Republic Of Algeria Democratic And People's
Ministry Of Higher Education And Scientific Research

University of KASDI Merbah - Ouargla

Faculty of New Technologies of Information and Communication
Department of Computer Science and Information Technologies

Professional Master Thesis

Domain: Computer Science

Specialty: Administration and Network Security

Presented by : Abazi Yahia

Aziza Akram Zakaria

Theme :

Hand gesture and sign language recognition based on deep learning

Presented on 17/06/2023 in front of the jury composed of :

Dr. MERBATI Hocine	President	University UKM Ouargla
Dr. BENKADDOUR Mohammed Kamel	Supervisor	University UKM Ouargla
Dr. BENBEZZIANE Mohamed	Examiner	University UKM Ouargla

Academic year: 2022-2023

Acknowledgements

We wish to thank first and foremost our Allah, the Almighty, the greatest of all. We would like to express our sincere gratitude to the following individuals and organizations that have contributed to the completion of our master's degree thesis: First and foremost, we would like to express our deepest appreciation to our thesis advisor, Dr. Benkaddour Mohammed Kamel, for his guidance, expertise, and unwavering support throughout this research journey. His valuable insights, constructive feedback, and dedication have played a pivotal role in shaping this thesis. We would like to acknowledge the support and encouragement from our families, especially our parents and siblings, for their love, patience, and understanding. Their constant support and belief in us have been a constant source of motivation. We extend our appreciation to our friends and colleagues who have provided assistance, insightful discussions, and encouragement throughout this thesis undertaking. Their camaraderie and intellectual exchange have been invaluable. Furthermore, we are thankful to the research participants who generously shared their time, knowledge, and experiences, without whom this study would not have been possible. In conclusion, the completion of this master's degree thesis would not have been possible without the support, guidance, and contributions of all these individuals and organizations. We are deeply indebted to each and every one of them.

Abstract

The recognition of Arabic sign language (ArSL) plays a crucial role in removing communication barriers between deaf-mute people and non-sign language speakers. In this study, we propose a dynamic model for Arabic sign language recognition using deep learning (DL) techniques. Our model utilizes a convolutional neural network (CNN) architecture to extract meaningful features from sign language (SL) images, for accurate classification of different signs. We also describe the dataset used for training and evaluating the model, which includes a collection of Arabic sign language images. After extensive experimentation and evaluation, the results prove the effectiveness of the proposed methods, achieving high recognition accuracy across multiple ArSL gestures.

Keywords: Sign language, Arabic sign language, Deep learning, Convolutional neural network, Recognition, Classification.

Résumé

La reconnaissance de la langue des signes arabe (ArSL) joue un rôle crucial dans l'élimination des barrières de communication entre les personnes sourdes et muettes et les locuteurs de langue sans signes. Dans cette étude, nous proposons un modèle dynamique pour la reconnaissance de la langue des signes arabe en utilisant des techniques d'apprentissage profond DL(Deep Learning). Notre modèle utilise une architecture de CNN (Convolution Neural Network) pour extraire des caractéristiques significatives des images du langage des signes (SL), pour une classification précise des différents signes. Nous décrivons également l'ensemble de données utilisé pour la formation et l'évaluation du modèle, qui comprend une collection d'images de la langue des signes arabe. Après une expérimentation et une évaluation approfondie, les résultats prouvent l'efficacité des méthodes proposées, atteignant une précision de reconnaissance élevée sur plusieurs gestes d'ArSL.

Mots clés: Langue des signes, Langue des signes arabe, Deep Learning, Convolution Neural Network, Reconnaissance, Classification.

ملخص

الاعتراف بلغة الإشارة العربية (ArSL) يلعب دوراً كبيرا في إزالة حواجز التواصل بين الصم -البكم و الذين لا يتكلمون بلغة الإشارة .نقترح في هذه الدراسة نموذجا ديناميكيا للتعرف على لغة الإشارة العربية باستخدام تقنيات التعلم العميق .(DL)يستخدم نموذجنا هندسة (CNN)لاستخراج التفاصيل و الخصائص من صور لغة الإشارة(SL)، من أجل التصنيف الدقيق لمختلف الاشارات .ونصف أيضا مجموعة البيانات المستخدمة في التدريب وتقييم النموذج، التي تتضمن مجموعة من صور لغة الإشارة العربية .وبعد إجراء تجارب وتقييمات واسعة النطاق، تثبت النتائج فعالية الأساليب المقترحة، مما يحقق درجة عالية من الدقة في الاعتراف على مختلف إشارات اللغة العربية.

كلمات مفتاحية: لغة الإشارة، لغة الإشارة العربية، التعلم العميق، CNN، التعرف، التصنيف.

Table of Contents

List of figures

List of Tables

Acronyms List

ASL	American Sign Language
SLR	Sign Language Recognition
ArSL	Arabic Sign Language
BSL	British Sign Language
ArASL	Arabic Alphabets Sign Language
CSIS	Center for Strategic International Studies
WHO	World Health Organization
CNN	convolutional neural network
HMM	hidden Markov model
ML	Machine Learning
SLR	sign language recognition
DNN	Deep neural networks
AI	Artificial intelligence
DL	Deep Learning
RNN	Recurrent neural networks
ReLU	Rectified linear unit
CSV	comma-separated values
GPU	Graphics processing unit
LSSVM	Least Square Support Vector Machine
LSTM	long short-term memory networks
MCSVM	Multiclass Support Vector Machine
NLP	Natural language processing
RAM	random-access memory
RGB	red, green and blue
SLID	Sign Language Identification

Introduction

General introduction

Deaf-mute individuals, those who are both deaf and unable to speak, represent a significant portion of the global population with disabilities, The World Health Organization (WHO) stated that approximately 70 million people around the world are deaf-mutes. A total of 360 million people are deaf, and 32 million of these individuals are children. And by 2050, it is expected that one in every four people will face some degree of hearing loss[1] . Those numbers are sad, knowing that many families are suffering, with one or multiple members are subject to this disability.

There are 220,000 deaf individuals in Algeria reported by the African Sign Language Resource Center, Out of the Middle East's 350 million people, over 11 million have a disabling hearing loss according to the Center for Strategic International Studies (CSIS) (2014). [2]

This highlights that even our Arabic world is widely affected, and as Arabic nations strive to reach the peak of civilization and compete in the new world order, their demands and needs to retrieve and utilize all the available resources are gradually increasing. Every contribution from any category of society is widely appreciated, and as we seen, the deaf-mute people represent a great percentage of the Arabic population, excluding them out off this task will cause significant harm and many opportunities are missed, recognizing their ideas and potential contributions to the general society and their own community can greatly benefit the nation.[3]

Sign language serves as the primary tool for communication within the deaf-mute community, and it can also be used as an alternative when lacking proficiency on the spoken language of a foreign country. It is a rich and expressive form of communication that utilizes visual gestures, hand movements, facial expressions, and body language to deliver the meaning. Each sign represents a specific word or concept, and the combination of signs allows for expressing complex ideas and sentences. It has its own grammar and vocabulary, it enables deaf-mute individuals to communicate effectively, engage in social interactions, access education, and participate fully in various aspects of life regarding their disability.

There is no standard form of Sign Language nor a universal one, although they may share certain similarities. It is estimated that there are over 300 distinct sign languages around the world [4], 130 are listed on The Ethnologue language database, reflecting cultural, regional, and historical differences worldwide, each developed within its specific region and different cultural

and historical background. For instance, while both Americans and British people share the English spoken language, American Sign Language (ASL) and British Sign Language (BSL) are different from each other. Similar to spoken languages, sign languages possess their own vocabularies, grammatical structures.

Developing such a tool for sign language recognition in Arabic countries holds significant importance. It can pave the way for the hard of hearing individuals to demonstrate their capabilities in many fields and contribute to society, economically by introducing a working-hand with new perspectives and special skills that will imply diversity into problem solving and introduce creative new ideas. Also, they can provide social stability by allowing these individuals to discuss their limitations and produce their own solutions to their problems without relying on specific figures, this will shorten the path to solving their issues. It also creates more understanding from others towards this community. All those points serve for the greater good of the society as a whole.

Due to the vastness and wide distribution of Arabic nations across the globe, Middle East North Africa, we witness the richness and diversity of cultures within this population. This diversity emphasizes a rise to various sub-languages based on historical, cultural, and regional factors. This leads to different dialects of Arabic Sign Language, such as Egyptian Sign Language and Saudi Sign Language, and many others. Consequently, there is no standard form of Arabic sign language, not even in the same Continents, African Arabic sign language or Middle Eastern Arabic Sign Language, there are no such things. This leads to another problem, which is the lack of a standard Arabic sign language. This made Creating a unified Arabic sign language recognition platform impossible without incorporating all language variations, which is an even bigger challenge. The only dataset available in a sufficient form is Arabic alphabets, because they are standardized across all Arabic spoken and sign languages, but datasets carrying actual words are none-existent. This put us in a position where we had to create our own dataset, it was a very time-consuming process that slowed the overall progress of this research even further.

The aim of this research is to create a platform for interpreting Arabic sign language gestures both isolated and continuous signs, taking the input form a live webcam feed, and performing live recognition.

The thesis will have numerous chapters, organized as follows :

- In chapter I, we introduce sign language , sign language types , talk about the Arabic sign language and in the end we gave a brief review about sign language recognition.
- In chapter II, we introduce deep neural network , digital image recognition and how deep learning is have so powerful methods for image recognition. Then we focus on Convolutional neural network (CNN), talked about The structure of a CNN and explained the CNN layers.
- In chapter III, we present related works on sign language recognition and talk about different techniques that they used.
- In chapter IV, We present the mythology and applied techniques that we went through, from gathering the DataSet, Pre-Processing and data augmentation. That's to to ensure consistency and remove potential biases.
- In chapter V, we start by presenting the experimental settings and tools. Then talk about the proposed architecture and the experiments. After that, we present the obtained results. Finally, we analysis and discussion them.
- In chapter VI, We gave a General Conclusion and present our Future Work.

Chapter I

Literature Review

I.1 Introduction

Sign language is the primary tool and Language for deaf-mute and hard hearing people to communicate with each other, and with external individuals outside of their community. In every society. This language is not very common outside of these communities, as it's not obligated upon them unless they know or have deaf-mute individuals in their families or as friends. Sign language recognition is a tool to translate and interpret sign language into a specific form of communication whether it's text or speech. The variety of techniques to perform sign language gestures poses unique challenges for recognition models.

I.2 Sign Language

Sign language is a visual and gestural form of communication used by individuals with hearing impairments or deafness to express and receive information. However, people who aren't deaf-mute also frequently utilize sign language to imply and further emphasize their emotions[5]. It is a complete and natural language with its own grammar, vocabulary, and syntax, except that it's not vocal. Instead of using spoken words, sign language utilizes a combination of hand shapes, facial expressions, body movements, and other non-verbal signals to convey meaning.

Figure I.1: example of signs and witch latter they mean [6]

There are multiple variations of sign language across the world as it is with spoken languages, those differences in languages are Provoked by regional and cultural aspects.

I.3 Sign Language Types

Sign language is composed of isolated signs, which deliver complete meaning through a single motion. And continued signs, which require a combination of gestures in a specific sequence to form words or phrases.

I.3.1 Isolated Signs

Such as alphabets and numbers, they can be considered as standalone gestures. Each sign represents a its own concept and meaning, allowing for direct translation between the sign and its corresponding meaning. These signs typically involve a single motion or hand shape which is easily recognizable.

Figure I.2: Isolated Signs

I.3.2 Continuous Signs

On the other hand, continued signs involve a series of consecutive signs performed in a specific order to form complete words or phrases. These signs require the recognition and interpretation of the each gesture and their sequential order to understand the full meaning. For

instance, in Arabic Sign Language, Al-salam alaykom is term which is formed of two signs that should be preformed consecutively. The continuous execution of these two signs, one after the other, creates the final word 'Al-Salam Alaykom'.

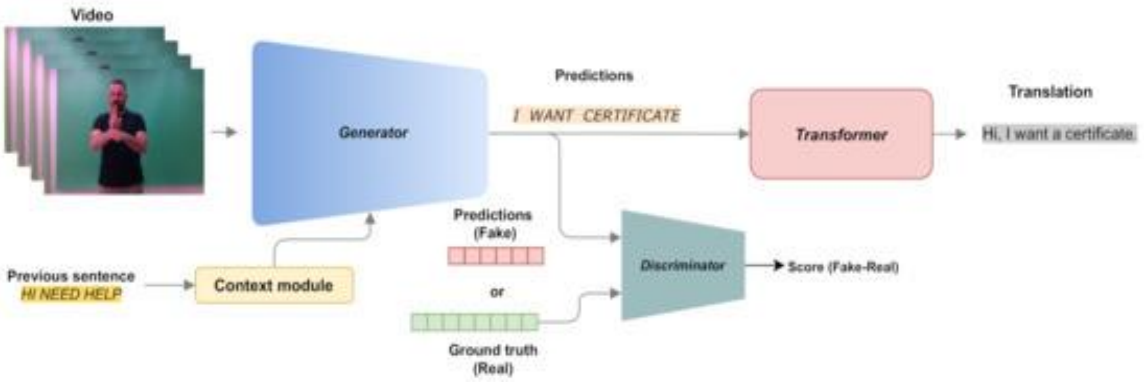

Figure I.3: Overview of the proposed framework that performs continuous sign language recognition and sign language translation [7]

Both isolated and continuous sign recognition can be approached using machine learning techniques such as deep learning, as well as computer vision. However, continuous sign recognition requires more complex methods and algorithms to handle the sequence of gestures performed.

