], they focused on recognizing individual signs. Their system, based on discriminative exemplar coding, analyzed 1971 samples from 73 signs, extracting features related to body posture, hand shape, and movement. This system leveraged a combination of multiple instance learning Support Vector Machine (mi-SVM) for training and the AdaBoost algorithm for classification.

Chuan et al. [27] investigated American Sign Language (ASL) recognition using a Leap Motion sensor. They explored two machine learning algorithms for classification: K-Nearest Neighbor and Support Vector Machine. Their system achieved an accuracy of 72.78% with K-Nearest Neighbor and 79.83% with Support Vector Machine, demonstrating the potential of Leap Motion for ASL recognition.

AlQattan and Sepulveda [28] explored a neural network-based system for recognizing dynamic American Sign Language (ASL) signs. Their system focused on six single-handed signs. To analyze the signs, they used a technique called the Discrete Wavelet Transform for feature extraction. Finally, they employed two classification algorithms, Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM), achieving the best accuracy of 76% with SVM.

Kim et al.[29] achieved high accuracy (greater than 90%) for American Sign Language (ASL) recognition using a novel approach: impulse radio sensors combined with a Convolutional Neural Network (CNN) for classification.

Islam et al.[30] focused on real-time hand gesture recognition using a mobile camera. Their neural network (NN)-based system achieved an accuracy of 94.32% for a dataset of 1850 single-handed static signs.

Ferreira et al.[31] explored multimodal fusion for sign language recognition, achieving a top accuracy of 97%. Their system combined data from Kinect (color and depth) and Leap Motion sensors to train a Convolutional Neural Network (CNN) for classifying 1400 single-handed static signs.

A summary of the reviewed ASL recognition systems is presented in Table 2:

Author	Acquisition mode	Single/double handed	Static/dynamic	Signing mode	Technique used	Recognition rate
Munib et al. [32]	Camera	Both	Static	Isolated	Neural networks	92.33%
Oz and leu [33]	Gloves	Single handed	Dynamic	Isolated	Neural networks	95%
Oz and Leu [24]	Gloves	Single	Static	Isolate	Neural networks	90%
Ragab et al. [34]	Camera	Single	Static	Isolated	SVM and random	94%

					forest	
Sun et al. [25]	Kinect	Both	Dynamic	Continuous	Latent support vector machine	86%
Sun et al. [26]	Kinect	Both	Dynamic	Isolated	Adaboost	86.8%
Chuan et al. [27]	Leap motion sensor	Single	Static	Isolated	KNN and SVM	72.78% (KNN) 79.83% (SVM)
Tangsuksant et al. [35]	Camera	Single	Static	Isolated	Neural network	95%
Zamani and Kanan [36]	Camera	Single	Static	Isolated	Neural network	99.88%
Wu et al. [37]	Arm sensors	Single	Dynamic	Isolated	Decision tree, SVM, NN, Naïve Bayes	81.88, 99.09, 98.56, 84.11%
Aryanie and Heryadi [38]	Camera	Single	Static	Isolated	KNN	KNN 99.8% for k=3 best
Kumar et al. [39]	Camera	Single		Isolated	SVM	93% (static), 100% (dynamic)
Kim et al. [29]	Impulse radio sensor	Single	Static	Isolated	CNN	>90%
Islam et al. [30]	Camera	Single	Static	Isolated	ANN	94.32%
Karayilan and Kiliç [40]	Camera	Single	Static	Isolated	NN	85% (histogram features)
Ferreira et al. [31]	Kinect	Single	Static	Isolated	CNN	97%
Oyedotun and Khashman [41] Single	Camera	Single	Static	Isolated	CNN	91.33%

Table 2: Summarized review of American Sign Language recognition systems

1.3.1 Discussion and Comments :

The data are analyzed and plotted the results in Figure 4. The analysis found that cameras are the most common data acquisition technique, used in 44% of American Sign Language research. This is followed by Kinect (23%), armband (13%), gloves (8%), and a combined usage of Leap Motion, electroencephalogram, and impulse radio sensors (12%).

This study analysis revealed that 75% of research focused on single-handed signs in ASL, followed by 21% investigating both single and double-handed signs, and only 4% concentrating on double-handed signs alone.

We observed that neural networks (33%) are the dominant classification technique used in ASL research. Support Vector Machines (SVM) follow closely at 21%, with hybrid techniques also accounting for 21%. Convolutional Neural Networks (CNNs) are used in 13% of the research. Techniques like AdaBoost, KNN, and DTW see the least usage. And Analysis of this sign language recognition system accuracy reveals that 65% achieve an average accuracy exceeding 90%. Additionally, 23% of systems demonstrate accuracy between 80% and 89%, while only 12% fall below 80% accuracy

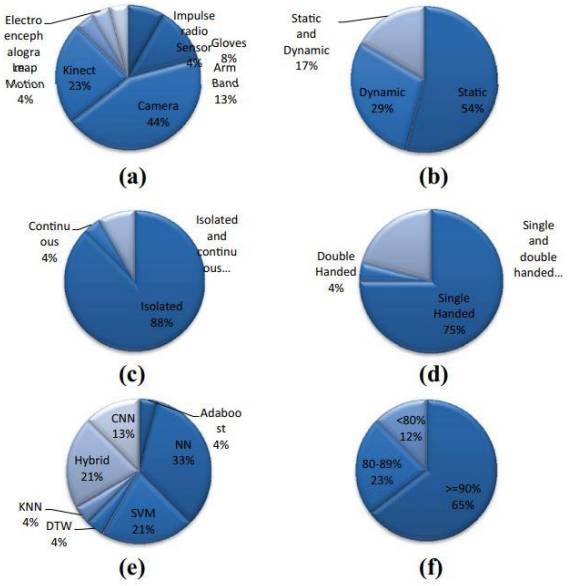

Figure 5: a Usage of different data acquisition techniques used in ASL systems. b Research work carried out on static/dynamic signs in ASL. c Percentage of research work carried out on the basis of signing mode in ASL. d Percentage of research work carried out on the basis of single/double handed signs in ASL. e Percentage of research work carried out on technique used for recognition of signs. f Accuracy of research for different ASL systems

1.4 Dataset acquisition :

1.4.1 Arabic Sign Language :

ArSL Datasets The availability of large and diverse datasets that capture the variability and complexity of sign language gestures is critical for the development and evaluation of ArSL recognition systems. These datasets typically contain video or sensor data of sign language gestures, along with corresponding annotations that indicate the meaning or label of each gesture. Although adequate video data is accessible online, it will be inappropriate for training ArSL recognition systems since it lacks annotations and the signs have not been segmented. The lack of a large-scale benchmarking dataset is a problem for ArSL recognition systems. It is challenging to locate a comprehensive dataset that meets the requirements of ArSL recognition. This is due in part to the scarcity of qualified ArSL specialists and the time and cost required to collect data on the sign language [42]. In addition, researchers may encounter challenges in obtaining valid ArSL datasets due to the inherent complexity of the Arabic language. Since some studies have created their own data that is generally limited or unavailable to other researchers, it may be difficult to directly compare the recognition accuracies of the various methodologies. Most of these datasets are also camera-based, meaning they lack depth information [42]. Therefore, the majority of researchers are required to manually generate datasets, which is a time-consuming and laborious procedure. Table 3 provides some examples of publicly available ArSL datasets.

Ref	
[54]/2011	Six gestures are used to generate 6,000 different sign images.
[55]/2012	The 80-word vocabulary is used to construct 40 sentences with no restrictions on syntax or sentence length; this process is repeated 19 times.
[56]/2013	There are 270 postures that make up the 200 gestures, with 189 postures involving two hands and 81 postures comprising only one hand. Every gesture is carried out ten times, each time by a different two person.
[57]/2014	There are a total of 2800 frames in the dataset generated from a single user's input of 28 alphabets, with 10 samples of each letter
[58]/2015	The database contains about five hundred static gestures, including "finger spelling, hand movements" (non-manual signs). Lip reading, body language, and facial expressions all play significant roles.
[59]/2016	Two sets of static alphabet data exist: 700 instances for each 28 characters written with naked hands and colored gloves.
[60]/2017	200 samples are taken from the unified ArSL lexicon, with each of the 25 signs being performed by two different signers four times. 125 for training and 75 for testing

[61]/2018	Thirty people are serious mobile photographers. Volunteers gesture these 30 ArSL alphabets. There are a total of 900 images spread across 30 letters.
[62]/2018	Captured 450 colorful ArSL videos
[63]/2019	28 Arabic letters and numerals (0-10) are represented by 7869 images for recognition.
[64]/2019	The dataset ArSL2018 comprises a collection of 54,049 images, which accurately depict the 32 alphabets and signs of ArSL. These images have been donated by a group of 40 signers.
[65]/2020	A total of 44 signs (29 single-handed and 15 double-handed) are executed by a group of 5 signers, where 80% are used for training and 20% for testing.
[66]/2021	There are 9240 images of the Arabic alphabet from 10 places and age groups. These images are organized in four separate datasets.
[67]/2021	There are eleven chapters totaling 502 signs that make up the words in the ArSL lexicon. Three signers are used for each sign. There are a total of 75300 samples, the result of 50 repetitions of each sign by each signer.
[68]/2021	There are a total of 220000 images in the dataset, split amongst 44 different classes (32 letters, 11 digits (0-10) and 1). There are 5000 images total, taken by various people, of each of the stationary signs.
[69]/2023	It contains 7,856 RGB images of ArSL alphabets. Data is collected from over 200 people in a wide range of shooting situations (including but not limited to: lighting, background, image orientation, size, and resolution).

Table 3: some examples of publicly available ArSL datasets.

1.4.1.1 Arabic sign language2018 :

The ArSL2018 [43] is the most up to date and comprehensive dataset of Arabic sign language images presented by a research team in Prince Mohammad Bin Fahd University, Al Khobar, Saudi Arabia. The data consists of 54,049 images for the 32 Arabic sign language sign and alphabets collected from 40 participants of different ages, in Grayscale data format and RGB format. The images are 64×64 pixels in (.jpg) format and were captured using a smart phone. Figure 5 shows the images of the used dataset.

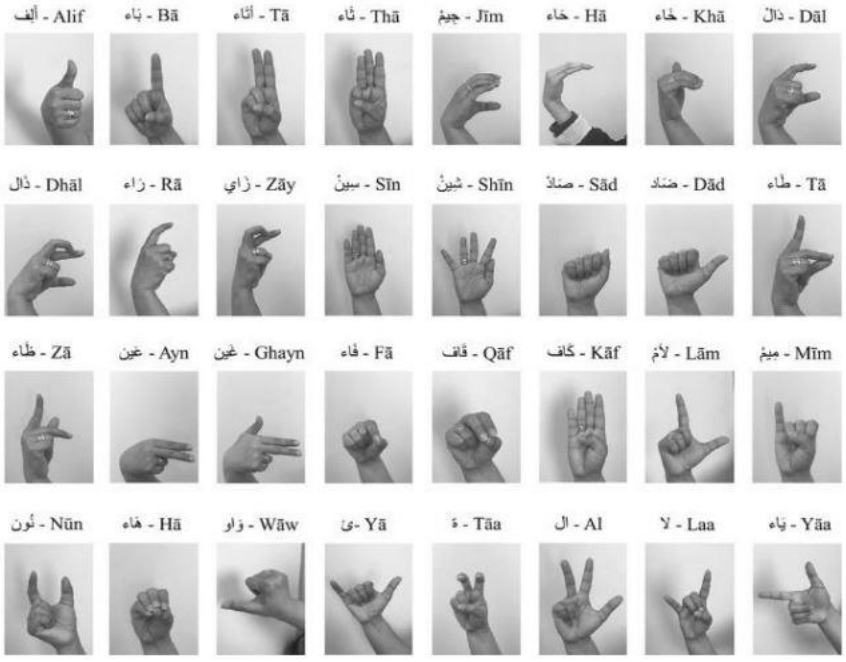

Figure 6 : representation of the Arabic sign language in the ArSL2018 dataset [6]

1.4.2 American sign language:

The system was trained on a dataset of 87,000 hand sign images obtained from a publicly available repository [44]. These JPG-format images are categorized into 29 folders, of which 26 are for the letters A-Z and 3 classes for SPACE, DELETE and NOTHING. with each folder representing a specific sign. Figure 3 provides a glimpse of some sample images from this dataset.

