**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE.NO** |
| **1.** | **Abstract** | **4** |
| **2.** | **Introduction**  2.1 Existing System  2.2 Proposed System | **5-6** |
| **3.** | **System Analysis**  3.1 Block Diagram  3.2 Architecture | **7-8** |
| **4.** | **Algorithms**  4.1 CNN  4.1.2 Relu’s Law  4.2 Pooling Layer  4.3 Flattening  4.4 Full Connection | **9-11** |
| **5.** | **SDLC** | **12-13** |
| **6.** | **System Study**  6.1 Feasibility study  6.2 Economic Study  6.3 Social Feasibility | **14** |
| **7.** | **System Requirements**  7.1 functional  7.2 Non-Functional | **15-16** |
| **8.** | **Software Specifications**  8.1 S/w Specifications  8.2 H/w Specifications | **17** |
| **9.** | **System Design**  **9.1** Input Design  9.2 Output Deesign | **18-19** |
| **10.** | **Modules**  **10.1** System Module  10.2 User Module | **20** |
| **11.** | **Diagrams**  **11.1** UML Diagram  11.2 Usecase diagrams  11.3 Class Diagrams  11.4 Sequence diagram  11.5 Collaboration  11.6 Development  11.7 Activity Diagram  11.8 Component  11.9 ER Diagrams  11.10 DFD Diagrams | **21-31** |
| **12** | **INTERFACE OUTLOOK** | **32-34** |
|  | **DATASET IMAGES** | **35-42** |
|  | **CONCLUSION** | **43-44** |
|  | **PROFILE** | **45** |

**HAND GESTURE AND VOICE CONVERSION FOR DEAF AND DUMB USING DEEP LEARNING**

**Abstract**

Despite a wide social group that could profit from it, the idea of sign language recognition by technology is underutilized. Several technologies are available that can be helpful in establishing a link between this social group and the rest of the world. One of the main tools for enabling sign language users to communicate with the rest of society is understanding sign language. Computers can recognize sign language with the use of image categorization and machine learning, which can then be translated by humans. This study uses convolutional neural networks to identify sign language motions. The static sign language gestures were photographed using an RGB camera and comprise the image dataset used. The photos underwent preprocessing before being used as the input that had been cleaned. Inception v3 convolutional neural network model was used to retrain and test this dataset of sign language motions, and the findings are presented in this study. Several convolution filter inputs are processed on a single input in the model.

The resulting validation accuracy was greater than 90%. The numerous attempts at sign language detection using machine learning and image depth data are also reviewed in this work. It assesses the different difficulties involved in solving the issue at hand and also describes the problem's potential future.

**Key words:** Sign Language Recognition, Deep Learning, Inception V3, Image Processing.

**Introduction**

A pattern recognition issue can be framed as comprehending human gestures. The required message can be conveyed if a computer can recognize and distinguish these human motion patterns reconstructed. Successful detection of static sign gestures for letters and numbers has been made. But, this system can also be expanded to recognize words and sentences. Here, American Sign Language (ASL) has been employed as the sign language whose gestures we are trying to recognize. This is also where many other sign languages have their roots. The bulk of people communicate with one another through spoken language. For a huge section of the population, it would not be conceivable without spoken language to exchange ideas. Nonetheless, despite the use of spoken language, certain people are unable to communicate with the majority of people. Those who are mute are unable to converse verbally. This segment of the community benefits from the use of sign language. Sign language provides the same instruments for communication as spoken language does, including facial emotions, static hand symbols, and hand gestures. There are numerous varieties of sign languages, just like there are in spoken languages. They have their roots in dialect and geography, just like spoken language. Polish Sign Language, American Sign Language, Indian Sign Language, etc. are some examples. It does have disadvantages of its own as a result of these variances. First of all, like all It is only appropriate for use among speakers of the language. Speech-impaired sign language users are unable to communicate with the general public, which is only fluent in spoken language. The distance between the two must be filled in order to enable greater and more effective communication. Human-computer interface is another challenge this group of people must overcome, particularly since sign language is their exclusive form of communication. There are between 250,000 and 500,000 ASL users. This is small compared to the size of the population. For the purpose of teaching and comprehending sign language, technology in the form of various software packages has been created.

Nonetheless, there has been good but limited progress in the use of contemporary technology for sign language recognition. A program that can effectively recognize and translate sign language is now required. Most importantly, it ought to serve as a link between sign language users and individuals who lack any immediate incentive to learn or comprehend the language. one such methodology in order to determine how well it recognizes sign language. In our experimentation, we have taken this into account and used American Sign Language (ASL).

**Existing system:**

Existing systems for sign language detection using deep learning. There are many other systems that use different deep learning architectures and techniques to recognize sign language gestures. This system uses a CNN to recognize sign language gestures. The CNN is trained on a large dataset of sign language gestures to learn to recognize the gestures.

**Disadvantages:**

* More Expensive.
* Difficult to scale up.
* Time consuming.

