Connect

Waddah Hassan,Sultan ALharbi,Ahmed Saleh

November 2022

Contents

List of Figures

1 Chapter 1

1.1 Introduction

Deaf and hearing-impaired persons utilize sign language as their primary language in daily life. There aren't enough interpreters available to help deaf individuals to communicate with others. But these interpreters are neither practical nor available in all situations. Sign Language may differ from country to country, geographical region to geographic region, and even from one deaf person's interests to another's interests. Sign language is formed from a combination of descriptive and non-descriptive signals, as well as alphabet signs (sometimes referred to as fingerspelling). However, Arabic sign language (ArSL) does not adhere to a standard structure that can be used to characterize the language. As a direct consequence of this, educating communities of deaf people who live in Arabic countries is a huge obstacle for them to overcome. The requirement for a bilingual education that places an emphasis on reading and writing, in addition to the dearth of teachers who are fluent in Arabic Sign Language, contributes significantly to the severity of this problem.In this chapter, we will walk you through various different aspects before discussing the fundamentals behind our project. Each will provide a few fundamental principles, and we will illustrate those ideas using examples from the relevant aspect of the project. As a consequence of this, we will provide a description of the issue, a suggestion for a solution that may be used to remedy the problem, an outline for a report, and a project plan that outlines the project's objectives and the deliverance's that it will produce.

1.2 General Context

By utilizing Computer Vision and Deep Learning algorithms to recognize hand gestures and translating it to Arabic language using Natural Language Processing techniques, this project hopes to break down the communication barrier between deaf and mute people and the general media audience. This will be accomplished by creating an understandable Arabic sentence based on Arabic sign language, which will then be communicated to the general media audience.

1.3 Problem Definition

We have a lot of talented people in our society, and they are getting the support and admiration they need; yet, how many of us have heard or seen on the news or in any other sort of media about deaf persons who are talented? Notwithstanding this, it is not accurate to say that they do not exist. Due to the expense and limited availability of hiring a Sign-Language interpreter, the majority of mainstream media would prefer to make an interview with a normal guy rather than a talented but deaf guy. This will lead those people of this group of our society to be unmotivated and have the feeling that they are unappreciated within our society. The problem with this is that it is a problem that most of the mainstream media has.

1.4 Aims and Deliverables

During the course of this project, we plan to create an interpreter for Arabic sign language that works in real time and makes use of a camera that can read a person's motions and convert them into Arabic. This interpreter will be developed using a camera that can read a person's motions and convert them into Arabic. Our deliverables will include a pre-trained deep neural network that has been put to use for the purpose of solving straightforward categorization issues. After that, we will include transfer learning strategies into the network in order to give it the ability to comprehend Arabic Sign Language. In the end, we will utilize natural language processing (NLP) to convert Arabic Sign Language into spoken Arabic in order to produce a phrase in the form of text that can be understood by the target audience.

Aims:

1. Translating Arabic Sign-Language in real time.
2. Break the communication barrier between deaf people and mainstream media.
3. Reducing the cost of hiring a real person to interpret sign-language.
4. Translating sentences rather than single words.

Deliverables:

1. Pre-trained neural network modified using Transfer-Learning technique.
2. Detecting hands using computer vision.
3. Gestures recognition using computer vision.
4. Sentence level signs translating.
5. GitHub repository for the work we will do.

1.5 Proposed Solution

By utilizing a camera and a computer, this project intends to bring awareness to those individuals and make it simpler for mainstream media to translate sign language in real time without the need to employ a person to serve as an interpreter. This will be accomplished by highlighting the contributions of those individuals.

1.6 Report Outline

The report contained three different chapters in total. Each consists of a number of pieces that have been designed to present them in an understandable manner. In the first chapter, there is a presentation that is more general in nature regarding the project. The second chapter will focus on the works that are connected to the project that have been finished and will discuss their significance. As a direct consequence of this, a comprehensive description of all the important topics and the existing systems will be produced. In the third chapter, we will start collecting the data and preprocessing it. In addition, we will discuss the benefits and cons of the already available systems, as well as the requirements for our project, both in terms of its functionality and its non-functional aspects.

1.7 Project Plan

The following is going to be the structure of the project plan that we will be working on: The project must be finished within the allotted time of eight weeks. During the first week, we will get everything ready, including the dataset, the camera, and the libraries that we will use in the project. After that, we will begin the process of locating an appropriate pre-trained deep learning model, performing transfer learning on it, and tweaking the hyper parameters. This process could take up to two weeks to determine the optimal setting combination. After that, we will use techniques from natural language processing to transform the neural network into text that can be understood by humans. This process will take a total of one month. During the final three weeks of the project, we will examine and make certain that there are no mistakes, such as over-fitting, and that the model is operating without any issues.

Planning phase	**13 days**	**Thu 9/8/22**	**Tue 9/20/22**		
the problem to solve	5 days	Thu 9/8/22	Mon 9/12/22	6	Waddah,Sultan,Dr. Usman,Ahme
information gathering	4 days	Tue 9/13/22	Fri 9/16/22	8	Waddah,Sultan,Ahmed
<meeting>	1 day				
Aims of the project	1 day	Sat 9/17/22	Sat 9/17/22	9	Waddah
project objectives	1 day	Sun 9/18/22	Sun 9/18/22	9	Sultan,Ahmed
feasibilty study	3 days	Mon 9/19/22	Wed 9/21/22	9,12	Waddah,Sultan
project plan	1 day	Thu 9/22/22	Thu 9/22/22	13	Waddah,Sultan,Ahmed
problem understanding	**12 days**	**Fri 9/23/22**	**Tue 10/4/22**		
Dfineing stakeholders	1 day	Fri 9/23/22	Fri 9/23/22	9	Waddah,Sultan
<meeting>	1 day				
background of the project domain	3 days	Sat 9/24/22	Mon 9/26/22	14	Waddah,Ahmed
lituraure review	5 days	Tue 9/27/22	Sat 10/1/22	13,14,18,16	Waddah
Comparison results with feasibilty study	3 days	Sun 10/2/22	Tue 10/4/22	13,18	Waddah
Analysis phase	**8 days**	**Wed 10/5/22**	**Wed 10/12/22**		
<meeting>	1 day				
functional & non-functional requirments	4 days	Wed 10/5/22	Sat 10/8/22	14,18	Waddah,Sultan
hardware requirement	4 days	Sun 10/9/22	Wed 10/12/22	13	Ahmed
Design phase	**16 days**	**Wed 10/12/22**	**Thu 10/27/22**		
Writing ch1,2,3	10 days	Fri 10/14/22	Sun 10/23/22	14,18	Waddah,Sultan,Ahmed
Review	2 days	Mon 10/24/22	Tue 10/25/22	18	Dr. Usman
<meeting>	1 day				
Writing ch4	3 days	Tue 8/23/22	Thu 8/25/22	27	Sultan,Waddah
Finel design of the papper	3 days	Fri 8/26/22	Sun 8/28/22	27,29	Ahmed
Editing	3 days	Tue 10/25/22	Thu 10/27/22	29	Sultan,Waddah
Implementation	**60 days**	**Mon 8/22/22**	**Thu 10/20/22**		
Coding the program	60 days	Mon 8/22/22	Thu 10/20/22		Ahmed,Sultan,Waddah
Testing	**14 days**	**Tue 10/11/22**	**Mon 10/24/22**	**32**	
Evaluate the requirment	7 days	Tue 10/11/22	Mon 10/17/22	33	
Test desighn	7 days	Tue 10/18/22	Mon 10/24/22	35	
Deployment	**3 days**	**Tue 10/25/22**	**Thu 10/27/22**		**Sultan**
Performance	3 days	Tue 10/25/22	Thu 10/27/22	36	
Update	2 days	Tue 10/25/22	Wed 10/26/22	35	
Version tracking	1 day	Thu 10/27/22	Thu 10/27/22	39	
Maintenance	**14 days**	**Fri 10/14/22**	**Thu 10/27/22**		**Ahmed,Sultan,Waddah**

Figure 1: This how we split the work.

