

***DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,***

***SCHOOL OF ENGINEERING AND TECHNOLOGY,***

***SHARDA UNIVERSITY, GREATER NOIDA***

**SIGN LANGUAGE DETECTION**

***A project submitted***

***In partial fulfillment of the requirements for the degree of***

***Bachelor of Technology in Computer Science and Engineering***

**By**

**NEHA CHALLA(2019003269)**

**VINAYAL ROHATGI(2018015118)**

**KRITI BAISHYA(2018007478)**

**Supervised by:**

**Dr.Keshav Gupta, Assoc. Professor (CSE)**

April-2022

**CERTIFICATE**

This is to certify that the report entitled **“Sign Language Detection”** submitted by

**“**Ms. NEHA CHALLA (201903269), KRITI BAISHYA (2018015118), VINAYAK ROHATGI (2018007478)**”** to Sharda University, towards the fulfillment of requirements of the degree of“Bachelor of Technology” is the record of bonafide final year Project work carried out by him/her in the **“**Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University**”**.

The results/findings contained in this Project have not been submitted in part or full to any other University/Institute for the award of any other Degree/Diploma.

Signature of Supervisor

Name: Dr. Keshav Gupta

Designation: Assoc. Prof (CSE)

Signature of Head of Department

Name: Prof. (Dr.) Nitin Rakesh

Place: Sharda University

Date:

**Signature of External Examiner**

**Date:**

**ACKNOWLEDGEMENT**

A major project is a golden opportunity for learning and self-development. We consider ourselves very lucky and honored to have so many wonderful people lead us through the completion of this project.

First and foremost we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

My grateful thanks to **Dr. Keshav Gupta & Dr.** for their guidance in our project work.

**Dr. Keshav Gupta**, in spite of being extraordinarily busy with academics, took time out to hear, guide, and keep us on the correct path. We do not know where we would have been without her help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students

NEHA CHALLA (2019003269)

KRITI BAISHYA (2018015118)

VINAYAK

ROHATGI(2018007478)

**Abstract**

In the field of Sign Language Detection, this paper will identify the best approach for achieving 97% accuracy (SLD). While a great deal of research has been done in SLD, the issue remains unanswered, notwithstanding the fact that we somehow need high accuracy. There is always an overwhelming desire in this fast-paced environment to digitize paperwork and to record data straight in modern form. And even today, there is quite an absolute lack in this field. From period to period, cochlear implant procedures and their endless improvisation attempt to fill this space. This research is about constructing an equation for handwritten character recognition, also known as SLD, putting aside the cochlear implant that deals with people suffering hearing loss.

This analysis will use a convolution neural network and TensorFlow for offline sign language character recognition. The goal is to improve the algorithm to achieve a much higher precision level while minimizing and optimizing time and space complexity. We support segmented sign language detection with neural networks for helping words and letter recognition. There are a variety of strategies for extracting roles and developing SLD programs in the literature, each with its own set of benefits and drawbacks. We put these theories to the test to see whether we can use language detection to enhance the recognition of English helping or emergency words.

***Index Terms*** Sign Language, Dynamic, Transfer Learning, Convolution Neural Network, Image Pre-processing,

|  |  |
| --- | --- |
| **Contents** |  |
| Title Page | i |
| [CERTIFICATE](#30j0zll) | [ii](#30j0zll) |
| [ACKNOWLEDGEMENT](#1fob9te) | [iii](#1fob9te) |
| [Abstract](#3znysh7) | [iv](#3znysh7) |
| [Chapter1: INTRODUCTION](#tyjcwt) | [6](#tyjcwt) |
| 1.1 [Problem Definition](#3dy6vkm) | [7](#3dy6vkm) |
| 1.2 [Project Overview/ Requirement Specifications](#3dy6vkm) | [7](#3dy6vkm) |
| 1.3 [Hardware Specifications](#2s8eyo1) | [10](#2s8eyo1) |
| 1.4 [Software Specifications](#17dp8vu) | [11](#17dp8vu) |
| [Chapter2: Literature Survey](#3rdcrjn) | [12](#3rdcrjn) |
| 2.1 [Existing System:](#3rdcrjn) | [12](#3rdcrjn) |
| 2.2 [Proposed System](#35nkun2) | [14](#35nkun2) |
| 2.3 [Feasibility Study](#1ksv4uv) | [16](#1ksv4uv) |
| 2.4 [Risk Management](#3j2qqm3) | [18](#3j2qqm3) |
| [Chapter 3: System Analysis and Design](#4i7ojhp) | [20](#4i7ojhp) |
| 3.1 [Software Requirement Specification](#4i7ojhp) | [20](#4i7ojhp) |
| 3.2 [Flowcharts/DFDs/ERDs](#3whwml4) | [23](#3whwml4) |
| 3.3 [Design and Test Steps/Criteria](#1pxezwc) | [27](#1pxezwc) |
| 3.4 Testing Process | 29 |
| Chapter 4: RESULTS / OUTPUTS | 33 |
| Chapter 5: Conclusion | 35 |
| 5.1 Further Improvement: | 40 |
| Chapter 6: References | 50 |

**Chapter1: INTRODUCTION**

This 'Sign Language Detection' project is a software algorithm that can effectively detect and recognize any sign language character on an input unit, whether it's an alphabet or anything more modern like emergency words, helping words or those words which can be used in times of need. Identifying sign language characters, especially those using help words as well as alphabets is a relatively new research field with a wide range of applications. The primary goal of this project is to create the "SLD utilizing Neural Network" expert system, which employs an Artificial Neural Network approach to define the use of sign language in case of emergency needs.

In the field of neural computation, it is relatively new. Data parallelism is introduced by neural computers. A trained (not scheduled) neural computer is used to assign input data to several classes or allow original data to expand to maximize the resource level due to a given starting state (data input).

The principle is to incorporate successful algorithms that obtain signs in real-time and not from any pre-trained datasets with higher accuracy rates. After that, the image is processed for better contrast, and then those images are labeled and updated on the basis of a label map. Then checkpoints are updated. Finally, signs are detected from the image, and text-to-speech conversion is done.

**Overview:**

To describe the learning model used, and the specification and implementation of algorithms, we needed to investigate the most recent approaches to the subject of sign language detection. We define the project and compare the strategies proposed in this chapter as potential candidates.

**Project Description:**

Users refer to online computer learning as virtual learning. This is not to be confused with the previously mentioned dynamic and static sign language detection. Static-based sign language can be developed using pre-trained datasets and applied to static images or pictures from where signs can be detected. The user gives the image as an input and the algorithm detects and tells the name of the sign from the image. On the other hand, a dynamic one can be used for not only pre-trained datasets but also for real-time data and can train the words and letters in real-time. Users train in real-time, give input in real-time, and with the help of wide-ranging algorithms, accuracy can also be improved. An application that can detect dynamic words, as well as alphabets, can be of great use for the user as there are not many applications that detect words and help people in need by using emergency words.

**1.1 Problem Definition**

The project's goal is to input, process, and train sign language’s English help words as well as alphabets in order to detect signs and then translate them to speech.

This project aims to create applications that will aid in the identification of signs in the English language. Only letters and English words are allowed in this project. Though numbers and patterns can also be trained, it will affect the accuracy rates.

One of the most critical applications of a neural network is for computers to mimic human abilities. Neural networks can be used to solve problems such as pattern recognition, category classification, sequence estimation, and data mining.

The most well-known application of neural grids is perhaps pattern recognition and sign language detection. The neural network has access to the objective vector and a pattern information vector that can be a manuscript image. The neural network then tries to fit the data into a previously saved sequence of the image.

To group input samples into groups, a neural network with classification capabilities is devised. These groups can be hazy if their distinctions aren't well established. This project aims to identify help words and characters.

**1.2 Project Overview/ Requirement Specifications**

**1.2.1 Functional Requirements**

**1.2.1.1 Introduction**

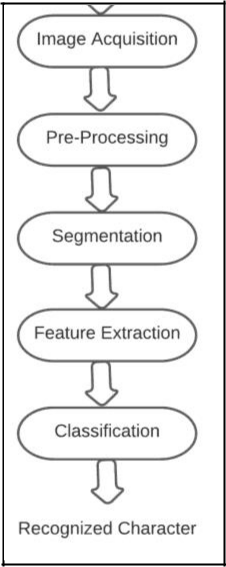
The purpose of this venture is to update a grouping calculation to perceive sign characters and words. Example detection has been shown to perform the best for all grouping problems reliably. Subsequently, in this particular issue of sign language detection, the scope of the undertaking also incorporated the rudimentary study of the distinctive classifiers and blend techniques and evaluated the admonitions around their execution.

**1.2.1.2 Input**

Letters/Alphabets, video sequence, words

Prior to using the framework must be fully prepared and a customer knows the specific framework. When the structure is prepared, you need to spare the structure before leaving, so the framework is stacked for further use. The quality has also been upgraded.

**1.2.1.3 Processing**





**1.2.1.4 Error Handling**

If the user shows an incorrect sign symbol, the machine detects it and shows a default message to verify symbol errors in sign input.

**1.2.2 Normal Requirements**

Criteria for our sign language detector are as follows:

**N1:** The program should provide a user interface with graphics.

**N2:** Feedback should identify alphabets with good accuracy.

**N3:** It should classify computer-based English help words or emergency words.

**N4:** The program should be able to fit the words as well as letters stored as the sign language input.

**1.2.3 Non Functional Requirements**

These are the specifications, as the name implies, that are not specifically correlated with particular functions offered by the device.

**1.2.3.1 Performance Requirements**

Execution based on the relation, high execution of the PC may involve one or more of the accompanying: fast reaction time for a certain bit of work. In comparison to the time and assets used, execution is defined by the measure of useful work done by a PC framework or PC system.

**1.2.3.2 Reliability**

Unwavering quality is a property of any component related to a PC (for example, programming, or equipment, or system) that performs consistently as its determinations indicate. For some time, it has been seen as one of three related characteristics that should be considered when making, buying, or using an object or part of a PC.