I.4 Arabic Sign Language (ArSL)

Arabic Sign Language (ArSL) is another form of arabic sign language, it shares both similarities and differences with other sign language forms used around the world.

Like other sign languages, ArSL relies on visual and gestural communication, utilizing hand shapes, facial expressions, and body movements to deliver the meaning. This visual representation produces a unique form of communication, Going beyond spoken language techniques.

However, despite these shared features, ArSL also has it own characteristics that differentiate it and set it apart from other sign language variations. The cultural and regional

diversity within the Arab world contributes to the variances in vocabulary, grammar, and even sign gestures performance.

Different regions may have their own unique signs or techniques in the way signs are performed, influenced by local dialects, cultural practices, and historical factors. For example, the signs used in Egypt which is Egyptian Sign Language may differ from those used in the Gulf (Gulf Sign Language) due to regional differences and influences we mentioned.

Arabic Sign Language (ArSL) has a rich history in the Arab world, although pinpointing its exact origins can be challenging. Sign language began to emerge organically within the Arabic-speaking deaf communities, evolving as a visual mode of communication to bridge the communication gap. As with many other sign languages around the world, ArSL developed through natural gestural communication and visual expressions within the deaf-mute communities.

I.5 Sign Language Recognition

The goal of sign language recognition is to translate sign language gestures and convert them into comprehensible form, text or speech. In order to provide an interface for people with hearing loss and deaf people, sign language recognition (SLR) systems are typically created to recognize hand motions and finger shapes of sign language discourse[5].

Sign language recognition utilizes multiple methods as Computer vision, machine learning, and pattern recognition in order to recognize and interpret the hand gestures, facial expressions, and body language that makes up sign language, those elements are examined throughout video records, images or input provided by electronic devices such sensors and gloves.

The process of sign language recognition goes through multiple steps, first input must be gathered, its formed of videos, images, sensors or gloves, it captures hand movement, facial expressions, body language. The input gathered gets passed through computer vision algorithms to extract relevant features from it.

once that is done, ML algorithms are trained to recognize and classify the different sign language gestures. This training process requires providing the algorithms with labeled

representation of those signs to learn patterns between extracted features and their equivalent meaning.

When performing the recognition, the system examines the gestures provided as input, and matches them against the learned patterns to identify and recognize the sign. This is done using techniques such as hidden Markov models (HMMs), deep learning architectures like convolutional neural networks (CNNs), or a combination of various machine learning approaches.

I.6 Conclusion

In this Chapter, we discussed what is sign language and its different types, being the isolated and the continues signs. We also talked about Arabic sign language and its similarities and differences with other version of sign language, and we gave a brief introduction about sign language recognition to prepare us to the upcoming Chapters.

Chapter II

Deep Neural Network for Sign Language

II.1 Introduction

Deep neural networks (DNN) is a class of machine learning algorithms similar to the artificial neural network and aims to mimic the information processing of the brain[8]. They excel in image recognition tasks and they are widely used for similar jobs, and are frequently used for their accuracy and adaptive nature in the research field of automatic classification tasks[9].

II.2 Artificial Intelligence and Machine Learning

Artificial intelligence (AI) and machine learning (ML) have been around for decades, but it wasn't until their development that these technologies started to gain population and flourish.[10]

Before DL and CNNs, AI and ML were used for a variety of tasks, but their capabilities were often limited and basic. For example, early AI systems were used to play games like chess, but they struggled to generalize to new tasks. ML algorithms were also used for tasks like spam filtering and fraud detection, but they required large amounts of labeled data to train.

After the development of Deep learning and CNNs all that changed. They were able to achieve state-of-the-art results on a wide range of tasks, even with relatively small amounts of training data. This made them very important for researchers and developers, and led to populate and direct interest into AI and ML.[11]

Today, deep learning and CNNs are used in a wide range of applications, including:

- **Image classification:** CNNs can be used to classify images into different categories, such as cats, dogs, or flowers.

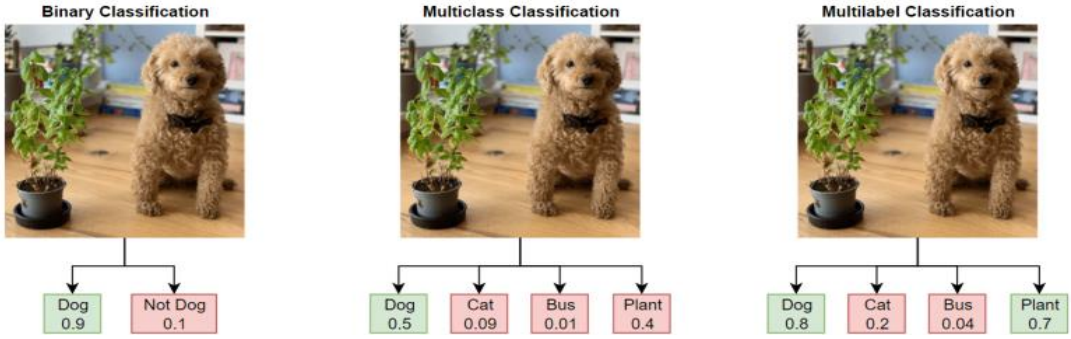

Figure II.1: Example of image classification[12]

- **Object detection:** CNNs can be used to: detect objects in images, such as faces, cars, or traffic signs.

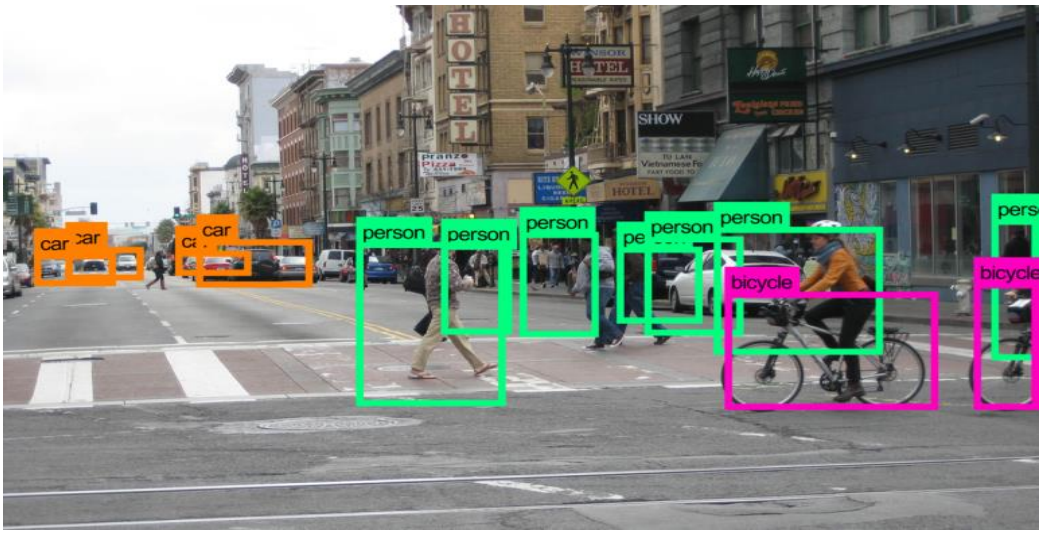

Figure II.2: A simple example of object detection in action [13]

- **Natural language processing:** CNNs can be used to process text, such as identifying sentiment or extracting keywords.

Deep learning and CNNs have revolutionized the field of AI and ML. They have made it possible to build powerful and accurate models that can be used to solve a wide range of problems.

II.3 Digital image Recognition

A digital image consists of pixels, which are small elements that collectively form the image. Each pixel has a specific numeric value representing its intensity or gray level.

Image recognition, is a digital image or video process to identify and detect an object or feature [14] , since software views the image as a numerical representation, it is upon artificial intelligence to recognize and classify the image correctly by recognizing the patterns and regularities in the numerical data representation of that image.

II.4 Deep Learning for Image Recognition

Deep neural networks have revolutionized the field of computer vision specially Image recognition. Traditional image recognition methods rely on features, such as edges, corners, and textures. Those features are extracted from images using multiple techniques such as edge detection and histogram equalization. These features are then used to train a classifier, like a support vector machine or a decision tree.

Deep neural networks (DNNs) have emerged and proved to be a powerful alternative to the traditional image recognition methods. They are inspired by the human brain, and can learn features directly from provided input images. They achieved state-of-the-art results on a variety of image recognition tasks, such as object classification, face recognition, and scene understanding.

There are two main types of deep neural networks which are used for image recognition: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are ideal spatial relationships tasks, such as object classification and scene understanding [11]. On the other hand RNNs are efficient for tasks like temporal relationships, such as video classification and natural language processing [32].

Deep Neural Networks are composed of three layers: Input Layer, Hidden Layers and Output Layer

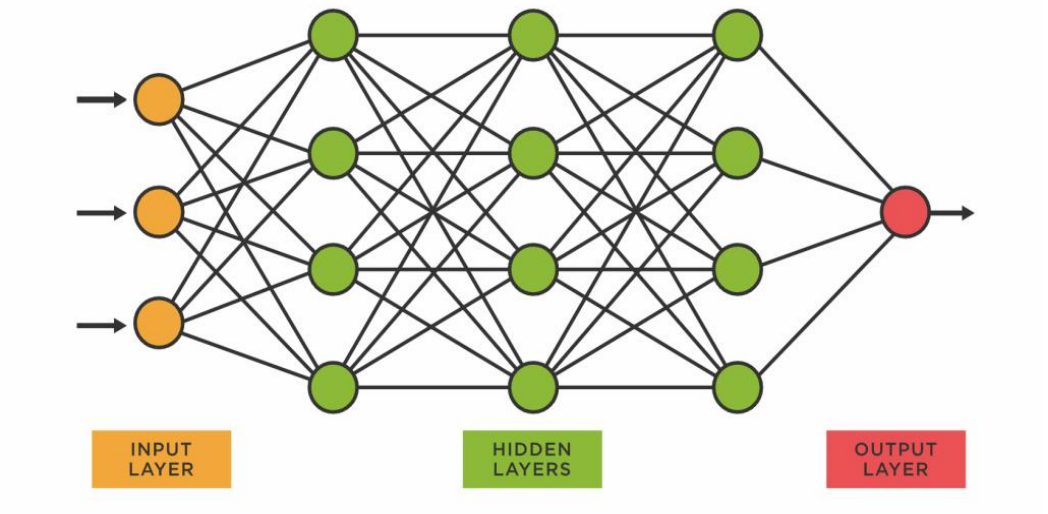

Figure II.3: The Layered Structure of Neural Networks[15]

II.4.1 Input Layer

This layer serves as the entry point for the data and acts as an interface between the input and the network. It receives the initial data, whether it's pixels from an image, words from text, or numerical values, and passes it forward to the following layers for further processing. The structure and characteristics of the input layer depends on the type and format of the data. Its main role is to receive the data and ease its flow through the network, without performing complex computations or transformations.

II.4.2 Hidden Layer

They are the intermediate layers between the input layer and the output layer. They play a crucial role in the network's ability to learn and extract complex representations from the input data. Hidden layers are composed of interconnected nodes, also known as neurons, where each neuron receives inputs from the previous layer and performs computations using activation functions. The number of hidden layers and the number of neurons in each layer are design choices that depend on the complexity of the problem being solved. These hidden layers progressively capture higher-level features and abstractions as the data moves through the network. They enable the network to learn non-linear relationships and extract complex patterns in the data, leading to enhanced model performance. The hidden layers are where the majority of the network's learning and computation happens, as they iteratively Improve the representations of the input data before ultimately producing the final output through the output layer.

II.4.3 Output Layer

They are the final layer that produces the network's predictions or outputs based on the computations and transformations performed by the preceding layers. The structure and characteristics of the output layer depend on the specific task demanded. For classification problems, the output layer often consists of nodes representing different classes or labels, where each node indicates the probability of the input belonging to a particular class. In regression tasks, the output layer typically contains a single node that produces a continuous value as the predicted

output. The activation functions used in the output layer depend on the nature of the problem, with softmax activation mostly used for multi-class classification and linear or sigmoid activation for regression tasks. The output layer's purpose is to provide the final output of the neural network, delivering the network's prediction or estimation based on the input data and the learned representations in the hidden layers

Deep networks can have hundreds of hidden layers, but standard neural networks have only 3.

II.5 Convolutional Neural Network CNN

Convolutional neural networks (CNNs) have in recent years achieved results which were previously considered to be purely within the human realm [16].

They are a type of deep learning algorithm that can be used for image recognition, spatial data analysis, computer vision, natural language processing, and other tasks. it is essentially a classification structure for classifying images into labeled classes [17]. CNNs are inspired by the organization of the visual cortex in the human brain, and their name comes from the convolution operation, which is one of their key components.

Convolutional neural networks are able to learn features from data directly, without the need to manually engineer the features, which makes them ideal for tasks where the features are not known, such as image recognition. They are one of the key forms of Deep Learning algorithms used to handle computer vision issues. This specific type employs what is known as a stack of convolution blocks to extract features and address a wide range of computer vision tasks.

Convolution is the systematic intertwining of two sources of information. Convolutions have been heavily used in image processing, primarily to blur and sharpen pictures, but also to perform other operations (e.g., improve edges and emboss). Convolutional neural networks, impose a pattern of local connection between neurons in neighboring layers. They employ filters (also known as kernels) to recognize which feature, such as edges, are present in a picture.