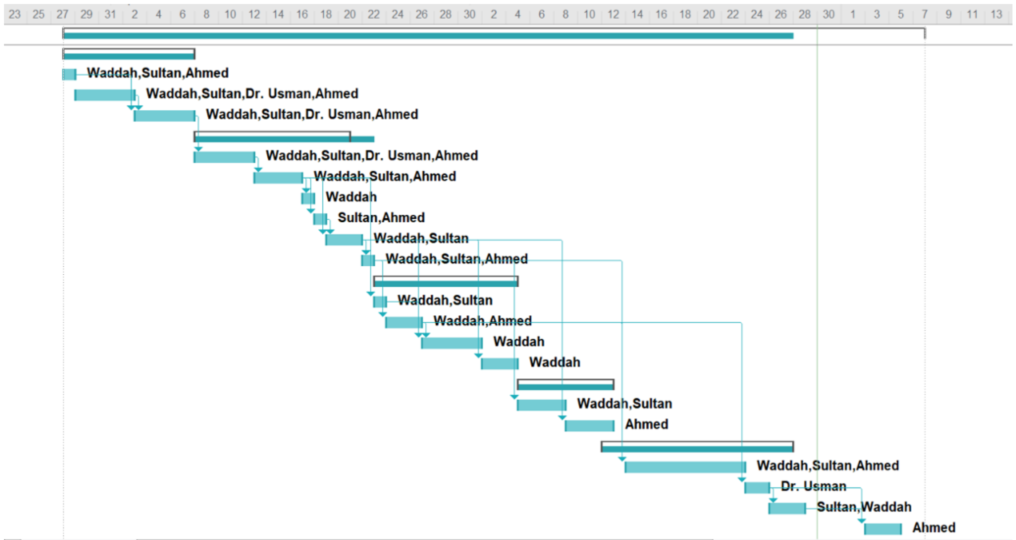


Figure 2: This is the time split.

2 Chapter 2

2.1 Introduction

This chapter will act as an introduction to the subsequent discussion of works that are linked to our paradigm. In addition, this chapter is broken up into three distinct sections, each of which is designed to explain essential ideas and make it easier to understand the overall content of the chapter. In the first section, "Motivations," we will provide a comprehensive description and illustration of the project's domains. In contrast, in the second section, "Demonstrating How the Project Domain Must Be Related to Critical Activities," we will demonstrate how the project domain must be related to critical activities in order to effectively and appropriately deliver project outcomes.This section is based on knowledge of similar models and monitoring the systems that are the subject of the study, in addition to published studies. In the second section, titled "Proposed systems," we delve deeper and provide an estimate of the existing systems that have been employed up to this point. In addition to this, it will include an adequate description of each system in addition to necessary facts, including an analysis of the benefits and drawbacks of each system. Because of this, we will be able to collect all of the data that is required in order to make a contribution to this study that is one of a kind.In the final portion, a comparison of the technical features and the details will be made, with an emphasis placed on any disparities, deviations, or causes. This will make it possible to provide a summary of references and ongoing projects along the rows of the table, and a comparison of features of the tools and technologies required for the project along the columns of the table.

2.2 Motivations

The field of artificial intelligence is expanding into every imaginable sector in order to assist human beings. In recent times, it has been put to use in order to provide assistance to the deaf community by way of improving the sign language detecting method. Sign languages are utilized extensively by deaf people all over the world as a means of communication for the purpose of information exchange with hearing people. As a result, the primary contribution that the technique for detecting sign language has made is to serve as a digital interpreter between those who are deaf and those who have hearing. There are methods of detection that involve the use of supplemental hardware such as specialized gloves, Kinect, etc.; nevertheless, the inputs are not 2D images, and the system is considerably more sophisticated than it initially appears. The majority of the other methods take an image (or sequence of images) containing signs as their point of entry, and the last step is to identify those signs and present them in a form that is relevant to the user. In later approaches, image processing techniques are applied to the input image, which locates the position of the gesture so that it may be detected. These latter systems do not rely on any external devices to function properly. Previous works were primarily based on antiquated image pre-processing approaches such as color-based foreground segmentation, morphological procedures, and so on. In current day and age of machine learning, convolutional neural networks give us with more sophisticated tools for object detection, which have surpassed the methodologies that were used in the past. In the context of our activity, we investigate and advance the field of sign language detection.

2.3 Background

In this section, we will take a quick look at each domain and outline how each domain will be involved in the process of constructing this project.

2.3.1 Computer Vision

The study of computer vision is a sub-field of computer science that focuses on the development of improved methods for gleaning information from still images and moving video. As a result of this, computer vision is able to assist us in recognizing hand motions and isolating the hand from things in the background, both of which are going to be primary focuses of our project.

2.3.2 Video Processing

A lot of focus is currently being put on developing better video processing technology so that accurate information may be extracted from videos without any information being lost in the process. As was demonstrated in the flow chart that came before it, the input video is cut up into individual frames. A portion of the input video is seen in Figure. The following paragraph provides an example of this procedure.

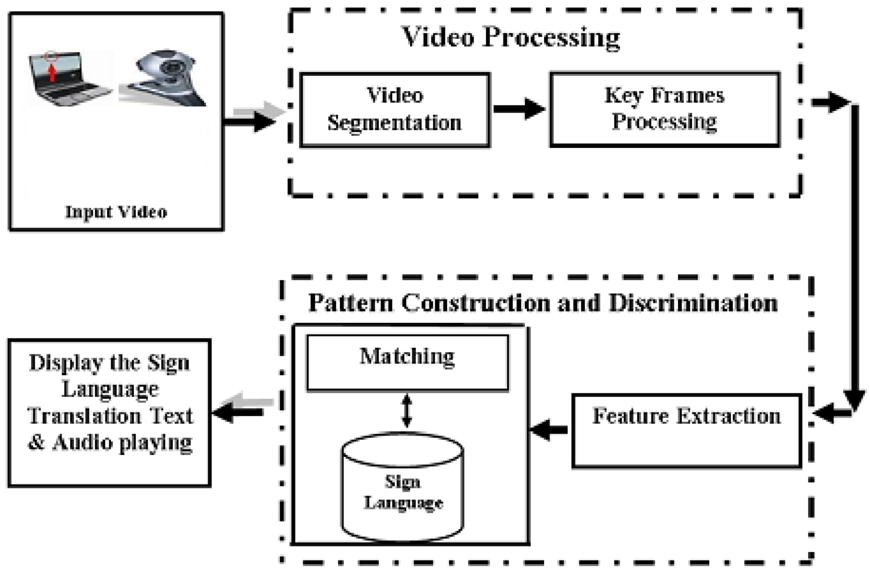


Figure 3: Proposed System Frame Work.

2.3.3 Deep learning

We will use transfer learning alongside CNNs in our project in order to make up for the lack of Arabic sign language data. We will use a pre-trained neural network that was trained on detecting simple objects and modify it to understand sign language. Deep learning is a technique that helps the computer act like a human by using raw data. In our project, we will use deep learning.

2.3.4 Natural Language Processing

It is the ability to make the computer understand human language, and we will use it to translate sign language into a correct word and combine them together to create a sentence that is understandable. Natural language processing is the technique that uses the computational power to process the human natural language. In simpler words, it is the ability to make the computer understand human language.

2.3.5 Convert RGB Image to Gray Scale

Converting an image, or a video frame, from RGB to a grayscale image is one of the most significant steps in the preprocessing of images.A conversion has resulted in a grayscale version of the image. Using methods such as digital image processing and computer vision, we are able to perform some operations to enhance the image, segment it, and perform morphology on it to remove noise. Following these steps, you are able to perform the stage of detecting important areas (the shape of a hand) in the image. In order to prepare for image processing and the extraction of important objects from the image, the image of the real color sign is converted into a gray image. This is accomplished by converting the three-dimensional image color matrix into the gray image matrix, with one dimension indicating the color density in the gray matrix saturation, hue, and ignoring luminance by preserving the color brightness with special weights according to (R, G, B) of the color compounds in the real image, where the color compound is the component of the image that contains the The research made use of gray images because the majority of image processing techniques, such as filtering, deal with gray images, and it is simpler to express image information as a single vehicle that indicates the intensity of the color in the image while maintaining the clarity of the image details. In addition, the image cutting techniques used in the research deal with gray images. These are the reasons why the gray images were used.

2.3.6 Data Preprocessing

In order to construct a deep learning model that actually works, the first step is to preprocess the data. It is utilized to transform the raw data into a format that is both usable and effective. The flow diagram for the preprocessing of data is shown in Figure 1. Due to the numerous changes that occur, real-time data are inherently unreliable and difficult to anticipate (rotating, moving, and so on). Image augmentation is a technique that can boost the performance of deep neural networks. It does this by employing a variety of processing techniques, such as shifting, flipping, shearing, and rotating, to produce images that are completely fabricated. With the help of this picture enhancement strategy, the images produced by the suggested system are given a random rotation between 0 and 360 degrees. A small number of the photos were additionally sheared at random within a range of 0.2 degrees, and a small number of the images were flipped horizontally.