**1.2.3.3 Availability**

Accessibility is a general concept used in PC systems and system management to describe the measure of time over a one-year span that the framework assets are available in the wake of partial system disappointments. A structure with all its properties that is continually available is seen as fruitful.

**1.2.3.4 Security**

Security (or PC security) in registration is the technique to ensure that information placed on a PC cannot be accessed or negotiated without approval by any person. Data encryption and passwords are the majority of PC efforts to develop security. Encryption of information is the interpretation of data into a structure that is indiscernible without a method of disentanglement. A watchword is a mystery word or phrase that gives a client access to a particular project or structure.

**1.2.3.5 Maintainability**

It is defined as the probability within a given time of conducting a successful repair operation. As such, practicality tests the straightforwardness and speed at which, after a disappointment occurs, a system can be returned to operating status. Convenience is a trademark that is credited to a PC application in the event that it may be used rather than the one in which it was developed as part of operating systems without the need for major reconstruction. Porting is the job of performing whatever work is necessary to maintain the PC program going in the new environment.

**1.2.3.6 Ability to learn**

It is simple to operate and reduces the learning function.

**1.3 Hardware Specifications**

|  |  |
| --- | --- |
| **Minimum Requirements** | **Windows** |
| **Operating System** | Windows 7 |
| **Processor** | Dual Core, Intel i3 |
| **RAM** | 2 GB RAM |
| **Disk Space** | The amount of disk space available depends on the partitioned size and whether or not online help files are allowed. The math works installer would tell you how much disk volume your partition needs. |
| **Graphics Adapter** | Any, preferably 8 bit graphic adapter and display |
| **CD ROM drive** | Any |

|  |  |  |  |
| --- | --- | --- | --- |
| **Recommended Requirements** | **Windows** |  |  |
|  | **Processor** | **Disk Space** | **Graphic Adapter** |
|  | Intel i3 | 2 GB RAM, 1 GB for VS Code only | A 32-bit or 64 bit openGL |

**1.4 Software Specifications**

|  |
| --- |
| **Tensorflow** |
| **HTML** |
| **CSS** |
| **JSON** |

**Language 1) JavaScript with Tensorflow**

**Chapter2: Literature Survey**

**2.1 Existing System:**

**“Optimal Approach for sign language for transliteration”**

* + To classify sign language characters in English, the article used Neural Networks.
  + Use of American Sign Language as the means
  + 24 static characters are detected using machine deciphered English manuscripts.
  + In the studies conducted, the corresponding English words appeared to be misclassified as a set of data where the studies conducted were minimal.
  + The more accurate the set of data used the most precise outcomes the system would achieve.

**“Dynamic Approach for Sign Language with feature extraction”:**

* + The Feed Forward methods, as well as the backpropagation learning method, are shown in this article.
  + To effectively identify characters, the paper utilizes a multilayer perceptron with one invisible layer.
  + In the back-propagation algorithm, greater efficiency can be achieved once all the hidden neurons to use it have been measured successfully.
  + Sign recognition for English ASL is known to be stronger and gives an accuracy of 90 percent and above.

**“Sign language variations and regional pattern accommodation”:**

* + This paper discussed the different steps of image processing which is used for the identification of signs, such as the reconstruction of characters, enhancement of the images, classification of recognition, segmentation, and extraction.
  + The NN uses these methods due to the incredibly high noise tolerance.
  + In images that are blurred through the years, this helps eliminate any unwanted signals. For such reports, effective language detection is also achievable and systems produce perfect results.

**“Interdisciplinary principle of American Sign Language”**

* + To classify letters, this paper used neural networks.



* Enhances the efficiency of recognition of English words-A to Z

To recognize and understand characters, this paper used feed-forward propagation and an NN to use backpropagation.

* + Since English characters have only been fed to NN to be processed further, they are represented in binary form.

**“Sign Language Linguistics”**

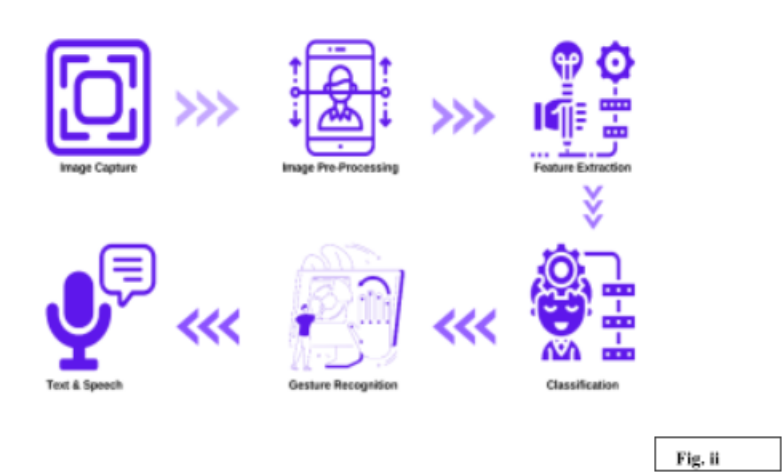
* + A correlation of the feature vectors is discussed in the paper, and the extraction of the feature is shown to improve performance over its standard counterparts.
  + The extraction gradient approach is best suited toward gray-scale images. Offering the "Normalization-Cooperated Feature Extraction" (NCFE) the most precise character rate, and also resulting in good results.
  + For gray-scale images, the gradient extraction method applies, and for binary images, other extraction methods apply.



|  |  |
| --- | --- |
| **Techniques or Algorithms Used** | **Accuracy rate [%]** |
| “Feedforward Techniques”. [21] | 85 |
| “Pre-processing & NN based recognizing work”. [3] | 82.5 |
| “Holistic and segmentation approach”. [22] | 98.75 |
| “Multi-layer feed-forward NN. Diag. based feature | 97, for 54 features and 98 percent for |
| extraction”. [23] | 69. |
| “Error Back Propagation Technique”. [24] | 70 |
|  |  |
| “Modified Hough transform”. [25] | 67.3 |
| “Segmentation, NN, & Statistical classifier”. [26] | 73.25 |
|  |  |
| “Determining multiple segmentation ” [27] | 67 for Upper Case, 75 for Lower |
|  | Case, and 76 percent Average |
|  | accuracy. |

**2.2 Proposed System**

The suggested recognition scheme is listed in this section. The below figure consists of a standard handwriting recognition system. Figure 2 shows a general schematic of the method.



**2.2.1 Image Preprocessing**

Various scanning algorithms are used to process the images and videos.

Image preprocessing is critical in the identification pipeline for accurate character prediction. Other techniques, such as noise reduction, image segmentation, cutting, and scaling, are also used. These techniques were used primarily in our project for 4-image recognition, but some were also used in touch mode, such as cutting the written character and scaling it to our input.

Noise is often introduced during the digital recording and transfer of an image, making it difficult to distinguish between what is and is not an object of interest. When it comes to character recognition, we want to reduce the number of essential noises for proper recognition. This can be accomplished in a variety of ways.

**2.2.2 Feature Extraction**

It is one of the most important steps in sign language detection as here all the feature vectors are extracted and obtained. Then, the classification task is assigned for a new feature vector and some of the trained modules are used to classify them as training.

The measurements' conservative properties are the features of the input data used to analyze or distinguish these instances of data. Feature extraction aims to choose the appropriate components.

According to the report, this could be the most crucial factor to choose a feature extraction tool. Sign language data is extracted using a variety of techniques, each with its own set of characteristics, invariance properties, and sign reconstruction ability. According to the paper, the question of which approach is ideally suited to a particular case must be tested experimentally.

Dong implemented MS Kinect based on American Sign Language in 2018, and they were used for section characters, words, and text lines.

**2.2.3 Classification**

We are seeking to create a training model and validate NN within this step.

Classification refers to the process of assigning marks (categories, classes) to findings that have yet to be categorized (instances of data). This is solved by running an algorithm via a series of machine learning training courses. Classification is a supervised learning problem in which a "teacher" associates a data case with one of several classes. A label is a unique number that distinguishes the class to which a certain instance belongs. It is normally a positive number.

The term "classifier" refers to a variety of machine learning classification models. To define a new observational event, classifiers have been designed to obey the judgment boundary in the space where the training examples are separated. The judgment limit is a hyper-surface of N1 dimensions that divides the field of N dimensions into the two categories shown below.

**2.2.4 Gesture Recognition**

A real-time setup is made using the trained data which was trained dynamically using the designed module.

Here, gestures are recognized according to the given input. If an alphabet is given as an input and if the symbol shown by the user is correct, the user will get the output.

**2.2.5** **Conversion of text to speech**

In the final module, the output which is obtained in the form of text is converted to speech for the convenience of the user.

**2.3 Feasibility Study**

Any comprehension of the major specifications for the scheme is necessary for a feasibility study. Feasibility Dimensions for Computers would be as shown in

* **Technology**

Is the project technically possible?

Is it a component of the state of the art?

Will failure be limited to the need for an implementation meeting the level?

* **Finance**

Is it financially practicable?

Is it realistic for the software company and its customer or company to achieve production at a reasonable pace?

* **Time**

Can the time for the idea to be sold, beat the competition?

* **Resources**

Will the corporation have the capital necessary for success?

Two major variables used in the study of viability are

* 1. Technological Feasibility
  2. Cost Feasibility

1. **Technical Feasibility**

The purpose of this analysis is to check the technological viability, that is to say, the system's technical requirements. Any built system does not have a strong need for the technological resources required. This will add to intense strains on the intellectual resources available. It would bring to the customer's already firm hopes. Since this system can only be applied with minor to no modifications, a bare minimum must be met.

In the following ways, a practical evaluation of feasibility can be carried out.

* 1. NP-Complete
  2. NP-Hard
  3. Satisfiability

1. **NP-Complete**

The **P Class** comprises those issues which can be solved in polynomial time. The **NP class** consists of such concerns which can be verified in polynomial time.