A Convolutional Neural Network has four primary operations:

- Convolution
- Non Linearity (ReLU)
- Pooling or Sub Sampling
- Classification (Fully connected layer)

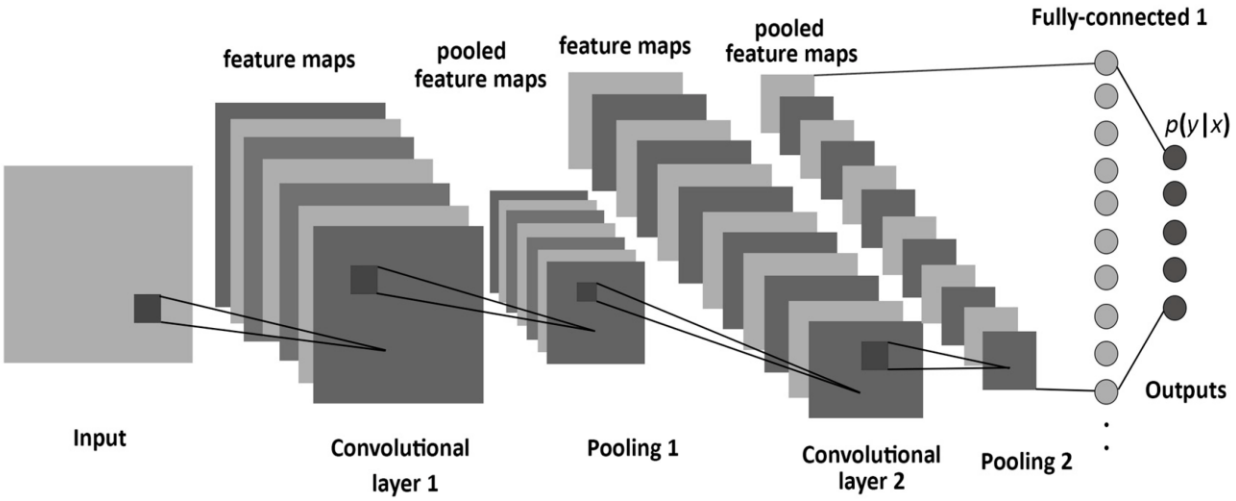

Figure II.4: The structure of a CNN [18]

II.6.1 Convolution Operation

A Convolutional Neural Network's initial layer is usually a Convolutional Layer. Convolutional layers perform a convolution operation on the input and transmit the output to the following layer. Convolutions combine all of the pixels in their receptive area into a single value. For example, if you apply a convolution to a picture, you will reduce the image size while also combining all of the information in the field into a single pixel. The convolutional layer's final output is a vector. We may employ several types of convolutions depending on the sort of issue we need to solve and the features we want to learn.

The 2D convolution layer, abbreviated as conv2D, is the most often used form of convolution. In a conv2D layer, a filter or kernel "slides" through the 2D input data, executing element-wise multiplication. As a consequence, the findings will be summed into a single output pixel. For each point it slides over, the kernel will conduct the same procedure, changing a 2D matrix (picture) into a distinct 2D matrix of features. [33]

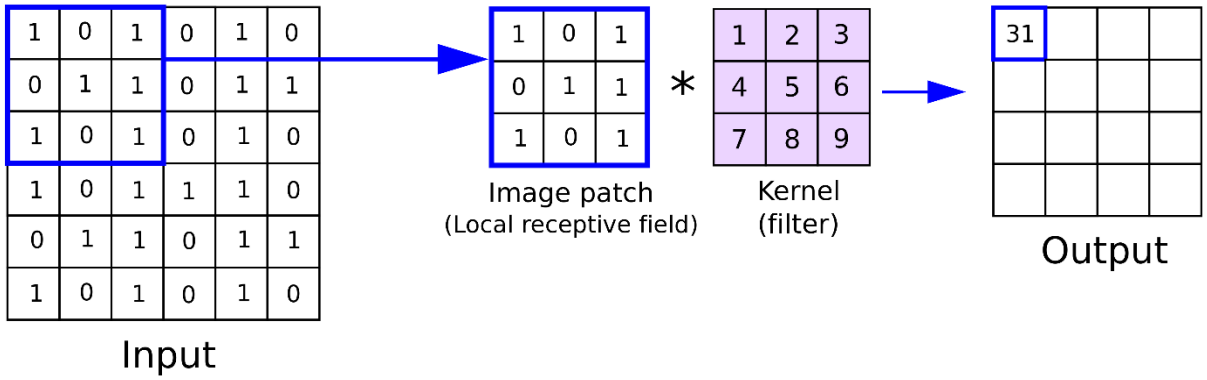

Figure II.5: Convolutional layer[19]

II.6.2 Non Linearity

After the convolution operation is completed and the feature maps are generated, the next step is to apply a non-linear activation function. This is because non-linearity helps to improve neural networks by speeding up training. A non-linear activation function introduces non-linearity into the neural network, which allows it to learn more complex relationships between the input and output data. This can help to improve the accuracy of the neural network and speed up the training process.

The gradient computation is pretty simple (either 0 or 1) depending on the sign of x. Furthermore, the computing step of a ReLU is straightforward: all negative components are set to 0.0, eliminating the need for exponential functions, multiplication, or division operations. "RelU(x)=max(0,x)" Despite this, the ReLU function discards negative values that may contain information. This is why the author proposes a new activation function.

II.6.3 Max Pooling

This phase entails replacing a (n x n) area with the maximum value inside, this surgery is carried out in order to:

Choosing the highest activation in a local region, thereby providing a small degree of spatial invariance.

It reduces the size of the activation for the next layer by a factor of n^2 With a smaller activation size, a smaller number of parameters need to be learned in the later layers.

Here are other types of polling such as global-average-pooling, winner-takes-all-pooling and stochastic-pooling.

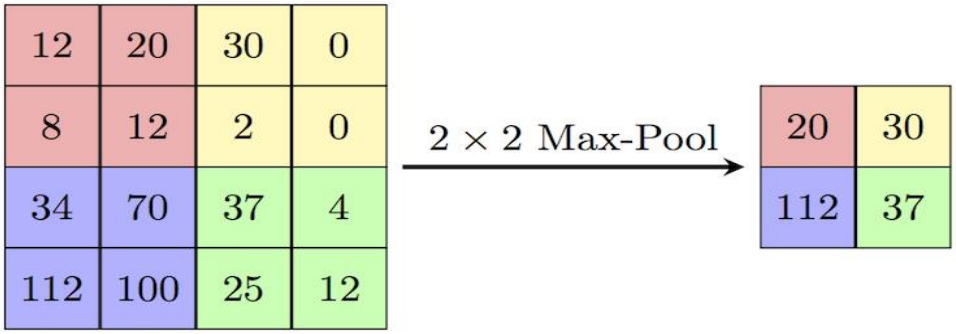

Figure II.6: example of max-pooling [20]

II.6.4 Fully connected layer

The final phase in the CNN architecture is the fully connected layer or a sub neural networks that has the right to decide or make segmentation after getting features from the previous convolution block**.**

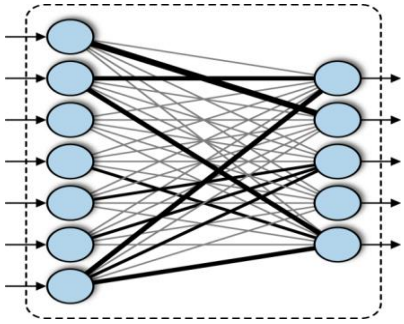

Figure II.7: A fully connected layer [21]

II.6 Conclusion

In this chapter, we have explored the fundamental concepts and background of Image recognition in the context of deep learning. We started by discussing the availability of digital images and their properties, which lead to various tasks such as recognition, and then we discussed Image recognition as a crucial task in various clinical applications.

We then dived into the foundations of deep learning, highlighting its revolutionizing impact on Image recognition and why we prioritize it over machine learning.

Furthermore, we showed the basic and fundamental concepts of deep learning and what a deep learning algorithm is composed of for both neural and convolutional neural network architectures.

Overall, this chapter provided a comprehensive overview of the background of Image recognition and deep learning. It laid the foundation for understanding the upcoming chapters, which will discuss more advanced titles.

Chapter III

Related work

III.1 Introduction

In this chapter, we talk about similar works of other people. The method and architecture they used, there trained model and the accuracy and witch language of signs they used.

III.2 Arabic Sign Language Recognition for Impaired People based on Convolution Neural Network

(Shahin, O. R., & Taloba, A. I. (2022)) they developed a system based on deep learning methodology, specifically utilizing Convolutional Neural Networks (CNNs), for recognizing Arabic Sign Language (ArSL) gestures and hand signs. The research focuses on addressing the challenge of variations in ArSL across different territories and states. The proposed system incorporates wearable sensors for data collection and utilizes a deep Convolutional network for feature extraction. The system is capable of recognizing 30 hand sign letters of ArSL with reasonable and moderate accuracy. The output of the system is vocalized speech based on the input of Arabic sign language hand gestures. The results of the system's recognition were successful for 90% of the people involved in the study.[22]

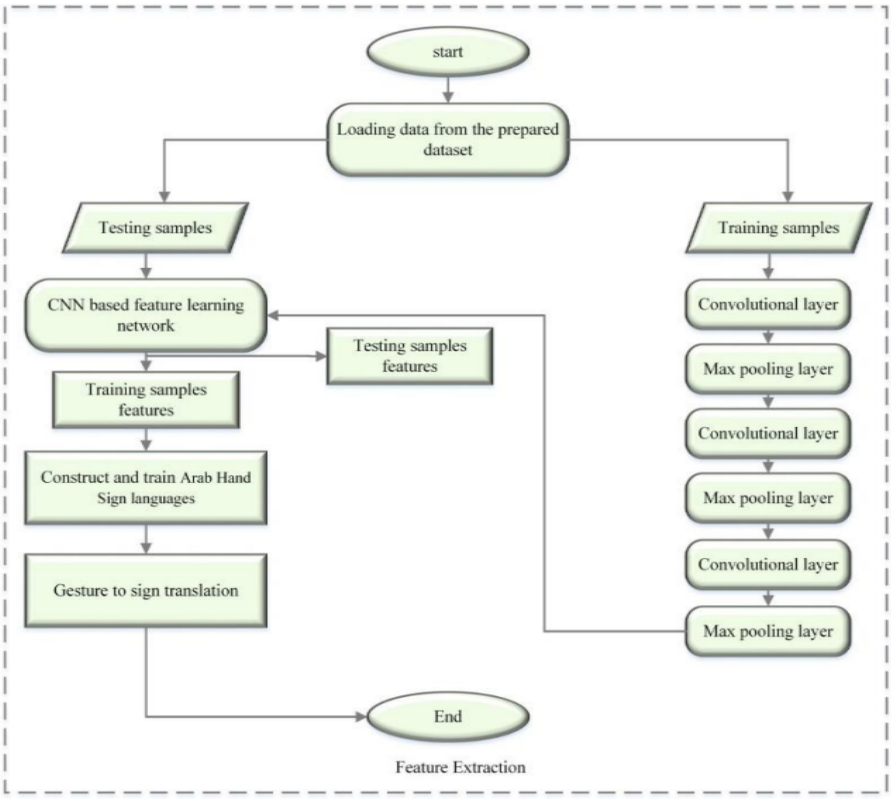

Figure III.1: Flow Chart for their CNN Proposed Model[22]

III.3 Hand Gesture Recognition for Sign Language Transcription

(Vazquez Lopez, I. (2017)) after a lot of research he developed a system that utilizes Convolutional Neural Networks (CNNs) to recognize hand gestures of American Sign Language (ASL) based on depth images captured by the Kinect camera. In the process of this research, the author created a new dataset consisting of depth images of ASL letters and numbers. They conducted an empirical assessment and compared their image recognition method with a similar dataset for Vietnamese Sign Language.[23]

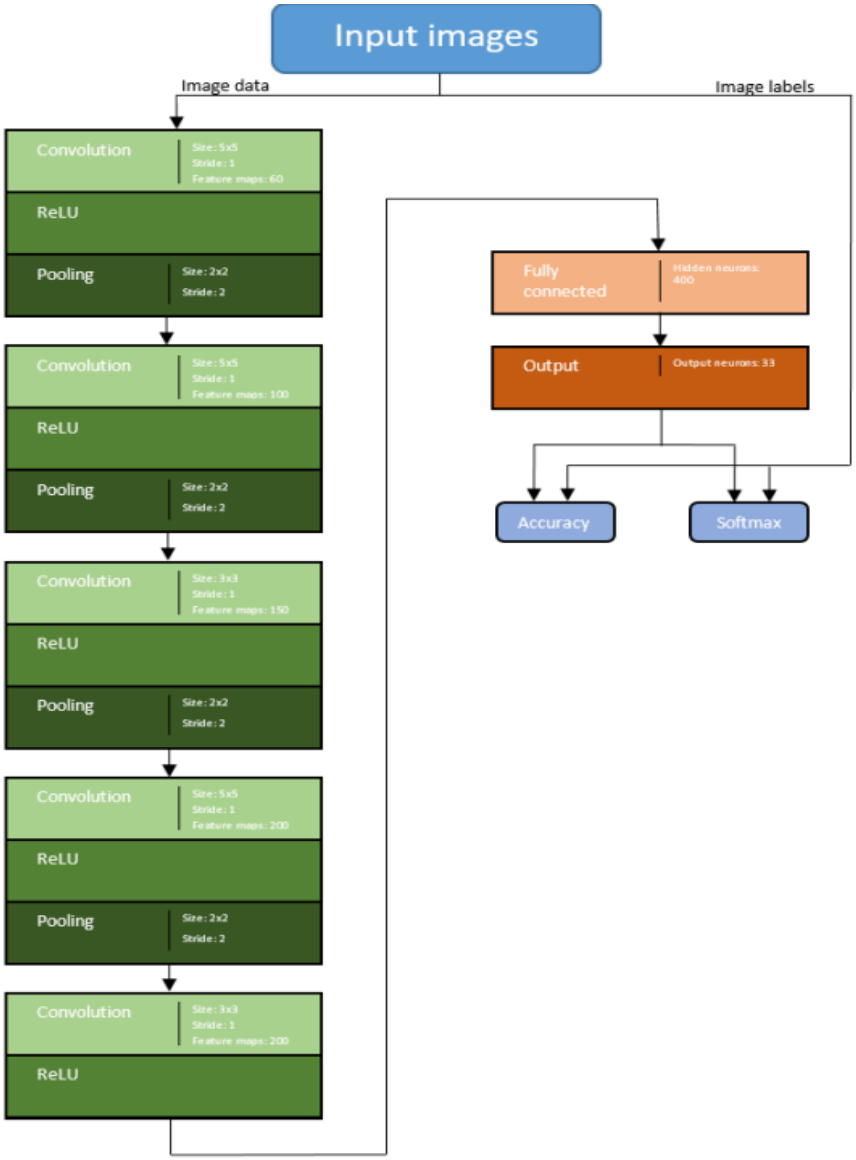

Figure III.2: Architecture of the CNN [23]

III.4 Deep Learning Application on American Sign Language Database for Video-Based Gesture Recognition

There are a lot of limitations of existing technologies for American Sign Language (ASL) translation.