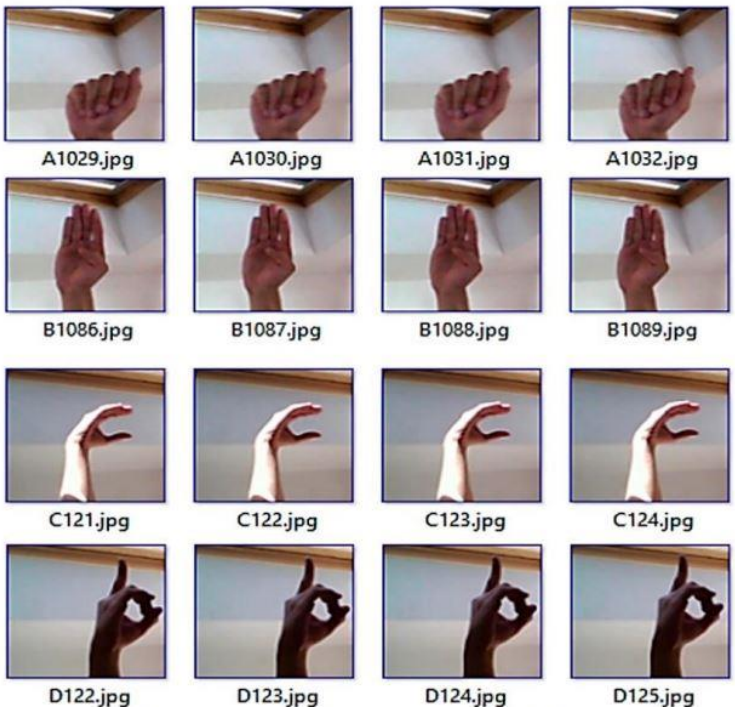

Figure 7 : Examples of American hand sign language [7]

1.5 Similarity of sign gestures in Arabic sign language and American sign language:

Similarities emerge across alphabets in English and Arabic Sign Languages. For instance, in American Sign Language (ASL), the hand gestures for letters A, M, and S share similarities. Likewise, the signs for (C and O) and (N and E) exhibit resemblances. Similarly, Arabic Sign Language features identical signs for "Dhal" and "Zay," while "Fa" and "Qaf" also show a resemblance (Figures 9). Interestingly, some signs even match in sound and form between the

two languages, such as "Lam" and "L" (Figure 8a), "Sad" and "S" (Figure 8b), and "Ya" and "Y" (Figure 8c).

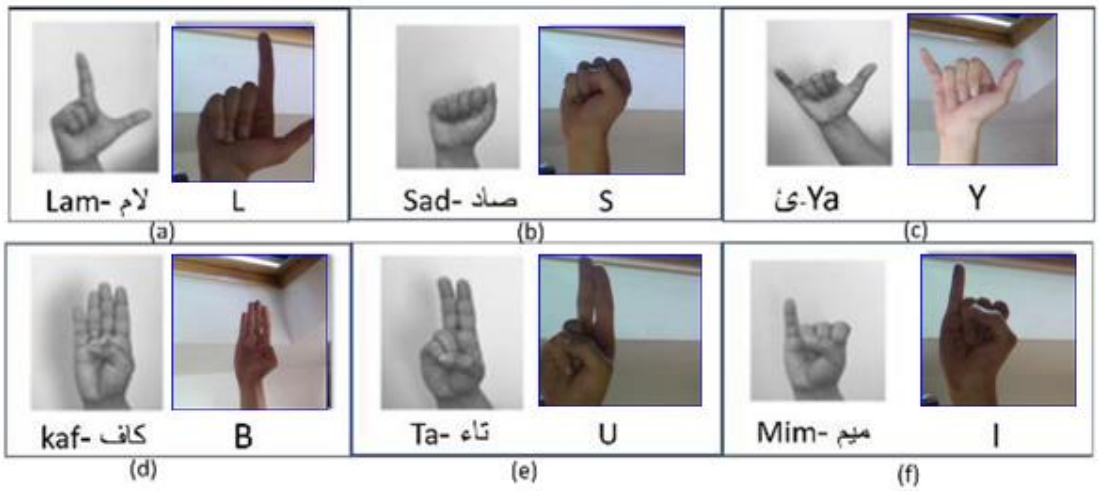

Figure. 8. Similarity of gestures in ASL and ArSL

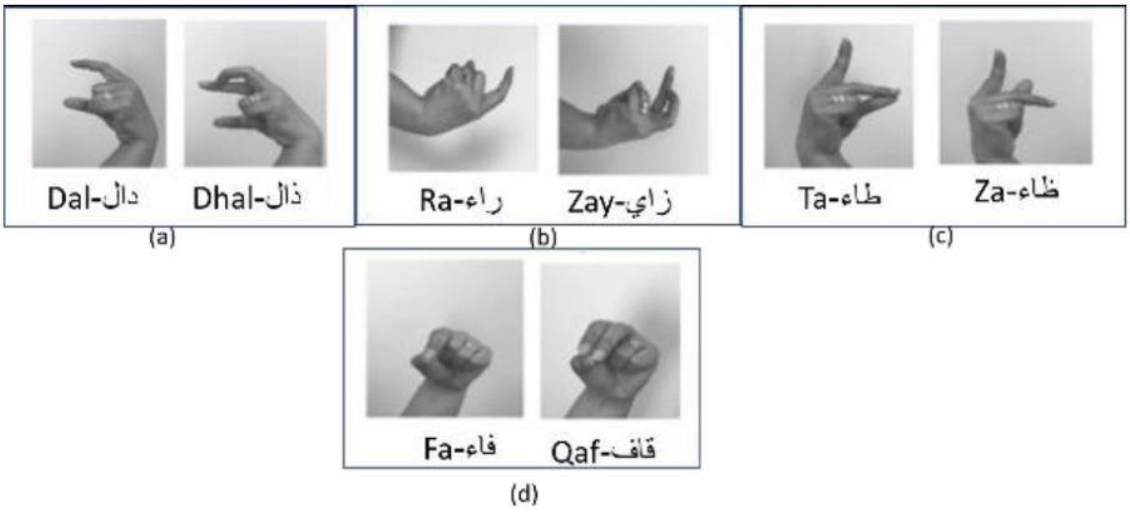

Figure 9. Similarity of gestures in ArSL

1.6 Conclusion :

This review explored techniques for sign language recognition, focusing on Arabic Sign Language (ArSL) alphabet identification and American Sign Language (ASL) research. Researchers have investigated various techniques like Support Vector Machines and Convolutional Neural Networks, achieving accuracies that vary depending on the approach. A significant challenge for both ArSL and ASL recognition is the limited availability of large, well-annotated datasets.

Chapter 2:

Gesture Recognition :

This chapter presents the definition of gesture recognition with its different approach, also the sign language and the type detection:

2.1 Definition of Gesture Recognition :

Gesture recognition is a computer or device's capacity to recognize and interpret human gestures as input. These gestures can be written symbols with the fingers as well as hand motions. Gesture recognition technology works by first capturing the gestures using cameras or other sensors, then analyzing and interpreting the recorded data using machine learning algorithms.

Many applications exist for gesture recognition technology. It can be utilized for entertainment purposes such as video games and virtual reality. Gesture recognition also allows for new methods to engage with interfaces, such as controlling a presentation or playing music by gesturing at a device.

Signs can be expressed in a multitude of ways by gestures, for example, sign language used by hearing impaired people. Other examples of gestures developed outside the computer field can be seen in use by traffic police, construction labours, and airport ground controllers.

Gestures can be static, which means that the user adopts a pose, or dynamic where the motion is a gesture by itself [45][46][47] . Attached devices such as gloves, data suits, Six Degrees of Freedom (6 DOF) trackers generally provide information along all the 3D geometries. For instance, hand and body gestures are used to by pilots to direct aircraft operations aboard aircraft carriers [45][46][47].

Mathematical models based on hidden Markov chains, or methods based on soft computing can handle gesture recognition [45][46][47]. The major advantage of using the hidden Markov model is the ability to recognize a variety of information for gesture recognition [45][46][47].

Any applied implementation of gesture recognition needs the use of diverse imaging and tracking devices or tools such as data gloves, body suits, and marker based optical tracking [45][46][47].

Gestures may be static or dynamic or both in certain cases such as sign language:

- **Static Gestures:** These gestures involve holding a specific hand shape for a brief period. They are simpler to recognize as they require less computational power because the hand position remains constant throughout the gesture. Static gestures are often used for representing individual letters in sign language, where each letter corresponds to a single image.

- **Dynamic Gestures:** In contrast, dynamic gestures rely on hand movement over time. They are more complex to recognize due to the continuous change in hand position. However, they are crucial for real-time sign language communication. Dynamic gestures involve tracking hand movement across multiple video frames, rather than relying on a single image.

There are many aspects that have been successfully used for many gesture recognition systems such as computer vision and pattern recognition techniques, including feature extraction, clustering, classification and object recognition. Analysis and detection of texture, shape, motion, color, image enhancement, optical flow, contour modeling and segmentation are image processing techniques that have been found to be effective

2.2 Different recognition approach:

The different recognition approaches studied are as follows:

2.2.1 PEN-BASED GESTURE RECOGNITION:

Pen and mouse gestures have been explored for decades as a way to interact with computers. The groundbreaking Sketchpad system (1963) utilized light-pen gestures, and commercial applications followed suit in the 1970s [70]. These gestures enhanced tasks like document editing, air traffic control, and design (e.g., spline editing).
More recent advancements, like the OGI Quick Set system, showcase the power of combining pen gestures with speech recognition for virtual environment control. Quick Set recognizes a wide range of gestures 68, including map symbols, editing tools, route and area indicators, and simple taps. Studies by Oviatt further highlight the significant benefits of using both speech and pen gestures for specific tasks [70]. Zeleznick and Landay and Myers developed interfaces that recognize gestures from pen-based sketching [50].

The past few years have seen a surge in Personal Digital Assistants (PDAs), starting with the Apple Newton and continuing with devices like the Palm Pilot and Windows CE models. Long and Rowe delved into the challenges and advantages of these gesture-based interfaces, offering valuable insights for interface designers.

While pen-based gesture recognition holds promise for various Human-Computer Interaction (HCI) environments, it relies on the availability and accessibility of a flat surface or screen. This can be a limiting factor in virtual environments, where users may not always have a suitable surface at hand

- techniques that allow the user to move around and interact in more natural ways are more compelling [50].

2.2.2 TRACKER-BASED GESTURE RECOGNITION

A variety of commercial tracking systems are available for gesture recognition, focusing on eye gaze, hand movements, and even full-body tracking. When it comes to interaction in virtual environments (VEs), each sensor offers unique advantages and limitations. While eye tracking interfaces hold potential, this discussion will delve into gesture-based input utilizing hand and body tracking.

2.2.3 DATA GLOVES

Hands, with their incredible dexterity (nearly 29 degrees of freedom), expressiveness, and convenience, are a natural choice for human-computer interaction. They can be powerful control devices, offering real-time, multi-dimensional control for complex tasks across various applications.

Sturman's analysis [51] highlights the importance of considering task requirements, hand capabilities, and device features when developing whole-hand input techniques. He proposes a taxonomy that categorizes techniques based on hand action style (continuous or discrete) and interpretation (direct, mapped, or symbolic) [51]. The optimal style depends on the specific interaction task, as discussed by Mulder [50] in his overview of hand gestures in HCI.

Several commercially available devices, known as data gloves, can be used to measure hand configuration and movement with varying degrees of precision, accuracy, and completeness.

Few advantages of data gloves, direct measurement of hand and finger parameters, provision of data, high sampling frequency, easy if use, line of sight, low cost version and translation independency feature of data.[70]

However with advantages of data glove there are few disadvantages like difficulty in calibration, reduction in range of motion and comfort, noise in inexpensive system, expensiveness of accurate system [70]. Moreover it's compulsory for user to wear cumbersome device. Many projects have used hand input from data gloves for "point, reach, and grab" operations or more sophisticated gestural interfaces. [70]

Despite these limitations, data gloves have been used effectively for various hand interaction tasks, including basic "point, reach, and grab" operations and more sophisticated gesture-based interfaces. Latoschik and Wachsmuth's work showcases a multi-agent architecture for detecting pointing gestures, while Väänänen and Böhm explore a neural network system for recognizing static gestures with the ability for user-defined additions [50]. Böhm et al. further expanded this

approach to recognize dynamic gestures [50]. The HIT Lab's Glove GRASP library exemplifies a software solution for integrating gesture recognition into applications, offering features like user-dependent training and one/two-handed gesture support [50]. A commercial version of this library is also available.

2.2.4 BODY SUITS

Motion capture (mocap) systems can analyze human movement by tracking strategically placed markers on the body. These markers enable recognition of postures, gestures, activities, and even identities.