**Proposed system:**

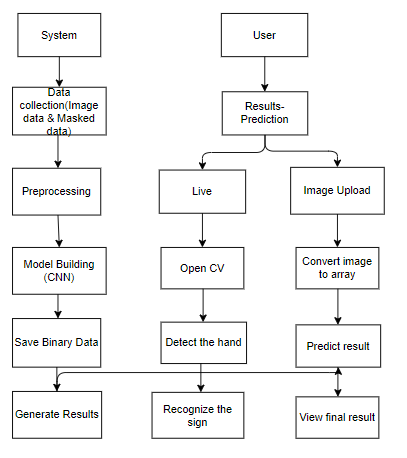
One potential enhancement to this proposed system could be to incorporate real-time sign language detection using a video stream. This would involve modifying the feature extraction and model training steps to accommodate real-time processing, as well as implementing a mechanism for handling video input and output. Another enhancement could be to incorporate multiple modalities, such as audio and facial expressions, to improve the accuracy of sign language detection.

**Advantages:**

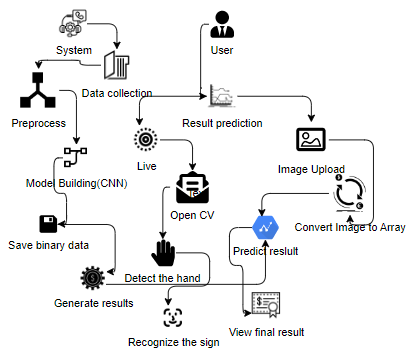
* Cheaper to operate.
* It can be scaled up quickly.
* Time minimising.

**System Analysis**

**Block Diagram:**

****

**Architecture:**

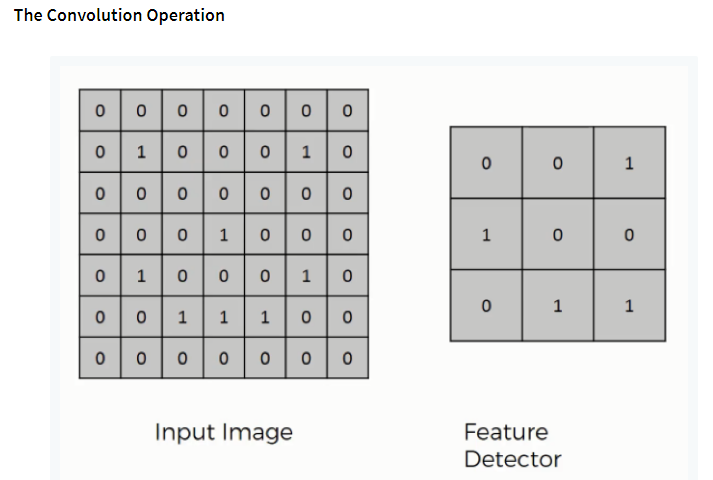
****

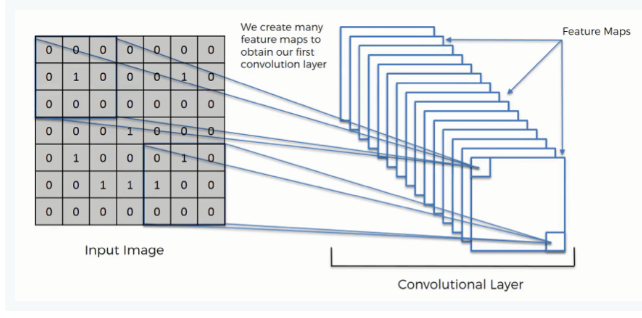
**Algorithm:**

**1. Convolutional Neural Network**

**Step1: convolutional operation**

The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out.

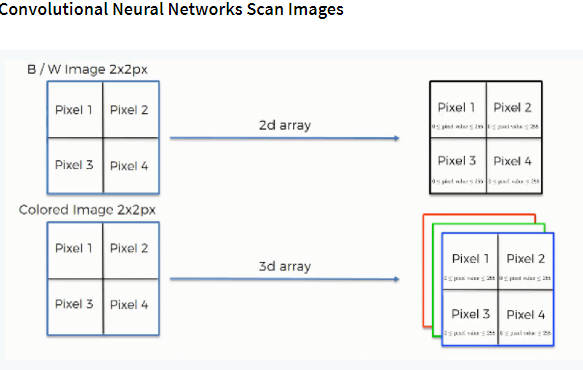




**Step (1b): Relu Layer**

The second part of this step will involve the Rectified Linear Unit or Relook. We will cover Relook layers and explore how linearity functions in the context of Convolutional Neural Networks.

Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.



**Step 2: Pooling Layer**

In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.

**Step 3: Flattening**

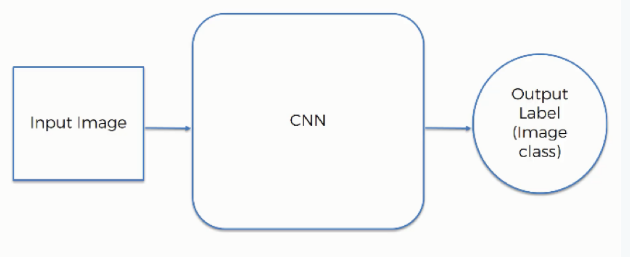
This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

**Step 4: Full Connection**

In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.