As an instance, (Saleem, M. M. (2020)) developed a model implemented in MATLAB 2020b and explores the utilization of current neural networks against a transformed data set of

ASL. He also analyzes the efficiency of pre-trained networks in combination with detection and segmentation algorithms. Additionally, he explores the impact of machine learning strategies like Transfer Learning on training a model for recognition. The research goals include manufacturing and augmenting the data set, applying transfer learning to create various models, comparing the accuracy of each model, and presenting a novel pattern for gesture recognition.[24]

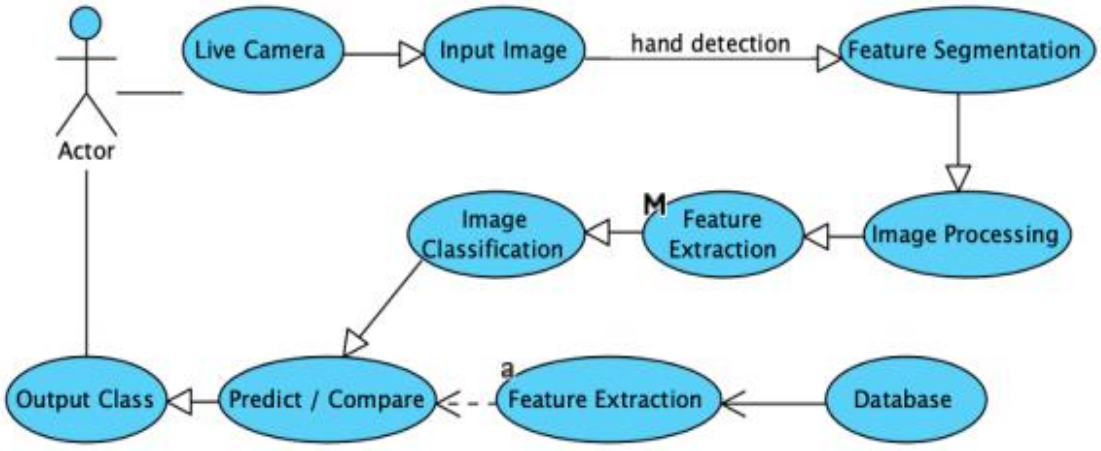

Figure III.3: The program workflow [24]

III.5 Architectures for Real-Time Automatic Sign Language Recognition on Resource-Constrained Device

Nowadays ,computing devices are so powerful, when combined with new cameras and sensors capable of detecting objects in three dimensional.

As an instance, (Blair, J. M. (2018)) developed a functional prototype of a sign language recognition system using three architectural patterns seen in speech recognition systems. He applied a Hidden Markov classifier and achieved an accuracy of 75-90%. The performance impact of each architecture, as well as the data interchange format, is then measured based on response time, CPU, memory, and network usage across an increasing vocabulary of sign language gestures. The results suggest that a partially-offloaded client-server architecture, where feature extraction

occurs on the client device and classification occurs in the cloud, is the ideal choice for most vocabularies.[25]

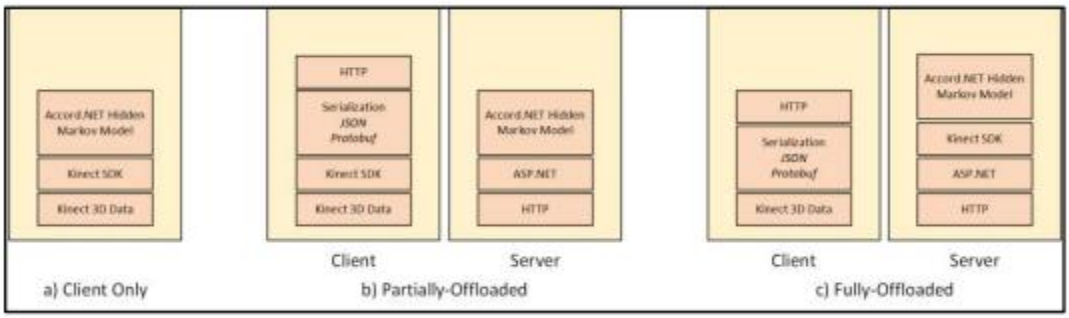

Figure III.4: Architecture diagram of ASLR system [25]

III.6 Arabic Sign Language Recognition and Generating Arabic Speech Using Convolutional Neural Network

Recent progress in the computer vision field has geared us towards the further exploration of hand signs/gestures' recognition with the aid of deep neural networks

As an instance, (Kamruzzaman, M. M. (2020)) proposes a vision-based system that utilizes deep neural networks, specifically Convolutional Neural Networks (CNNs), for recognizing hand signs and gestures in Arabic Sign Language (ArSL). The system aims to automatically detect hand sign letters and translate them into spoken Arabic using a deep learning model. The proposed system achieves a 90% accuracy in recognizing Arabic hand sign-based letters. He suggests that using more advanced hand gesture recognition devices, such as Leap Motion or Xbox Kinect, can further improve the system's accuracy. Finally, the recognized hand sign-based letters are converted to text and fed into a speech engine to produce audio output in the Arabic language.[26]

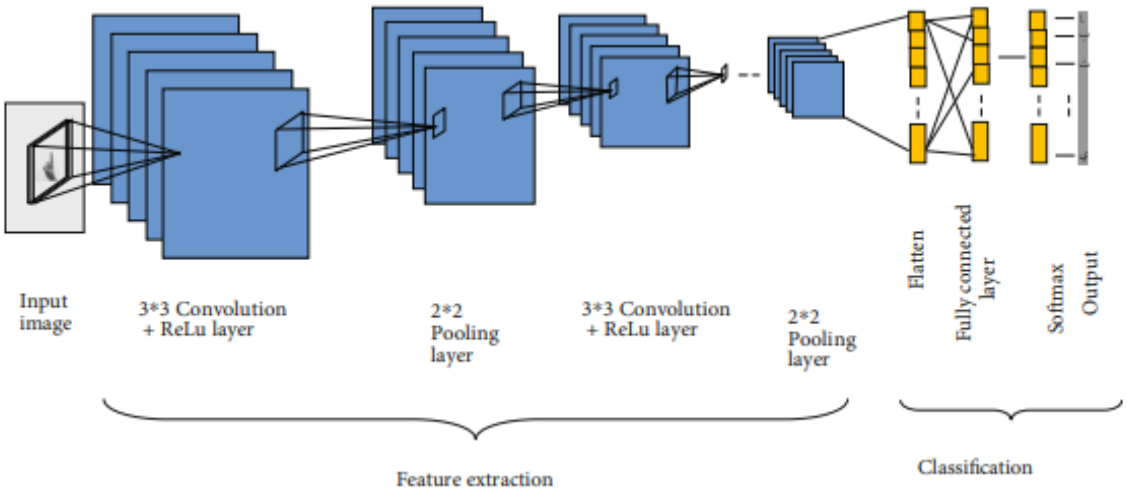

Figure III.5: Architecture of Arabic Sign Language Recognition using CNN [26]

III.7 A Study on Hand Gesture Recognition Technique

(Meena, S. (2011)) presented a technique for human computer interface through hand gesture recognition. Is able to recognizing 25 static gestures from the American Sign Language hand alphabet. The objective of the research is to develop an algorithm with reasonable accuracy for hand gesture recognition. The process involves segmenting the grayscale image of the hand gesture using the Otsu thresholding algorithm, which classifies the image into hand and background. Morphological filtering techniques, including dilation, erosion, opening, and closing operations, are used to remove background and object noise from the segmented image. Canny edge detection is applied to find the boundary of the hand gesture, and a contour tracking algorithm tracks the contour in a clockwise direction. The contour of the gesture is represented by a Localized Contour Sequence (L.C.S) using perpendicular distances between contour pixels and a chord connecting the end-points. These extracted features are used as input to classifiers, including linear classifier, Multiclass Support Vector Machine (MCSVM), and Least Square Support Vector Machine (LSSVM). The experimental results show recognition accuracies of 94.2% with the linear classifier, 98.6% with MCSVM, and 99.2% with LSSVM.[27]

III.8 Arabic Sign Language Recognition: A Deep Learning Approach

(ALMAHRI, H. G. A. A. (2022)) review revealed that the main challenge in Arabic Sign Language Recognition is data collection, as existing datasets lack variety and do not represent real-life scenarios. Current methods for data collection are either expensive or easily influenced by the surrounding environment.To address this challenge, He proposes a solution using MediaPipe, which enables data collection through a webcam. The framework is leveraged to build a recognition system for Emirati Sign Language, specifically recognizing signs for the seven Emirates. An LSTM model is employed, achieving 100% accuracy in the testing dataset.[28]

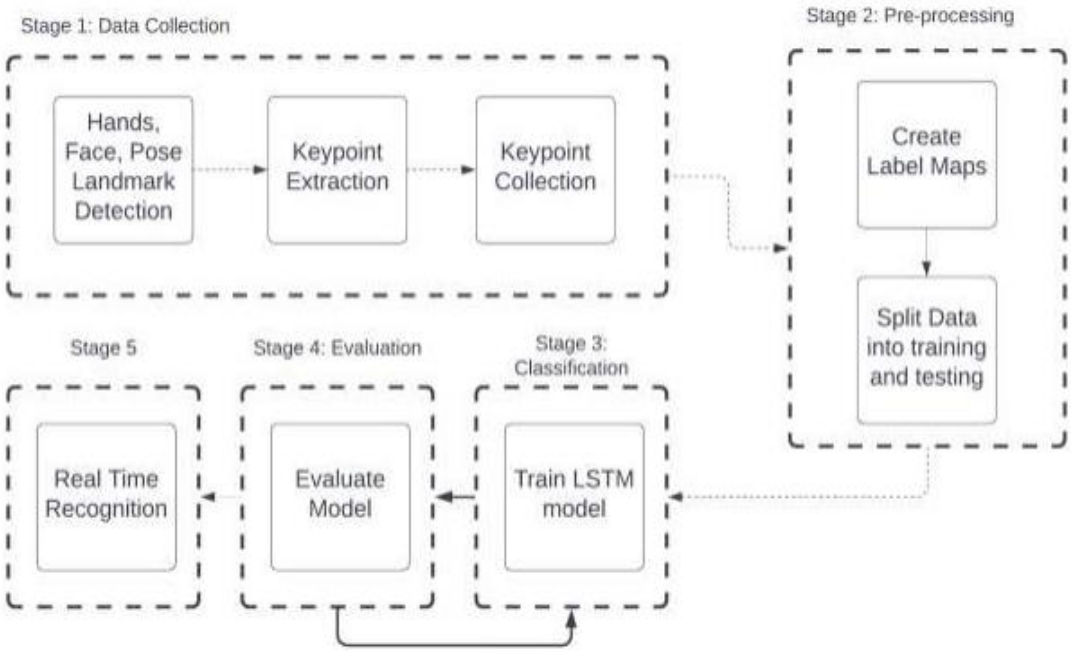

Figure III.6: System Architecture [28]

III.9 Sign Language Identification and Recognition: A Comparative Study

In this study, (Sultan, A., Makram, W., Kayed, M., & Ali, A. A. (2022)) [23] focuses on two primary tasks in SL processing: Sign Language Recognition (SLR) and Sign Language

Identification (SLID). SLR aims to translate the signer's conversation into sign tokens, while SLID is focused on identifying the language being used by the signer. The article delves into the commonly used datasets in the literature for these tasks, encompassing both static and dynamic datasets collected from various sources. These datasets include numerical, alphabetical, word, and sentence representations from different SLs.[29]

The devices necessary for building these datasets are discussed, as well as the preprocessing steps applied before training and testing the models. The article compares different approaches and techniques employed on these datasets, encompassing both vision-based and data-glove-based approaches. Special attention is given to analyzing and focusing on the main methods used in vision-based approaches, such as hybrid methods and deep learning algorithms. they presents graphical depictions and tabular representations of various SLR approaches.[29]

III.9 Conclusion

We talked about different architectures from other articles and how they used them for sign language recognition and the accuracy of their models.

