FINAL YEAR MINOR PROJECT REPORT (PROJECT STAGE – I)

SKETCH RECOGNITION

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Computer Science and Engineering from Rajasthan Technical University, Kota

MENTOR:

Mr. Basant Agrawal
(Dept. of Computer Science & Engineering)

SUBMITTED BY:

Archit Mathur	**(15ESKCS033)**
Aviral Agarwal	**(15ESKCS040)**
Charchit Sharma	**(15ESKCS050)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SWAMI KESHVANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN
Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2018-19

SWAMI KESHVANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN

Ramnagaria (Jagatpura), Jaipur – 302017

CERTIFICATE

This is to certify that Final Year Minor Project Report (Project Stage – II) entitled "**SKETCH RECOGNITION**" has been duly submitted by

• Archit Mathur	(15ESKCS033)
• Aviral Agarwal	(15ESKCS040)
• Charchit Sharma	(15ESKCS050)

For partial fulfillment of the Degree of Bachelor of Technology of Rajasthan Technical University. It has been found satisfactory and hence approved for submission as Minor Project during academic session 2018-19.

MENTOR	**HEAD OF DEPARTMENT**
Mr. Basant Agrawal (Dept. of Computer Science & Engineering)	**Prof. Dr. C.M.Choudhary** (Dept. of Computer Science & Engineering)

ABSTRACT

In our day to day life, we come various objects and structures which are unknown to us. To identify and recognize that object, we can draw a sketch of that object. This image which is created may or may not be identified by a common person. We have designed a system which incorporates machine learning and can help us to identify that object through the sketch created.

DECLARATION

We hereby declare that the report of the project entitled “SKETCH RECOGNITION” is a record of an original work done by us at **Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur** under the mentorship of **Mr. Basant Aggarwal**This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology**(B.Tech) in the **Department of Computer Science & Technology.** It has not been submitted anywhere else, under any other program to the best of our knowledge and belief

ACKNOWLEDGEMENT

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization. We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor and project coordinator **Mr. Basant Aggarwal.** He has been a guide, motivator & source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We thank him for his co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to **Prof. C.M. Choudhary,** HOD, Department of Computer Science & Engineering, for facilitating, motivating and **s**upporting us during each phase of development of the project. Also, we pay our sincere gratitude to all the **Faculty Members** of Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur and all our Colleagues for their co-operation and support.

INDEX

UNIT – 1 PROJECT CHARTER

1.1 Problem Statement & Objective

Problem: Identification of various unknown objects in day to day life.

Solution: Instead of wasting our time and energy by asking around other people, we can get more accurate results through implementing technology and thus identify the object.

Objective

To eliminate the problems faced by people in recognizing unknown objects and various structures.

1.2 Proposed Logic or Algorithm/ Solution

Due to the various advancements in machine learning we are able to detect to a certain degree the features of what makes an object itself. With the power of convolutions, derived from convolutional neural networks, and long short-term memory, derived from recurrent neural networks, our project aims to eliminate the difficulties and thrive on the opportunities that arise in areas such as convict detection from witness' account, teaching drawing to kids, helping doctors understand mental illness of a person by their sketching sense etcetera.

1.3 Scope of the Project

Using sketch, various objects can be identified which can even include people, especially for the criminal recognition used by the police agencies. This can be very helpful in tracking the criminals, and help to bring rightful justice to people.

UNIT – 2 SYSTEM REQUIREMENT SPECIFICATION

Overall Description

This section and its subsections contain the description of the project components such as interfaces, performance requirements, design constraints, assumptions and dependencies etc.

Product Perspective

The application will be a Windows / Linux based, self contained and independent product.

System Interfaces

List each system interface and identify the functionality of the system (hardware and software both) to accomplish the system requirement and interface description to match the system.

User Interfaces

The application will have a user friendly and menu based interface. Following screens will be provided:

a. A Login screen to enter the username, password and role.

b. A screen for capturing and displaying information regarding subjects offered in each semester.

c. Following reports will be generated:

   - Students’ list report: Printable report generated to show the list of the students enrolled in a particular batch year.

   - Marks List Report: Printable report showing the subjectwise marks

   - Marksheet: A printable marksheet will be generated for each student of each semester.

Hardware Interfaces

Screen resolution of at least 800 x 600 pixels is required for proper and complete viewing of screens. Higher resolutions in wide-screen mode will be better for a better view.
Support for printer (dot-matrix / deskjet / inkjet / laser) is required. This implies that appropriate drivers should be installed and printer device should be connected for printing of reports and marksheets.