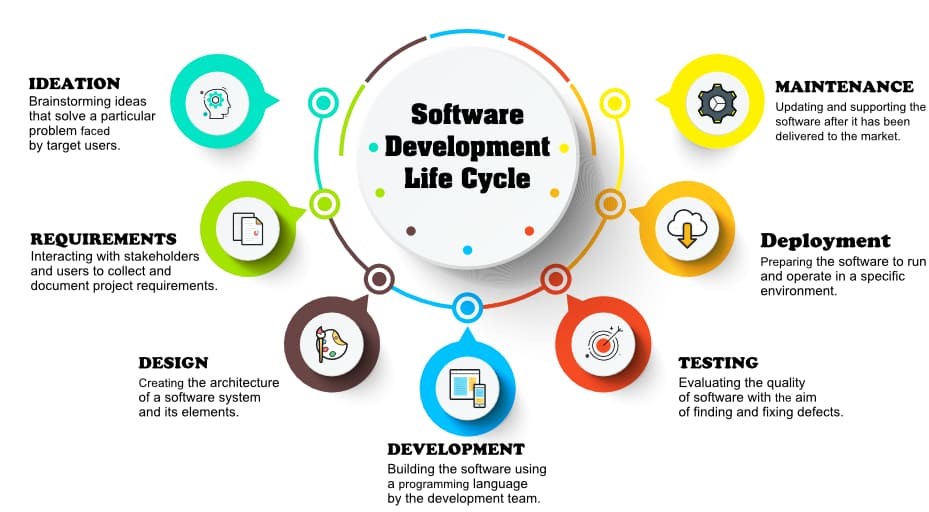
**Summary:**

In the end, we'll wrap everything up and give a quick recap of the concept covered in the section. If you feel like it will do you any benefit (and it probably will), you should check out the extra tutorial in which Soft ax and Cross-Entropy are covered. It's not mandatory for the course, but you will likely come across these concepts when working with Convolutional Neural Networks and it will do you a lot of good to be familiar with them.



**Software development life cycle – sdlc:**

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.



**Fig 3**: SDLC Model

* **Requirement Gathering and analysis** − all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**Feasibility study**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**Economic feasibility:**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased. Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social feasibility:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**System Requirements Specification**

**Functional and non-functional requirements:**

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

**Functional Requirements**: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1. Authentication of user whenever he/she logs into the system
2. System shutdown in case of a cyber-attack
3. A verification email is sent to user whenever he/she register for the first time on some software system.

**Non-functional requirements**: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Examples of non-functional requirements:

1. Emails should be sent with a latency of no greater than 12 hours from such an activity.
2. The processing of each request should be done within 10 seconds
3. The site should load in 3 seconds whenever of simultaneous users are > 10000

**SYSTEM SPECIFICATIONS:**

# H/W Specifications:

# Processor : I5/Intel Processor

# RAM : 8GB (min)

* Hard Disk : 128 GB

**S/W Specifications:**

* Operating System : Windows 10
* Server-side Script : Python 3.6
* IDE : PyCharm, Jupyter notebook
* Libraries Used : Numpy, IO, OS, Flask, Keras, pandas, tensorflow, Segmentation

**System Design:**

## Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

### Objectives for Input Design:

The objectives of input design are

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### Objectives of Output Design:

The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**Modules:**

**System:**

**Data collection:** First, System will collect the user data through image or masked data.

**Preprocessing:** System will preprocess the data which was collected by the system.

**Model Building:** CNN algorithm is used in this project for model building.

**Save Binary data:** The system will save the binary data of the project.

**Generate results:** Finally, it will generate the results.

**User:**

**Result prediction:** User, will predict the result which was saved by the system.

**Live**: The predicted result will show the live result to the user.

**Image upload:** The user will upload the image and the system will show the exact result.

**Open CV:** Here, the cam will open to detect the Gesture.

**Recognize the sign:** The cam will recognize the sign of the hand.

**UML DIAGRAMS**

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

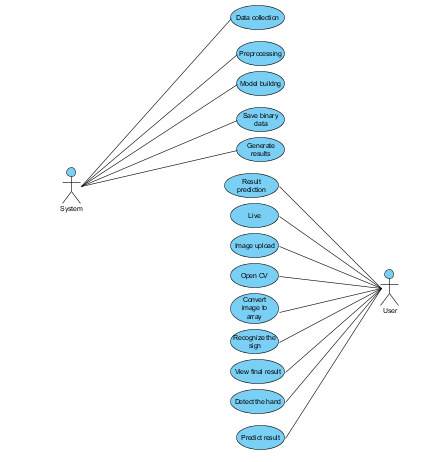
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