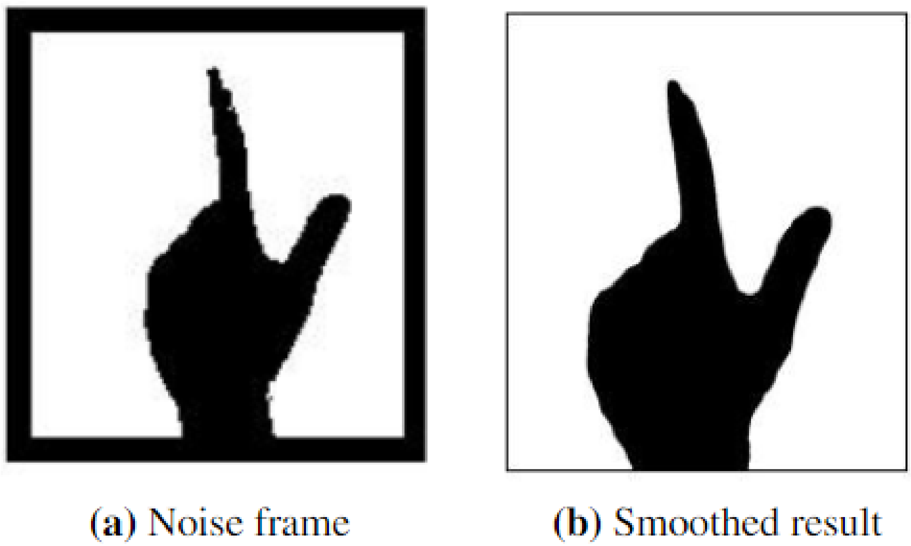


Figure 4: Pre-processing results.

2.3.7 Hand Detection

The procedure known as preprocessing, which is also known as the data preparation process, is the second stage of our system. The datasets have been made more robust by taking distinct photographs of additional signers with varying manual sizes and complex backdrops. Additionally, the images comprise round samples and various angles ranging from 60 to -60 degrees to make the system more resilient. The processing of color images helps to improve image quality. The processing includes converting the color image to a grayscale image with 256 levels of density and resizing the image. It is possible to get rid of unwanted noise using various filtering strategies. This stage consists of several methods of optimization and image enhancement, segmentation, and morphological filtering, and its goal is to convert the data into a format that can be processed more easily and effectively. In order to accomplish this goal, the preprocessing stage must first convert the data into a format. This stage is dependent on the earlier steps of hand detection, which involve applying various methods of optimization and image enhancement, segmentation, and morphological filtering to the color image in a variety of different ways in order to obtain the best sample that will subsequently assist us in extracting the best features and achieving the highest level of accuracy.

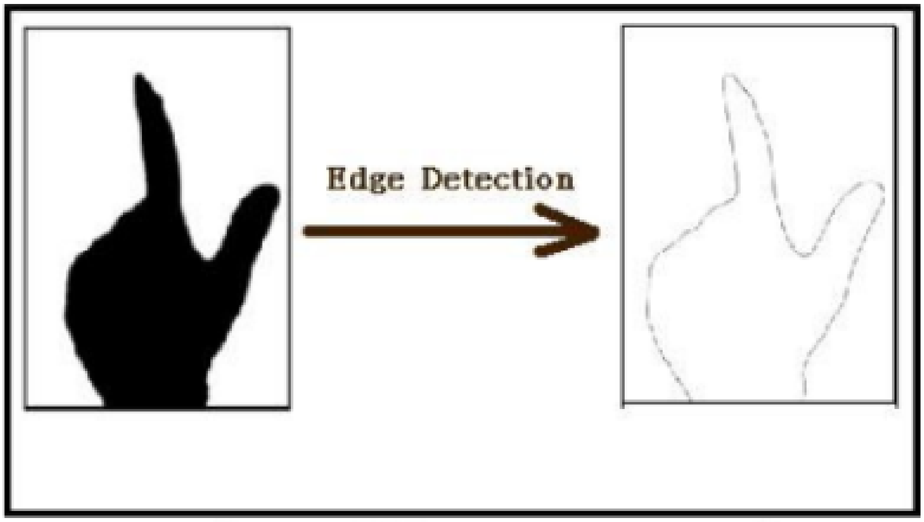


Figure 5: Edge-detection stage.

2.3.8 Features Extraction.

In the previous phase, the hand was characterized and extracted using one of the methods that had been used previously; in this phase, we determine the best features in the gesture of a sign character and distinguish it from other gestures, which are then used in the training or testing process for the dataset. In the previous phase, the hand was characterized and extracted using one of the methods that had been used previously. After it has been determined that a sign is present, the next phase, which is called the feature extraction phase, is required because specific features have to be removed in order for them to be distinct for each gesture or sign.

2.4 Existing Systems

A number of different systems in the field of sign language recognition make use of very similar concepts in order to understand sign language through the application of computer vision and deep learning. In this part of the article, our goal is to provide an acceptable explanation of each system, as well as sufficient specifics, including an analysis of its benefits and drawbacks.

2.4.1 Existing Systems one

Article:Sign Language Recognition Using Convolutional Neural Networks Work done by: Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen.

Brief:*The project focus on the recognition of the signs or gestures. There are two main steps in building an automated recognition system for human actions in spatio-temporal data. The first step is to extract features from the frame sequences. This will result in a representation consisting of one or more feature vectors, also called descriptors. This representation will aid the computer to distinguish between the possible classes of actions. The second step is the classification of the action. A classifier will use these representations to discriminate between the different actions (or signs). In our work, the feature extraction is automated by using convolutional neural networks (CNNs). An artificial neural network (ANN) is used for classification.*

Advantages:

1. Transfer learning for modelling the BSL shows promising results and should be investigated further.

Disadvantages:

1. Their pre-processed BSL corpus lacks constructs that are essential for a sign language, such as classifier signs and others.

2.4.2 Existing Systems two

Article: Transfer Learning for British Sign Language Modelling Work done by: Boris Mocialov, Graham Turner, Helen Hastie

Brief:*This paper focuses on language modelling, a common technique in the field of ASR and Natural Language Processing to model the likelihood of certain words following each other in a sequence. They tend to improve modelling of the BSL glosses by proposing to use transfer learning approaches, such as finetuning and layer substitution. The use of transfer learning technique can overcome the data sparsity issue in statistical modelling for scarce resource languages by using similar resources that can be found in large quantities and then further training the models on a specific low resource data.*

Advantages:

1. The ability to use it on a limited computational power computer.

Disadvantages:

1. Unfortunately, we couldn't find any notable cons.

2.4.3 Existing Systems three

Article: Learning to Estimate 3D Hand Pose from Single RGB Images Work done by: Christian Zimmermann, Thomas Brox University of Freiburg

Brief:*Low-cost consumer depth cameras and deep learning have enabled reasonable 3D hand pose estimation from single depth images. In this paper, we present an approach that estimates 3D hand pose from regular RGB images. This task has far more ambiguities due to the missing depth information. To* *this end, we propose a deep network that learns a network-implicit 3D articulation prior. Together with detected keypoints in the images, this network yields good estimates of the 3D pose. We introduce a large scale 3D hand pose dataset based on synthetic hand models for training the involved networks. Experiments on a variety of test sets, including one on sign language recognition, demonstrate the feasibility of 3D hand pose estimation on single color images.*

Advantages:

1. The Local approach incorporates the kinematic model of the hand and uses the network to estimate articulation parameters of the model.

Disadvantages:

1. the used dataset was built based on 3D hand figures , not real hand
2. he performance seems mostly limited by the lack of an annotated large scale dataset with real world images and diverse pose statistics.

2.4.4 Existing Systems four

Article: gesture recognition in the wild with iterative visual attention

Work done by: Bowen Shi1 , Aurora Martinez Del Rio2 , Jonathan Keane2 , Diane Brentari2 Greg Shakhnarovich1 , Karen Livescu1 1Toyota Technological Institute at Chicago, USA 2University of Chicago, USA

Brief: *Sign language recognition is a challenging gesture sequence recognition problem, characterized by quick and highly co-articulated motion. In this paper we focus on recognition of gesture sequences in American Sign Language (ASL) videos collected in the wild, mainly from YouTube and Deaf social media. Most previous work on sign language recognition has focused on controlled settings where the data is recorded in a studio environment and the number of signers is limited. Our work aims to address the challenges of real-life data, reducing the need for detection or segmentation modules commonly used in this domain. We propose an end-to-end model based on an iterative attention mechanism, without explicit hand detection or segmentation. Our approach dynamically focuses on increasingly high-resolution regions of interest. It outperforms prior work by a large margin. We also introduce a newly collected data set of crowd sourced annotations of gesture in the wild, and show that performance can be further improved with this additional data set.*

Advantages:

1. they developed a new model for ASL fingerspelling recognition in the wild, using an iterative attention mechanism

Disadvantages:

1. the approach does not rely on any hand detection, segmentation, or pose estimation modules.