If any other issue in NP can be converted (or reduced) into p in polynomial time, a question p in NP is **NP-complete.**

1. **NP-Hard**

There are problems where no such viable solutions have been identified. The complexities of these topics are usually more complex unlike P, NP, and NP-Complete. Relatively high multiplicative constants, exponent terms, or polynomials of a high order can be involved in this.

1. **Satisfiability**

In order to make it valid, if there is at least one way to add value to its vector and we denote it by using SAT, the Boolean formula is satisfied. The dilemma of evaluating whether or not the given formula is satisfactory.

1. **Cost Feasibility**

This study evaluates the economic impact of the scheme on the business. It restricts the amount of money that can spend on the research and development of its strategy. It is necessary to justify the expenses. Thus, within the budget, the developed system was also developed and this was done because much of the technology used is readily accessible. It was only appropriate to buy personalized items

**2.4 Risk Management:-**

**2.4.1 Risk Identification**

**2.4.1.1 Product Size Related**

R1 Memory may be squandered as a result of additional lines of code or redundant algorithms.

**2.4.1.2 Customer Related**

R2 Since its consumer isn’t a professional individual and it poses a challenge in interpreting the customer's additional specifications.

R3 If the consumer offers unnecessary details; it can result in an undisclosed danger.

**2.4.1.3 Process Risk**

R4 A fuzzy or disruptive image may be analyzed throughout segmentation.

**2.4.1.4 Technical Risk**

R5 The difficulty of ANN would increase if character features are not extracted.

**2.4.1.5 Development Environment Related**

R6 When a client requests a change or makes an unnecessary alteration later in the implementation process, it is impossible to change the whole system configuration to accommodate the request.

R7 Inexperience and a lack of tool training can make it challenging to complete project modules.

**2.4.2 Strategies used to manage Risks**

S1 By reducing redundant coding, we can prevent Chance R1.

S2 Meeting with the customer regularly reduces the risk to some extent.

S3 R3 properly develops the system to incorporate modifications at a later stage and retains all necessary paperwork to minimize the risk, as previously stated

S4 Use an appropriate noise reduction algorithm prior to segmentation processing.

S5 Extract the character's feature to reduce the ANN's complexity.

S6 We have chosen to complete a phased operation model and to transfer the number of employees to our client in the same manner. The ANN component of the project is demanding and extended.

S7 As consumer demand changes, we will continue to increase the software's functionality.

S8 We will prevent R7 by providing adequate tool instruction

**Chapter 3: System Analysis and Design**

**3.1 Software Requirement Specification**

**3.1.1 Product Perspective**

In view of its composition, SLD is not the first of its kind but certainly the first open-source attempt to interpret the user as every person has an alternate sign that can also be detected at the time of emergency. The aim of the venture will be to add to a system that can be efficiently used by associations everywhere.

**3.1.2 Product Functions**

SLD's real talent is to interpret an author's pool of users. At various organizations, individual confirmation is essential; this venture could be used as part of those places. In observable division, individually identifiable evidence is likewise imperative.

**3.1.3 User Characteristics**

**3.1.3.1 Large Organizations**

When fully developed, the system can be used as part of offices, households, and the separation of the misconduct scene.

**3.1.3.2 Academic Organizations**

In order to get an understanding of essential parts used as part of the framework such as sign language handling techniques, neural systems, and manually used signs, colleges and students are relying on adding to an extensive class of customers, of this structure.

**3.1.4 Design and Implementation Constraints**

Determine the imperatives that can be enforced by various models, confines of equipment, accuracy, etc.

**3.1.4.1 Standards Compliance**

Decide on the basics of current websites or legislation. They can include: Report bunch Data naming Accounting procedures Audit tracing Case in point, this may show the importance of programming to take after getting ready for activity. For a couple of uses, such takes after are necessary to meet minimum regulatory or money-related requirements.

**3.1.5 Assumptions and Dependencies**

* There should be an Internet link that is secure

**3.1.6 Requirement Specification**

**3.1.6.1 User Interfaces**

The user interface architecture consists of three basic interfaces, namely the main window, the window of training, and the window of acknowledgment and testing. The components of each of the system interfaces are listed below.

**The Main Window**

The main interface of the window offers an option for the user to select the detection option. If the user chooses the training option, the device will run the training module by opening the training window.

**The Training window**

Whenever the user wishes to train the device with his or her sign language patterns, the training window is used. It is designed primarily to have a region to join the sign language patterns of the user and a position to show the next character to be trained with extra buttons to pause the training and switch back to the training window.

**The Recognition Window**

When the user decides to enter his or her sign as the input of the SLD, the recognition window is used. The recognition window also includes a region for entering the sign of the user and displaying the recognized symbol, much like the training window.

**Hardware Interfaces**

A good detector is needed so that the composed specimens can be examined and the structure can be provided as details. On the off chance that the detector is available but only composed characters or symbols are taken as input.

**3.1.6.2 Software Interfaces**

The SLD must provide interfaces with applications for

* The program that Admin uses.
* The program used by various users is

**3.1.6.3 Communications Interfaces**

As already mentioned, an Internet connection is obviously important for a complete service, but for your network adapter, you do not need any special configuration.

**3.1.7 System Features or Functional Requirements**

**3.1.7.1 Introduction**

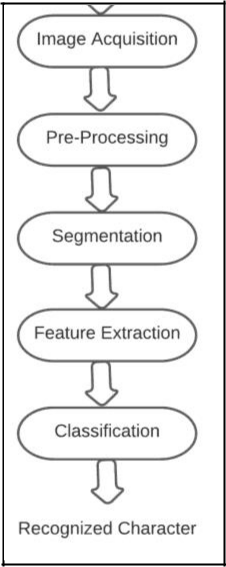
The aim of this venture is to upgrade a grouping calculation to interpret sign language characters and words. Example recognition has been shown to work the best for all grouping problems reliably. Subsequently, the scope of the undertaking also included the distinctive classifiers and blends techniques in the rudimentary analysis and assessed the alerts about their execution in this particular issue of digital sign language detection.

**3.1.7.2 Input**

Character, Digit, order, pictures

Prior to using the system must be thoroughly prepared and a customer understands the basic framework. When the structure is packed, you need to spare the structure before leaving, so the framework is stacked for further use. The standard has also been improved.

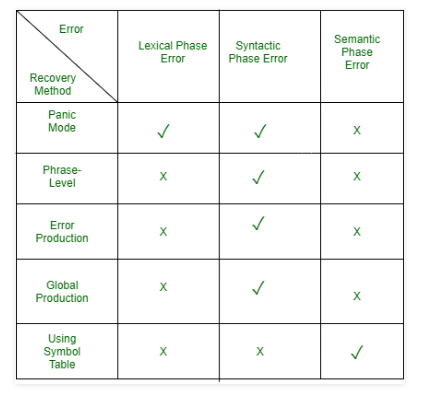
**3.1.7.3 Processing**

 Fig iii

**3.1.7.4 Error Handling**

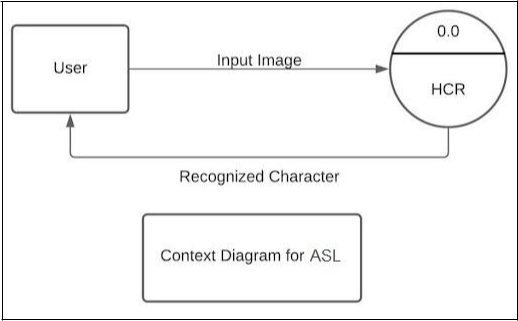
**The monitoring of errors is a very crucial part of developing any kind of application or website because it is very important to keep a check on the important and higher level prospects of the developed programs as well as the design of the application. Integrating everything together while handling of errors according to the input given by the user and to check if it is giving out the right output is considered one of the major aspects of error handling. From earlier classes, even while practicing C language we have known an approach called Try-Catch error handling method which can be used in any methodology or language to keep a check on the amount of errors and if the program is working efficiently according to our expectations or not.**

Expectation and Try Catch is used to monitor the handling of errors. If the user enters incorrect characters, the machine detects it and shows errors to verify errors in character input.