Chapter IV

Methodology and Applied Techniques

IV.1 Introduction

To develop a robust model for Sign Language Recognition that translates it into readable interpretation, there are multiple approaches combining multiple steps towards acquiring this model.In this chapter we will talk about our approach.

IV.2 DataSet

As the dataset for Arabic words was unfortunately unavailable we had to construct our own dataset, our first step was to find a reliable and practical source that could provide us with accurate signs and their corresponding meanings in spoken Arabic, and that source had to be efficient on delivering the data without any protocols. After a long research, we discovered a community called (the Community Development Authority in Dubai) [30], which had a YouTube channel offering Emirate Sign Language tutorials accompanied with the translation of the sign's meaning in Arabic.

Once we had identified the source, we carefully selected the most commonly used words then we followed the tutorials provided by the organization and performed the signs exactly as demonstrated using a webcam.

As we discussed earlier there are several approaches to capture the continues signs, we chose to use the advantages that deep learning offers, DL takes the data provided and tries to extract all the possible features from it without asking questions, so we gathered the combinations of the continues signs and put them together in one folder creating a single class named with the full meaning of that sign, now the DL will extract the features from the different combination that form the signs in those folders and assigns them each to its corresponding class. this method requires a lot of data on each class so the DL can learn efficiently and extract the correct features from each sign. We used this approach since the gestures that form the continues signs that we collected ('kayfa al hal' and 'Al salam alaykom') don't represent any meaning if performed individually, for example if you take 'kayfa al hal' combination of signs, the first sign and the second don't have any meaning if performed by them self, another reason for choosing this approach was when this model is deployed live, if someone is performing a continues sign, and somehow for any reason the frames captured from the live feed where a component of a continues

sign wasn't clear enough, the model could recognize the meaning of the performed sign from recognizing other components of the sign.

then after gathering the data we implemented some algorithms to retrieve and cover only the essential parts and features necessary for the model to apprehend the fundamental factors of these signs and remove any unbalance in the images that can affect the model and mislead it into extracting irrelevant features that will harm the accuracy of the classification.

Unfortunately, we couldn't find the number dataset either, so we had to go with out it. But, obtaining the dataset for Arabic alphabets was comparatively easier, as they are generally standardized across various Arabic sign language variations. We utilized the (Arabic Alphabets Sign Language Dataset (ArASL)) [31] published on 5th November 2018. This dataset consists of a total of 54,049 images of Arabic sign language alphabets performed by over 40 individuals, covering all 32 standard Arabic alphabets. The number of images per alphabet varies across the classes. Additionally, a sample image showcasing all Arabic language signs is included. The accompanying CSV file provides the corresponding labels for each Arabic sign language image based on the image file names.

The dataset we obtained was quite extensive, containing all the arabic alphabets and a selection of words from both isolated and continuous signs.

In total, we compiled 39 classes, with each class containing an average of 1300 to 1700 images.

Figure IV.1: Representation of the Arabic Sign Language for Arabic Alphabets [31]

Figure IV.2: images from data set before pre-processing

IV.3 Pre-Processing

To ensure consistency and remove potential biases, we performed some pre-processing techniques on the dataset. We converted all the images to grayscale to avoid relying on skin color as a distinguishing feature. Additionally, we used the mediaPipe library for hand detection and tracking. This library allowed us to accurately pinpoint the location of the hands in the captured images and crop them accordingly to eliminate any external information that could mislead the model's gesture recognition.

Furthermore, we utilized the same library to draw a shape that follows the path of the fingers, it almost represents the skeleton of the hand. This approach proved beneficial as it leverages the hand shape as a primary feature. Even when dealing with lower-quality images, the model could rely on the drawn hand shape to accurately recognize and classify the gestures since drawing the handSkeleton is pretty consistent even in bad lighting conditions.

this step was very challenging to execute because of the alphabets dataset was already pre-processed, it was reshaped into 64x64 shape, which makes reshaping it to a normal shape impossible without losing some impotent details, also the providers of this dataset removed RGB colors using the grayscale method, recoloring the images back to their original colors is very difficult even when using AI, all these modified features out of these images are necessary for the mediaPipe library to detect and draw the hand-skeleton shape, as it was predicted the library failed to draw that shape on the majority of the images because of the pre-processing modifications on them, and reversing these modifications is impossible without using another specified AI models which are hard to find especially high quality ones, neither do we have the time to spend searching and trying them, we had no choice but to do it manually, so we resized the images to the biggest size possible without losing necessary details for the hand-skeleton function to execute effectively.

Even after the efforts of reversing the pre-processing, the hand-skeleton method failed to draw on almost 60-70% of the images, so we chose the ones with the successful operation and ignored the rest.

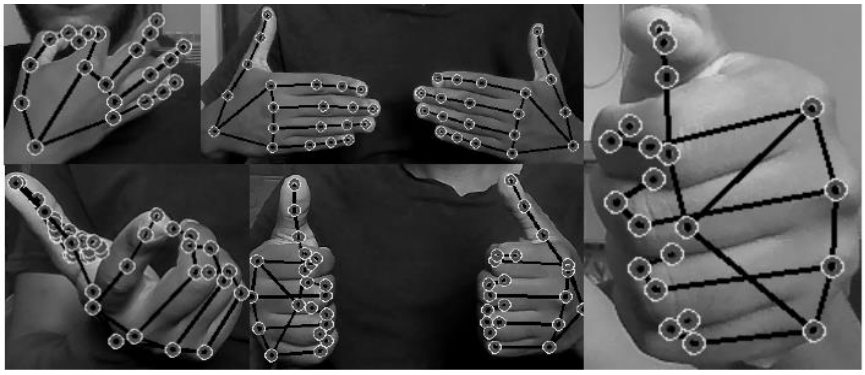

Figure IV.3: Images from data set after Pre-Processing

IV.4 Data-augmentation

Since we lost a considerable amount of the alphabets dataset, and the insufficient the numbers of the images of the words, we ended up with a small dataset, at least some classes didn't contain enough images to achieve good results. so we had to use data augmentation.

In our augmentation process, we employed several key parameters. First, we used the **rotation_range** parameter, which allowed us to rotate each image by a range of specified angles, ranging from negative to positive degrees.

Next, we utilized the **zoom_range** parameter, which enabled us to zoom the images by a specified percentage. This zooming effect added further diversity to the dataset, as it simulated images captured from different distances or perspectives.

To introduce additional variability, we employed the **height_range** and width_range parameters. These parameters allowed us to shift and move the content of the images horizontally or vertically, creating the impression of slight displacements. This was useful in replicating different hand positions or gestures, further enriching the dataset.

Finally, we used the **fill_mode** parameter, which was set to 'nearest'. This mode ensured that any empty or newly created pixels resulting from the augmentation process were filled in using the nearest neighboring pixel values. This preserved the overall appearance and integrity of the augmented images.

we were able to significantly expand the size and diversity of our words dataset, removing the limitations imposed by the limited size of captured images

The acquired dataset was split by 80% 20% for training and testing respectively.

IV.5 Technologies and Environment

For our CNN architecture and data pre-process we used Python as the main programming language, we also used TensorFlow and its sub-libraries as well as several other matrix manipulation libraries.

The environment that we implied this research on is **google colab** with a t4 GPU and 12G RAM. We used google Drive to contain our Dataset.

IV.6 Conclusion

In this chapter we discussed our approach towards pre-training methods, we talked about our dataset and different methods that we used to pre-process it and how we augmented it. This will lay foundation to ease the understanding of the next chapters.

Chapter V

Experiment, Results and Discussion

V.1 Introduction

After obtaining a model it is required to test it and experiment with it to understand it's behavior and locate the areas that needs improvement. In this chapter we will talk about our CNN architecture and it's components, then we will view our experiment with it, then discuss how we evaluated the model and the results we achieved.

V.2 Proposed Architecture

The model used in this experiment is a Convolutional Neural Network (CNN) implemented with the Keras library. CNNs are well-suited for image classification tasks due to their ability to capture spatial hierarchies and local patterns in images, this our Architecture:

1. Input Layer:

- The input shape is determined by the shape of the training data (64, 64, 1).
- The input shape consists of the image dimensions, such as height, width, and channels.

2. Convolutional Layers:

- The architecture begins with a Conv2D layer with 64 filters and a kernel size of (3, 3).
- Each Conv2D layer applies a set of learn-able filters to the input, extracting relevant features from the images.
- ReLU activation is applied after each Conv2D layer, introducing non-linearity to the network.

3. Batch Normalization:

- After each Conv2D layer, BatchNormalization normalizes the activations of the previous layer, helping to stabilize and speed up the training process.

4. Max Pooling Layers:

- MaxPooling2D layers follow each Conv2D layer, reducing the spatial dimensions of the output feature maps while retaining the most important information.
- Each MaxPooling2D layer uses a (2, 2) pooling window to downsample the input.

5. Flatten Layer:

- The Flatten layer converts the 2D feature maps from the previous layer into a 1D feature vector, preparing it for input into the dense layers.

6. Dense Layers:

- Two dense layers follow the Flatten layer, each with different units (512 and 256) and ReLU activation.
- The dense layers are fully connected layers, responsible for learning high-level representations based on the extracted features from the convolutional layers.

7. Dropout:

- Dropout layers with a rate of 0.5 are applied after each dense layer.
- Dropout randomly sets a fraction of input units to 0 during training, which helps prevent overfitting by introducing regularization.

8. Output Layer:

- The final dense layer has units equal to the number of classes in the dataset (len(classes)), and it uses softmax activation.
- Softmax activation produces a probability distribution over the classes, indicating the model's predicted probabilities for each class.

9. Compilation and Training:

- The model is compiled with the Adam optimizer, which is an adaptive learning rate optimization algorithm.
- The loss function used is sparse categorical cross-entropy, suitable for multi-class classification tasks.
- The accuracy metric is used to monitor the model's performance during training.
- The model is trained using the fit() function, with a batch size of 128, 30 epochs, and a validation split of 20% is used to evaluate the model's performance on a portion of the training data during training.

This architecture aims to capture and learn hierarchical features from the input images through multiple convolutional and pooling layers. The dense layers provide a high-level representation of the extracted features, leading to improved classification performance. Dropout is employed to prevent overfitting, and batch normalization enhances the stability and speed of training

Layer (name)	Layer (type)	Output Shape	Param #
conv2d	Conv2D	(None, 62, 62, 32)	320
max_pooling2d	MaxPooling2D	(None, 31, 31, 32)	0
conv2d_1	Conv2D	(None, 29, 29, 64)	18496
max_pooling2d_1	MaxPooling2D	(None, 14, 14, 64)	0
conv2d_2	Conv2D	(None, 12, 12, 128)	73856
max_pooling2d_2	MaxPooling2D	(None, 6, 6, 128)	0
flatten	Flatten	(None, 4608)	0
dense	Dense	(None, 128)	589952
dense_1	Dense	(None, 40)	5160
Tottal			687784

Table V.1: Model Summary

V.3 Experiment

As we described earlier, our architecture uses a CNN model with three sets of convolutional layers followed by batch normalization and max pooling. It includes two fully connected layers with ReLU activation and dropout regularization. The final output layer uses softmax activation for multi-class classification. The model is trained using the Adam optimizer with sparse categorical cross-entropy loss.

We trained this architecture for 30 epochs, using a batch size of 128, we had 273180 samples in the training set. the model took about 35 minutes to complete the training phase.

After obtaining astonishing results, we experimented with the model to test it in difficult situations, so we introduced noise into the testing images with variant levels. We picked about four images form four different classes and we added noise to them using Gaussian Filter with different levels of intensity up to 100%.

This experiment was quite interesting because of the variety of results from class to class, which enhances our understanding of the model. The class we chose in this experiment were 'al salam alaykom', 'alhamdoulilah', 'ism-2', and 'ain'.