2.4.5 Comparison Table

In this section, we will talk about the differences and similarities between our project and the related systems. The first paper that we listed in the "Related works" section will be referred to as system 1, and subsequent articles will be numbered sequentially going forward. Our evaluation will focus on the ability to show sentences, as well as identify and recognize words, evaluate data quality, and make use of applications.

	display sentence	Detect & recognize	Low Resolution Data	Use application
System 1	No	Yes	No	Yes
System 2	No	No	Yes	No
System 3	No	Yes	Yes	No
System 4	No	Yes	Yes	Yes
Our System	Yes	Yes	No	No

Figure 6: Table of Comparison.

3 Chapter 3

3.1 Introduction

Gathering requirements, having functional requirements, and having non-functional needs are going to be the topics that we cover in this chapter. During the process of gathering requirements, we will be looking at current platforms that are comparable to ours, their limitations and problems, and how to avoid them while also making our platform more effective and easier to use in the real world. After that, we will review our objectives and decide whether or not we will change them or add a new one based on our search of related systems. After that, we will go through the functional requirements that the platform needs in order to meet our objectives. These functional requirements will be grouped into three categories: basic-level, mid-level, and high-level requirements. After that, we will go through the functional requirements that the platform needs in order to meet our objectives. In the end, we will talk about the non-functional needs, which should be requirements that are reasonable and fit for our platform.

3.2 Requirements Gathering

Translation at the sentence level: much like spoken languages, sign languages are distinct languages with their own linguistic structures. This is true even when sign languages are similar to spoken languages. A system that detects individual signals one at a time is only capable of providing a translation into the specific sign language that is being delivered, regardless of the surrounding circumstances. When a signed sentence is turned into a question, for example, the entire translation as well as the words that were used could end up being completely different as a result of a sign that appears later in the sentence. In addition, if the signer is required to stop after each signature and wait for the result, the overall user experience may be negatively impacted.

Accessibility of the platform Despite the fact that our platform makes use of technologies such as Deep Learning and Computer Vision, this platform must be available on a regular computational power in order to assist a greater number of customers in interpreting Arabic Sign Language.

Programming environment: We made the decision to construct this platform using Python due to the fact that it is the most popular programming language when it comes to deep learning and computer vision. Additionally, Python is known for its open-source libraries and its large community.

3.3 Functional Requirements

Basic-level	Mid-level	High-level
Detecting human hands.	Translating Hand gestures into text.	Gathering text to create a whole sentence (Sentence Level Signs).
Display the translated sentence.	Features Extraction.	Model deployment
Model building	Convert RGB Image to Gray Scale	
Camera	Video Processing	

Figure 7: Functional Requirements Table.

3.4 Non-functional Requirements

Non-functional Requirements	
Run on 32 bit (x86) or 64 bit (x64).	**Easy to use for non-tech users.**
Getting high accuracy result.	Avoiding over-fitting in the data.
Development environment	Data integrity
Privacy	Memory Optimization

Figure 8: non-Functional Requirements Table.

3.5 Conclusion

Unfortunately, we did not observe a true application of deep learning for the purpose of translating Arabic Sign Language, despite the fact that this is a promising and emerging field and that it represents the current state of the art in the interpretation of sign languages. We are aware that the road ahead of us with this project will not be an easy one; however, we have faith that if we are able to overcome all of the challenges that lie ahead, our project will be able to achieve widespread success and propagate itself across the entirety of the nation, allowing us to demonstrate the more positive aspects of our culture.

4 Chapter 4

4.1 Introduction

The topic of sprint 1 development will be the focus of the discussion that takes place in Chapter 4, which features its own structure of six distinct sections. These sections are the Use-case Diagram, in which we will create a use-case diagram to represent all of the functional requirements; the Sprint 1 Static Aspect Design, in which we must create an E-R diagram to design all of the functional requirements; the Sprint 1 Architectural Aspect Design, in which we must create a sequence diagram to represent the various hardware components of the functional requirements; and the Sprint 1 Dynamic Aspect Design, in which we must create a flowchart that represents how the functional requirements will be implemented.

4.2 Use Case

A Use Case is a representation of the functionality of the system that is used during the process of requirement elicitation and analysis. This representation is utilized in the context of a Use Case. A use case is a description of a function that is carried out by the system and results in observable output for an actor. This output can be measured or otherwise evaluated. The boundary of the system is defined as a result of the identification of actors and use cases, which differentiates the tasks that are completed by the system from the tasks that are completed by its environment. The system's environment is responsible for completing the tasks that are not completed by the system. To accomplish this, first identify the activities that are carried out by the system and then identify those that are carried out by its environment. Actors are positioned on the outermost edge of the system's perimeter, whereas use cases are located on the interior of the system. A document that describes the operation of a system as seen from the point of view of an actor is called a use case. The actor's point of view illustrates this behavior beautifully. This provides an explanation of the role that the system performs as a chain of events that lead to a conclusion that is clear for the player. To What End Are Use Case Diagrams Put to Use, and What Are Their Benefits? It is possible to record the dynamic nature of a system using a diagram called a use case diagram. However, due to the fact that it is overly general, this definition cannot be used to provide an acceptable description of the aim. A number of other diagrams, including the activity diagram, the sequence diagram, the cooperation diagram, and the status chart, can also be used to describe the objective. In this section, we are going to look into a specific objective that will differentiate this graphic from the other four that were presented earlier.

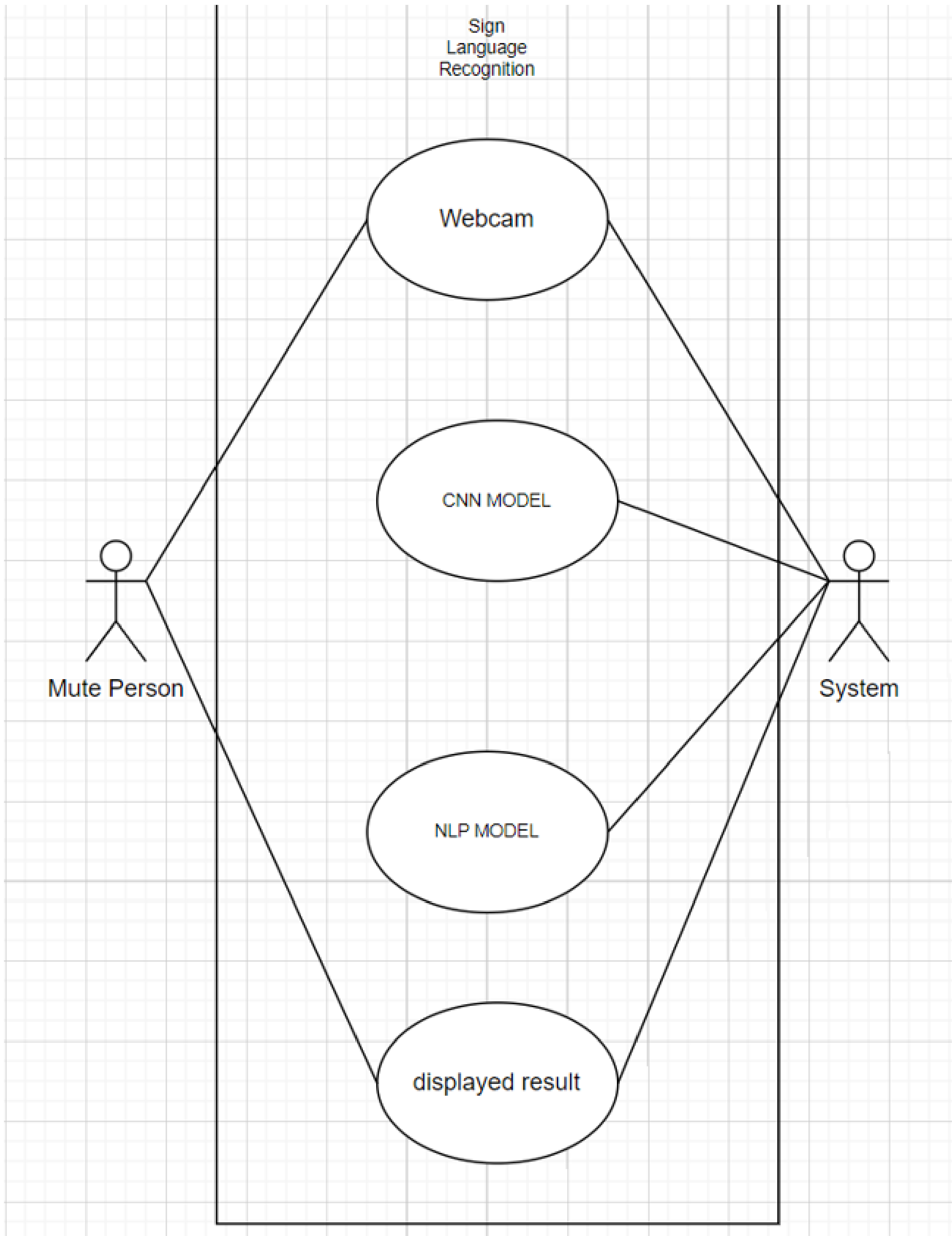


Figure 9: This Is System Use Case.