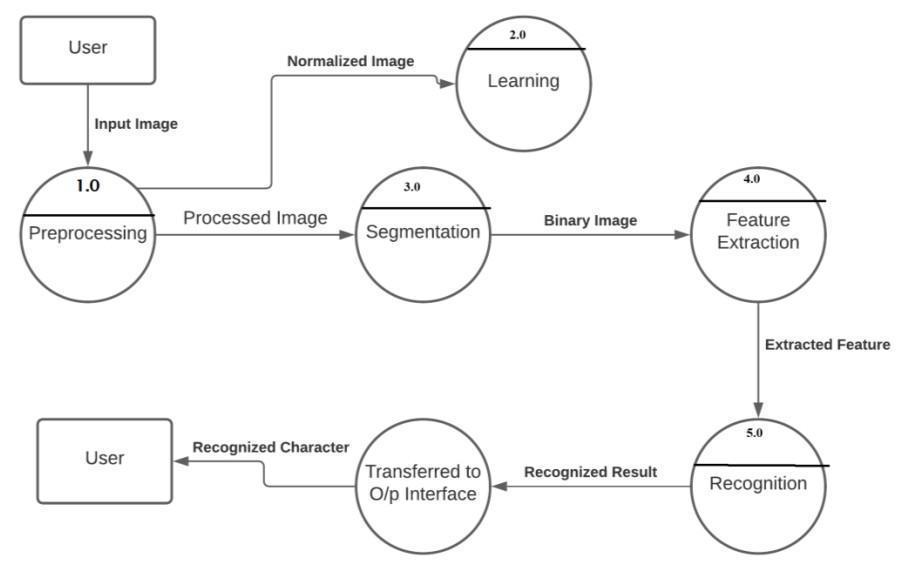
**3.2 Flowcharts/DFDs/ERDs**

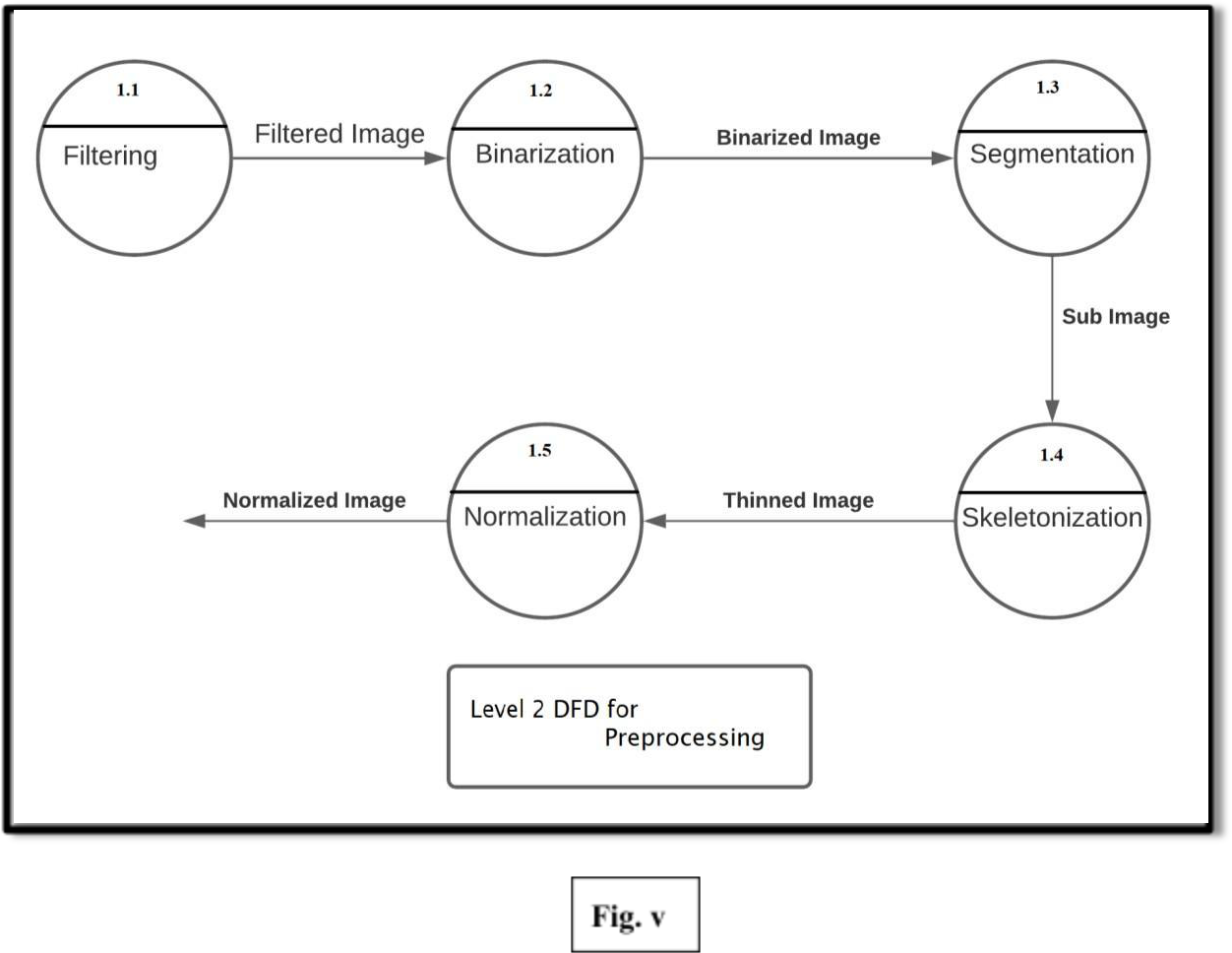
**3.2.1 DFDs**

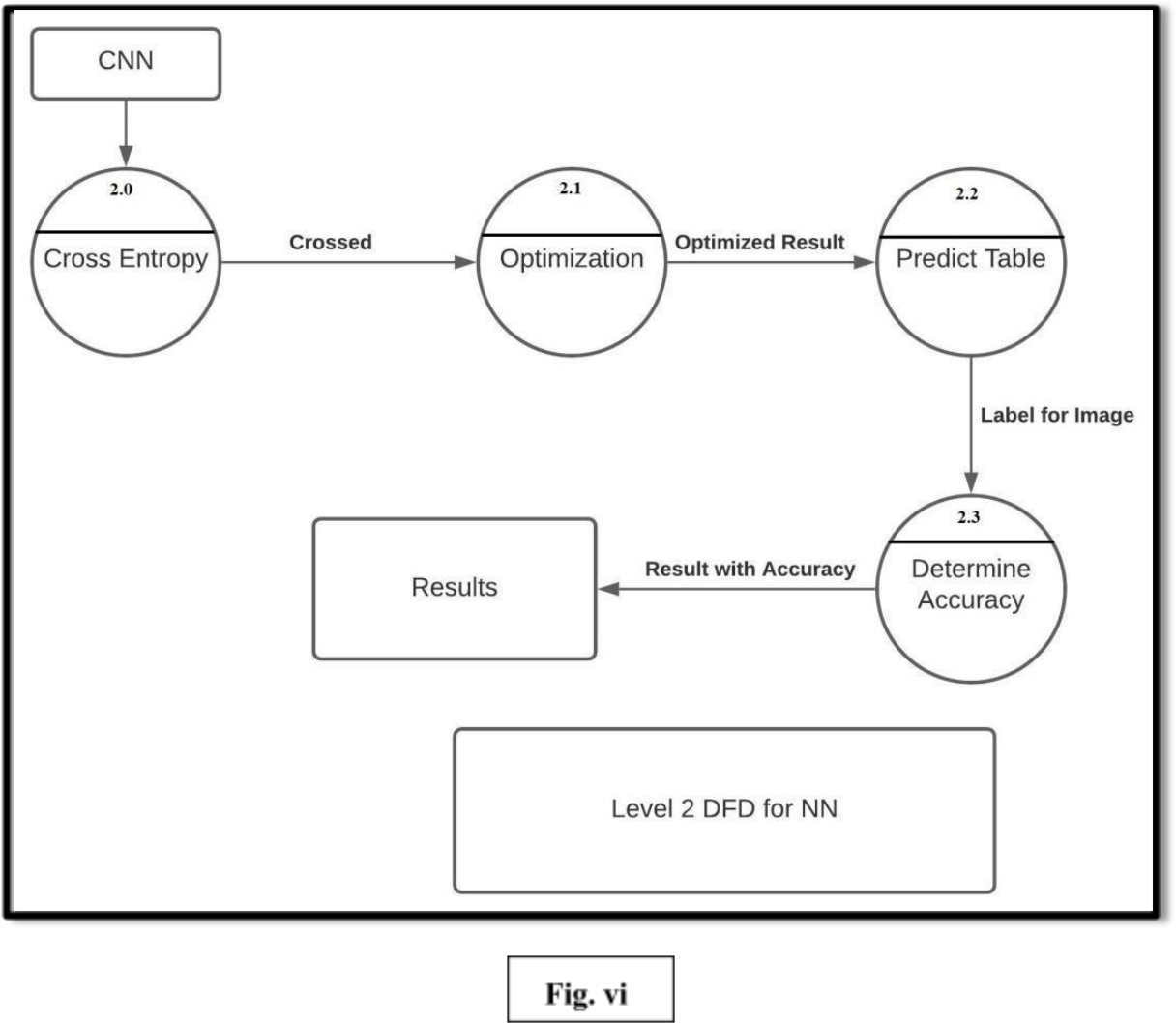


**Fig iii**

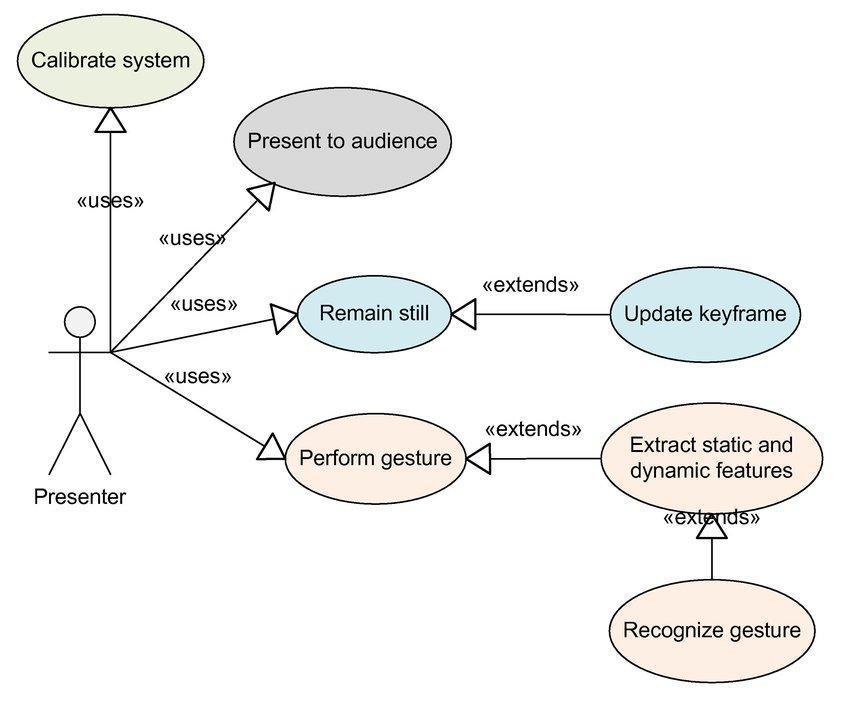
**Fig iv**







**3.2.2 Use Case Diagram**



**3.3 Design and Test Steps/Criteria**

**3.3.1 Process Model**

A process is grouped into a model of the same kind by the Process Model. As a consequence, a model describes a mechanism on a type-level basis. Even though the paradigm has now reached the type stage, it is still a process of instantiation. The same method model is often used to create multiple iterations and has various instantiations. A system model should be used to prescribe how tasks can be carried out concerning the currently taking place.