One common approach involves attaching markers to the body and using optical cameras to measure their 3D positions over time. This data is then processed to reconstruct the body's movements and joint angles, creating a digital representation of the person's actions. Another method, articulated sensing, utilizes electromechanical sensors to directly measure joint angles. While some systems employ small markers attached to clothing, this discussion focuses on body suits, which offer a more comprehensive solution.

Body suits have advantages and disadvantages similar to those of data gloves. At high sampling rate it provides reliable results but they are cumbersome and very expensive.

While these limitations exist, body suits remain a powerful tool for motion capture in various applications, particularly when high-fidelity data is essential.

2.2.5 HEAD AND FACE GESTURES:

When people interact with one another, they use an assortment of cues from the head and face to convey information [70]. These gestures may be intentional or unintentional, they may be the primary communication mode or back channels, and they can span the range from extremely subtle to highly exaggerate.[70] Some examples of head and face gestures include: nodding or shaking the head, direction of eye gaze, raising the eyebrows, opening the mouth to speak, winking, flaring the nostrils and looks of surprise, happiness, disgust, anger, sadness, etc [71].

People display a wide range of facial expressions. Ekman and Friesen developed a system called FACS for measuring facial movement and coding expression; this description forms the core representation for many facial expression analysis systems [72].

A real-time system to recognize actions of the head and facial features was developed by

Zelinsky and Heinzmann, who used feature template tracking in a Kalman filter framework to recognize thirteen head/face gestures [72].

Essa and Pentland used optical flow information with a physical muscle model of the face to produce accurate estimates of facial motion. This system was also used to generate spatio-temporal motion-energy templates of the whole face for each different expression - these templates were then used for expression recognition [50].

2.2.6 HAND AND ARM GESTURES:

These two parts of body (Hand & Arm) have most attention among those people who study gestures in fact much reference only consider these two for gesture recognition [70]. The majority of automatic recognition systems are for deictic gestures (pointing), emblematic gestures (isolated signs) and sign languages (with a limited vocabulary and syntax). Some are components of bimodal systems, integrated with speech recognition. Some produce precise hand and arm configuration while others only coarse motion [50].

The recognition of hand and arm gestures has been applied to entertainment applications [50]. Freeman developed a real-time system to recognize hand poses using image moments and orientations histograms, and applied it to interactive video games. Cutler and Turk described a system for children to play virtual instruments and interact with life like characters by classifying measurements based on optical flow [50].

2.2.7 BODY GESTURES:

This section dives into full body motion analysis, encompassing gesture recognition and broader human activity understanding. Activities can span longer durations than gestures; for instance, two people meeting, conversing, and parting ways could be classified as a recognizable activity. Bobick proposed a taxonomy classifying motion into three levels: "Movement" (basic elements), "Activity" (sequences of movements or static postures), and "Action" (high-level contextual understanding). Most research to date has focused on the first two levels [50] .

The Pfinder system, adept at body tracking and gesture recognition, has been used in various applications. It creates a 2D body representation using statistical color and shape models. This paves the way for applications in video games, interactive dance, navigation, and virtual character interaction. Researchers have also combined Pfinder with speech recognition for virtual world manipulation and navigation, or used it for interactive dance performances where body movements trigger musical changes. Motion analysis systems in virtual environments hold promise for rehabilitation and athletic training. For example, gait recognition systems could be

used to assess rehabilitation progress. Additionally, view-based approaches using "temporal templates" to capture motion history have been explored for interactive children's environments.

2.2.8 VISION-BASED GESTURE RECOGNITION:

Vision-based interfaces are gaining traction for gesture recognition due to their unobtrusive nature. Unlike cumbersome tracker systems, they use cameras to capture and interpret human motion. While challenges like occlusions (hidden body parts) exist, vision offers advantages. Cameras can serve multiple purposes beyond gesture recognition, and advancements in CMOS technology promise miniaturized, low-cost options. Feature extraction techniques analyze captured images to classify gestures, as demonstrated by our own hand gesture recognition project using a webcam. This approach holds promise for a future of natural and immersive human-computer interaction.

2.3 Fingerspelling Definition:

Fingerspelling, using handshapes to represent letters of the alphabet, is a crucial tool for deaf individuals. It bridges communication gaps by providing a way to spell out new terms, names lacking established signs, or written words. Each letter has a distinct hand form, making fingerspelling a clear visual representation of written language. Research suggests it's the fastest method for deaf people to visually recognize words [53]. Moreover, fingerspelling builds on existing literacy skills, making it easier to learn after mastering lipreading, pronunciation, and writing the alphabet [53]. Therefore, fingerspelling serves as a valuable supplement to Sign Language, facilitating learning and communication for deaf individuals.

2.4 Application Areas of Hand Gesture Recognition Systems:

Research in hand gesture recognition has become a hot topic due to its potential for natural and intuitive interaction with technology. This approach offers a significant advantage over traditional methods that rely on various devices like mice, keyboards, touch screens, joysticks, and console controllers. These conventional interfaces can be cumbersome and expensive, particularly when data gloves are involved. Figure 10 shows the most common application area deal with hand gesture recognition techniques

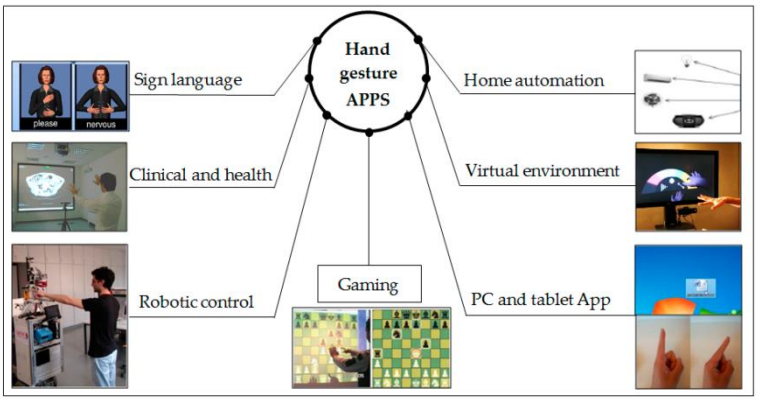

Figure 10 : most common application area of the hand gesture interaction system recognition techniques [3]

2.4.1 **Sign language :**

Sign language is a visual language that conveys meaning and communicates with others through the use of hand gestures, facial expressions, bodily movements, and other visual cues [76]. It is also defined as a visual language that is accessible to people who are deaf or hard of hearing, and also used by others to communicate with them [76].

2.4.2 **Types of sign language :**

There are various types of sign languages in the world [76]. There are over 140 distinct sign languages used throughout the world that developed independently in different communities. There are regional sign languages too. Every person can have their own sign language according to the regional variations of the language [76]. Sign languages are not universal so people who generally speak different languages can speak in the same sign language [76]. Following are few types of sign languages and where are they used in:

1. American Sign Language (ASL) - used in the United States and Canada

2. British Sign Language (BSL) - used in the United Kingdom

3. Australian Sign Language (Auslan) - used in Australia

4. French Sign Language (LSF) - used in France

5. Japanese Sign Language (JSL) - used in Japan

2.4.3 **Types of Sign Language Detection :**

Different Sign Language Detection Types There are two distinct methods for recognizing sign language, and it's obvious that they differ from one another—one being harder to identify than the other. In the initial scenario, often referred to as isolated sign language recognition, the system is taught to identify just static input, such as a single motion. The technology is able to identify a word or phrase by recognizing a certain gesture, digit, or letter of the alphabet. In the second case, in continuous sign language recognition the system can recognize and synthesize whole sentences [75]. Even though a lot of study has been done in the past, there are still many shortcomings (weaknesses) that must be fixed in order to overcome the problems. Some of these challenges include:

- When labeling words, isolated SLR approaches should be as precise as feasible.

- In continuous SLR methods, the program is mainly based on the isolated systems with time segmentation as preprocessing, which is non-trivial and this can cause many errors in the successive steps and eventually in the sentence synthesis[75]

- The devices that have to be used to create sign language recognition systems are expensive and it is difficult for the public to create a good model to commercialize sign language recognition systems.[75]

- The camera we use on our computer doesn’t give the best quality of video, so the dataset gets collected might have degraded quality.[75]

- Acquiring data from sensors could be very helpful in creating such a technology but problems arise such as noise, poor human handling, poor ground connection etc.[75]

- Vision based methodologies may occur plenty of inaccuracies because hands and fingers are overlapping; this problem could be solved by some 3D visualization of the mentalist.[75]

2.5 Conclusion:

Gesture recognition has become an effective method for human-computer interaction. A variety of techniques provide distinct capabilities, ranging from camera-based vision systems to data gloves and body suits. One of the main applications is sign language recognition, where studies are being done on both single signals and whole sentences. Deep learning techniques have shown promise, but difficulties remain. By solving these obstacles, gesture recognition will progress in the future. Through the creation of large datasets, investigation of depth sensors to overcome camera constraints, and possible integration of vision and sensor-based methods, scientists can usher in a new era of reliable gesture detection. This will not only empower sign language communication but also revolutionize human-computer interaction across diverse fields.

Chapter 3:

Proposed Approaches and Result discussion:

3.1. Introduction:

This project aims to build a model that can convert the alphabetical image in ArSL and ASL to the corresponding written letter in the Arabic language (American language) by applying some algorithms that can extract features and differences in the image. Choosing the right algorithm depends on the nature of data, complexity, and required resolution of the images.

Before constructing any code, we should do a general flowchart or pseudo code to plan the model's steps to know the path on it to achieve the task. It helps to be systematically through programming the code. Figure 11 shows the general flowchart for the project:

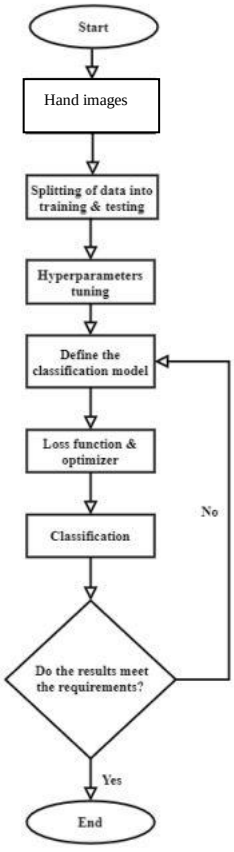

Figure 11 : flowchart of recognition model

The explanation of the flowchart will be in the following steps:

- **Splitting Data:** We need to split data perfectly into training, and testing datasets. These datasets assure to generalize the data and examine the model performance.

- **Hyperparameters Tuning:** The models in deep learning have several parameters like; the number of epochs, the number of batches, learning rate, and dropout. Parameter's manipulation can make the model better or worse, and selecting parameters depends on the experience or testing several trials until satisfying the results.

- **Defining the Model:** It means constructing the model's architecture, including the type of layers, number of layers.

- **Loss Function and Optimizer:** To assist the model, we need a loss function to ensure that the training is doing well. Also, we need an optimizer to update the weights inside the network.

- **Classification:** At this stage, the model is trained and tested. After that, we see the results if they satisfy our target or not

3.2. Related work:

Figure 12 illustrates the typical workflow of a sign language identification system. Data capture, a crucial step, can be achieved through various techniques. These techniques include cameras, data gloves , and sensors such as motion sensors, EMG sensors, or EEG sensors.[74,75]

Data for analysis can come in various forms like images, signals, or video streams. Before processing, a preprocessing step is often necessary. This step aims to remove unwanted noise or clutter from the data. Techniques like filtering, cropping, or resizing images can be used to achieve this, ultimately improving the quality and accuracy of the analysis . For image data, segmentation may be employed to isolate the sign itself from the background. This step helps focus the analysis on the relevant information. Once preprocessed, the system extracts features from the data. These features can be one-dimensional signals (like time series from sensors) or two-dimensional characteristics captured in images.

Classifiers play a vital role in deciphering sign language alphabets across various languages, as shown in the reviewed literature. Researchers have explored a diverse set of classification techniques, including artificial neural networks (ANN) known for their ability to learn complex patterns, support vector machines (SVM) offering strong performance in high-dimensional data,

hidden Markov models (HMM) adept at handling sequential data like gestures, tree-based classifiers for creating decision trees based on features, and the simpler yet effective K-nearest neighbor (KNN) approach that classifies signs based on their similarity to stored examples. This variety of classifiers demonstrates the ongoing exploration of optimal methods for accurate sign language recognition.