4.3 Sprint 1 Static Aspect Design

Class diagrams are used to illustrate the structure and content of classes through the utilization of design components such as classes, packages, and objects. Class diagrams are used to represent the various design viewpoints, such as conceptual, specification, and implementation, which are utilized. The name of the class, the students' qualities, and the assignments they complete make up the three aspects that make up a class. In addition, class diagrams illustrate several types of relationships, including containment, inheritance, association, and others. The association relationship is the type of relationship that appears the most frequently in class diagrams. The association exemplifies the relationship that exists between the class instances.

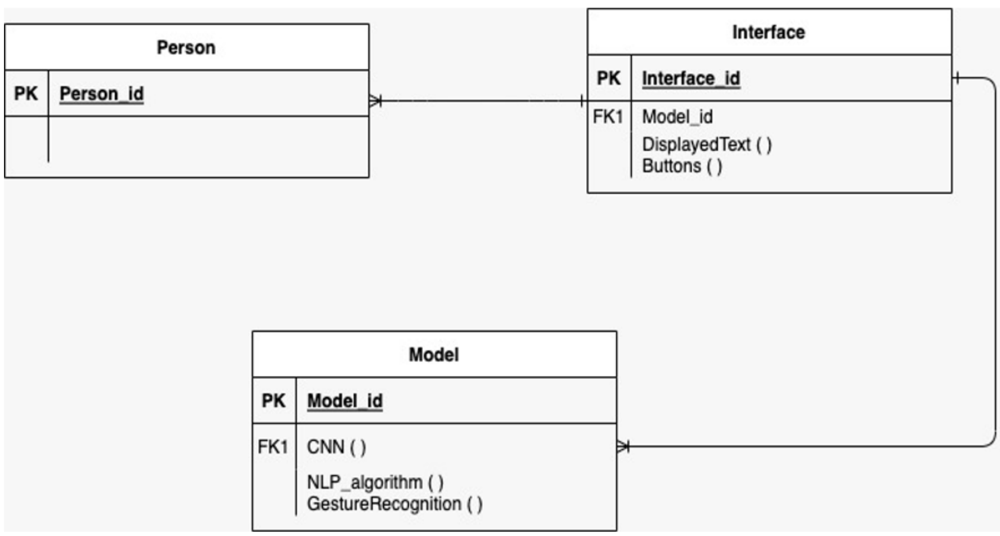


Figure 10: This Is System Use Case.

4.4 Sprint 1 Architectural Aspect Design

In addition to the necessary hardware for the configuration, we have compiled a table that describes each component and the reasons why we require it.

Functional Requirements hardware
Camera to capture hand movement
High end RAM 16gb
High end GPU for CNN
Monitor to display the sentence

Figure 11: This Is Hardware that is required.

4.5 Sprint 1 Dynamic Aspect Design

A sequence diagram is a type of diagram that displays the chronological order in which the various events that occurred during the interaction between the objects took place. This takes into consideration both the vertical dimension (time) as well as the horizontal dimension (different objects). Objects: An entity is said to be an object if it can be located at a particular time, has a value that can be quantified, and can also retain an identity. One way to conceptualize an object is as an entity that is present at a particular point in time and with a particular worth. In a sequence diagram, the item interactions are depicted according to the chronological order in which they took place. It illustrates the classes and objects that are engaged in the scenario, in addition to the order in which messages are sent between the various components in order to carry out the functionality of the scenario. In the Logical View of the system that is currently being created, sequence diagrams and use case realizations are typically coupled to one another as part of the system's overall architecture. Sequence diagrams go under a few other names, including event diagrams and event scenarios, among others. Multiple processes or items that exist at the same time are depicted in a sequence diagram as parallel vertical lines (lifelines), and the messages that are sent between them are depicted as horizontal arrows. This is carried out according to the sequence in which the activities or events take place. As a consequence of this, it is now feasible to graphically express uncomplicated run time scenarios.

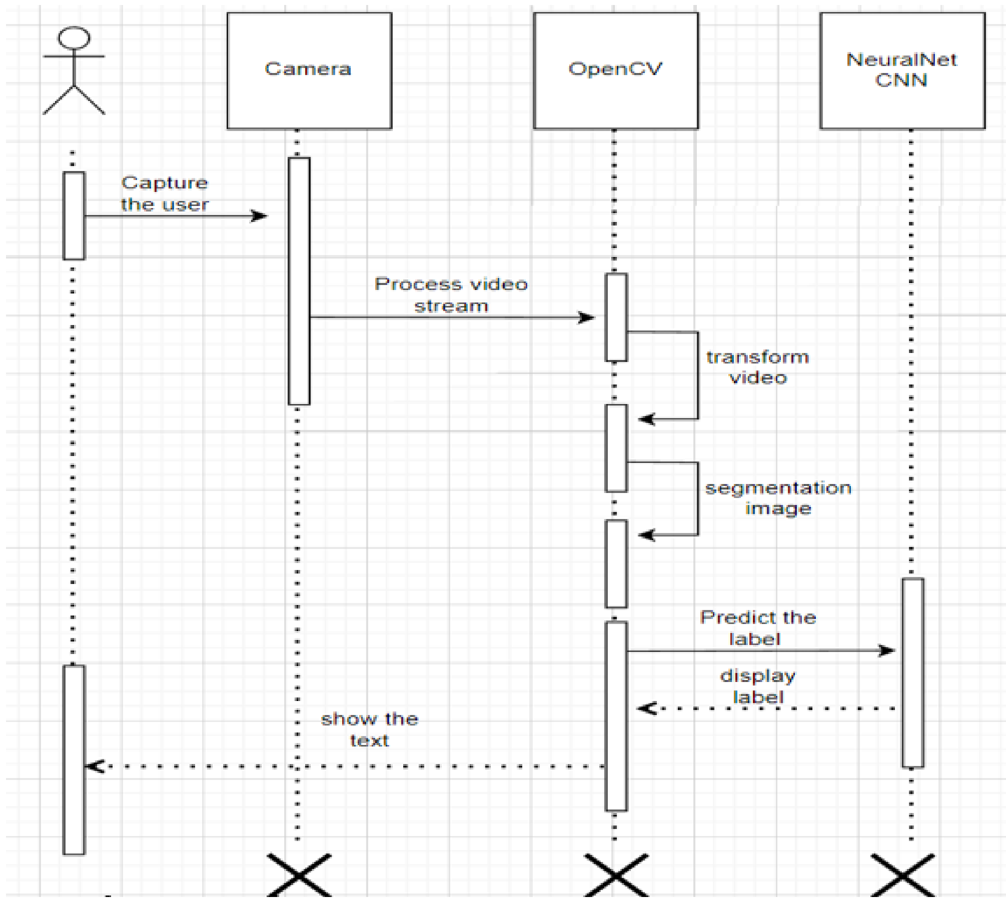


Figure 12: Sequence diagram.

4.6 Sprint 1 Implementation

A data flow diagram can also be referred to as a bubble chart in some circles. It is a straightforward graphical formalism that can be used to depict a system in terms of the data that is input into the system, the various processing operations that are performed on the data, and the data that the system generates as output. In other words, it shows how a system works from the perspective of the data. You can use it to illustrate a system in terms of the data that is fed into the system, the various processing activities that are carried out on the data, and the data that is produced as a result of those processing processes. It provides a visual representation of the flow of information for every particular entity.