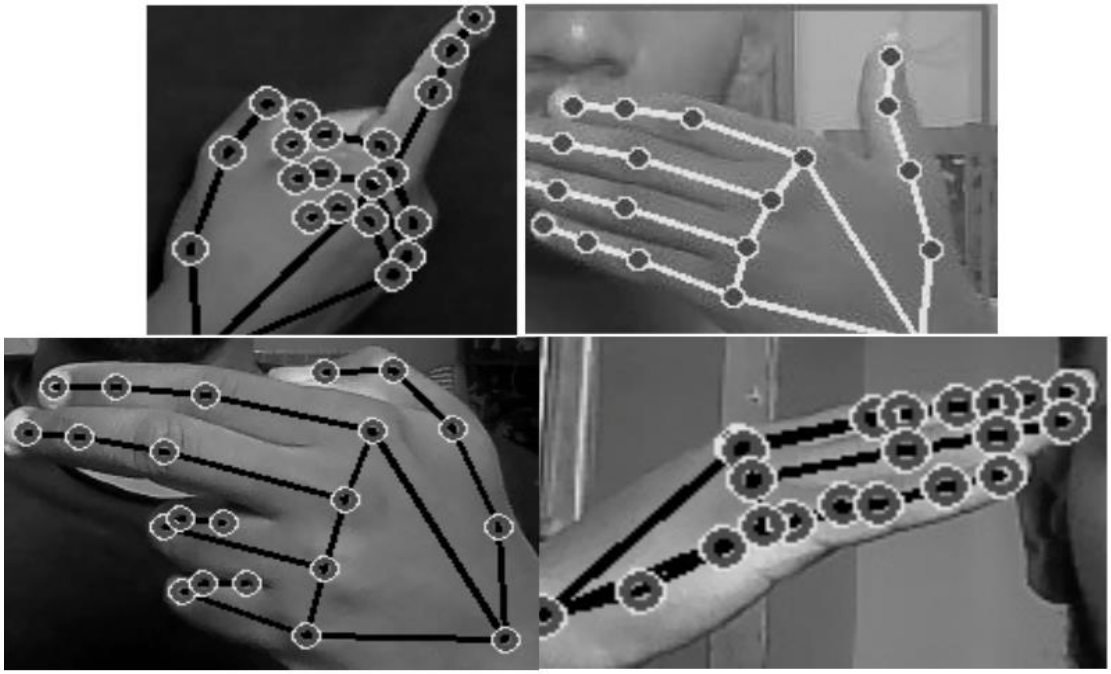

Figure V.1: Images of signs we chose from our data set

These are the images we chose respectively from our dataset. starting with 'al salam alaykom' sign, first we used 30% of noise intensity.

Here is the image of the sign after the noise

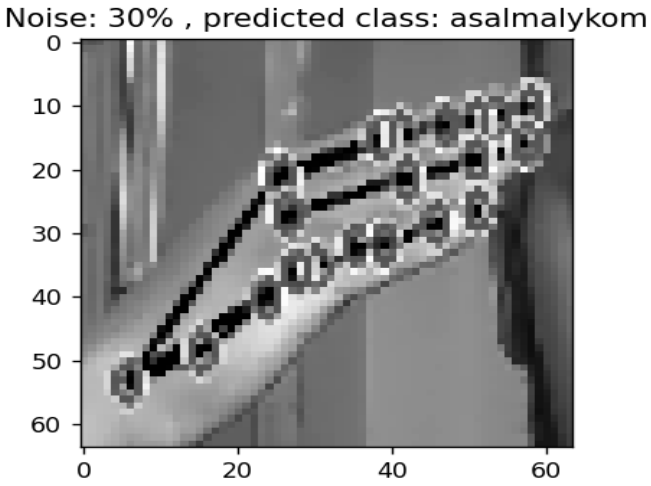

Figure V.2: 'al salam alaykom' sign (30% of noise intensity)

The model was able to predict it accurately. Next we raised the intensity into 50%

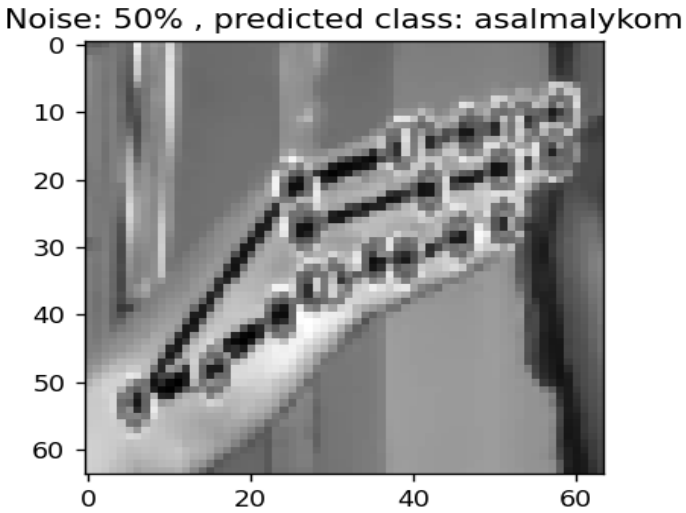

Figure V.3: 'al salam alaykom' sign (50% of noise intensity)

The model was also able to recognize and predict it correctly, now we increased the intensity into 80%

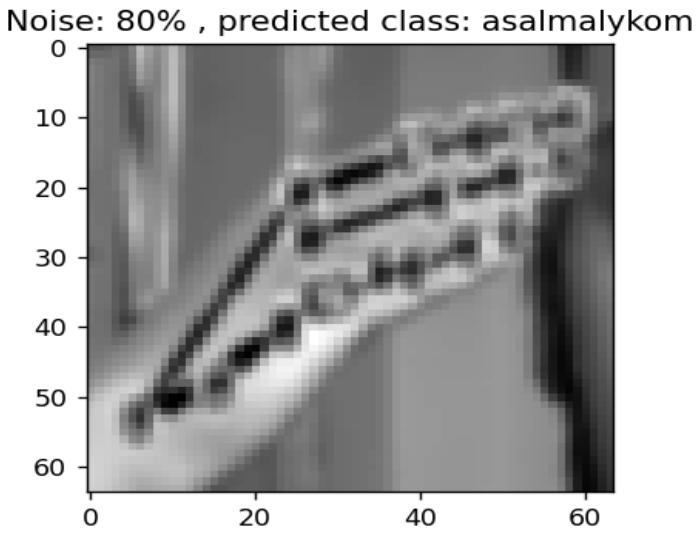

Figure V.4: 'al salam alaykom' sign (80% of noise intensity)

The model also predicted the image correctly. Now, since the model handled the noise consistently, we used 100% noise intensity.

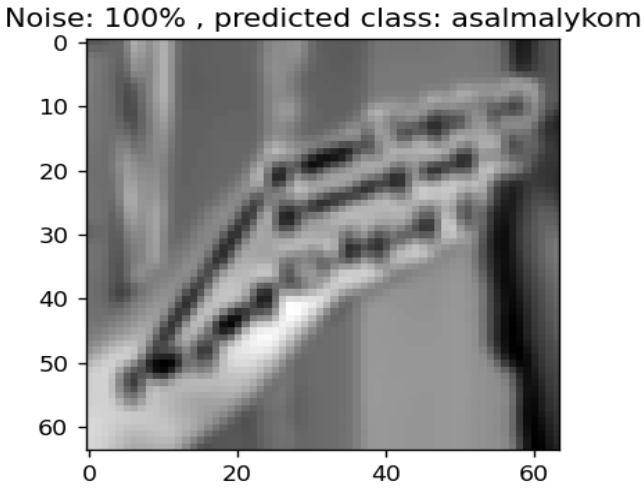

Figure V.5: 'al salam alaykom' sign (100% of noise intensity)

As you can see from the picture we barley can see the hand clearly, but amazingly the model predicted the sign accurately.

The second sign was 'alhamdoulilah', as usual we started with 30% noise intensity, the model accurately predicted the sign

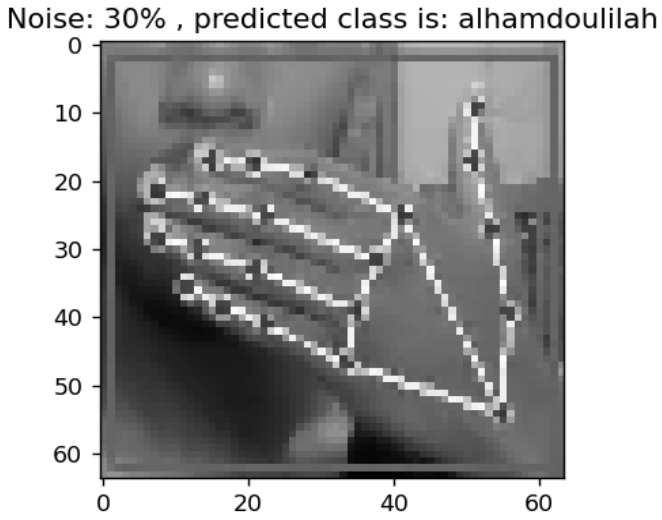

Figure V.6: 'alhamdoulilah' Sign(30% of noise intensity)

Then we augmented the intensity into 50%, here is what the sign looks like after adding the noise.

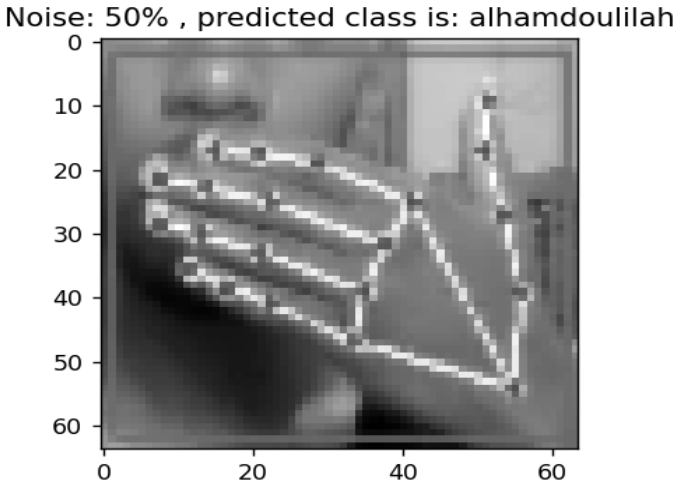

Figure V.7: 'alhamdoulilah' Sign(50% of noise intensity)

The model predicted this one correctly as well. Third one we used 80% intensity, this is the outcome after the noise

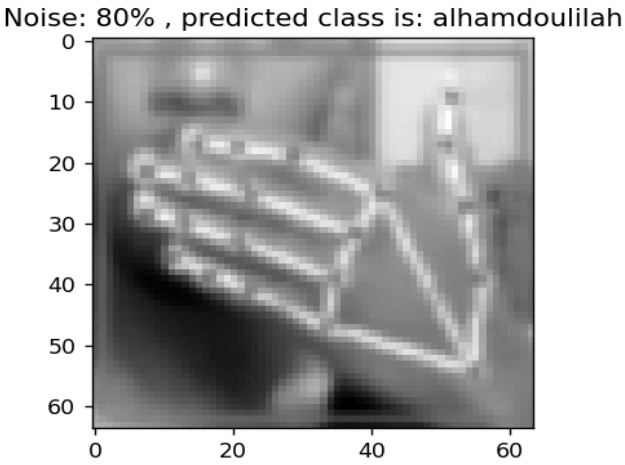

Figure V.8: 'alhamdoulilah' Sign(80% of noise intensity)

The model succeeded again in predicting the sign. The final level of noise intensity was the max percentage, 100%.

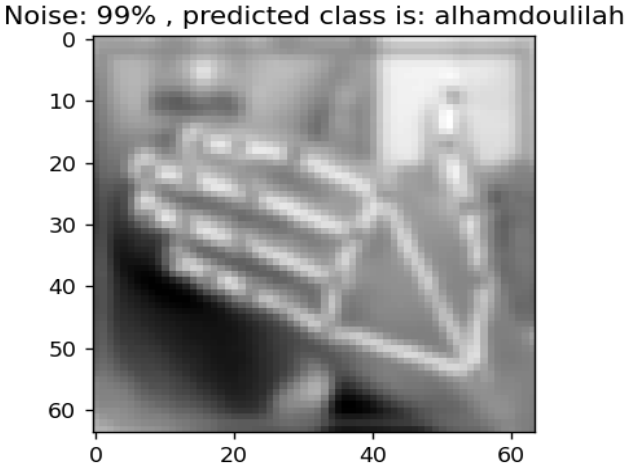

Figure V.9: 'alhamdoulilah' Sign(99% of noise intensity)

The model failed in classifying the noised sign, we found out that at the intensity of 99% the model still predicted the sign correctly.

The third sign was 'ain', as every previous sign, the model tolerated the noise at the first intensity which is 30%

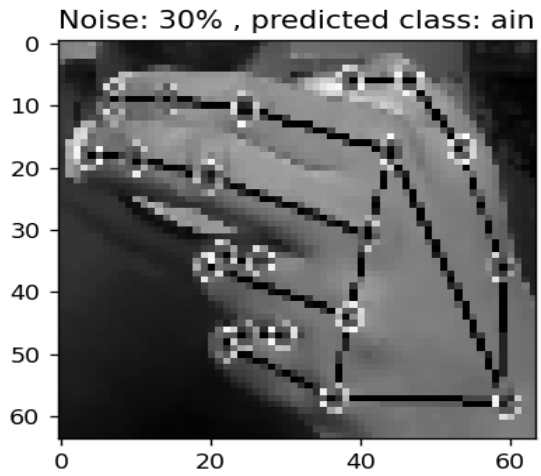

Figure V.10: 'ain' sign(30% of noise intensity)

At the second level of intensity, 50% the input image look like this

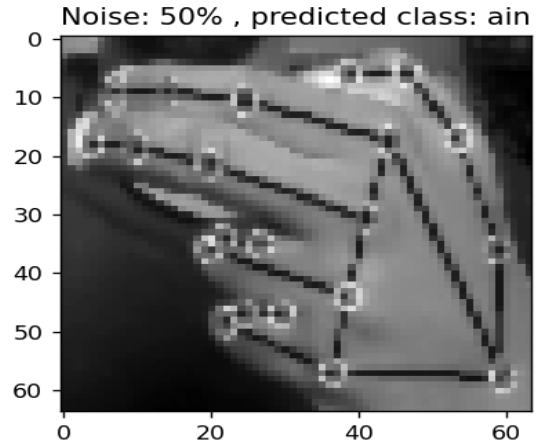

Figure V.11: 'ain' sign(50% of noise intensity)

The model was able to classify it correctly. The third stage which is 80%, this is the image.