Deep learning architectures are established players in machine learning research and are experiencing a surge in popularity due to their effectiveness in real-world applications. Notably, supervised learning tasks often leverage Convolutional Neural Networks (CNNs) for image analysis

Figure 13 illustrates the workflow of automatic sign language classification using deep learning. Unlike traditional methods that separate feature extraction and classification, deep learning architectures like Convolutional Neural Networks (CNNs) can perform both tasks simultaneously. Convolutional layers within CNNs act as feature extractors, automatically identifying key characteristics from sign language images.

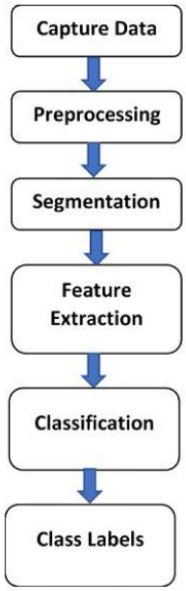

Figure 12 : flow diagram of sign language identification

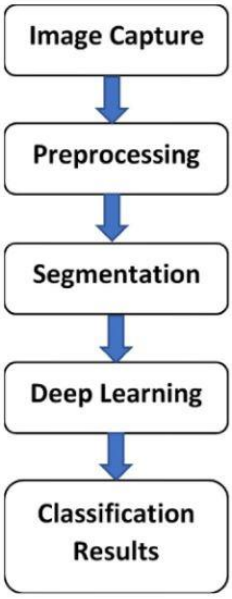

Figure 13 : flow diagram for sign language identification using deep learning architectures

3.3. Design Specification :

3.3.1. Artificial Intelligence :

AI is the ability of a machine to perform cognitive tasks and act intelligently. The field of AI tries to understand intelligent entities [74].

AI is a new discipline that began in 1956. With a help of AI, it is possible for machines to learn from their own experience, adapt to new inputs and perform human-like tasks. AI is widely used in finance, education, healthcare, transportation fields and in other industries such as computer vision, medical diagnosis, robotics and remote sensing [77].

Alan Turing, a pioneer in both computer science and artificial intelligence, proposed the 'Turing test' in 1950. This test aimed to define intelligence based on a machine's ability to exhibit human-like conversation.

Computers and robots can exceed the human ability at some tasks that are considered to be 'intelligent' using techniques such as data mining and pattern recognition etc. Lower cognitive tasks that are natural for humans can be extremely complex for machines. For example, a vision system and object recognition, partially concealed objects, same object, different shape, color, texture and size consistency [77].

3.3.2. Deep Learning :

Deep learning is introduced in Artificial Intelligence (AI) as a sub-group of machine learning. For instance, Deep learning is the fundamental technology that enables driverless cars to detect people and traffic signals. This principle also underlies the ability to recognize voices and audio on various devices, like tablets and cell phones. Deep learning is becoming well-known for its ability to complete tasks that were previously impossible. A deep learning model is built on the division of data into discrete, small-scale classification layers, which may include text, images, or audio.

Deep learning algorithms have offered effective solutions for picture recognition problems. Convolutional neural networks (CNNs) are a class of deep neural networks that are frequently used in computer vision applications in the field of deep learning.

3.3.3. Define the Model :

We trained many models to classify the letters. Many models outperform others, and problems found in one model can be solved in another. Our issue is deemed a complicated problem, hence typical ways cannot reach our aim.

3.3.3.1. CNN Model [78]:

A Convolutional neural network (CNN) is a type of artificial neural network specifically designed for image recognition. A neural network following the activity of human brain neurons is a patterned hardware and/or software system. CNN is also defined as a different type of multi-layer neural network and each layer of a CNN converts one amount of activations to another through a function. CNN is a special architecture used for deep learning . CNN is frequently used in recognizing scenes and objects, and to carry out image detection, extraction and segmentation

CNN has two phases: training and inference. CNN architecture consists of three layers: convolutional, pooling, and fully connected. The first layer is a convolutional layer, which is the primary component of CNN. It applies multiple filters to an image and generates unique activation features. The second layer is pooling, which is utilized for down sampling. The algorithm uses non-linear activation to get input and outputs based on window size. The last layer is fully connected and identifies a target to define the category of output. The three layers eliminate the need for feature extraction through image processing, allowing CNN to learn directly from visual input.

a. Convolution Layers [8] :

It is an essential component of the CNN architecture used for feature extraction. The neurons in the first convolution layer do not connect to every neuron in the input data, but neurons in one layer are connected to other neurons in their receptive field (specific patterns in small regions of the visual field). Each neuron in the second convolution layer links neurons located within a small rectangle in the first convolution layer, The first layer in the architecture is responsible for extracting some features, the next layer for other features, etc.

b. Feature Maps [8] :

The convolutional layer can be represented in 3D which every layer has multiple feature map. A feature map is considered a filter that explores features like vertical lines and horizontal lines
The equation that shows how the convolutional layer computes the output of a given neuron is:

$$Z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} X_{i',j',k'} \cdot W_{u,v,k',k} \qquad with \begin{cases} i' = i \times s_h \times u \\ j' = j \times s_w \times w \end{cases}$$

Where:

- $Z_{i,j,k}$: the neuron's output in row i, column j in feature map k of the convolutional layer.
- $X_{i',j',k'}$: the output of the neuron in layer L – 1, row i', column j'.
- b_k : the bias term in layer L.
- $W_{u,v,k',k}$: the connection weights.

c. Pooling Layers [8]:

This layer decreases the computations by shrinking the inputs. The pooling layer is like the convolutional layer, connected partially to the previous layer's outputs, located with the small rectangle receptive field. There are two forms of pooling: Max pooling and Mean pooling. Max pooling is the most popular form, which takes the maximum value in the higher-level feature layer. Mean pooling takes the average of all the elements in the higher-level feature layer.

d. Dropout Layer [8] :

A huge number of parameters in the network gives it the flexibility to tend to overfit. One solution to avoid overfitting is using the early stopping technique, where the network stores the parameters at the best values when the validation set worsens. With unlimited computations, the early stopping technique will be aggressive and consume more time so, the best way to avoid overfitting is using the regularizes. Dropout is one of the most regularization techniques. The term dropout refers to dropping out the neuron and incoming and out coming connections temporarily from the network, as shown in Figure 14.

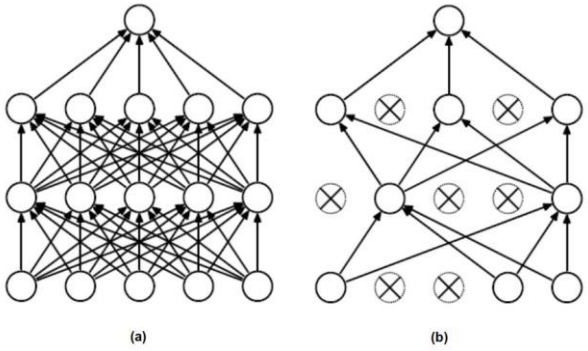

Figure 14 : (a) Network Without Dropout, (b) Network With Dropout

e. Fully Connected Layer :

This layer transforms the last convolutional layer into a one-dimensional array and connects to one or more dense layers, in addition to a dropout layer after each dense layer, with a 0.5 dropout rate will reduce overfitting. A non-linear activation function follows the final fully connected layer to estimate inputs classification according to the output probabilities

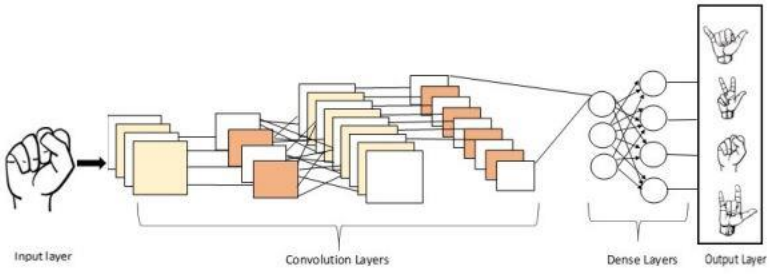

Figure 15 : Basic CNN Architecture

Figure 15 shows the general architecture of CNN. It is noticed that there are convolutional layers followed by a ReLU function or others, then another pooling layer, and so on. The previous steps are considered feature layers and make the image smaller and smaller through the architecture until it reaches the classification layers. Stages do the classification: flattening the last layer to pass it into a fully connected layer and then passing the fully connected layer into softmax function to classify the images according to the estimated probabilities.

3.3.3.2. AlexNet [79] :

AlexNet was the first convolutional network to employ the graphics processing unit (GPU) to improve performance. AlexNet has five convolutional layers, three max-pooling layers, two normalization layers, two fully connected layers, and one SoftMax layer in its design. Convolutional filters and a nonlinear activation function ReLU are used in each convolutional layer. Max pooling is conducted using the pooling layers. Due to the presence of fully connected layers, the input size is fixed. The input size is usually stated as 224 × 224 × 3, however due to padding, it can add up to 227 × 227 × 3. The neural network has 60 million parameters in total. Max Pool is used to down-sample an image or a representation. Overlapping Max Pool layers are like Max Pool layers with the exception of the adjacent windows over which the maximum determined overlaps, as shown in Figure. AlexNet's employed pooling windows with a size of 33 and a stride of 2 between adjacent windows. Figure 16 shows AlexNet architecture.

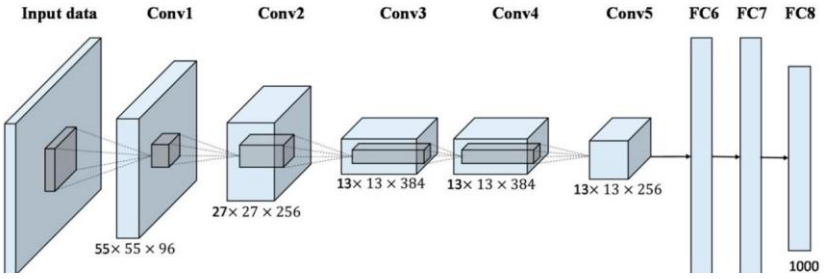

Figure 16 : AlexNet Architecture

3.3.3.3. LeNet 5 [80] :

LeNet is a convolutional neural network that Yann LeCun introduced in 1989. LeNet is a common term for **LeNet-5**, a simple convolutional neural network.
The LeNet-5 signifies CNN's emergence and outlines its core components. However, it was not popular at the time due to a lack of hardware, especially GPU and alternative algorithms, like SVM, which could perform effects similar to or even better than those of the LeNet.

The LeNet-5 CNN architecture has seven layers. Three convolutional layers, two subsampling layers, and two fully linked layers make up the layer composition.

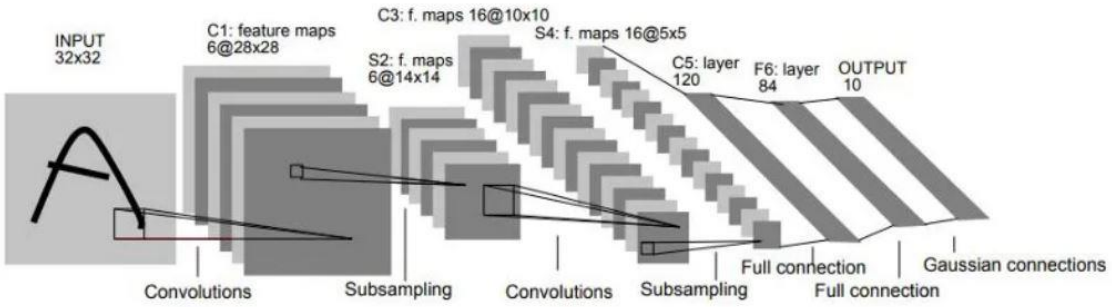

Figure 17 : LeNet-5 Architecture

3.3.3.4. ResNet 50 [81] :

ResNets or Residual networks are a type of deep convolutional neural network architecture that was first introduced in Dec 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun @ MSRA(Microsoft Asia).

One of the most well-known ResNet architectures is ResNet50, which consists of 50 layers and achieved state-of-the-art performance on the ImageNet dataset in 2015. ResNet50 consists of 16 residual blocks, with each block consisting of several convolutional layers with residual connections. The architecture also includes pooling layers, fully connected layers, and a softmax output layer for classification.