A process or system is often what one refers to when referring to the manner in which data is processed in terms of its inputs and outputs. It displays the data inputs, outputs, storage places, and paths between each destination using predefined symbols such rectangles, circles, and arrows. These symbols are used to depict the data flow. They are helpful not only for modeling new systems but also for performing analysis on ones that already exist. The ability of a DFD to "communicate" things graphically that are difficult to explain in words makes it valuable for applications in both the technical and non-technical realms. Because of this, it can be utilized in a wide variety of settings. The DFD can be broken down into the following four sections:

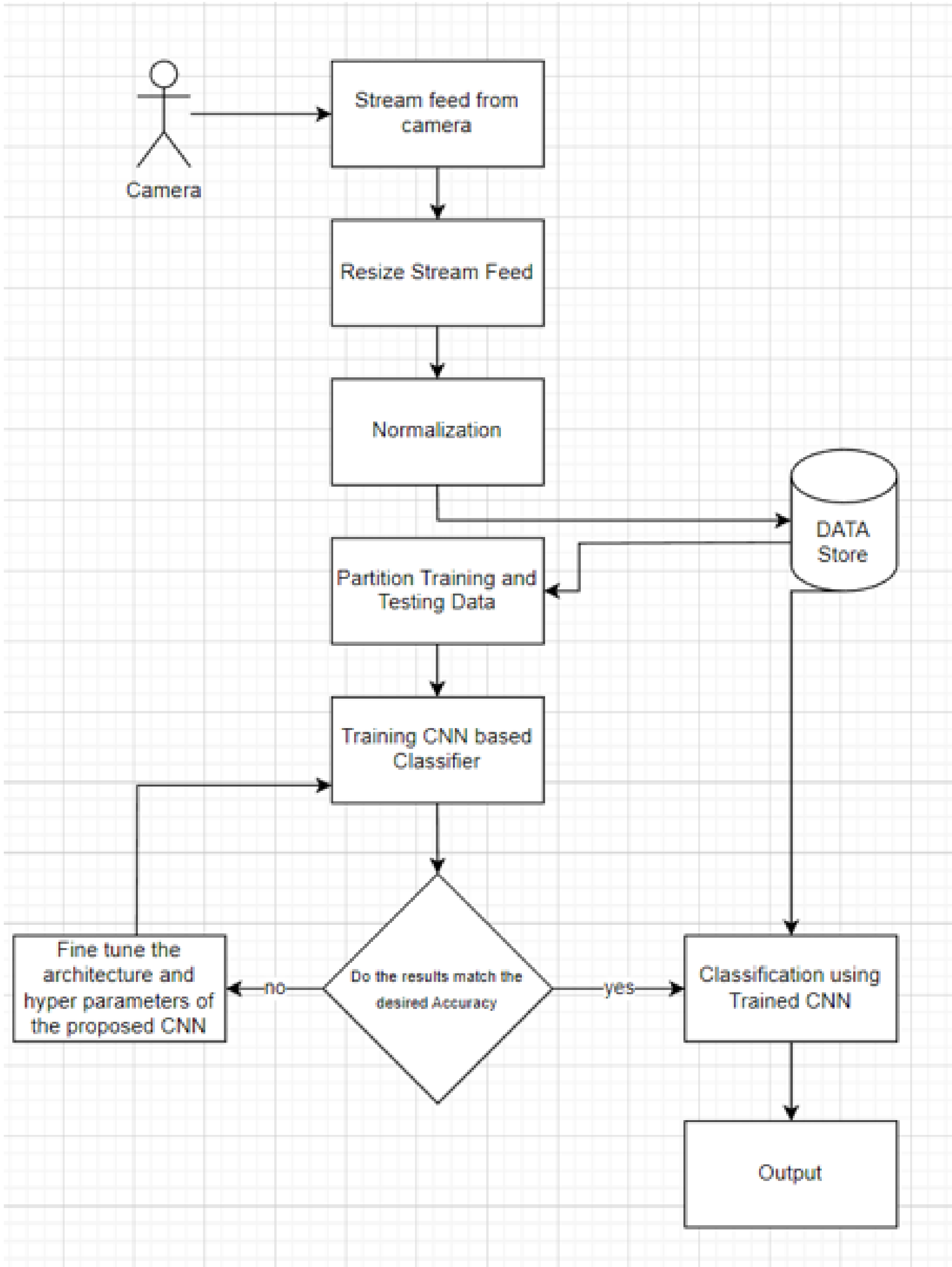


Figure 13: : Hardware and software for implementation.

4.7 Sprint 1 Testing Evaluation

The great majority of research publications focus on precise recognition of sign language content, and the principal metrics that these studies use seek to measure this skill. As a result of this, some studies make use of standard accuracy indicators that are based on percentages, such as precision and recall, as well as their combination, which is referred to as the F1 score.

Test Case	Testing Type	Test Scenario	Test Steps	Expected Result
failure	Functionality Testing	recognizing	1)After setting up the user and the camera 2)we try to detect user then extract key frame 3)extract features and try to classify gesture 4)failed to recognition or display the text	Won't be able to display any text or will miss some text
success	Functionality Testing	recognizing	1)After setting up the user and the camera 2)we try to detect user then extract key frame 3)extract features and try to classify gesture 4)succeeded to recognition and display the text	It will be able to display a full sentence in real time

Figure 14: : Testing Evaluation table.

4.8 Conclusion

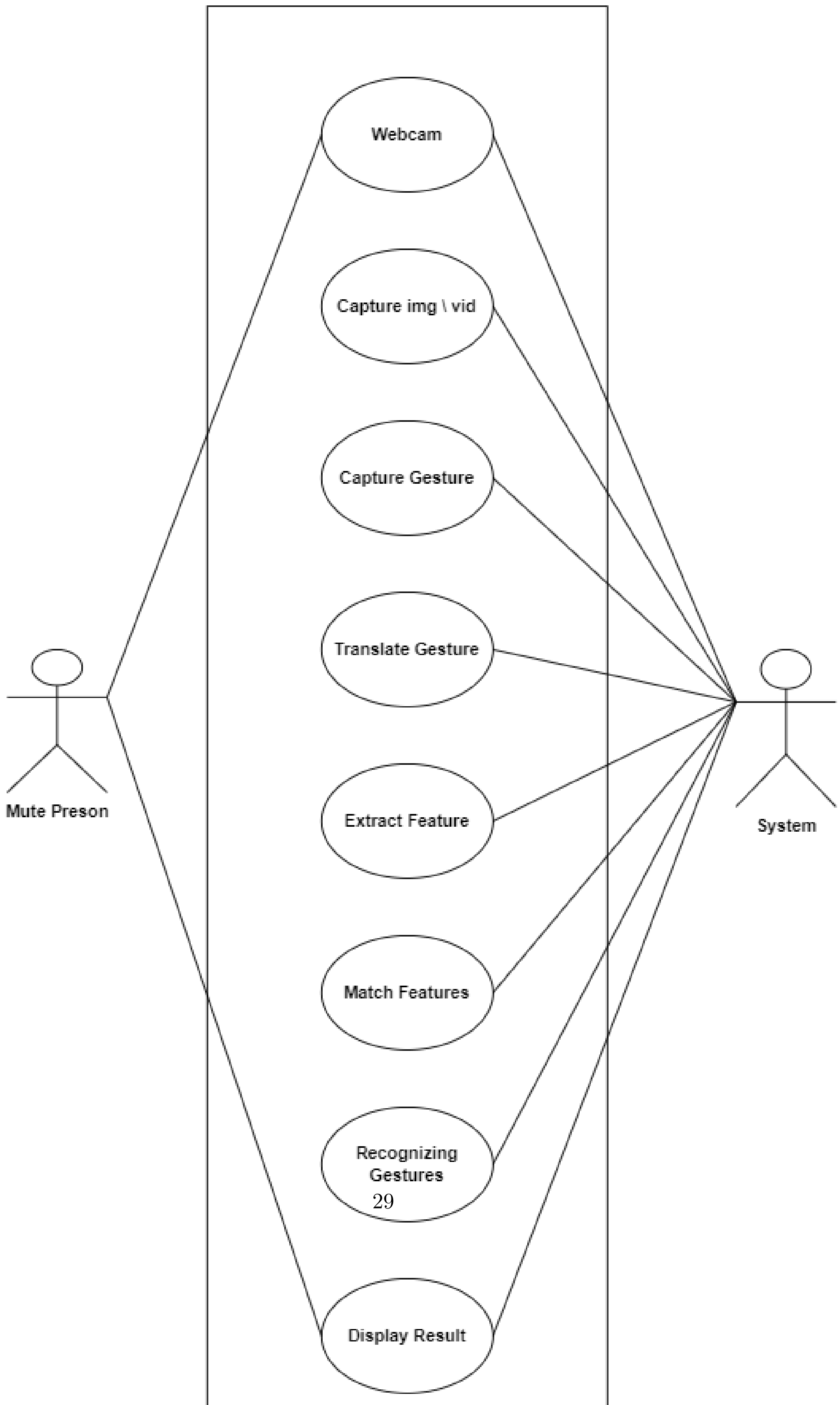
The project diagrams were generated, and this chapter discusses the construction of the diagrams as well as the coverage they provide. Because of this, it is now lot easier for us to track the process from the use case diagram all the way up to the sprint. This process includes the Static Aspect Design to create a detailed class diagram for our functional requirements, the Architectural Aspect Design to determine the hardware required for this project, the Dynamic Aspect Design, Implementation, Testing and Evaluation, where we estimate scenarios to test and evaluate the project, and the Architectural Aspect Design to determine the hardware required for this project. In addition, this procedure includes the Architectural Aspect Design.