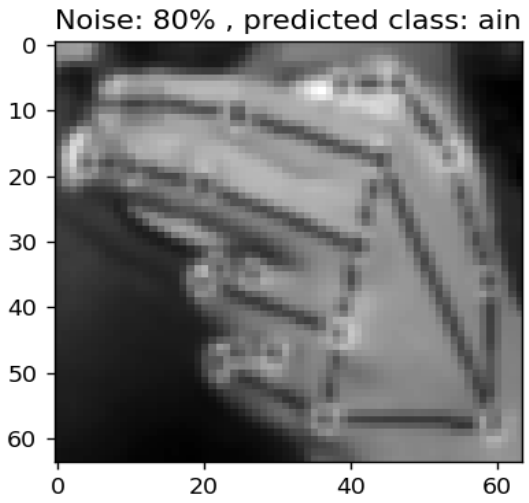

Figure V.12: 'ain' sign(80% of noise intensity)

The final stage, the maximum intensity 100%

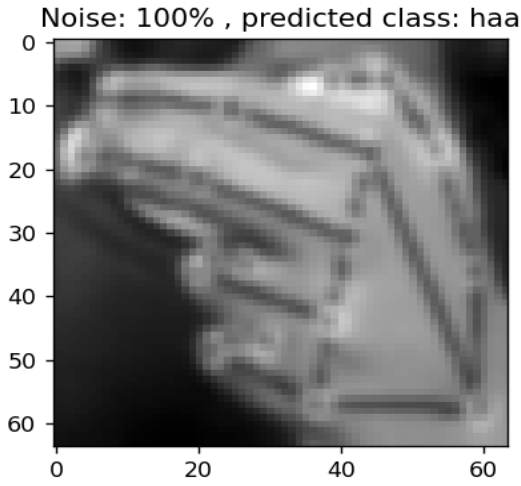

Figure V.13: 'ain' sign(100% of noise intensity)

The model failed to identify and recognize the image at this level of noise intensity. After the testing between the last two levels we found that the model recognizes the images until the noise reaches 86% then it starts falsely classifying it as 'haa'.

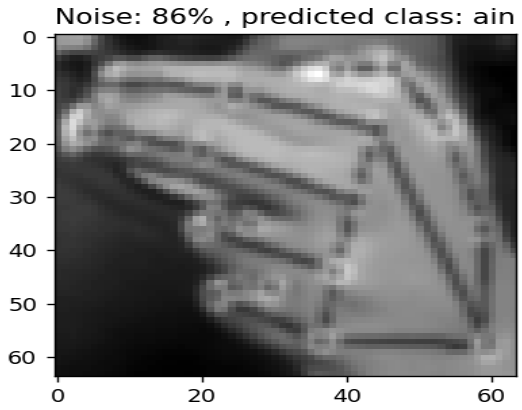

Figure V.14: 'ain' sign(86% of noise intensity)

The last sign was 'ism-2' which is the second way to preform the sign 'ism' which means 'name' in English. First, starting with the 30% intensity, the model succeeded to predict the image.

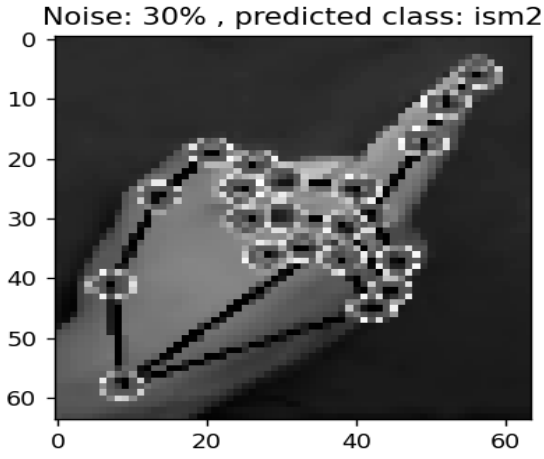

Figure V.15: 'ism-2' sign (30% of noise intensity)

The second level with 50% noise intensity, the model also succeeded to classify the image correctly.

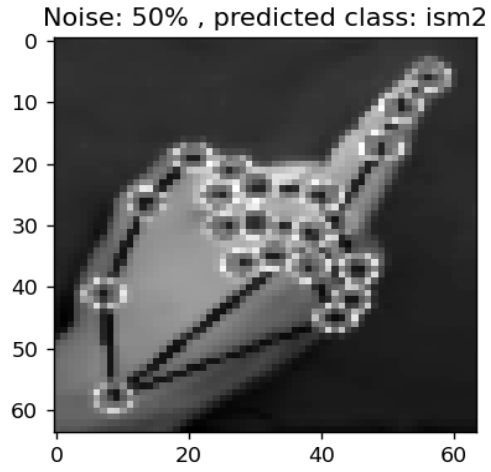

Figure V.16: 'ism-2' sign (50% of noise intensity)

The third level, with 80% intensity

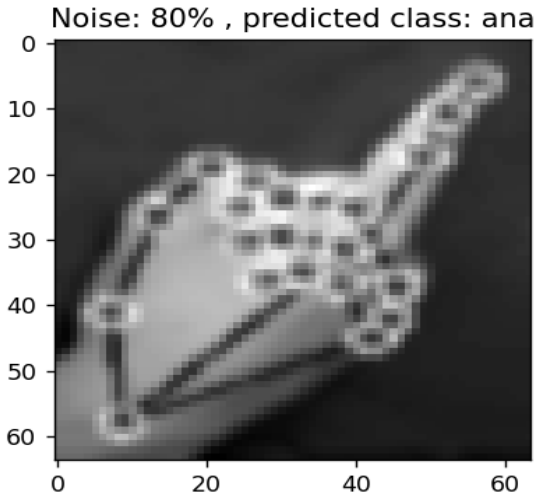

Figure V.17: 'ism-2' sign (80% of noise intensity)

The model mis-classifies the image, after searching for the break point of the model with this sign we found that the model fails to recognize and classify correctly any images that surpass the noise intensity of 73%.

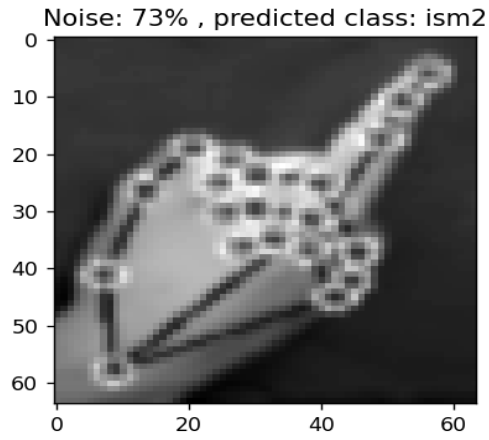

Figure V.18: 'ism-2' sign (73% of noise intensity)

V.4 Results

The performance evaluation of our model using the classification report demonstrates its high accuracy and effectiveness in recognizing sign language gestures. The model achieved an overall accuracy of 99.8%, indicating its amazing ability to classify gestures correctly, on the first step of testing.

After the experiments performed on the model, we come to a conclusion that our model is variant with noise even in extreme conditions, we saw some images that reached maximum intensity of noise which is 100%, but the model succeeded predicted it correctly which confirms the model ability to tolerate significant levels of noise and perform effectively.

V.4.1 The plot Graph

The accuracy plot graph displayed that the training and validation accuracy were extremely close, this means that the model is neither underfitting nor overfitting which signifies that the model is performing robustly across both the training and the unseen validation data, which indicates a well generalized model that achieved a good balance between capturing the complexities of the training data and being able to generalize well and apply those conclusions

on new unseen data. Consequently its very likely that the model will perform it's classification task effectively on new data beyond training and validation data.

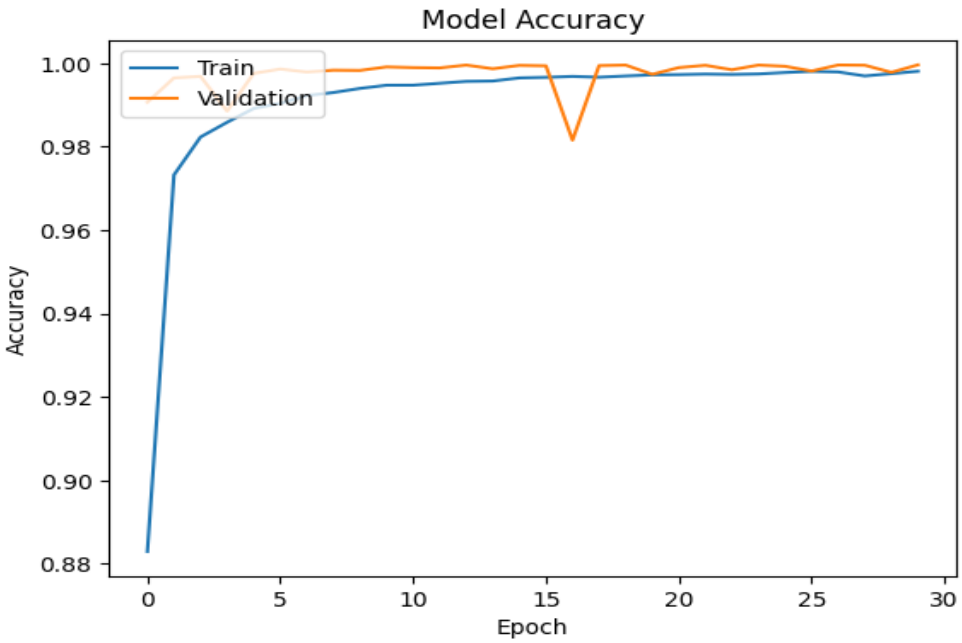

Figure V.19: accuracy plot graph

The loss plot graph shows the same pattern with the training loss and the validation loss are quite close to each other, which means the model is effectively learning from the training data without excessively overfitting or underfitting.

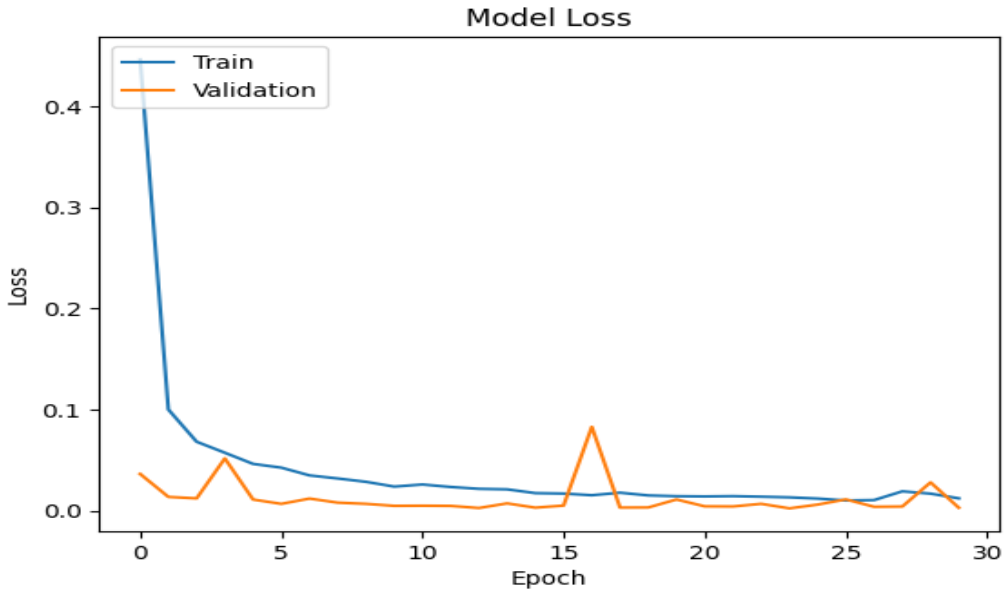

Figure V.20: loss plot graph

V.4.2 Classification Report

Analyzing the precision, recall, and F1-score metrics for each class

- **Precision, Recall, and F1-Score:**

Overall, the model achieves high precision, recall, and F1-scores for most classes, with many of them reaching a perfect score of 1.00. This indicates that the model is able to accurately predict the majority of instances for each class.

The classes "dal," "la," "ha," "ain," "asalmalykom," "al," "khaa," "ta," "dha," "fa," "taa," "meem," "mahoua," "thaa," "gaaf," "ism," "jeem," "anta," "waw," "kayfa al hal," "saad," "haa," "thal," "ana," "ghain," "zay," "ism2," "alhamdoulilah," and "kaaf" all have perfect precision, recall, and F1-scores of 1.00, indicating exceptional performance for these classes.

The classes "yaa," "nun," "dhad," "seen," and "ra" have slightly lower scores, but still achieve high performance overall with precision, recall, and F1-scores above 0.97.

- **Accuracy:**

The overall accuracy of the model reached 1.00 - 100%, indicating that the model accurately predicts the sign language gestures with a high degree of precision, which is not strange to classification tasks if the data is sufficient and pretty easy to differentiate between it's classes.