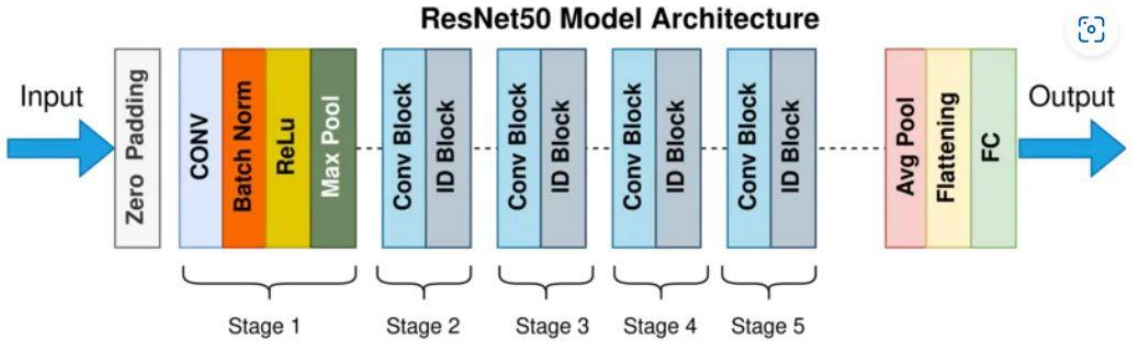

Figure 18 : ResNet50 Architecture

3.3.3.5. Efficientnet B0 [79] :

EfficientNet is a convolutional neural network design and scaling method that uses a compound coefficient to scale all depth/width/resolution dimensions evenly. The EfficientNet scaling method consistently scales network breadth, depth, and resolution with a set of predefined scaling coefficients, unlike traditional methods, which arbitrary scales these elements. Before the EfficientNets, the most popular technique to scale up ConvNets was to increase the depth (number of layers), the breadth (number of channels), or the image quality. The EfficientNet started by creating a baseline network using a technique called neural architecture search which

automates the building of neural networks. On floating-point operations per second (FLOPS) basis, it optimizes both accuracy and efficiency. The movable inverted bottleneck convolution is used in this architecture.

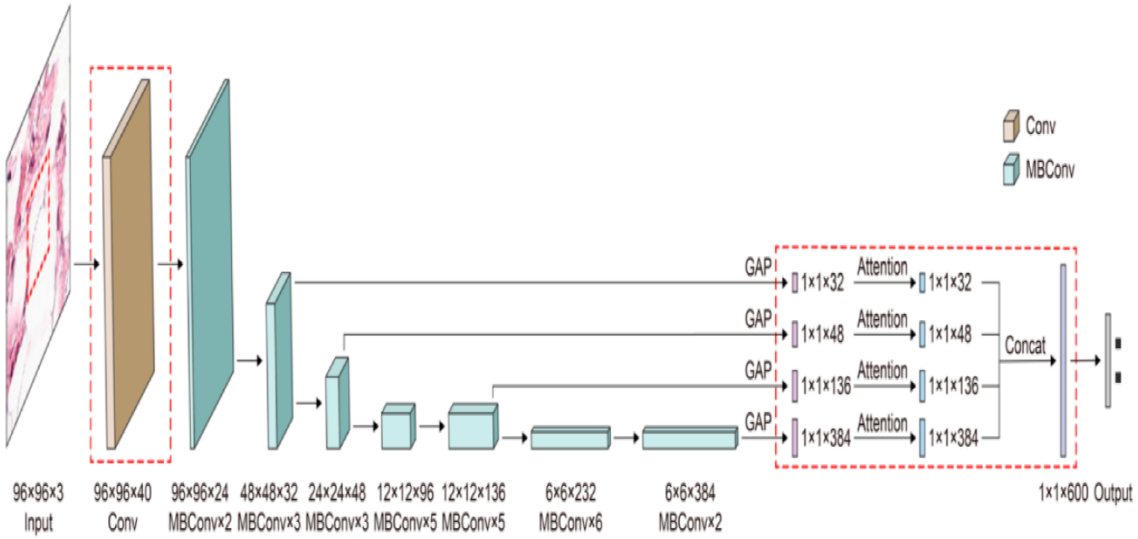

Figure 19 : Efficientnetb0 Architecture

Results and Discussion:

3.4. Arabic sign language:

3.4.1. Adopted Methodology:

In this section, we'll explore the research methodology used in this study. We'll provide a step-by-step breakdown of the process, from how we gathered data to how we arrived at our findings. This will include a detailed look at the key steps outlined in the flowchart, along with explanations of the pre-trained models we employed.

We began by importing necessary libraries like Keras, Pandas, and Matplotlib. Next, we loaded the ArASL image dataset directly from Kaggle. As a reminder, this dataset contains 54,049 images representing 32 Arabic Sign Language characters.

After loading the dataset, we tackled data preprocessing to address potential biases and inconsistencies in the model's output. Since the dataset has varying image counts per category (imbalance), we balanced the class sizes by allocating a fixed number of samples for each category. Additionally, we resized all images to a uniform size to ensure compatibility with the model.

For model development, we split the preprocessed data into training and testing sets. We used a 70/30 split, allocating 70% of the data for training the model and 30% for testing its performance.

The training set was fed into three pretrained models, taking advantage of their efficiency and stability in extracting complicated patterns from data. These models are Efficientnetb0, AlexNet , Resnet50 and Lenet.

We added a prediction layer with a softmax activation function to the pre-trained models' final fully connected layer. To optimize their performance, we fine-tuned these models by adjusting hyperparameters like different number of epochs. Following fine-tuning, we assessed each model's effectiveness on a validation set using accuracy scores. Visualization techniques helped us understand these results better. Ultimately, we selected the model that achieved the best performance.

3.4.2. Training :

Convolutional Neural Networks (CNNs) have achieved impressive results in sign language recognition. To capitalize on this potential, we conducted a study comparing several pre-trained CNN models. Our goal was to identify the most effective model for sign recognition using transfer learning. We leveraged the extensive ArASL dataset containing 54,049 images across 32 Arabic signs for training and validation.

The efficientnetb0 architectural specification employed in this investigation is shown in Table 04.

Layer (type)	Output Shape	Param #
efficientnetb0 (Functional)	(None, None, None, 1280)	4049571
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 512)	655872
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 32)	2080
dense_5 (Dense)	(None, 32)	1056

Total params: 4,881,059
Trainable params: 831,488
Non-trainable params: 4,049,571

Table 4 : Efficientnet B0 Model Architecture (Arsl)

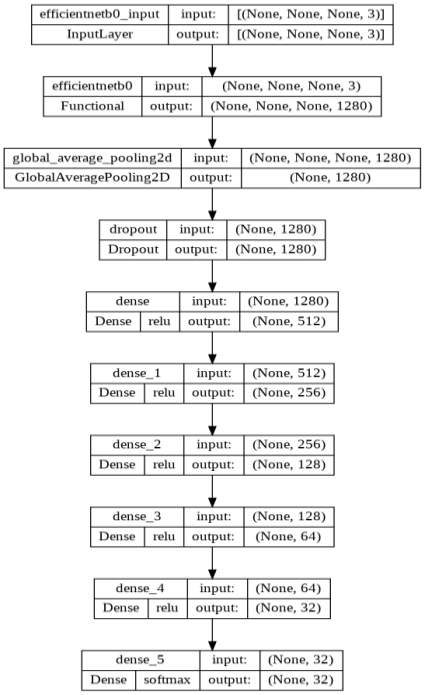

Figure 20 : Efficientntb0 Model Plot

This architecture utilizes a "sequential" model, meaning the layers are stacked one after another. The core of the network is an EfficientNetB0 block, a pre-trained convolutional neural network with over 4 million parameters. This pre-trained block acts like a feature extractor, automatically learning complex patterns from images fed as input (shape: "(None, None, None, 1280)"). "None" indicates flexibility in the input image size.

Following the pre-trained block, a GlobalAveragePooling2D layer reduces the spatial dimensions (height and width) of the feature maps into a single vector, preserving the channel information (1280 channels). A Dropout layer with a rate of likely 0.5 (not explicitly shown) randomly drops a certain percentage of activations during training to prevent overfitting.

The network then transitions to a series of fully-connected (dense) layers. The first dense layer (512 neurons) performs a significant dimensionality reduction, followed by layers with a decreasing number of neurons (256, 128, 64, 32, 32). These layers learn increasingly complex relationships between the extracted features. The final layer with 32 neurons likely represents the number of classes the network is trying to predict.

Overall, this architecture leverages a powerful pre-trained model for feature extraction and fine-tunes it with additional dense layers for the specific classification task. The use of dropout helps prevent overfitting, and the gradual decrease in neuron count in the dense layers promotes efficient learning.

Table 05 is Lenet architecture specification used in this study:

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 60, 60, 6)	156
max_pooling2d_8 (MaxPooling2D)	(None, 30, 30, 6)	0
conv2d_13 (Conv2D)	(None, 26, 26, 16)	2416
max_pooling2d_9 (MaxPooling2D)	(None, 13, 13, 16)	0
flatten_3 (Flatten)	(None, 2704)	0
dense_9 (Dense)	(None, 120)	324600
dense_10 (Dense)	(None, 84)	10164
dense_11 (Dense)	(None, 32)	2720

Total params: 340056 (1.30 MB)
Trainable params: 340056 (1.30 MB)
Non-trainable params: 0 (0.00 Byte)

Table 5 : LeNet model architecture (Arsl)

The Lenet architectural design is as shown in Table 05, It starts with a convolutional layer that extracts features from the input image using a set of learnable filters. These features are then downsampled by a pooling layer to reduce image size and computational cost , This reduces the image size to (None, 30, 30, 6). It has no trainable parameters (Param # = 0) as it only performs a fixed operation. A second convolutional layer with more filters extracts more complex features from the downsampled data, followed by another pooling layer for further reduction. The flattened output from the pooling layers is fed into fully-connected layers with a decreasing number of neurons, This layer reshapes the multidimensional data from the previous layer (feature maps) into a single long vector for feeding into fully connected layers. The output shape becomes (None, 2704), which is the product of the width, height, and number of feature maps

(13 * 13 * 16). The final layer, with 32 neurons, likely represents the number of classes the network is trying to predict. While the exact filter and window sizes are missing here, this architecture demonstrates a common approach in CNNs: using convolutional layers to learn spatial features from images, followed by pooling for dimensionality reduction and fully-connected layers for classification. This specific configuration, with its moderate complexity, could be suitable for tasks with limited data or computational resources, potentially recognizing handwritten digits with additional information about variations or classifying objects into 32 categories.

The model training process is carried out in a hardware and software environment with specifications for Dell Latitude E7470 Laptop, 08 GB RAM, Processor Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz, 256 GB SSD GPU Intel(R) HD Graphics 520. The operating system used is Windows 10 Professional and the training and testing model algorithms are implemented using python code by utilizing the Tensorflow library.

The results obtained with the proposed architecture (efficientnetb0) are show below:

```
Epoch 1/30
296/296 [==============================] - 179s 533ms/step - loss: 1.5031 - accuracy: 0.5285 - val_loss: 0.5692 - val_accuracy: 0.8272
Epoch 2/30
296/296 [==============================] - 153s 516ms/step - loss: 0.7576 - accuracy: 0.7485 - val_loss: 0.3511 - val_accuracy: 0.8962
Epoch 3/30
296/296 [==============================] - 154s 522ms/step - loss: 0.5941 - accuracy: 0.8021 - val_loss: 0.2643 - val_accuracy: 0.9216
Epoch 4/30
296/296 [==============================] - 154s 520ms/step - loss: 0.5024 - accuracy: 0.8350 - val_loss: 0.2221 - val_accuracy: 0.9332
Epoch 5/30
.....
296/296 [==============================] - 160s 542ms/step - loss: 0.1870 - accuracy: 0.9409 - val_loss: 0.0858 - val_accuracy: 0.9733
Epoch 29/30
296/296 [==============================] - 152s 514ms/step - loss: 0.1763 - accuracy: 0.9433 - val_loss: 0.0897 - val_accuracy: 0.9726
Epoch 30/30
296/296 [==============================] - 152s 513ms/step - loss: 0.1757 - accuracy: 0.9440 - val_loss: 0.0892 - val_accuracy: 0.9718

43/43 [==============================] - 18s 409ms/step - loss: 0.0980 - accuracy: 0.9717

✓ Loss ==> 0.09801185876131058
✓ Accuracy ==> 0.9716547131538391
```

Table 6: Efficientnet B0 Model Accuracy (Arsl)

Experiments show that varying the number of training epochs (from 0 to 30) results to high accuracy of 94.40%.

Figure 21 shows an increase in training and validation accuracy, as well as loss (figure 22) . Show blue and orange lines.