5 Chapter 5

5.1 Introduction

The discussion in Chapter 5, which has its own structure of six unique sections, will be focused on the subject of sprint 2 development. The Sprint 2 Static Aspect Design section requires us to create an E-R diagram to design all of the functional requirements. The Sprint 2 Architectural Aspect Design section requires us to create a sequence diagram to represent the various hardware components of the functional requirements. The Sprint 2 Dynamic Aspect Design section requires us to create a use-case diagram to represent all of the functional requirements.

5.2 Use-case diagram



5.3 Sprint 2 Static Aspect Design

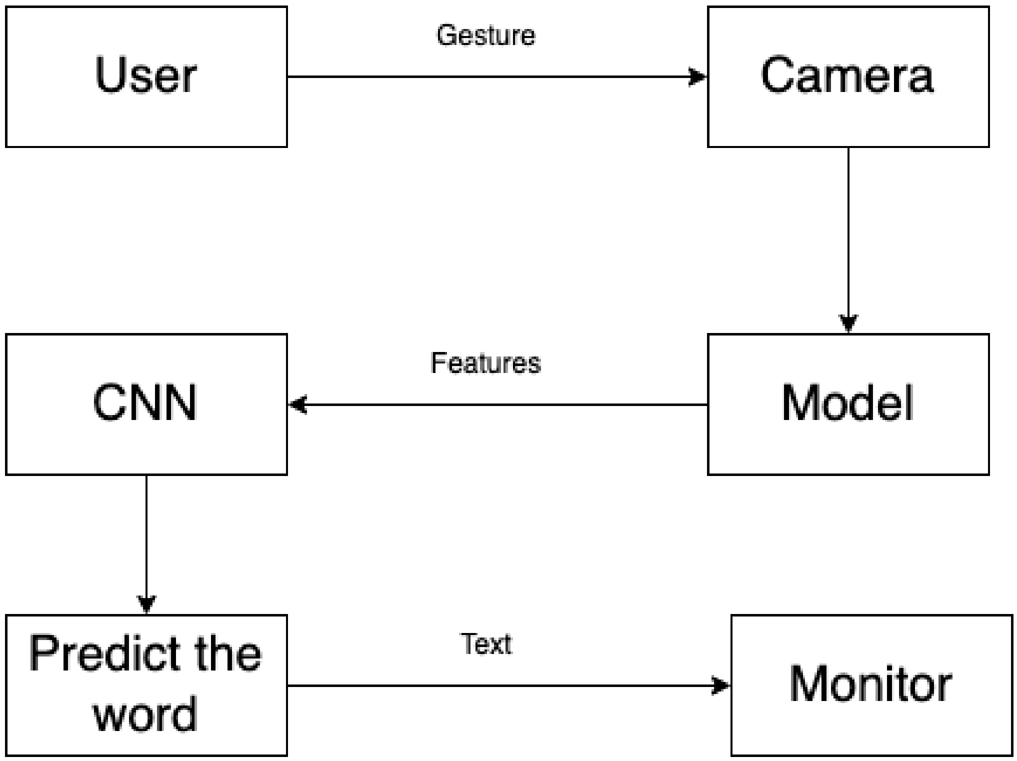


Figure 16: : This Is System Use Case Mid Level.

5.4 Sprint 2 Architectural Aspect Design

Functional Requirements hardware
Camera to capture hand movement
High end RAM 16gb
High end GPU for CNN
Monitor to display the sentence

Figure 17: This Is Hardware that is required. mid level

5.5 Sprint 2 Dynamic Aspect Design

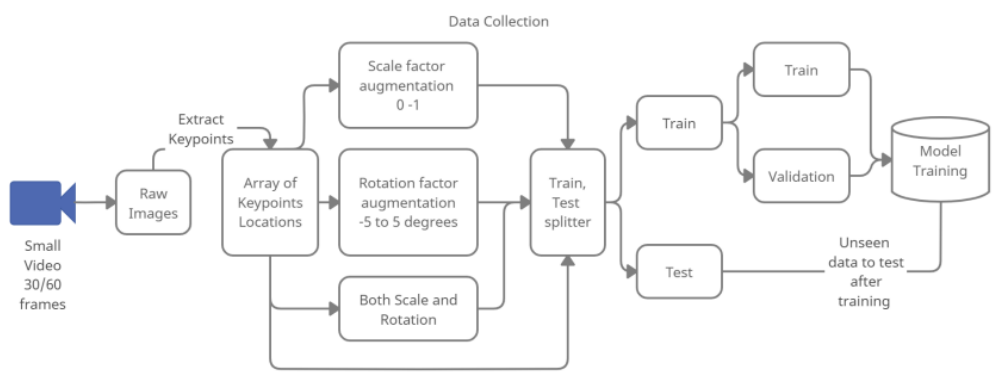


Figure 18: :Dynamic Aspect Design.

5.6 Sprint 2 Implementation

Figure 19: : Implementation Software.

5.7 Sprint 2 Testing Evaluation

Test case	Testing Type	Test scenario	Test steps	Expected Result
Failure	Unit testing	Recognition	-Setting up the camera. -Checking if the module can detect the hand even with not enough light in the room. -Extracting features from gesture. -Checking if the text output is the same as expected.	It would not be able to determine the gesture probably.
Success	Integrating testing	Recognition	-Setting up the camera. -Checking if the module can interpret gestures. -Extracting text from gesture. -The module can display text extracted from signs.	It will be able to display a word per gesture.

Figure 20: : Sprint 2 Testing Evaluation.

5.8 Sprint 2 Testing Evaluation

5.9 Conclusion

The diagrams for the project were created, and this chapter covers how they were made as well as the coverage they offer. As a result, tracking the process from the use case diagram all the way up to the sprint is much simpler for us now. This process entails the Static Aspect Design to produce a thorough class diagram for our functional requirements, the Architectural Aspect Design to establish the hardware needed for this project, the Dynamic Aspect Design, Implementation, Testing and Evaluation, where we estimate test and evaluation scenarios, and the Architectural Aspect Design to establish the hardware needed for this project. This process also includes designing the architectural aspect.

6 Chapter 6

6.1 Introduction

The discussion in Chapter 6 , has its own structure like Chapter 5 but more ADVANCED in the six unique sections, will be focused on the subject of sprint 2 development. The Sprint 2 Static Aspect Design section requires us to create an E-R diagram to design all of the functional requirements. The Sprint 2 Architectural Aspect Design section requires us to create a sequence diagram to represent the various hardware components of the functional requirements. The Sprint 2 Dynamic Aspect Design section requires us to create a use-case diagram to represent all of the functional requirements