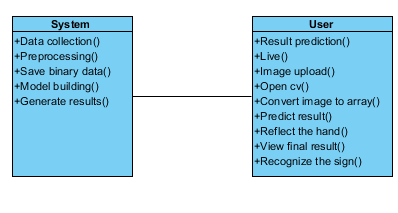
**USE CASE DIAGRAM**

* A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
* Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.
* The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



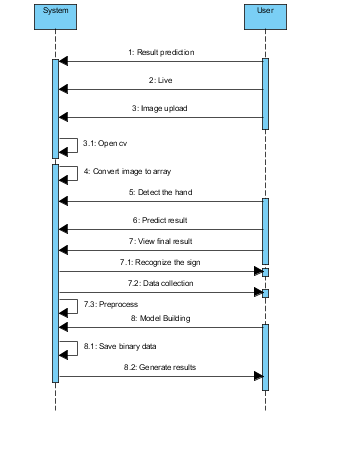
**CLASS DIAGRAM**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information



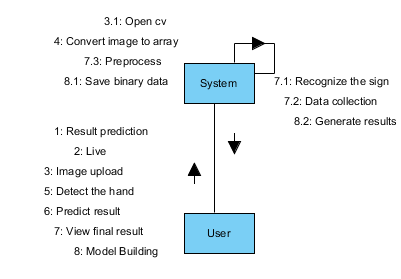
**SEQUENCE DIAGRAM**

* A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.
* It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams



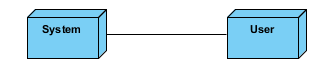
**COLLABORATION DIAGRAM:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



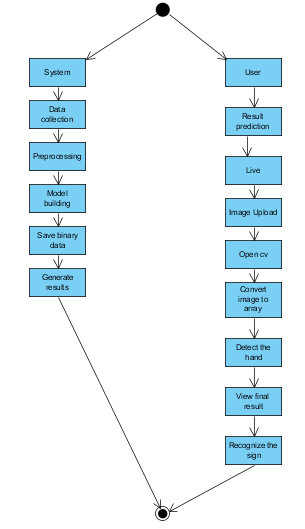
**DEPLOYMENT DIAGRAM**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**COMPONENT DIAGRAM**:

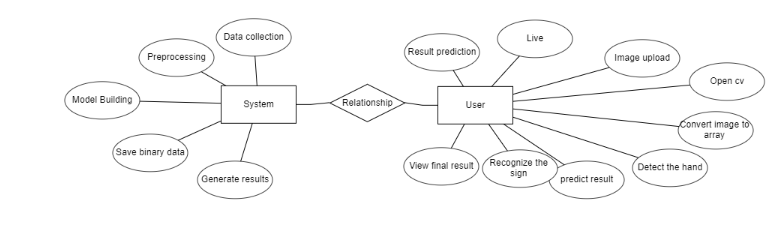
 component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



**ER DIAGRAM:**

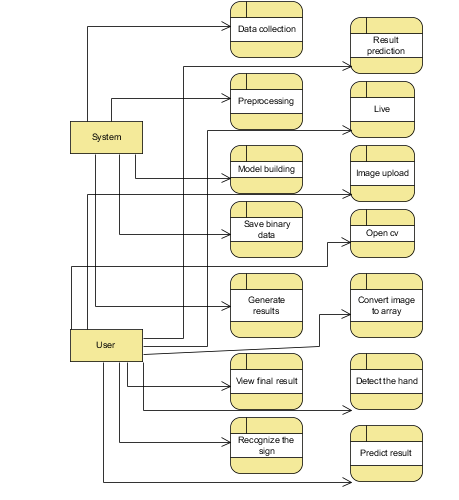
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

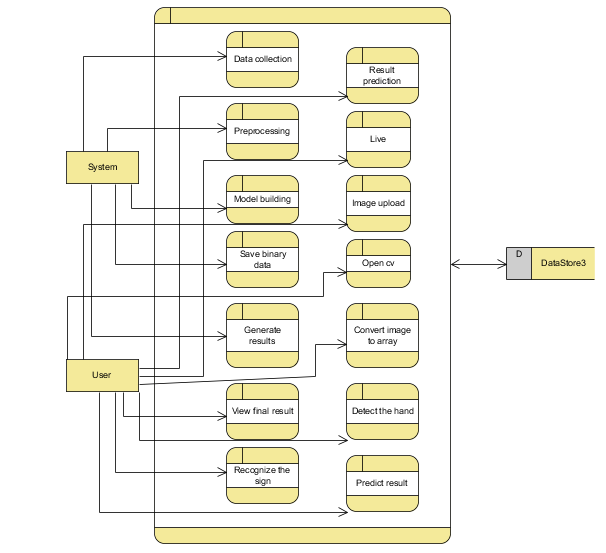
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.