The objective of a model is as follows:

* **Descriptive:**
  1. Keep track of what occurs during a procedure.
  2. Consider an outside expert's perspective who examines how an operation is carried out and determines whether changes are to be made to make it more successful or reliable.
* **Prescriptive:**
  1. Definition of the procedures needed including how they're being performed.
  2. Set laws, procedures, and patterns of action that will contribute to the desired performance of the process is applied. It can vary between strict adherence and fluid guidance.
* **Explanatory:**
  1. Provide details on the rationale behind such methods.
  2. Centered on logical reasoning, analyzing, and comparing various potential courses of action.
  3. Make a strong connection between both the procedures and the standards which that model would meet.
  4. Also before the positions at which tracking data can be obtained.

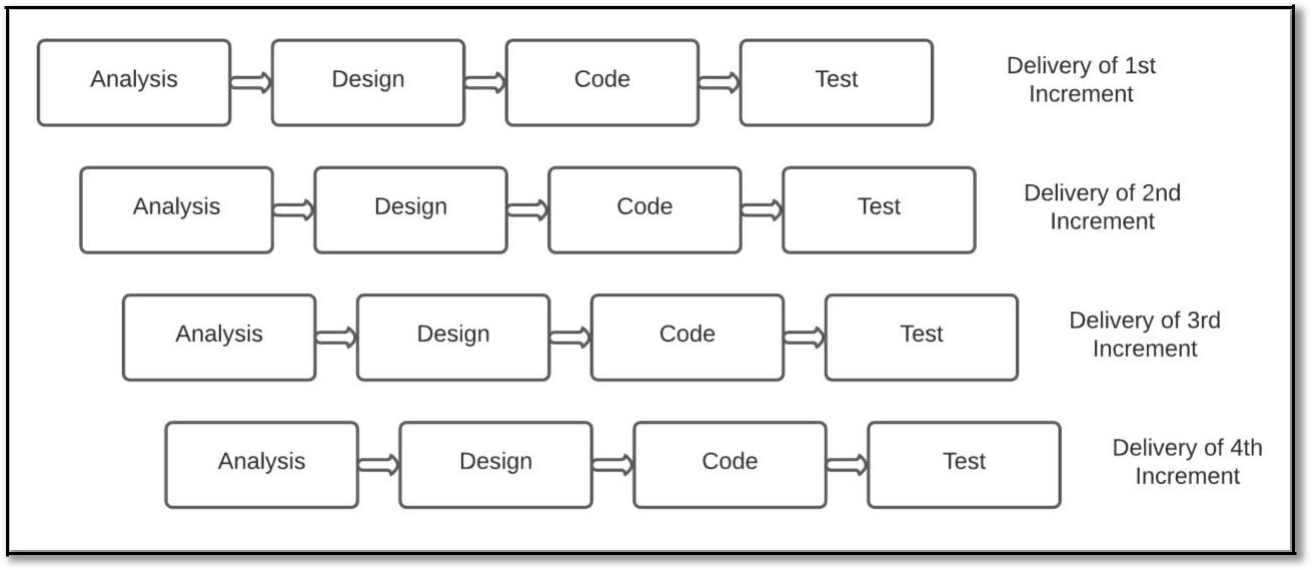
**3.3.1.1 Incremental Model**

The loop model in our method is seen as an incremental solution. (Pictured) On the basis of the design and implementation of the project is chosen the S/w engineering process model. We have chosen an Incremental Model for our project.

A small collection of specifications are enforced easily and distributed to the authority/customer using the Incremental model.

* Changed & extended demands can be added step by step.
* It combines elements with the iterative prototyping theory of the linear sequential paradigm.
* A deliverable increment of the S/w is generated by each linear sequence.
* The Linear Sequence is divided into four sections:-

1. **Analysis:** Device & software specifications are reported and reviewed.
2. **Design:** Includes four software attributes: Data structure, S/w Architecture, representation of the interface & procedural information.
3. **Coding:** This step is used to convert the design into machine code.



1. **Testing:** Works with S/w logical internals and guarantees that all declarations are right to detect all secret errors.

**Advantages of Incremental Model:**

* Generates S/w function rapidly & early during the life cycle.
* More versatile & less expensive for changing specifications.
* Easier for checking & debugging
* Customers will react to each designed product.

**Why is the Incremental Approach used?**

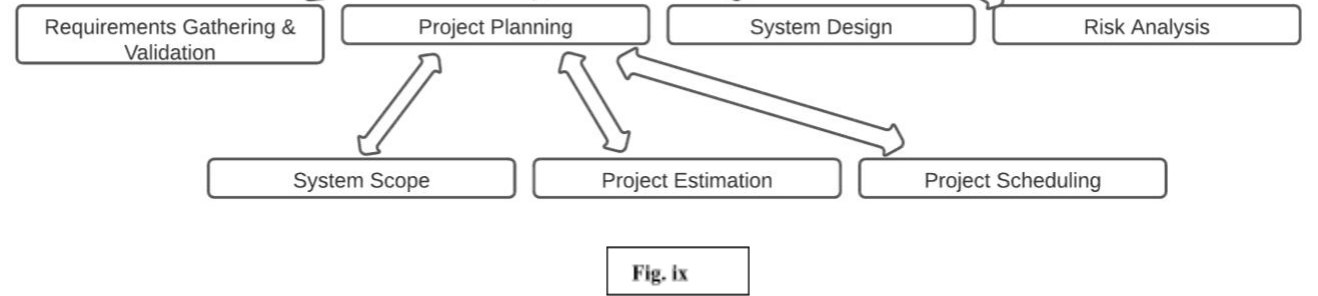
In order to boost the project's performance and usability, the key aim of using the model is to add additional features to the current modules. Using this model, we will adapt to changing consumer needs, which helps to expand the project in a very short period. The next increment in the previous raise incorporates input from consumers and several extra requirements. The process is replicated before the project is completed.

**Characteristics of Incremental Model:**

1. These models allow the rapid implementation and delivery of a new set of industry requirements to clients and then updating and expanding functionality step by step.
2. Each increment generates the commodity sent to the consumer and proposes certain adjustments and increments that differ with certain extra criteria compared to previous ones.
3. The radical model prevents the initiative from being completed all at once. This is useful for designing and checking components, enabling the project to be modularized for easier management.

Ultimately, the growth of the project in increments is easier. We will create a working prototype form 1 with only core tasks and then in subsequent increments, expand on this layout. Splitting the entire system into separate priority groups will serve to reduce system complexity.

**3.3.2 Breakdown Structure (Modules\_Analysis)**



* **Communication:**

The phase of product creation begins with user and developer interactions. We also gathered the project-related specifications according to work requirements.

* **System Design:**

A process model that is used in the implementation of the system. This activity also determines the Breakdown Structure (Modules). In the Breakdown Structure, various components used in the framework are shown.

* **Project Planning:**

Full calculation and timing of the entire timeline diagram for project development and for monitoring are included. Tasks are often expected to identify tools, timeline, and other details relevant to the project.

* **Modeling (Analysis & Design):**

It entails a thorough review of specifications and project planning. In the analysis of demands, system analysis is done in accordance with customer requirements, and what the start of the system will be in which direction it moves and what the destination will be is provided by the analysis process. In architecture, device design takes place according to research.

**3.3.2 Breakdown Structure (Modules\_Implementation)**

The proposed work is split into the following modules, as shown in Fig:

* + Image Preprocessing
  + Segmentation
  + Feature Extraction
  + Classification

1. **Image Preprocessing**

Various detection algorithms are used to process the videos,images such as sign detection.

**2. Feature Extraction**

Binary glyphs are created until the character is segmented, and the sum of the values of each row and column is computed as a function.

**3. Classification**

We are seeking to create a training model and validate NN within this step.

**Algorithms and Pseudo Code**

The following is an overview of the different development measures that have been taken to achieve the ultimate aim of our project.

**Module 1: Image Processing**

Processing involves phases, required to create a segmentation-appropriate type of the input image. The hue of the image is translated to a Greyscale. The graphic, meaning a black image in white, is converted into binary images.

**Algorithm:** Gray-scale conversion

**Input:** Different signs as an input

**Output:** Recognition in the form of text and speech.

**Phase 1:** Starting off

**Phase 2:** Select the Image Input Text

**Phase 3:** Repeat from x=0 to Image Width

**Phase 4:** Repeat to the height of the image with y=0

**Phase 5:** Extract the RGB value as an RGB value for each pixel

**Phase 6:** Set the Computed Gray Level Pixel

**Phase 7:** View the Picture GrayScale and using the graphical user interface to detect the sign.

**Phase 8:** Make a stop

**Module 2: Feature Extraction**

Since each character has been separated, the image can be resized to 15 x 20 pixels. The accuracy of recognition is better if the features are accurately derived. The 15 x 20 indicates 300 pixels, as it does with the feature vector.

**Module 3: Training and Recognition**

As an input, the features extracted from previous modules are given.

**Recognition**

Offer a pattern to recognize the symbols and test it using various test cases.