6.2 Use-case diagram

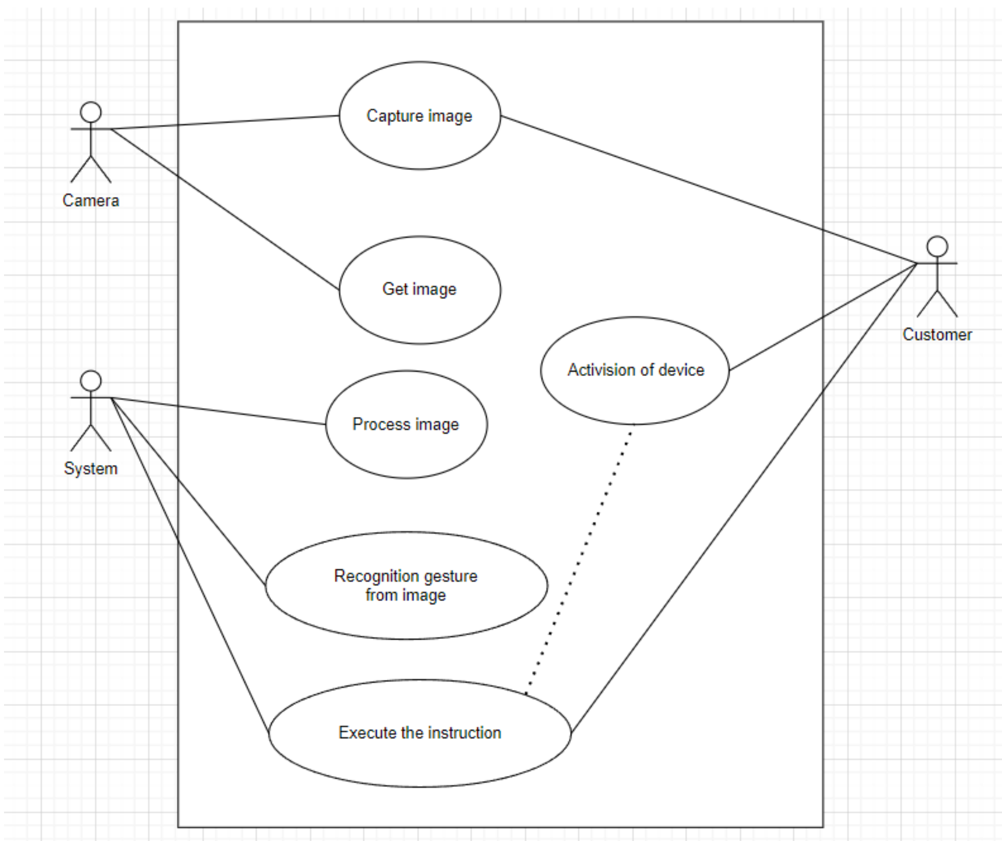


Figure 21: : This Is System Use Case in High level.

6.3 Sprint 2 Static Aspect Design

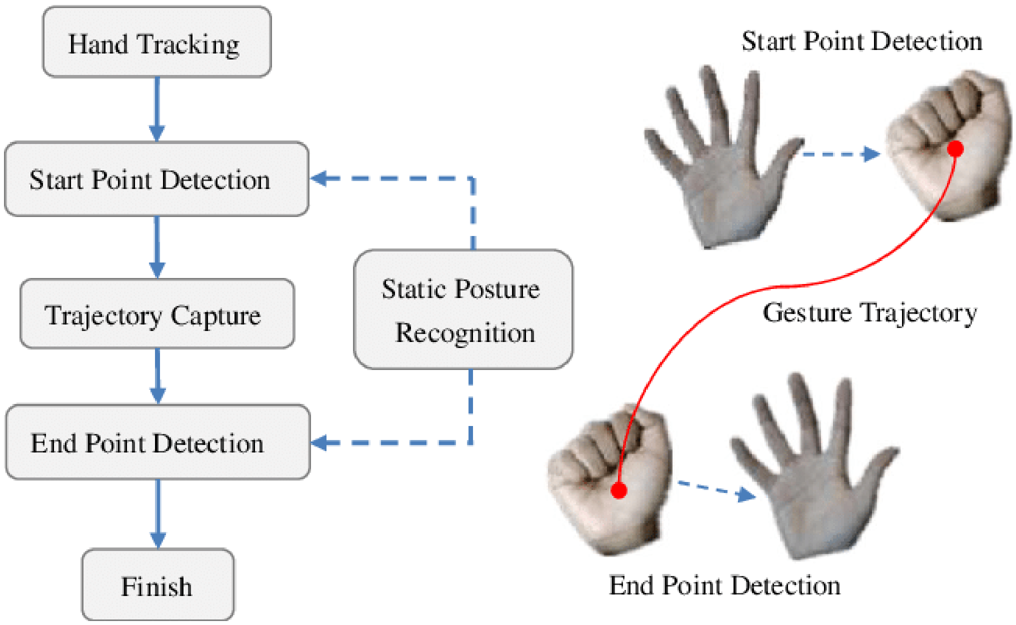


Figure 22: : This Is System Aspect Design high Level.

6.4 Sprint 2 Architectural Aspect Design

Functional Requirements hardware
Camera to capture hand movement
High end RAM 16gb
High end GPU for CNN
Monitor to display the sentence

Figure 23: This Is Hardware that is required. hign level

6.5 Sprint 2 Dynamic Aspect Design

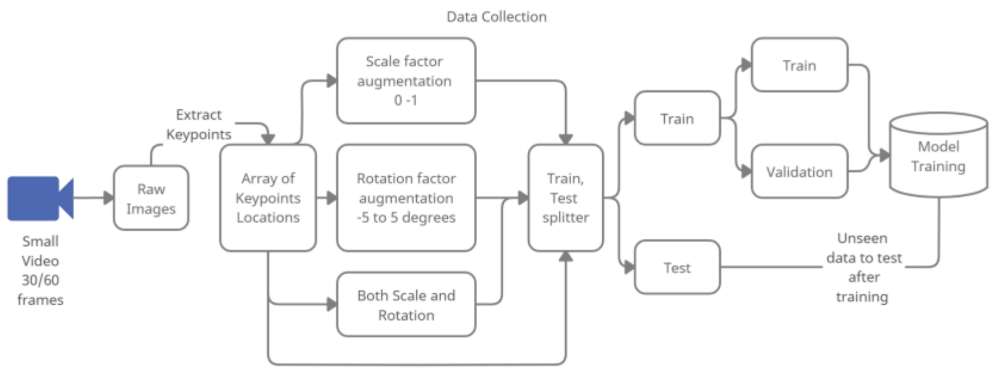


Figure 24: :Dynamic Aspect Design.

6.6 Sprint 2 Implementation

Figure 25: : Implementation Software.

6.7 Sprint 2 Testing Evaluation

Test case	Testing Type	Test scenario	Test steps	Expected Result
Failure	Unit testing	Recognition	-Setting up the Camera -Checking if the module can detect the hand even with not enough light in the room. -Extracting features from gesture. -Checking if the text output is the same as expected.	It would not be able to determine the gesture probably.
Success	Integrating testing	Recognition	-Setting up the camera -Checking if the module can interpret gestures. -Extracting text from gesture. -The module can display text extracted from signs.	It will be able to display a word per gesture.
Success	System testing	Recognition	-Setting up the camera -Extracting features from gesture. -Checking if the module can interpret gestures. -Activating of device -Checking if the text output is the same as expected. -Execute the instruction	The system will operate, translate a sign language, and translate the language as it is in the data

6.8 Conclusion

In conclusion, the development of a standardized structure for Arabic Sign Language (ArSL) and the creation of a bilingual education program are critical steps towards addressing the challenges faced by deaf and hearing-impaired individuals who use sign language in Arabic-speaking countries. By leveraging technology and working with experts in the field, we can create digital platforms and resources that make learning and using ArSL more accessible and user-friendly. By partnering with local organizations and institutions, we can ensure that our efforts are aligned with the needs and priorities of the communities we aim to serve. Through research and evaluation, we can measure the impact of our work and identify opportunities for further development and improvement. By promoting the use and recognition of ArSL, we can help ensure that all individuals, regardless of their hearing ability, have the opportunity to communicate and thrive.

References

[1] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan,A. Thangali, H. Wang, and Q. Yuan. Large lexicon project:American Sign Language video corpus and sign languageindexing/retrieval algorithms. In Workshop on the Represen-tation and Processing of Sign Languages: Corpora and SignLanguage Technologies (CSLT), 2010.

[2] S. C. W. Ong and S. Ranganath, —Automatic sign language analysis: A survey and the future beyond lexical meaning, IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 873– 891, Jun. 2005.

[3] L. Ding and A. M. Martinez, —Modelling and recognition of the linguistic components in American sign language, Image Vis. Comput., vol. 27, no. 12, pp. 1826– 1844, Nov. 2009.

[4] N. Purva, K. Vaishali, Indian Sign language Recognition: A Review, IEEE proceedings on International Conference on Electronics and Communication Systems, pp. 452-456, 2014.

[5] G. Fang, W. Gao, and D. Zhao, —Large vocabulary sign language recognition based on fuzzy decision trees, IEEE Trans. Syst., Man, Cybern. A Syst. Humans, vol. 34, no. 3, pp. 305–314, May 2004

[6] F. Pravin, D. Rajiv, HASTA MUDRA An Interpretation of Indian Sign Hand Gestures, 3rd International conference on Electronics Computer technology, vol. 2, pp.377-380, 2011

[7] N.B. Ibrahim et al. / Journal of King Saud University – Computer and Information Sciences 30 (2018) 470–477

[8] H. Matsuo, S. Igi, S. Lu, Y. Nagashima, Y. Takata, and T. Teshima, "The recognition algorithm with noncontact for Japanese sign language using morphological analysis," Proceedings of Gesture Workshop, pp. 273–284, 1997.

[9]] J. S. Kim, W. Jang, and Z. Bien, "A dynamic gesture recognition system for the Korean sign language (KSL)," IEEE Transations on Systerms, Man, Cybern. Part B, Vol. 26, pp. 354–359, April 1996.