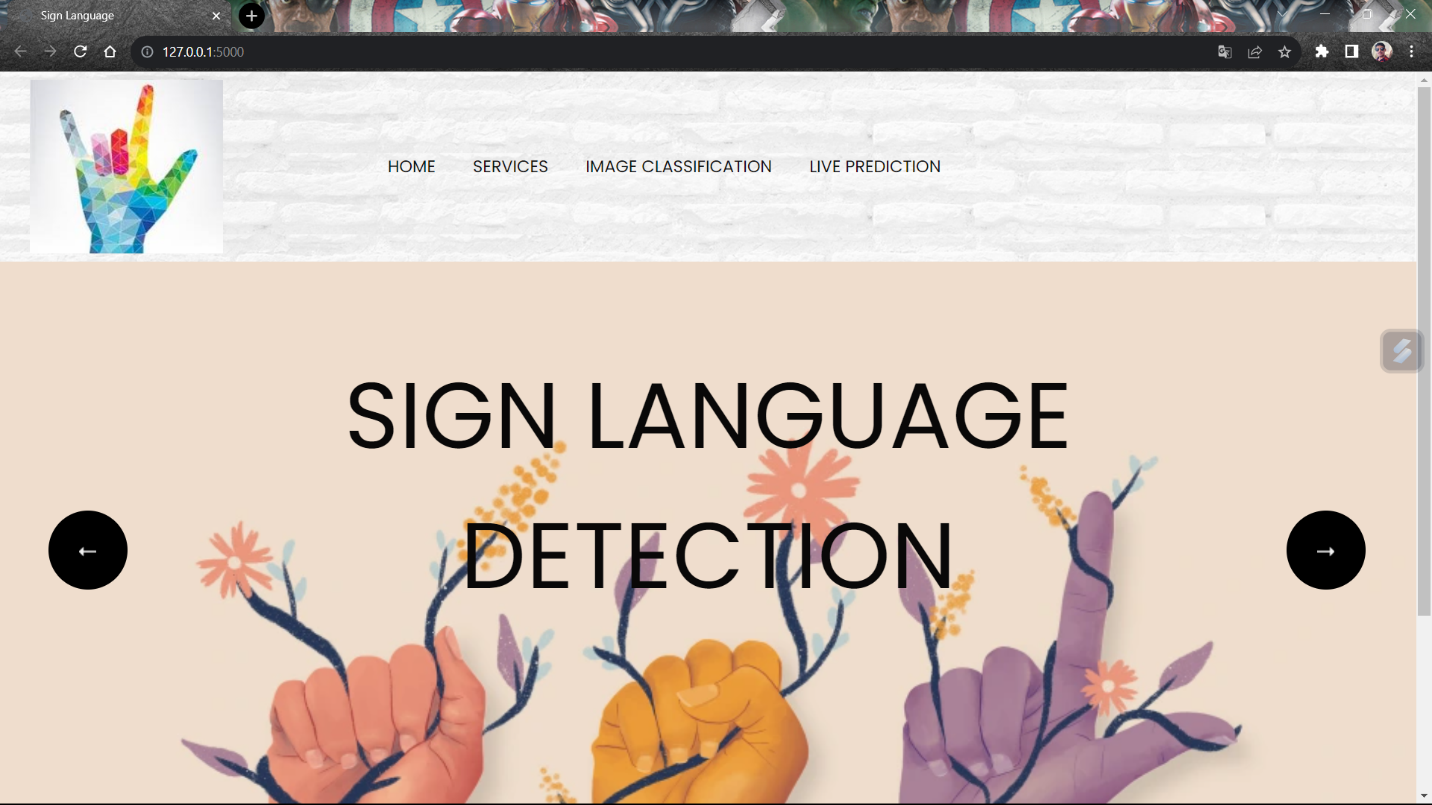
**DFD DIAGRAM:**

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

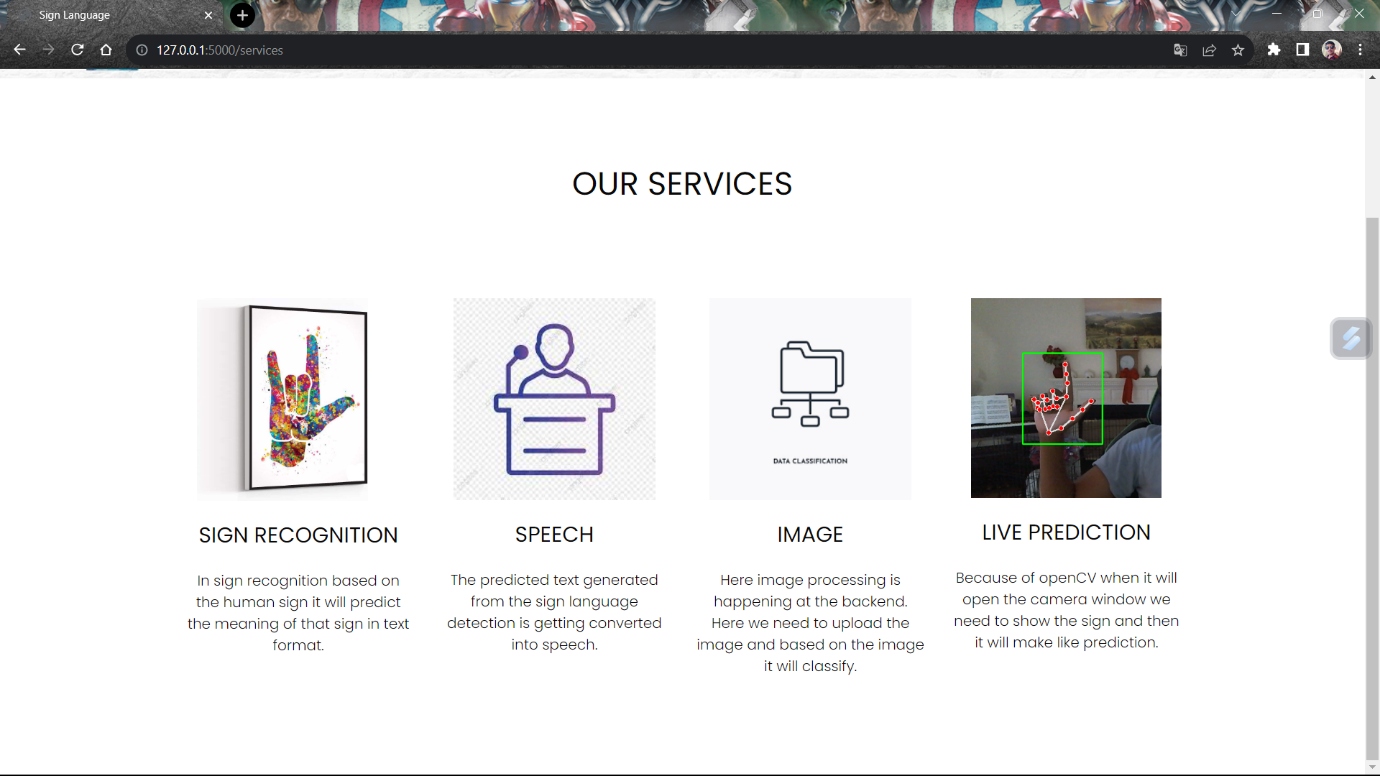




**Interface outlook**

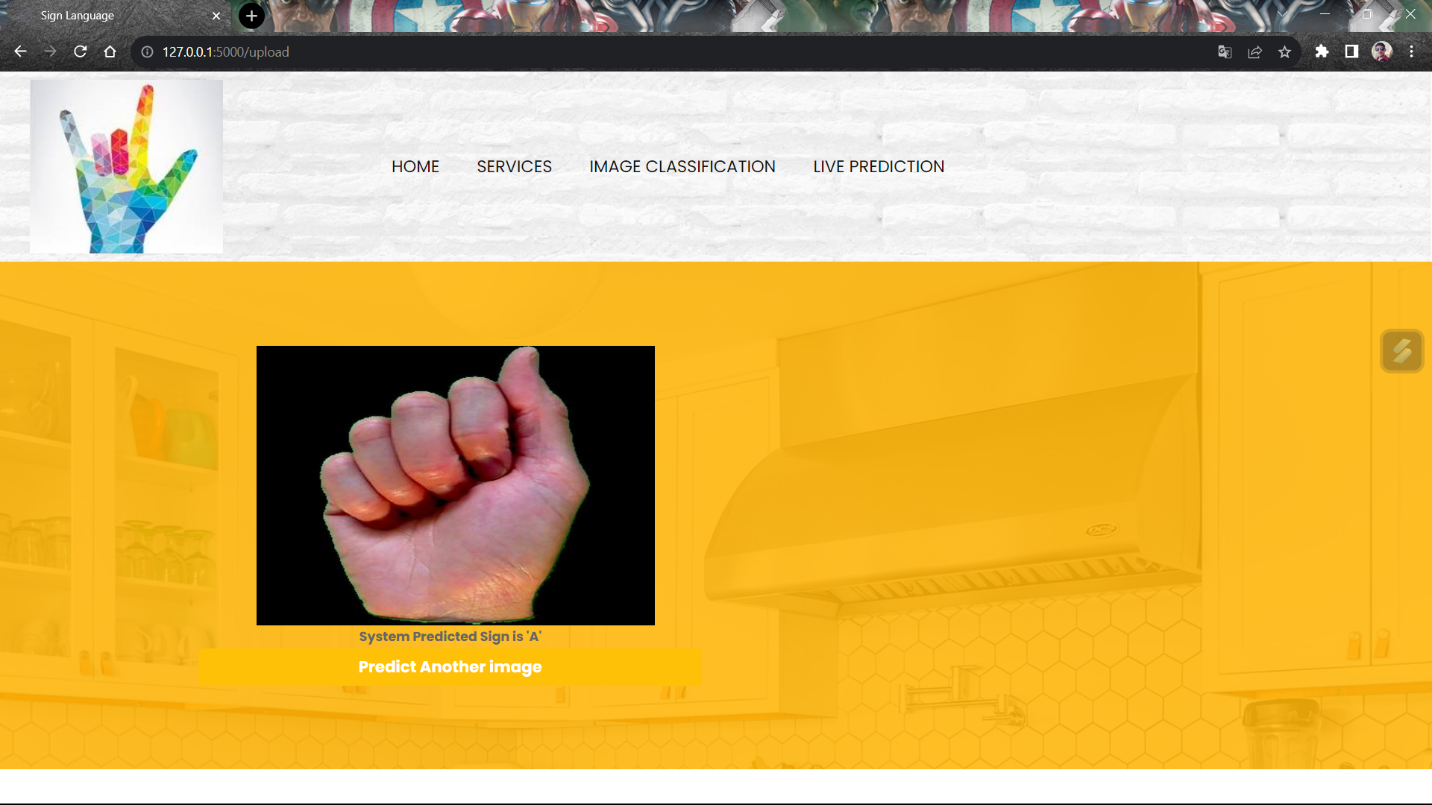
****

This is the interface when login into the website