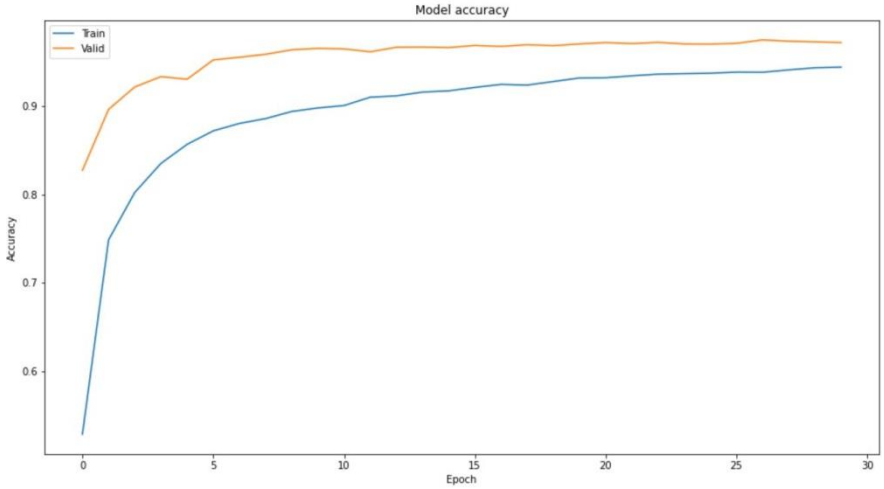

Figure 21 : Training accuracy vs validation accuracy

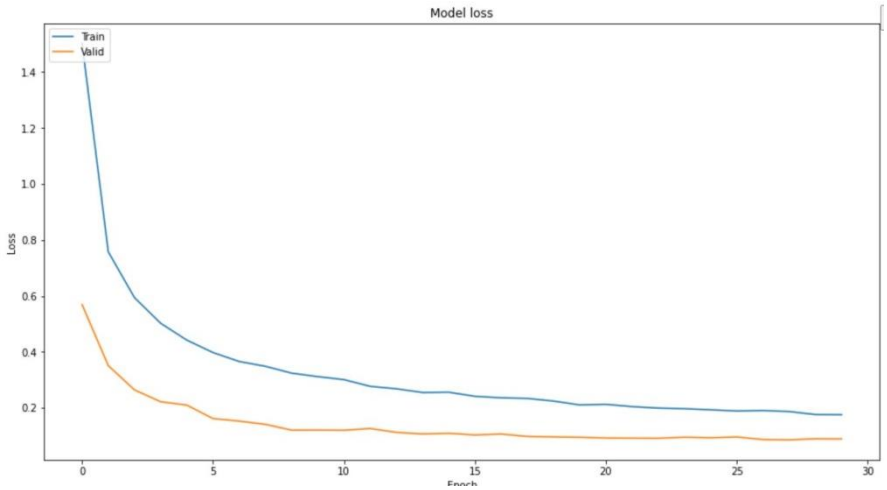

Figure 22 : training loss vs validation loss

As demonstrated in Fig 22,23 , There are 30 epochs in the training process. The first epoch has a training accuracy of 52.85% and a validation accuracy of 82.72%, along with a training loss of >90% and a validation loss of 56.92%. Ultimately, in the final epoch, the training and validation accuracy are 94.40% and 97.18%, respectively, while the training and validation losses are 17.57% and 08.92%, respectively.
This study found that using resizing, background correction, and hyperparameters improves model accuracy.

We tested numerous algorithms to determine which strategy was optimal for the gesture recognition application. We compared with other deep learning models (AlexNet , Lenet, Resnet50)

In the end, we produced the best results and was chosen, with an accuracy of 94% during training and an outstanding 97.16% during testing.

Table 07 describes the accuracy and loss of each model,

Model	Optimizer	Accuracy		Loss	
		Train (%)	Test (%)	train	Test
Efficientnetb0	Adam	94.40	97.18	0.1757	0.0892
AlexNet	Adam	98.17	82.52	0.0597	0.6389
Lenet	Adam	99.52	74.77	0.0023	0.0952
Resnet50	Adam	74.97	71.58	0.0432	0.0532

Table 7 : Accuracy and loss of each Model (Arsl)

3.4.3. Impact of Optimizers:

In these experiments, We tried to examine four distinct optimizers in these experiments: Adam, RMSprop, AdamW and Adadelta optimizers. To determine which optimizer among them produced the best results for the Lenet model in terms of performance rate for the ArSL recognition process. The outcomes of this scenario with a different optimizer in the training loop are compiled in Table 08 .

Lenet Model				
Optimizer	Accuracy		Loss	
	Train	Test	Train	Test
Adam	99.52%	74.77%	0.2%	09.52%
RMSprop	95.62%	73.10%	01.06%	6.15%
Adam w	98.74%	73.93%	0.3%	7.56%
Adadelta	97.25%	73.65%	0.6%	5.93%

Table 8 : Model Performance on testing data using different optimizer (ARSL)

Adam combines Adadelta's and RMSpro's beneficial features. It employs an adaptive learning rate and momentum, which means that the learning rate is gradually changed over time. It is still the most widely used optimizer in deep learning. Similar to this, the Adam optimizer is modified by the AdamW optimizer, which carries out optimization for the learning rate and weight decay independently. It is considered to hVave a faster convergence rate than Adam under some conditions. Adam does better than everyone else, as expected.

The Confusion matrix value from the model results is displayed in Figure 23.

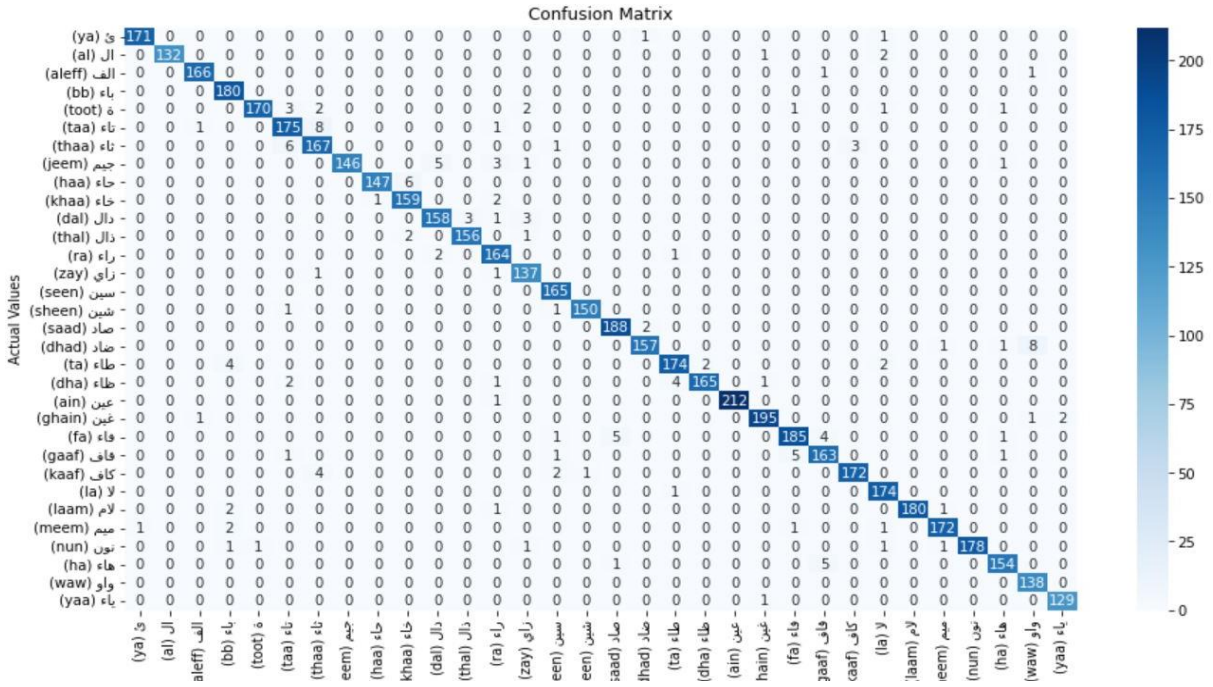

Figure 23 : Confusion matrix of Efficientnet model

3.4.4. Comparative Study :

The proposed recognition techniques were compared to some of the most relevant studies that produced the best accuracy for ArSL recognition.

References	Accuracy
Maraqa and Abu-Zaiter []	95.11%
Mohandes et al. []	98%
Elons et al. []	88%
Our proposed	97.16%

Table 9 : Comparison of the results obtained by the proposed approach and other pervious methods (Arsl)

3.5. American Sign language:

3.5.1. Data Acquisition

The data comes from the ASL Alphabet repository on Kaggle.com. It consists of 29 folders containing 3000 hand sign photos in JPG format. A total of 87,000 photos were included in the acquired image data [32]. The 21,000 images were distributed for the training process . we are take juste 04 classes [“A” , “B” , “C” , “D”]

3.5.2. Preprocessing :

Using the bicubic interpolation method, the resizing process is done during the preprocessing step .

```
new_array = img.resize((img_size,img_size), Image.BICUBIC)
```

The output of this procedure is an image that is 64 by 64 pixels in size, compared to the original 200 by 200 pixels. This step is used to decrease the temporal complexity during model training.

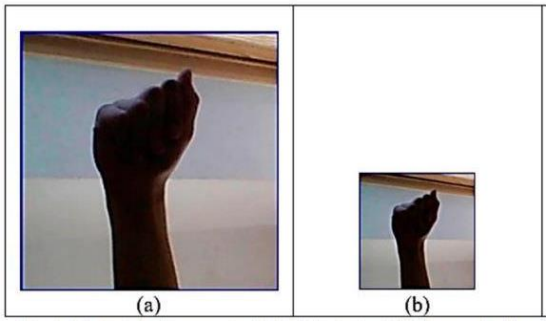

Figure 24 : (a) .Original image size 200x200 pixels (b) The result of resizing

3.5.3. Training :

At this stage, the CNN model design used in the training process is carried out to produce an appropriate model to classify hand sign language images. The CNN model applied uses hyperparameter values consisting of the learning rate, epoch, loss function, and optimizer. Table 10 is CNN architecture specification used in this study.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 4)         304
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 4)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 16, 16, 15)        1515
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 15)          0
_________________________________________________________________
```