**Module 1 :-**

The following improvements were made:-

* Place image into imgSize's intended img, transpose for TF and normalize gray-values.
* Extend the dataset by stretching the images at regular intervals.
* If the line text is low contrast and the line width is thin, just improve the contrast.
* Create a target image and insert the sample image.
* Initialization
* Computation

**Module 2 :-**

The following improvements were made:-

Filenames and data routes

A Sample from the dataset

A batch of photos and ground truth texts

loader for the dataset at the specified venue, preprocessing images and text depending on parameters separate into training and validation sets

**Module 3 :-**

The following improvements were made:-

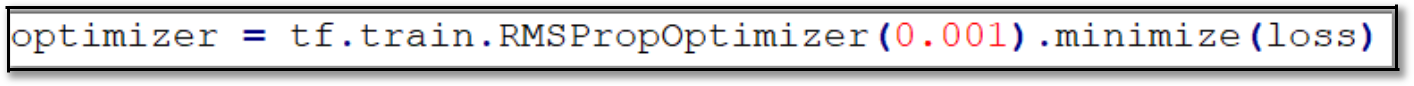
* Model Constants
* Configure optimizer to train NN
* Tensor-Flow initialization
* Return the results.
* Initialization of the Tensor-Flow

**Implementation:-**

The user trains the model on the basis of a processing module which trains the available signs.Then, the user trains it using may epochs which thus achieves about 99 percent of accuracy using static images but the same is not possible if we predict them dynamically.Then,in the test module,all the trained words are tested.

**Training**

To train the NN, the batch part loss values' mean is fed into an optimizer like RMSProp.



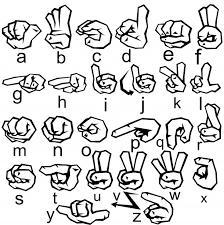
**SYSTEM DEVELOPMENT:**

**Dataset**

* There are 60,000 photographs of training.
* It does have a 10,000 picture test range
* The images are black and white
* There is a box of 28x28 pixels

**ASL dataset**

* This has various photographs of all the alphabets in training.
* The pictures are black and white
* There is a box of 28x28 pixels



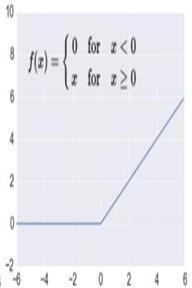


**ASL dataset sample image**

**Activation function**

**RELU function**

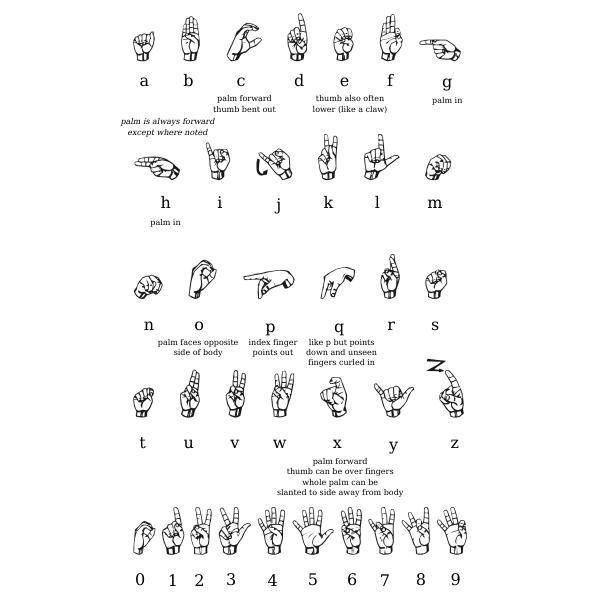
They use our network to solve problems with non-linear properties. They convert a node input signal into a NN signal



**Design of a system for ASL dataset**

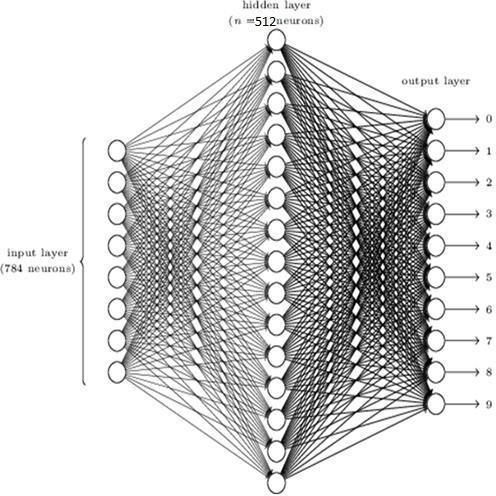
The data set is split into two sections: preparation and research. The training set contains 60000 images of the 28\*28 grid standard, while the test set contains 10000 images of the same dimension.

The first 25 images in the training set are as follows:



**First 25 images of the training set for ASLdataset**

​



**Optimizer**

Adam's optimizer is employed. The Adam Optimizer is a stochastic down gradient variant. The term "adaptive time estimation" is abbreviated as Adam. For gradient-based convex optimization algorithms, it is currently the better solution. Hyperparameters have the disadvantage of being easier to tune.

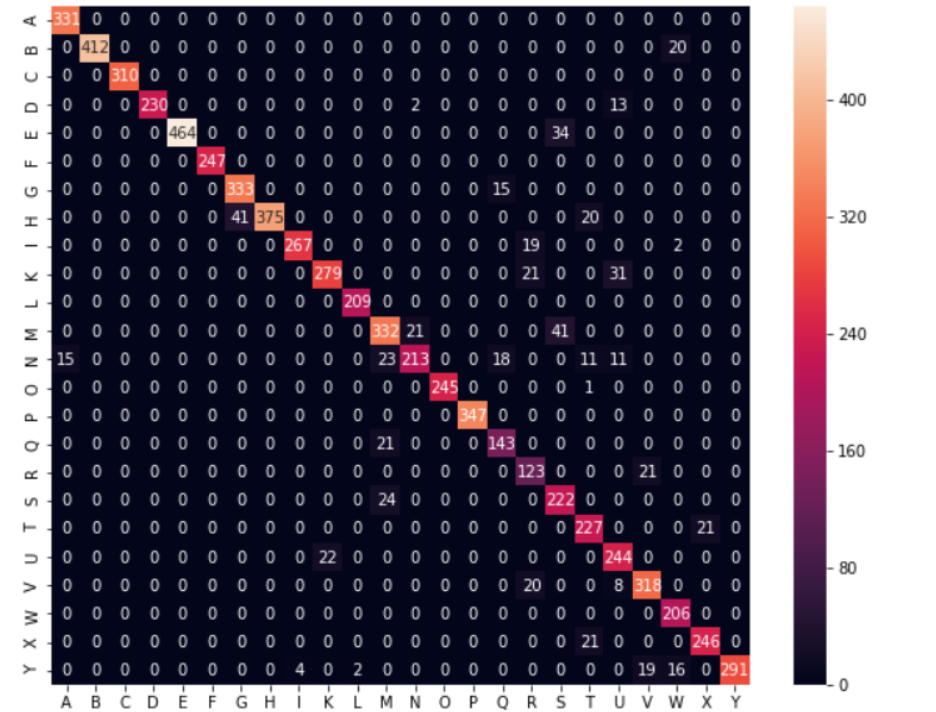
**Loss function**

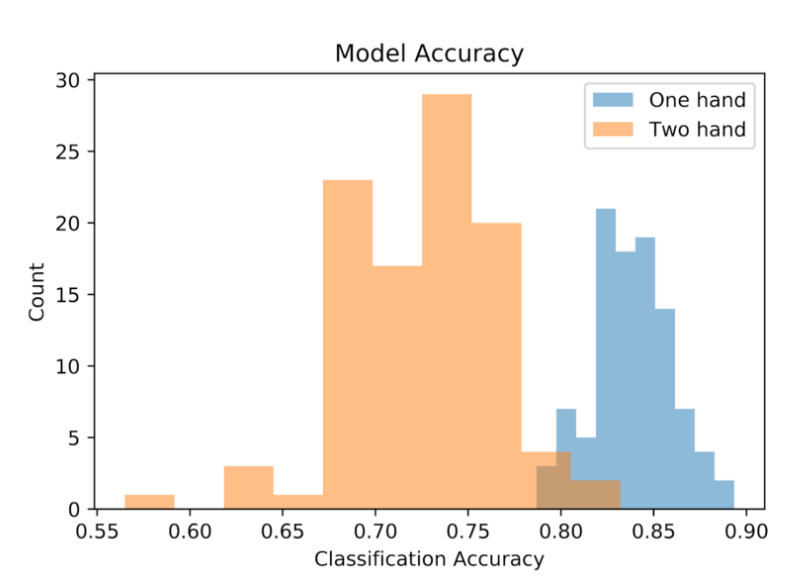
* logarithmic loss function is used to calculate the loss (sparse categorical cross-entropy). The aim of each training process is to reduce this failure function to the greatest extent possible. This loss function is used where the groups we're calculating are mutually exclusive.

**Metric of evaluation**

Our model's efficiency is measured using precision measurements, also known as classification accuracy.