**Services Provided by the Website**

****

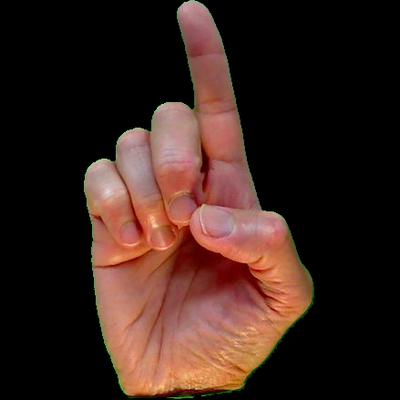
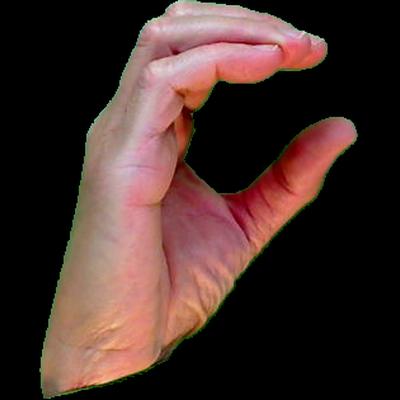
** Uploding the Image tells the word/Letter in this interface**

**Output of the given image**

**DATASET IMAGES**

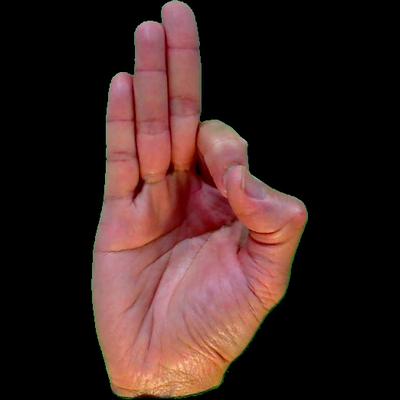
****

**A B**

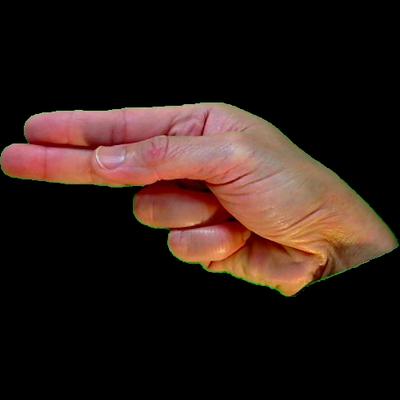
****

**C D**

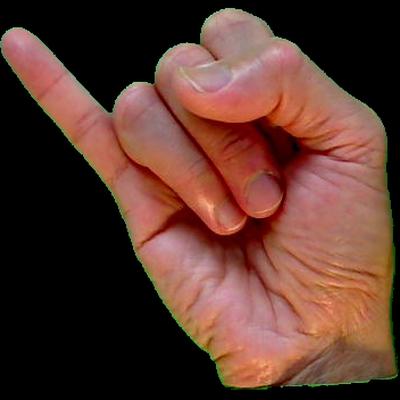
****

****

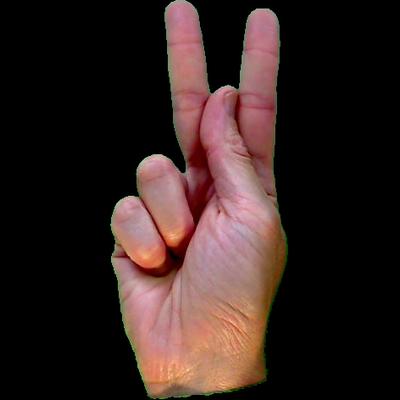
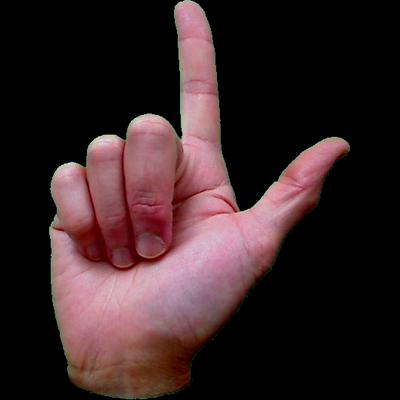
**E F**