```
flatten_1 (Flatten)          (None, 240)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 964
=================================================================
Total params: 2,783
Trainable params: 2,783
Non-trainable params: 0
_________________________________________________________________
```

This table summarizes the architecture of a convolutional neural network (CNN).

The network starts with a convolutional layer (conv2d_1) that takes an input of unspecified size (likely images) with 4 channels (e.g., RGB + alpha) and applies 4 filters (kernels) of size unspecified. This results in an output with the same height and width as the input, but with a depth of 4 (number of filters). It has 304 trainable parameters (weights and biases for each filter).

Next, a max pooling layer (max_pooling2d_1) downsamples the feature maps, reducing their height and width by a factor of 2 while retaining the most significant activation (usually the maximum value) within a predefined window. It doesn't introduce new parameters (0 trainable params).

This pattern of convolution (conv2d_2 with 15 filters) and max pooling (max_pooling2d_2) repeats, progressively extracting higher-level features and reducing the spatial dimensions.

The flatten_1 layer transforms the multi-dimensional data from the previous layer into a single long vector, preparing it for a fully-connected layer. It doesn't have any trainable parameters (0 trainable params).

Finally, a dense layer (dense_1) with 4 neurons takes the flattened features and performs classification, generating an output of size (batch_size, 4). This layer has 964 trainable parameters (weights connecting each input feature to each neuron and biases for each neuron).

The table shows a total of 2,783 trainable parameters, which the network will learn during training to perform the desired classification task.

ASL-CNN achieved the highest testing accuracy (99.04%) at 10 learning epochs. Figure shows the model accuracy when the proposed ASL-CNN model is trained with 10 epochs. The training and testing performances are close to each other during different epochs, indicating that the model has not been overtrained. The training and testing accuracy values are summarized in Table 11.

No.epochs	Training Acc. (%)	Testing Acc. (%)
10	99.52	99.04

Table 11 : Summarize the training and testing accuracy (ASL)

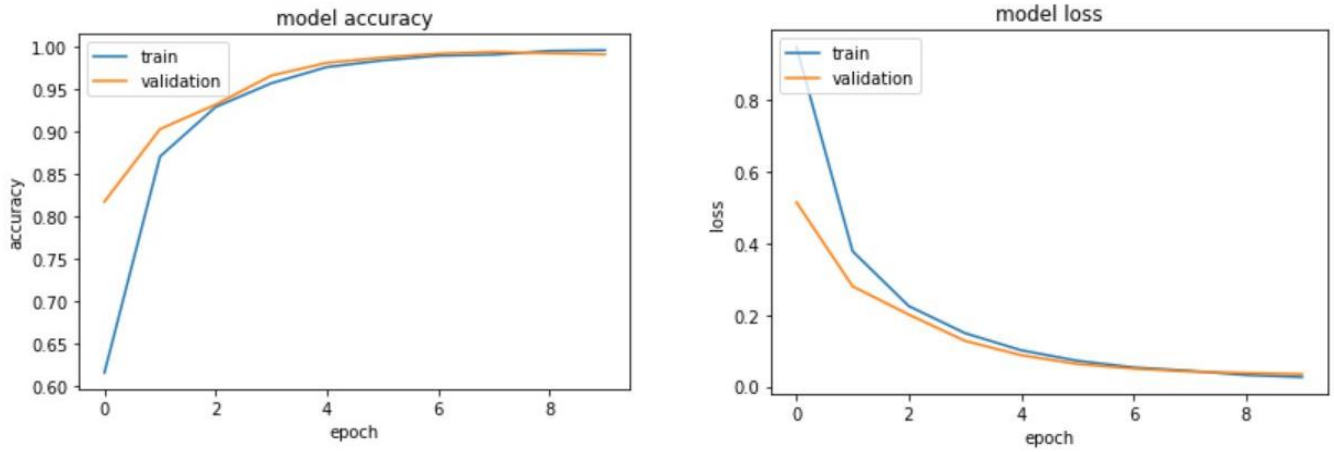

Figure 25 : Graphics of the training CNN model process : (a) Accuracy of training and validation (b) Loss of training and validation

We tested numerous algorithms to determine which strategy was optimal for the gesture recognition application. We compared with other deep learning models (AlexNet , Lenet, CNN)

Table 12 describes the accuracy and loss of each model,

Model	Test Accuracy	Test Loss	Train Accuracy	Train Loss
AlexNet	96.91%	0.1025	99.49%	0.0234
LeNet	99.87%	0.0045	99.97%	0.0027
CNN	99.04%	0.0348	99.52%	0.0267

Table 12 : Accuracy and Loss of each model (ASL)

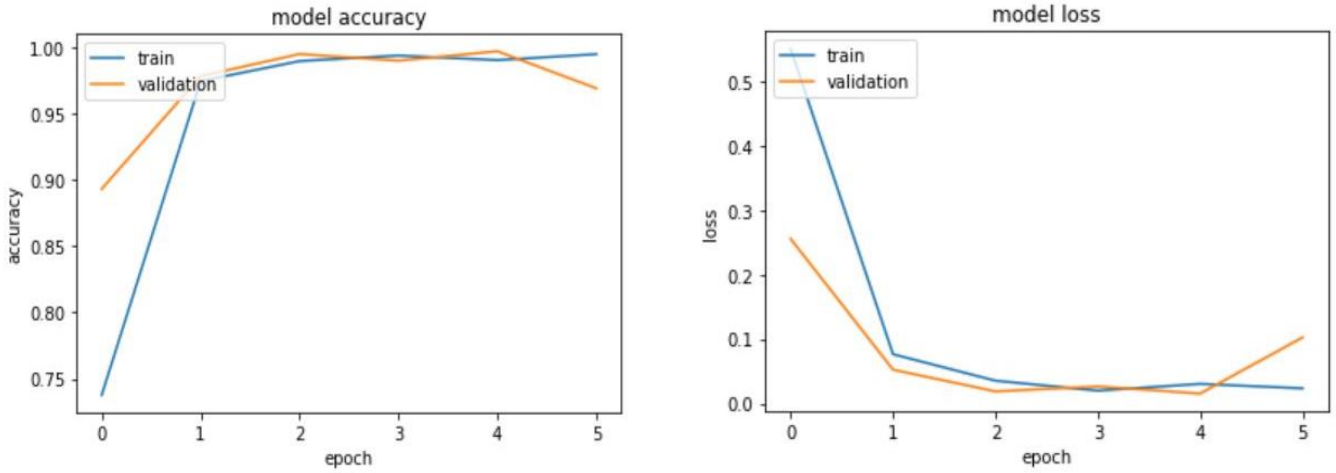

Figure 26 : Graphics of the training AlexNet model process : (a) Accuracy of training and validation (b) Loss of training and validation

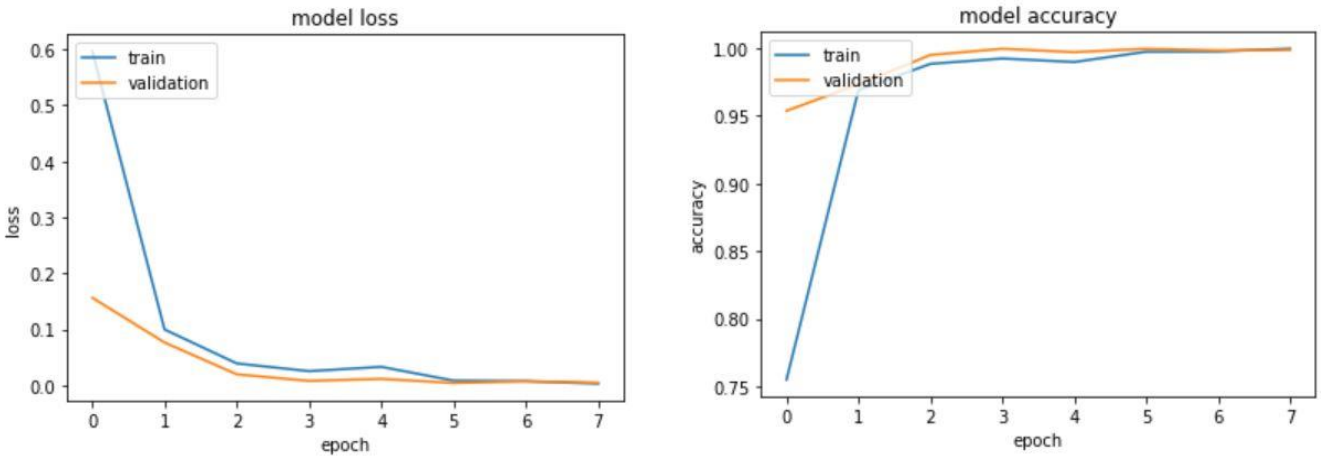

Figure 27 : Graphics of the training LeNet model process : (a) Accuracy of training and validation (b) Loss of training and validation

3.5.4. Comparative Study :

To show the performance of the proposed system, we have compared the results obtained from recent literature with our method.

References	Algorithm	Accuracy
Ferreira et al.	CNN	97%
Oyedotun and Khashman	CNN	91.33%
Islam et al.	ANN	94.32%
Our proposed	CNN	99.04%

Table 13: comparison of the results obtained by the proposed approach and other previous methods (ASL)

In comparison to the outcomes of the models mentioned earlier.

With a validation accuracy of 99.04%, our recommended architecture produced the greatest results and has shown to be useful in the recognition of sign language.

CONCLUSION :

Conclusion and Future Work:

CONCLUSION:

This thesis proposes a Convolutional Neural Network (CNN) model and CNN architecture for recognizing hand signs representing letters of the alphabet.

The proposed CNN-based classification models are applied to two sign language datasets: American Sign Language (ASL) and Arabic Sign Language. Our model achieves high accuracy. This improvement builds upon previous research in hand sign recognition.

We compare different training and testing approaches to identify the optimal algorithm for hand gesture recognition.

The comparison between the proposed system and other related works proved that the proposed system is more effective and accurate than others

This research aims to develop a robust system for sign language recognition, facilitating communication between deaf and hearing individuals so this approach contributes to bridging the communication gap between these communities.

Future Work :

We will expand the project to include:

- We will extend the dataset beyond just letters of the alphabet to include basic words and common phrases. This will significantly improve the system's practical applications.
- We aim to integrate real-time object detection using libraries like OpenCV for a more seamless user experience.
- We plan to translate this research into a real-time application that facilitates communication between deaf and hearing individuals.

REFERENCES :

[1] Y. Li, J. Huang, F. Tian, H.-A. Wang, and G.-Z. Dai, "Gesture interaction in virtual reality," Virtual Reality & Intelligent Hardware, vol. 1, no. 1, pp. 84–112, Jan. 2019

[2] Noor A. Ibraheem, Rafiqul Z. Khan, Vision Based Gesture Recognition Using Neural Networks Approaches: A Review , Aligarh, 202002, India ; pp 02

[3] Munir Oudah ,Ali Al-Naji and Javaan Chahl, Hand Gesture Recognition Based on Computer Vision: A Review of Techniques, Submission received: 23 May 2020 / Revised: 15 July 2020 / Accepted: 21 July 2020 / Published: 23 July 2020,

[4] Gupta, H.P.; Chudgar, H.S.; Mukherjee, S.; Dutta, T.; Sharma, K. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. *IEEE Sens. J.* 2016, *16*, 6425–6432

[5] Ankita Wadhawan , Parteek Kumar; Sign Language Recognition Systems: A Decade Systematic Literature Review; Barcelona, Spain 2019 ; pp 789

[6] Abeer Alnuaim ,Mohammed Zakariah ,Wesam Atef Hatamleh, Hussam Tarazi, Vikas Tripathi and Enoch Tetteh Amoate; Human-Computer Interaction with Hand Gesture RecognitionUsing ResNet and MobileNet ; pp 04

[7] Bambang Krismono Triwijoyo, Lalu Yuda Rahmani Karnaen, Ahmat Adil ; Deep Learning Approach For Sign Language Recognition ;Bumigora University, Jl. Ismail Marzuki No.22, Mataram 83127, Indonesia ; pp 15

[8] Muhammad Al-Barham and Ahmad Jamal , Design of Arabic Sign Language Recognition Model, 26/05/2021

[9] Mohandes M, Quadri SI, Deriche M (2007) Arabic Sign Language recognition an image-based approach. In: 21st IEEE international conference on advanced information networking and applications workshops, AINAW'07, vol 1, pp 272–276

[10] Maraqa M, Abu-Zaiter R (2008) Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. In: First IEEE international conference on the applications of digital information and web technologies, 2008. ICADIWT, pp 478–481

[11] Al-Rousan M, Assaleh K, Tala'a A A (2009) Video-based signerindependent Arabic Sign Language recognition using hidden Markov models. Appl Soft Comput 9(3):990–999

[12] Shanableh T, Assaleh K (2011) User-independent recognition of Arabic Sign Language for facilitating communication with the deaf community. Digit Signal Process 21(4):535–542

[13]. Elons AS, Ahmed M, Shedid H, Tolba MF (2014) Arabic Sign Language recognition using leap motion sensor. In: 9th IEEE international conference on computer engineering & systems (ICCES), pp 368–373

[14]. Mohandes M, Deriche M, Johar U, Ilyas S (2012) A signerindependent Arabic Sign Language recognition system using face detection, geometric features, and a Hidden Markov Model. Comput Electr Eng 38(2):422–433

[15] Assaleh K, Shanableh T, Fanaswala M, Bajaj H, Amin F (2008) Vision-based system for continuous Arabic Sign Language recognition in user dependent mode. In: 5th IEEE international symposium on mechatronics and its applications, ISMA, pp 1–5

[16] Dahmani D, Larabi S (2014) User-independent system for sign language fnger spelling recognition. J Vis Commun Image Represent 25(5):1240–1250

[17] Ahmed AA, Aly S (2014) Appearance-based Arabic Sign Language recognition using hidden markov models. In: IEEE international conference on engineering and technology (ICET), pp 1–6

[18] Mohandes M, Aliyu S, Deriche M (2014) Arabic Sign Language recognition using the leap motion controller. In: 23rd IEEE international symposium on industrial electronics (ISIE), pp 960–965

[19] Tubaiz N, Shanableh T, Assaleh K (2015) Glove-based continuous Arabic Sign Language recognition in user-dependent mode. IEEE Trans Hum Mach Syst 45(4):526–533

[20]. Sarhan NA, El-Sonbaty Y, Youssef SM (2015) HMM-based Arabic Sign Language recognition using Kinect. In: Tenth IEEEinternational conference on digital information management (ICDIM), pp 169–174

[21]. Hassan M, Assaleh K, Shanableh T (2016) User-dependent sign language recognition using motion detection. In: IEEE international conference on computational science and computational intelligence (CSCI), pp 852–856

[22] Hamed A, Belal NA, Mahar KM (2016) Arabic Sign Language alphabet recognition based on HOG-PCA using microsoft Kinect in complex backgrounds. In: IEEE 6th international conference on advanced computing (IACC), pp 451–458

[23]. Darwish SM (2017) Man–machine interaction system for subject independent sign language recognition. In: Proceedings of the 9th international conference on computer and automation engineering. ACM, pp 121–125

[24] Oz C, Leu MC (2011) American Sign Language word recognition with a sensory glove using artifcial neural networks. Eng Appl Artif Intell 24(7):1204–1213

[25] Sun C, Zhang T, Bao BK, Xu C (2013a) Latent support vector machine for sign language recognition with Kinect. In: 20th IEEE international conference on image processing (ICIP), pp 4190–4194

[26]. Sun C, Zhang T, Bao BK, Xu C, Mei T (2013) Discriminative exemplar coding for sign language recognition with kinect. IEEE Trans Cybern 43(5):1418–1428

[27] Chuan CH, Regina E, Guardino C (2014) American Sign Language recognition using leap motion sensor. In: 13th IEEE international conference on machine learning and applications (ICMLA), pp 541–544

[28] AlQattan D, Sepulveda F (2017) Towards sign language recognition using EEG-based motor imagery brain computer interface. In: 5th IEEE international winter conference on brain–computer interface (BCI), pp 5–8

[29] Kim SY, Han HG, Kim JW, Lee S, Kim TW (2017) A hand gesture recognition sensor using refected impulses. IEEE Sens J 17(10):2975–2976