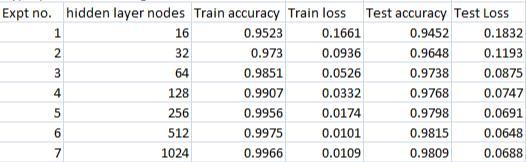
**ASL DATASET USAGE GRAPH**

****

****

**PERFORMANCE ANALYSIS**

**Hyperparameter tuning of NN for ASL dataset**

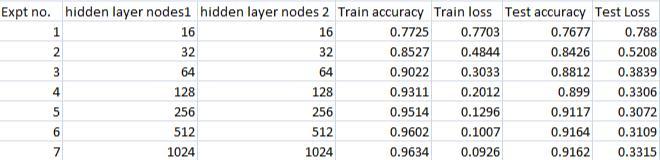


Data Collection NN Hyperparameter Tuning

**Tuning of NN for ASL dataset**



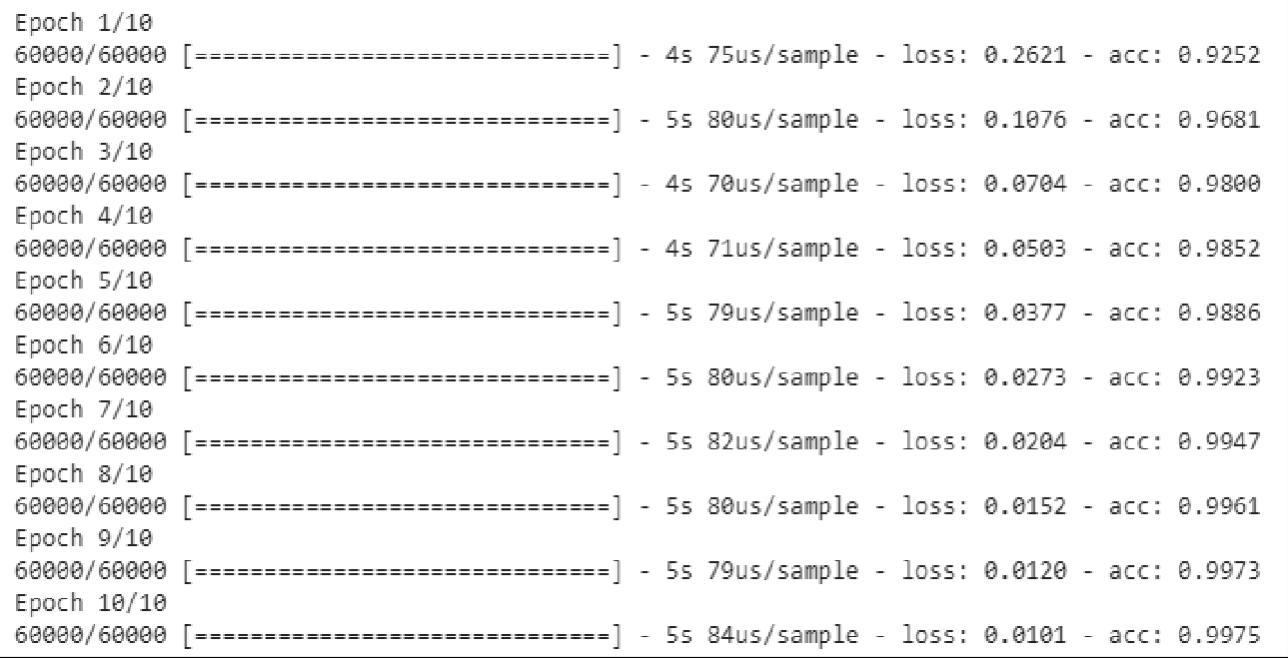
If the number of hidden layers is 16, 32, 64 and so on. That is the below case:



**Results of the ISL dataset using**

**Neural Network Training step**

The training phase:



Dataset train failure and train accuracy. The precision is 0.9975 and the loss is 0.0101.

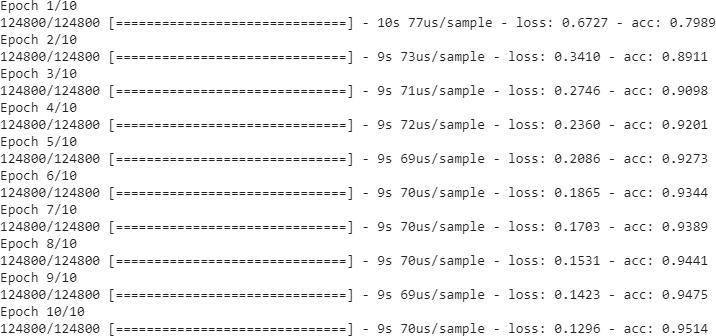
**Testing step**

In the research phase, 10,000 photographs are used.

The precision is 0.9815 and the loss is 0.0648.

**Results of Sign language detection**

The preparation phase:



For the ISL dataset, train failure and train accuracy were calculated. The precision is 0.9514 and the error is 0.1296.

**Testing step**

In the research phase, many video sequences are

used. The precision is 0.9417 and the loss is

0.3072.

**Testing Process**

**3.4.1 Software Testing**

**3.4.1.1 Introduction:-**

The role of software testing is to ensure that programs are efficient and accurate. Software testing is an observational science investigation conducted to provide consumers with information regarding a product's quality in the environment in which it is intended to function. This can include but is not limited to running a program or application to detect errors.

**3.4.2 Unit Testing:-**

In this case, each module is evaluated independently. The standards for defining unit test modules were selected to identify modules that have key functionality. A module may be either an individual or a method.

The unit testing functions that will be tested are as follows:

Choose the handwritten document's scanned input image.

* Preprocessing can be used.
* We are using Feature Extraction to extract features.
* Take out a sign character.

**3.4.3 Integration Testing:-**

Relevant components are integrated and analyzed as a group during integration planning. Integration testing takes unit-tested elements like data, groups them into larger aggregates, applies integration test plan tests to those aggregates, and produces the integrated testing framework.

**3.4.4 Validation Testing:-**

At the start or end of the production process, this approach is used to determine if the software satisfies the specified specifications.

**3.4.5 GUI Testing:-**

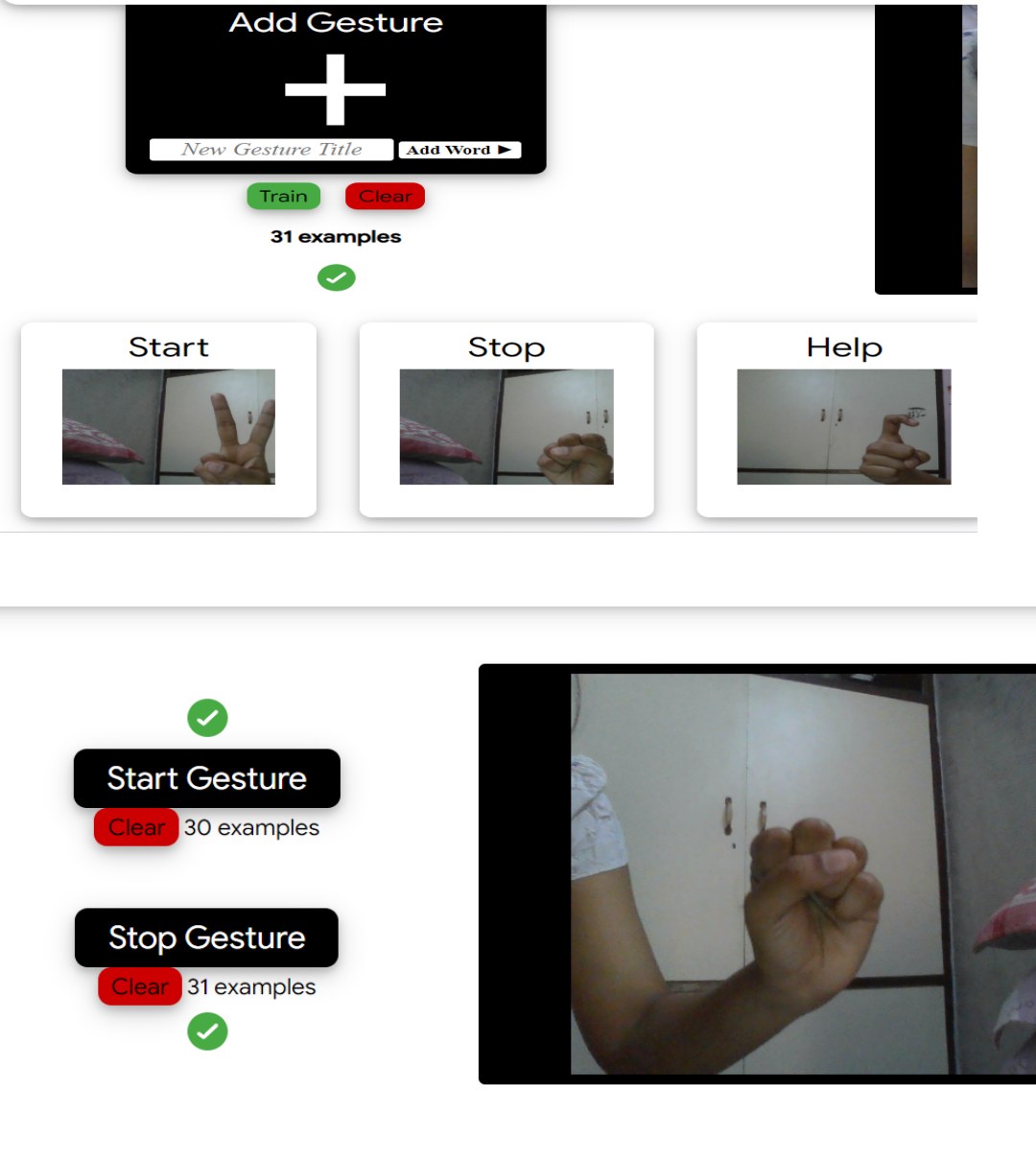
GUI testing is the process of examining a product's graphical user interface to ensure that it complies with standards, such as retaining navigation between icons/buttons with source code.