****

**G H**

****

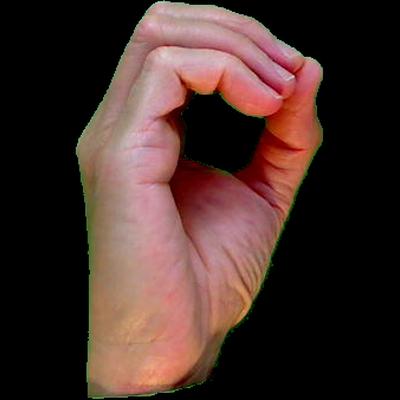
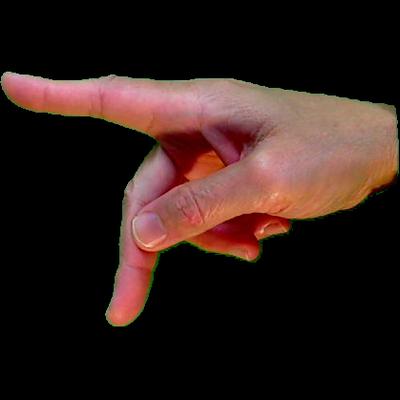
**I J**

****

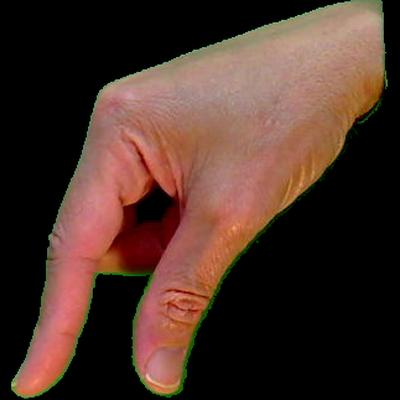
**K L**

****

**M N**

****

**O P**

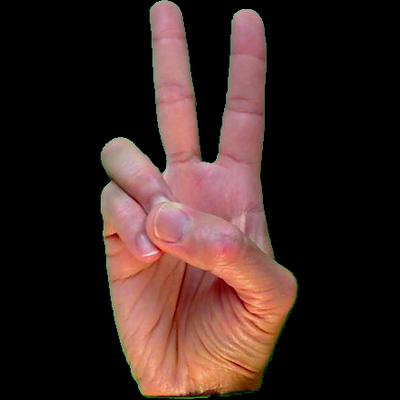
****

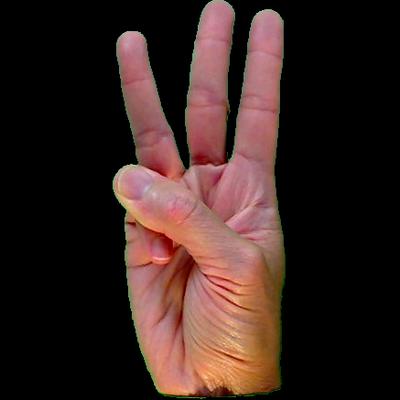
**Q R**

****

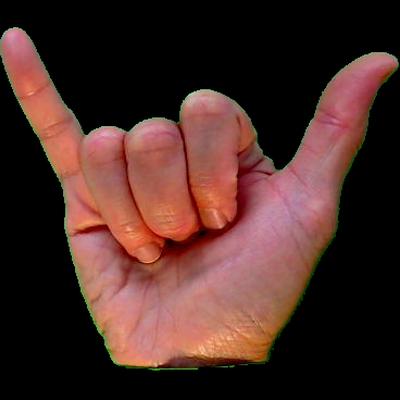
****

**S T**

****

** U V**

**W X**

****

****

**Y Z**

**CONCLUSION**

**Conclusion:**

In conclusion, deep learning has shown great potential in sign language detection. With the help of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), accurate recognition of sign language gestures can be achieved. Deep learning models have been trained on large datasets of sign language videos, allowing them to learn the subtle differences between signs and interpret them correctly. Sign language detection using deep learning has numerous applications, including improving communication between the deaf and hearing communities, creating more accessible technologies for the deaf and hard-of-hearing, and facilitating communication in noisy environments where speech recognition may be challenging.

**PROFILE**

NAME : Pamuru Rakesh

Reg. No : 210200001020

FATHER NAME : Pamuru Siva Deva Kumar

MOTHER NAME : Pamuru Sirisha

ADDRESS : Main Road,

Pothireddy Palem,

Kovur(M),

Nellore (Dt),

PINCODE:524002, AP

Temporary ADDRESS : Main Road,

Pothireddy Palem,

Kovur(M),

Nellore (Dt),

PINCODE:524002, AP

ICET RANK : 3661

EMAIL ID : rakeshrakhi4046@gmail.com

PHONE NO : 9014949078

MCA Percentage : 80