A network connection (internet / intranet) is required to make the web service accessible on other systems connected over the network.
Other hardware interface specifications are as follows

HARDWARE INTERFACES - CLIENT SIDE (Minimum)	
HARDWARE	SPECIFICATIONS
Intel Pentium 4 and Higher Processor	
QWERTY Keyboard (U.S. Design)	
USB 2.0 / PS2 Mouse	
Table 2.1 – Minimum Client Side Hardware Interfaces	

HARDWARE INTERFACES - SERVER SIDE (Minimum)		
HARDWARE	**RAM**	**DISK SPACE**
Intel Core i3 / i5 / i7 2.27 GHz and higher Or AMD 4XXX and higher	2048 MB	20 GB
Table 2.2 – Minimum Server Side Hardware Interfaces		

HARDWARE INTERFACES - CLIENT SIDE (Recommended)		
HARDWARE	**RAM**	**DISK SPACE**
Intel Core i3 / i5 / i7 2.27 GHz and higher Or AMD 4XXX and higher	1024 MB	1 GB
QWERTY Keyboard (U.S. Design)		

USB 2.0 Optical Mouse		

Table 2.3 – Recommended Client Side Hardware Interfaces

HARDWARE INTERFACES - SERVER SIDE (Recommended)		
HARDWARE	**RAM**	**DISK SPACE**
Intel Xeon higher Or AMD equivalent	4096 MB	40 GB

Table 2.4 – Recommended Server Side Hardware Interfaces

2.1.1.4 Software Interfaces

Any Microsoft Windows 7 and higher (Windows 7 / 8 / 8.1 / 10) or equivalent Linux based operating system with minimum kernel support 3.X.
Crystal Reports 8 for generation and viewing of reports
J2EE (JSP, Java Bean, Servlets, HTML, XML, AJAX) & jQuery for coding and developing of the application.

SOFTWARE INTERFACES (Minimum)		
Software Tool	Version	Purpose of Use
Operating system	Windows 7 and higher or Linux with Kernel 3.x and higher	Installation and operational platform
Web Browser	Internet Explorer 6 and other higher compatible	Access to the web application
Web Server	Apache Tomcat 6.0 and higher	Running the web application over internet / intranet
Database	MySQL 5.1.x	Running and linking the database over internet /

		intranet to the online web application

Table 2.5 – Minimum Software Interfaces

SOFTWARE INTERFACES (Recommended)		
Software Tool	Version	Purpose of Use
Operating system	Windows 8 & higher or Linux with Kernel 4.x & higher	Installation and operational platform
Web Browser	Internet Explorer 11 and other higher compatible	Access to the web application
Web Server	Apache Tomcat 8.0 and higher	Running the web application over internet / intranet
Database	MySQL 5.5.x	Running and linking the DB on internet / intranet

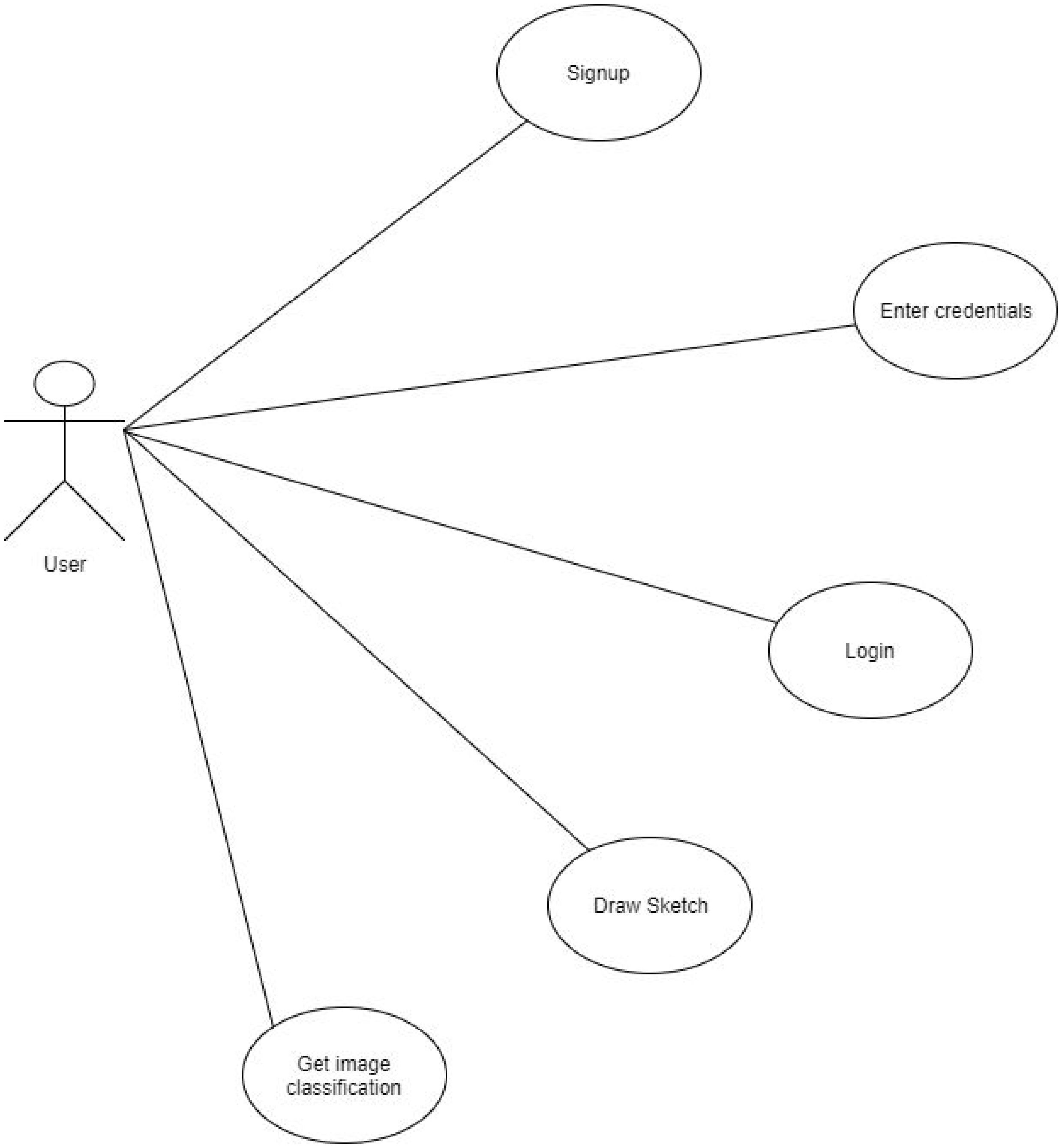
Table 2.6 – Recommended Software Interfaces