[30] Islam MM, Siddiqua S, Afnan J (2017) Real time hand gesture recognition using diferent algorithms based on American Sign Language. In: IEEE international conference on imaging, vision & pattern recognition (icIVPR), pp 1–6

[31] Ferreira PM, Cardoso JS, Rebelo A (2017) Multimodal learning for sign language recognition. In: Iberian conference on pattern recognition and image analysis. Springer, Cham, pp 313–321

[32] Munib Q, Habeeb M, Takruri B, Al-Malik HA (2007) American sign language (ASL) recognition based on Hough transform and neural networks. Expert Syst Appl 32(1):24–37

[33] Oz C, Leu MC (2007) Linguistic properties based on American Sign Language isolated word recognition with artifcial neural networks using a sensory glove and motion tracker. Neurocomputing 70(16):2891–2901

[34] Ragab A, Ahmed M, Chau SC (2013) Sign language recognition using Hilbert curve features. In: International conference image analysis and recognition. Springer, Berlin, Heidelberg, pp 143–151

[35]. Tangsuksant W, Adhan S, Pintavirooj C (2014) American Sign Language recognition by using 3D geometric invariant feature and ANN classifcation. In: 7th international conference on biomedical engineering (BMEiCON), pp 1–5

[36]. Zamani M, Kanan HR (2014) Saliency based alphabet and numbers of American Sign Language recognition using linear feature extraction. In: 4th IEEE International eConference on computer and knowledge engineering (ICCKE), pp 398–403

[37]. Wu J, Tian Z, Sun L, Estevez L, Jafari R (2015) Real-time American Sign Language recognition using wrist-worn motion and surface EMG sensors. In: IEEE 12th international conference on wearable and implantable body sensor networks (BSN), pp 1–6

[38] Aryanie D, Heryadi Y (2015) American Sign Language-based fnger-spelling recognition using k-nearest neighbors classifer. In: 3rd IEEE international conference on information and communication technology (ICoICT), pp 533–536

[39]. Kumar A, Thankachan K, Dominic MM (2016) Sign language recognition. In: 3rd IEEE international conference on recent advances in information technology (RAIT), pp 422–428

[40] Karayılan T, Kılıç Ö (2017) Sign language recognition. In: IEEE international conference on computer science and engineering (UBMK), pp 1122–1126

[41] Oyedotun OK, Khashman A (2017) Deep learning in visionbased static hand gesture recognition. Neural Comput Appl 28(12):3941–3951

[42] Sidig, A. A. I., et al. (2021). KArSL: Arabic sign language database. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 20(1), 1-19

[43] Latif G., Mohammad N., Alghazo J., Roaa A.,Rawan A. Arasl: Arabic alphabets sign language dataset. Data in brief. 2019;23:103777

[44] A. Nagaraj, "ASL Alphabet." 2022, Kaggle. https://doi.org/10.34740/kaggle/dsv/29550

[45] N. A. Ibraheem and R. Z. Khan. "Survey on various gesture recognition technologies and techniques". International Journal of Computer Applications 50(7), 2012. Available: http://search.proquest.com/docview/1032039156. DOI: 10.5120/7786-0883.

[46] P. Premaratne, Human Computer Interaction Using Hand Gestures, 2014th ed. Singapore: Springer, 2014.

[47] S. Mitra and T. Acharya. Gesture recognition: A survey. Tsmcc 37(3), pp. 311-324. 2007. Available: http://ieeexplore.ieee.org/document/4154947. DOI: 10.1109/TSMCC.2007.893280.

[50] Kay M. Stanney HANDBOOK OF VIRTUAL ENVIRONMENTS Design, Implementation, and Applications, Gesture Recognition Chapter #10 by Matthew Turk

[51] Daniel Thalman, Gesture Recognition Motion Capture, Motion Retargeting, and Action Recognition

[52] J. Heinzm ann and A. Zelinsky Robust Real - Time Face Tracking and Gesture Recognition

[53] Maria Papatsimouli ,Panos Sarigiannidis and George F; A Survey of Advancements in Real-Time Sign Language Translators: Integration with IoT Technology ; 22 June 2023

[54] Nagi, J., et al. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. Paper presented at the 2011 IEEE international conference on signal and image processing applications (ICSIPA), Kuala Lumpur, Malaysia

[55] Assaleh, K., et al. (2012). Low complexity classification system for glove-based arabic sign language recognition. Paper presented at the Neural Information Processing: 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part III 19.

[56]Elons, A. S., et al. (2013). A proposed PCNN features quality optimization technique for pose-invariant 3D Arabic sign language recognition. Applied Soft Computing, 13(4), 1646-1660.

[57] Mohandes, M., et al. (2014). Arabic sign language recognition using the leap motion controller. Paper presented at the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE).

[58] Shohieb, S. M., et al. (2015). Signsworld atlas; a benchmark Arabic sign language database. Journal of King Saud University-Computer and Information Sciences, 27(1), 68-76.

[59] Ahmed, A. M., et al. (2016). Automatic translation of Arabic sign to Arabic text (ATASAT) system. Journal of Computer Science and Information Technology, 6, 109-122.

[60]ElBadawy, M., et al. (2017). Arabic sign language recognition with 3d convolutional neural networks. Paper presented at the 2017 Eighth international conference on intelligent computing and information systems (ICICIS).

[61] Alzohairi, R., et al. (2018). Image based arabic sign language recognition system. International Journal of Advanced Computer Science and Applications (IJACSA), 9(3), 185-194.

[62] Ibrahim, N. B., et al. (2018). An automatic Arabic sign language recognition system (ArSLRS). Journal of King Saud UniversityComputer and Information Sciences, 30(4), 470-477.

[63] Hayani, S., et al. (2019). Arab sign language recognition with convolutional neural networks. Paper presented at the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco.

[64] Latif, G., et al. (2019). ArASL: Arabic alphabets sign language dataset. Data in brief, 23, 103777.

[65] Alnahhas, A., et al. (2020). Enhancing the recognition of Arabic sign language by using deep learning and leap motion controller. Int. J. Sci. Technol. Res, 9, 1865-1870.

[66]Tharwat, G., et al. (2021). Arabic sign language recognition system for alphabets using machine learning techniques. Journal of Electrical and Computer Engineering, 2021, 1-17

[67]Sidig, A. A. I., et al. (2021). KArSL: Arabic sign language database. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 20(1), 1-19.

[68] Ismail, M. H., et al. (2021). Static hand gesture recognition of Arabic sign language by using deep CNNs. Indonesian Journal of Electrical Engineering and Computer Science, 24(1), 178-188.

[69] Al-Barham, M., et al. (2023). RGB Arabic Alphabets Sign Language Dataset. arXiv preprint arXiv:2301.11932.

[70] Muhammad Inayat Ullah Khan , 2011 .Hand Gesture Detetction & Recognition System

[71] Hatice Gunes, Massimo Piccardi, Tony Ja, 2007, Research School of Information Sciences and Engineering Australian National University Face and Body Gesture Recognition for a Vision-Based Multimodal Analyzer

[72] J. Heinzm ann and A. Zelinsky Robust Real - Time Face Tracking and Gesture Recognition

[73] Savur C., Sahin F. American sign language recognition system by using surface emg signal. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE 2016, 002872–002877.

[74] Luqman H., Mahmoud S.A., et al. Transform-based arabic sign language recognition. Procedia Computer Science. 2017;117:2–9

[75] Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W.: Video-based Sign Language Recognition without Temporal Segmentation

[76] Srushti Satardekar, December 2023. SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING , p 01, 02

[77] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach. Hoboken: Pearson, 1995.

[78] S. Pang, J. J. D. Coz, Z. Yu, O. Luaces, and J. Díez, “Deep learning to frame objects for visual target tracking,” Engineering Applications of Artificial Intelligence, vol. 65, pp. 406–420, 2017

[79] A Real Time Arabic Sign Language Alphabets (ArSLA) Recognition Model Using Deep Learning Architecture , https://doi.org/10.3390/computers11050078

[80] LeNet 5 Architecture Explained. In the 1990s, Yann LeCun, Leon Bottou… | by Siddhesh Bangar | Medium

[81] Detailed Explanation of Resnet CNN Model. | by TANISH SHARMA | Medium

ملخص

ملخص باللغة العربية

بالنسبة للصم وضعاف السمع، تشكل لغة الإشارة، وهي لغة تعتمد على الإيماءات وتعبيرات الوجه، لغةً أساسيةً للتواصل .ويأتي نظام التعرف الآلي على لغة الإشارة (SLR) ليلعب دورًا هامًا في سد الفجوة بين عالمهم الصامت وعالم الأشخاص الذين يتمتعون بحاسة السمع.

يركز هذا البحث بشكل خاص على لغتين رئيسيتين من لغات الإشارة :لغة الإشارة العربية (ArSL) ولغة الإشارة الأمريكية .(ASL) ثم تم استكشاف تقنية تعلم عميق عالية الأداء تسمى CNN للتعرف على هاتين اللغتين .تتميز تقنية CNN بقدرتها على تمييز أشكال اليدين ومواضعهما وحركاتهما بدقة عالية في لغة الإشارة.

سنقوم بتحليل النتائج والتحديات المرتبطة باستخدام بنيات CNN مختلفة مثل Efficientnetb0 و LeNet و AlexNet للتعرف على ArSL و ASLيتمتع SLR بإمكانية هائلة لتغيير العالم من خلال تمكين التواصل ما وراء الكلمات، وتسهيل إجراء المحادثات المهمة التي لم تتمكن من الحدوث من قبل أو التي كانت ضرورية لفترة طويلة جدًا.

Abstract

Your summary in English

For the deaf and hard-of-hearing, sign language, a language of signs based on gestures and expression. The automatic Sign language recognition system is bridging the gap between their silent and hearing world.

This thesis is particularly focused on two different sign languages, Arabic Sign Language (ArSL) and American Sign Language (ASL). Next, a formidable deep learning technique called CNNs has been studied to recognize the languages. CNNs do a great job of recognizing hand shapes, positions, and movements in sign language.

We will discuss the results and hurdles for using various CNN architectures such as Efficientnetb0, Lenet , Alexnet on ASL and ArSL recognition. SLR has the possibility to change the world by talking beyond talking, and by having those important conversations that have never been had or have needed to be had for far too long.

Résumé

Votre résumé en français

Pour les personnes sourdes et malentendantes, la langue des signes, qui est un langage gestuel et facial, est leur langue maternelle. Le système de reconnaissance automatique de la langue des signes (SLR) constitue un pont pour relier le monde sans bruit au monde du son.

Ce mémoire porte principalement sur deux langues des signes majeures, à savoir la langue des signes arabe et la langue des signes américaine. Ensuite, nous avons examiné CNN, une technique d'apprentissage profond de haute performance, pour distinguer les deux langues des signes. Les CNN se spécialisent dans la distinction des formes, des postures et des mouvements des mains en utilisant la langue des signes.

Nous analyserons les résultats et les défis liés à l'utilisation de différentes architectures CNN telles que Efficientnetb0, LeNet et AlexNet pour la reconnaissance de la ArSL et de l'ASL. Le SLR a le potentiel de changer le monde en permettant la communication au-delà des mots, en facilitant des conversations importantes qui n'ont jamais pu avoir lieu ou qui ont été nécessaires pendant bien trop longtemps.