Class	Precision	Recall	F1-Score	Support
dal	1.0	1.0	1.0	2130
yaa	0.99	0.98	0.99	142
la	1.0	1.0	1.0	3679
ha	1.0	1.0	1.0	1367
ain	1.0	1.0	1.0	1130
asalmalykom	1.0	1.0	1.0	2289
al	1.0	1.0	1.0	172
aleff	0.99	1.0	1.0	165
khaa	1.0	1.0	1.0	1183
ta	1.0	1.0	1.0	2934
dha	1.0	1.0	1.0	3361
nun	1.0	0.96	0.98	132
fa	1.0	1.0	1.0	2345
dhad	1.0	0.97	0.98	191
taa	1.0	1.0	1.0	2969
meem	1.0	1.0	1.0	2781
mahoua	1.0	1.0	1.0	1299
thaa	1.0	1.0	1.0	2842
bb	0.98	0.99	0.98	127
sheen	1.0	1.0	1.0	194
gaaf	1.0	1.0	1.0	1653
ism	1.0	1.0	1.0	1858
jeem	1.0	1.0	1.0	4000
anta	1.0	1.0	1.0	1487
ya	0.98	0.99	0.98	146
waw	1.0	1.0	1.0	1385
kayfa al hal	1.0	1.0	1.0	1238
saad	1.0	1.0	1.0	4177
haa	0.97	0.97	0.97	1301
seen	1.0	1.0	1.0	185
thal	0.99	0.99	0.99	1664
ra	1.0	1.0	1.0	149
ana	1.0	1.0	1.0	1669
ghain	1.0	1.0	1.0	2829
zay	1.0	1.0	1.0	5266
toot	0.98	0.98	0.98	122
ism2	1.0	1.0	1.0	1036
alhamdoulilah	0.98	0.98	0.99	2863

Table V.2: classification report

V.5 Discussion

The comprehensive evaluation of our model's performance showcases its remarkable accuracy and effectiveness in classifying a wide range of sign language gestures. These results underscore the potential of our model to facilitate effective communication for individuals using sign language in various contexts and applications.

While our model performed exceptionally well, there were some challenges and limitations encountered during building this model.

The first challenge we encountered was the lack of a dataset for arabic words which forced us to create our own, then we couldn't perform the pre-processing approach that we desired on the alphabets dataset, so we handpicked the images that were successfully pre-processed which cut down the size of that dataset over its half.

Another significant challenge was the impact of lighting conditions on the recognition results. Despite taking the precautions to enhance performance under poor lighting conditions, the model's accuracy was occasionally affected due to the low-quality of the images under bad lighting. In these conditions, the recognition may be compromised, leading to potential mis-classifications of the gestures.

Another challenge arises from certain signs that share similarities, especially when combined with poor lighting conditions. The model may struggle to differentiate between these similar signs, resulting in potential inaccurate classifications.

V.6 Conclusion

in this chapter we talked about the metrics for evaluating our model and their results then we discussed them. The evaluation process is very important to better understand the model's behavior and pinpoint it's weaknesses and strong points.

Chapter VI

General Conclusion and Future Work

VI.1 General Conclusion

Arabic sign language is as diverse and rich as the regular spoken arabic language and due to that diversity, an absence of a standard version of arabic sign language is implied.

Our study explored the task of arabic sign language recognition using Deep Learning techniques. We aimed to develop a high performing model that classifies and translates arabic sign language gestures into clear regular arabic words and phrases. CNN proved its effectiveness in those types of tasks as we succeeded to build an efficient model with a 99.98\%~ accuracy which is remarkable.

Our approach was based on using pre-possessing methods that highlights the most significant areas of the hands and removing all the unbalances that can mislead the model into apprehending and relying on unimportant features, mainly we used hand tracking technique to shift the attention only to the hands and remove unbalances which contains varying useless informations. The other one is drawing a shape that simulates the skeleton of the hand which leverages the shape of the gesture regardless of the shape of the hand(tall, short, skinny or fat hands) also the library that provides this skeleton shape proved to be consistent in low-quality lighting, which helps the models in those types of scenarios to stay efficient. The dataset we used was a mix of arabic language alphabets (ArASL_Database_54K_Final) and a selection of common used words which we created after using (Community Development Authority in Dubai) tutorials on how perform certain arabic sign language Emirate's version, that dataset was fairly sufficient after preforming several data-augmentation variations. We utilized a Convolutional Neural Network architecture that emphasizes learning hierarchical features from images with multiple convolution and pooling layers, which performed magnificently providing satisfying results.

We used multiple standard methods for evaluating classification models on our model, they all came back great and of course we performed live execution and it was also positive.

Those types of models will help to facilitate the communication for deaf-mute and hard hearing individuals with ones whom has no knowledge with sign language in different realms of social life and vice versa, providing a tool that will magnify the voice of individual in this community can benefit the whole society in multiple forms, such as enhancing the workforce and

refreshing it by introducing new ideas with new perspectives, also it will strengthen familial relationships by connecting the deaf-mute persons with none sign language speakers in the family, and also with outside of these communities in the different parts of the social life.

Quite obviously we encountered many challenges which limited the potential of the project, one was the lack of available dataset for images that represent the performance of arabic words in arabic sign language, contrary the dataset for alphabets was significantly easier to obtain because its standardized across all variations and sub-versions of arabic sign language throughout the arabic nations. We had to improvise and form our own dataset using a webcam, which couldn't provide enough images to sufficiently train a model that contains 40 classes. The second problem we faced was due to the shortness of time, we couldn't build a Natural Language Processing model NLP, which would've complimented the model by taking it's classification outputs and combine them to create a correct understandable phrase, which will ease engaging in full conversations with sign gestures.

VI.2 Future Work

Reaching those great results motivated us to start aiming high and start pushing the potential areas that we can improve on to deliver a full product that can do our society and the Arabic nation a favor.

First, We are eager to develop a natural language processing model NLP which will combine results of the classification into forming a fully understandable and complete phrases, this will promote the model into translating full conversations performed with sign gestures.

second of all, we will expand the list of words this model can cover, which will enhance the ability to use this model and engage in any desired topic .

We will study the possibility to add another language, American sign language ASL, this will widen the reach of this model and bring together different parts of the globe and remove not only the language barrier but also the disability and help them grow together.

We will develop a natural language processing model NLP which will combine results of the classification into forming a fully understandable and complete phrases, this will promote the model into translating full conversations performed with sign gestures.

add sign to voice algorithm which will convert the word or sentences interpreted by the model into Arabic sound. this will enable communication with people with vision problems and blind individuals.

all of those improvements will require great dedication to fulfill them efficiently, and lots of enhancements on the current structure.

References

[1] "WHO: 1 in 4 people projected to have hearing problems by 2050." https://www.who.int/news/item/02-03-2021-who-1-in-4-people-projected-to-have-hearing-problems-by-2050 (accessed Jun. 13, 2023).

[2] "Algeria," *African Sign Languages Resource Center*. https://africansignlanguagesresourcecenter.com/algeria/ (accessed Jun. 13, 2023).

[3] "Reading the Signs: Diverse Arabic Sign Languages," Aug. 2014, Accessed: Jun. 13, 2023. [Online]. Available: https://www.csis.org/analysis/reading-signs-diverse-arabic-sign-languages

[4] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," *Intell. Syst. Appl.*, vol. 12, p. 200056, Nov. 2021, doi: 10.1016/j.iswa.2021.200056.

[5] A. Walelign, "Ethiopian Sign Language Recognition Based on Hand Gesture and Facial Expression Using Convolutional Neural Network," Thesis, 2020. Accessed: Jun. 13, 2023. [Online]. Available: http://ir.bdu.edu.et//handle/123456789/12725

[6] "ASL in Sign Language (Video + important explanations)," *Lingvano ASL*, Oct. 06, 2020. https://www.lingvano.com/asl/blog/asl-in-sign-language/ (accessed Jun. 12, 2023).

[7] I. Papastratis, K. Dimitropoulos, and P. Daras, "Continuous Sign Language Recognition through a Context-Aware Generative Adversarial Network," *Sensors*, vol. 21, no. 7, Art. no. 7, Jan. 2021, doi: 10.3390/s21072437.

[8] A. R. N. Aouichaoui, R. Al, J. Abildskov, and G. Sin, “Comparison of Group-Contribution and Machine Learning-based Property Prediction Models with Uncertainty Quantification,” in *Computer Aided Chemical Engineering*, M. Türkay and R. Gani, Eds., in 31 European Symposium on Computer Aided Process Engineering, vol. 50. Elsevier, 2021, pp. 755–760. doi: 10.1016/B978-0-323-88506-5.50118-2.

[9] A. Halnaut, R. Giot, R. Bourqui, and D. Auber, “Chapter 3 - Compact visualization of DNN classification performances for interpretation and improvement,” in *Explainable Deep Learning AI*, J. Benois-Pineau, R. Bourqui, D. Petkovic, and G. Quénot, Eds., Academic Press, 2023, pp. 35–54. doi: 10.1016/B978-0-32-396098-4.00009-0.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Jun. 13, 2023. [Online]. Available: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[12] “Multilabel Image Classification Using Deep Learning - MATLAB & Simulink.” https://www.mathworks.com/help/deeplearning/ug/multilabel-image-classification-using-deep-learning.html (accessed Jun. 13, 2023).

[13] V. Pusarla, “Object Detection using Regions with CNN features,” *Analytics Vidhya*, Mar. 02, 2021. https://medium.com/analytics-vidhya/object-detection-using-regions-with-cnn-features-557392e22f84 (accessed Jun. 13, 2023).

[14] H. Bhardwaj, P. Tomar, A. Sakalle, and U. Sharma, “Chapter 20 - Principles and Foundations of Artificial Intelligence and Internet of Things Technology,” in *Artificial*

Intelligence to Solve Pervasive Internet of Things Issues, G. Kaur, P. Tomar, and M. Tanque, Eds., Academic Press, 2021, pp. 377–392. doi: 10.1016/B978-0-12-818576-6.00020-4.

[15] "What is a Neural Network?," *TIBCO Software*. https://www.tibco.com/reference-center/what-is-a-neural-network (accessed Jun. 12, 2023).

[16] P. Abolmaesumi *et al.*, "Contributors," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, S. K. Zhou, D. Rueckert, and G. Fichtinger, Eds., in The Elsevier and MICCAI Society Book Series. Academic Press, 2020, pp. xvii–xxvi. doi: 10.1016/B978-0-12-816176-0.00005-3.

[17] V. E. Balas *et al.*, "List of contributors," in *Artificial Intelligence for Future Generation Robotics*, R. N. Shaw, A. Ghosh, V. E. Balas, and M. Bianchini, Eds., Elsevier, 2021, pp. xi–xii. doi: 10.1016/B978-0-323-85498-6.00018-6.

[18] S. Albelwi and A. Mahmood, "A Framework for Designing the Architectures of Deep Convolutional Neural Networks," *Entropy*, vol. 19, no. 6, Art. no. 6, Jun. 2017, doi: 10.3390/e19060242.

[19] A. H. Reynolds, "Anh H. Reynolds," *Anh H. Reynolds*. https://anhreynolds.com/ (accessed Jun. 12, 2023).

[20] "File:MaxpoolSample2.png - Computer Science Wiki." https://computersciencewiki.org/index.php/File:MaxpoolSample2.png (accessed Jun. 12, 2023).

[21] "4. Fully Connected Deep Networks - TensorFlow for Deep Learning [Book]." https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html (accessed Jun. 12, 2023).

[22] R. E. Rwelli, O. R. Shahin, and A. I. Taloba, “Gesture based Arabic Sign Language Recognition for Impaired People based on Convolution Neural Network.” arXiv, Mar. 10, 2022. doi: 10.48550/arXiv.2203.05602.

[23] I. Vazquez Lopez, “Hand Gesture Recognition for Sign Language Transcription,” *Boise State Univ. Theses Diss.*, May 2017, doi: https://doi.org/10.18122/B2B136.

[24] M. M. Saleem, “Deep Learning Application On American Sign Language Database For Video-Based Gesture Recognition,” *Stud. Theses,* Nov. 2020, [Online]. Available: https://source.sheridancollege.ca/fast_sw_mobile_computing_theses/2

[25] J. Blair, “Architectures for Real-Time Automatic Sign Language Recognition on Resource-Constrained Device,” *UNF Grad. Theses Diss.*, Jan. 2018, [Online]. Available: https://digitalcommons.unf.edu/etd/851

[26] M. M. Kamruzzaman, “Arabic Sign Language Recognition and Generating Arabic Speech Using Convolutional Neural Network,” *Wirel. Commun. Mob. Comput.*, vol. 2020, p. e3685614, May 2020, doi: 10.1155/2020/3685614.

[27] S. Meena, “A Study on Hand Gesture Recognition Technique,” MTech, 2011. Accessed: Jun. 12, 2023. [Online]. Available: http://ethesis.nitrkl.ac.in/2887/

[28] H. G. A. A. Almahri, “Arabic Sign Language Recognition: A Deep Learning Approach,” May 2022, Accessed: Jun. 12, 2023. [Online]. Available: https://bspace.buid.ac.ae/handle/1234/2043

[29] A. Sultan, W. Makram, M. Kayed, and A. A. Ali, “Sign language identification and recognition: A comparative study,” *Open Comput. Sci.*, vol. 12, no. 1, pp. 191–210, Jan. 2022, doi: 10.1515/comp-2022-0240.

[30] "هيئة تنمية المجتمع." https://www.cda.gov.ae/ar/pages/default.aspx (accessed Jun. 13, 2023).

[31] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, "ArASL: Arabic Alphabets Sign Language Dataset," *Data Brief*, vol. 23, p. 103777, Apr. 2019, doi: 10.1016/j.dib.2019.103777.

[32] Yoshua Bengio, Geoffrey Hinton, and Ilya Sutskever ," On the Importance of Input Representations in Neural Language Models" 2011.

[33] Digital Image course by Dr.Aiadi Ossama.