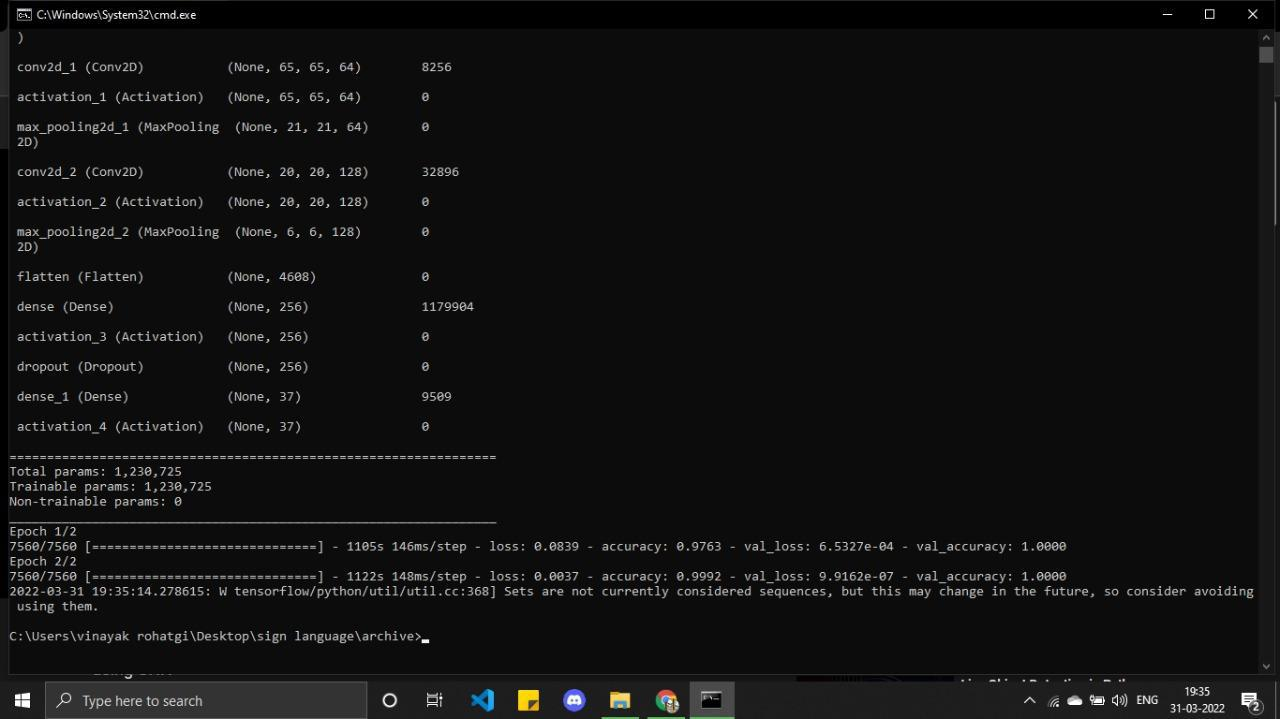
**Test-Cases:-**

|  |  |  |
| --- | --- | --- |
| Use Case ID | 1 |  |
| Test Case Name | Check Image Format | |
| Test Case Description | Valid Image format must be processed to continue | |
| Steps | 1. | Open necessary code |
|  | 2. | Give input as video sequence with proper format |
| Expected Results | Input video accepted and display | |
| Actual Results | As expected | |

|  |  |
| --- | --- |
| Use Case ID | 2 |
| Test Case Name | Start Training |
| Test Case Description | Start Training Characters |
| Steps | 1. Start training of characters |
| Expected Results | Training should be done and accurate |
| Actual Results | As expected |

|  |  |
| --- | --- |
| Use Case ID | 3 |
| Test Case Name | Sign character’s detection from the video |
| Test Case Description | Sign character from the video should be displayed |
| Steps | 1. Accept any video sequence. |
| Expected Results | Application should be able to display recognized sign from video |
| Actual Results | As expected |

**Chapter4:RESULTS/OUTPUTS**



**Chapter 5: Conclusion**

The architecture of the sign language detector and how it recognizes text in images is discussed. Loss function and accuracy are also calculated. We also spoke about how data augmentation techniques can increase accuracy and allow the development of sign language detection system with detailed code. Finally, proposals for further changes to this system were made.

**5.1 Further Improvement:-**

* Sign language can also be done using face gestures that are not yet popularized.
* Image detection in real-time can also be improvised using many modern-day algorithms like LDA and PCA.
* How about using an end-to-end solution to localize and interpret sign language on the website along with a pre-trained dataset for the ease of use. The object localization concept is used to localize different types of signs and classification

**Chapter 6: References**

1. “Sujitshree, C Venugopal- Irregular Pattern Recognition Using Artificial Neural Network, International Journal of Advanced research in Computer Science and Management Studies Volume 3 Issue July 2015.”
2. “Nikita Sarkar, Harsh Kishore - Handwritten Charachter Recognition Using Method Filters-International Journal of Research in Engineering and Technology pISSN: 2321-7308 ”
3. “J Pradeep, E Srinivasan, S Himavathi.- Diagonal Feature Extraction Based Handwritten Character system using Neural Network. International Journal of Computer Application Volume 8 No October 2010.”
4. “Anilkumar Holambe, Dr Ravinder C Thool, Dr S.M Jagade - Printed and Handwritten Character & Number Recognition of Devanagari Script using Gradient Features. International Journal of Computer Application Volume 2 No- June 2010.”
5. “Dayashankar Singh, Sanjay Kr Singh, Dr Maitreyee Dutta. - Hand Written Character Recognition Using Twelve Directional Feature Input and Neural Network, International Journal of Computer Application, Volume 1- No 3”
6. “Muhammad Naeem Ayyaz, Imran Javed and Waqar Mahmood - Handwritten Character Recognition Using Multiclass SVM Classification with Hybrid Feature Extraction, Pak J Engg & Appl Sci Vol 10 Jan 2012”
7. “Rakesh Kumar Mandal, N R Manna.- Hand Written English Character Recognition using Column-wise Segmentation of Image Matrix (CSIM)”
8. “Aini Najwa Azmi, Dewi Nasien (2014), “Feature Vector of Binary Image using Freeman Chain Code (FCC) Representation based on Structural Classifier”, 2014”
9. “D. Nasien, H. Haron, and S. S. Yuhaniz (2010), “Support Vector Machine (SVM) for English handwritten character recognition,” 2010 2nd Int. Conf. Comput. Eng. Appl. ICCEA 2010, vol. 1, pp. 249–252, 2010.”
10. “S. B. Hallale and G. D. Salunke (2013), “Twelve Directional Feature Extraction for Handwritten English Character Recognition, ” no. 2, pp. 39–42, 2013.”
11. “J.Pradeep, E.Srinivasan, S.Himawathi,” Perfomance analysis of hybrid feature extraction technique for recognizing english handwritten character”,978-1- 4673-4805-8/12/2012IEEE.”
12. Anshul Gupta, Manish Srivastva, Chitraleka Mahanta,”Offline handwritten character recognition using neural network”, 2011 International conference on computer application and industrial electronics.
13. “Aiquan yuan, Gang Bai, Lijing Jiao, Yajie Liu, Offline handwritten english character recognition based on convolution neural network, 2012, 10th IAPR International workshop on document analysis system.”
14. “N.M. Noor, M.Razaz and P.Manely Cooke,Global geometry extraction for fuzzy logic based handwritten character recognition ,ICPR,2004.”
15. “Rajib Lochan Das, Binod kumar Prasad and goutam Sanyal,HMM based offline handwritten writer independent english character recognition using global and local feature extraction, International Journal of Computer Application,Volume 46-No 10,May 2012.”
16. “Huiqin Lin, Wennuan Ou,Tanglin Zhu,The research of algorithm for handwritten charcter recognition in correcting assignment system,”.
17. D. K. Patel, T. Som, Manoj Kumar Singh, “ Multiresolution technique to handwritten english charcter recognition using learning rule and Euclidean distance matric,”.
18. Huihung Zhao, Dejian Zhou and Zhaohua Wu, “SMT product charcter recognition based on BP neural network,” 2010 Sixth international conference on natural computation(ICNC 2010).
19. Peng Xu,” Research of standard and handwritten english letter recognition system based on the PSOBP neural network,” IEEE,2011.
20. P.M. Patil, P.S. Dhabe, U.V. Kulkarni and T. R. Sontakke,”Recognition of handwritten charcter using modified fuzzy hyperline segment nueral network,”.
21. Singh, Sameer, Mark Hewitt,“Cursive Digit And Character Recognition on Cedar Database”, Pattern Recognition, 2000. Proceedings. 15th international conference on. Vol. 2. IEEE 2000.
22. Vijay Patil, Sanjay Shimpi,“Handwritten English Character Recognition Using Neural Network”, Elixir Computer Science Eng 41, pp. 5587-5591, 2011.
23. Mahmud S. Alkoffash, Mohammed J. Bawaneh, HasanMuaidi, ShihadehAlqrainy, MuathAlzghool,“A Survey of Digital Image Processing Techniques in Character Recognition”, International Journal of Computer Science and Network Security (IJCSNS), 14(3), (2014) 65.
24. Ankit Sharma, Dipti R Chaudhary,“Character Recognition Using Neural Network”, International Journal of Engineering Trends and Technology (IJETT), 4.4, pp. 662-667, 2013.
25. Anshul, Manisha Shrivastava, Chitralekha Mahanta. “Offline handwritten character recognition using neural network.” Computer Applications and Electronics (ICCAIE), 2011 IEEE International Conference on, IEEE, 2011.
26. F. Kimura, N. Kayahara, Y. Miyake, M. Shridhar. “Machine And Human Recognition Of Segmented Characters From Handwritten Words”, 4th International Conference On Document Analysis And Recognition (ICDAR ’97), Vol. 2. IEEE, 1997, pp. 866-869.
27. Yamada, Hirobumi, Yasuaki Nakano,“Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis”. IEICE Transactions on Information and Systems, Vol. E79-D, 1996, pp. 464-470
28. “Huiqin Lin, Wennuan Ou,Tanglin Zhu,The research of algorithm for handwritten charcter recognition in correcting assignment system,”.
29. D. K. Patel, T. Som, Manoj Kumar Singh, “ Multiresolution technique to handwritten english charcter recognition using learning rule and Euclidean distance matric,”.
30. Huihung Zhao, Dejian Zhou and Zhaohua Wu, “SMT product charcter recognition based on BP neural network,” 2010 Sixth international conference on natural computation(ICNC 2010).
31. Peng Xu,” Research of standard and handwritten english letter recognition system based on the PSOBP neural network,” IEEE,2011.
32. <https://www.researchgate.net/publication/262187093_Sign_language_recognition_State_of_the_art>
33. “Rajib Lochan Das, Binod kumar Prasad and goutam Sanyal,HMM based offline handwritten writer independent english character recognition using global and local feature extraction, International Journal of Computer Application,Volume 46-No 10,May 2012.”
34. “Huiqin Lin, Wennuan Ou,Tanglin Zhu,The research of algorithm for handwritten charcter recognition in correcting assignment system,”.
35. D. K. Patel, T. Som, Manoj Kumar Singh, “ Multiresolution technique to handwritten english charcter recognition using learning rule and Euclidean distance matric,”.