UNIT– 3 SYSTEM DESIGN SPECIFICATION

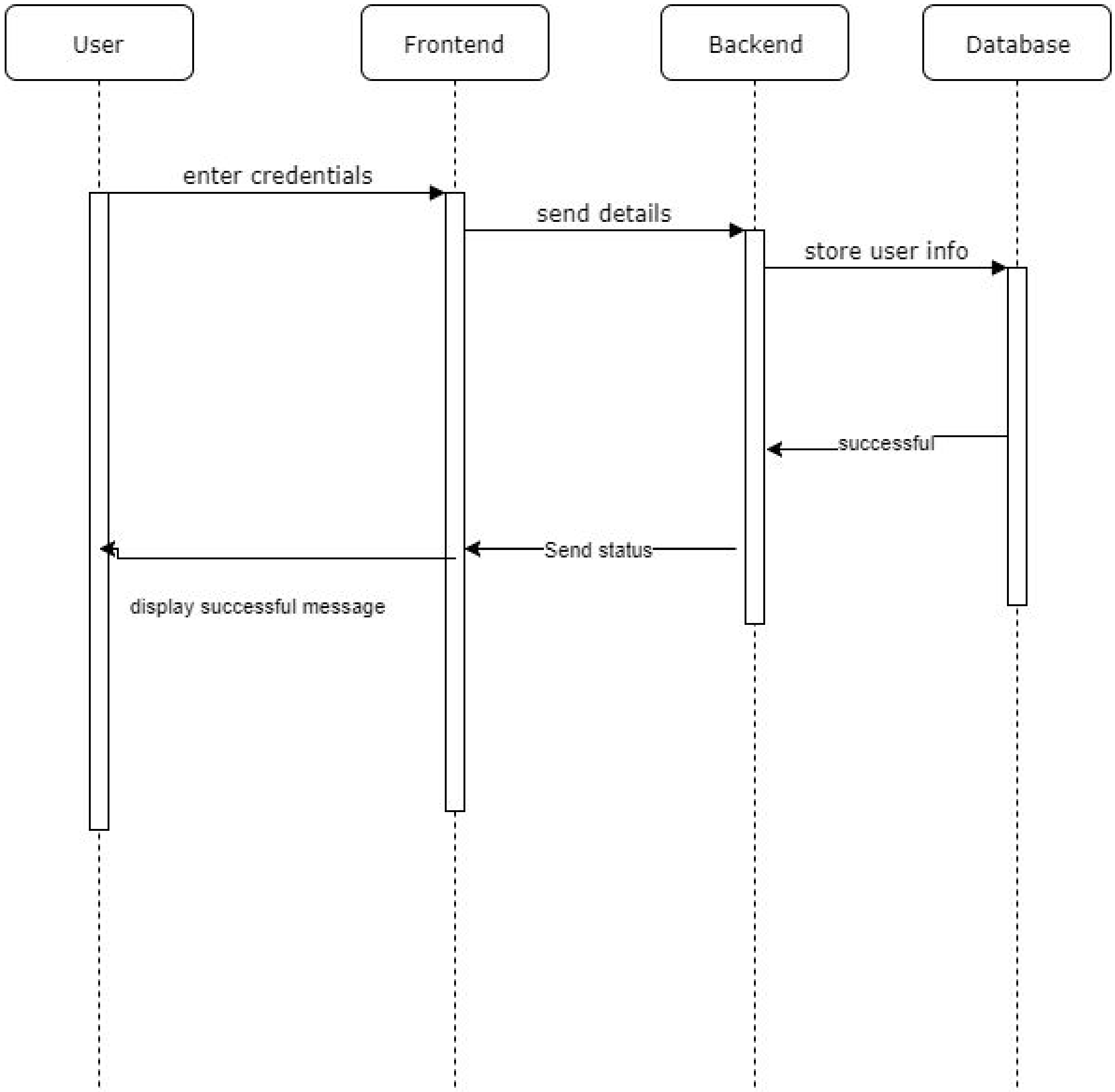
3.1 System Architecture

System architecture presents the schematic view of the complete system along with its major components and their connectivities. The overall architecture of the proposed system will be as follows.

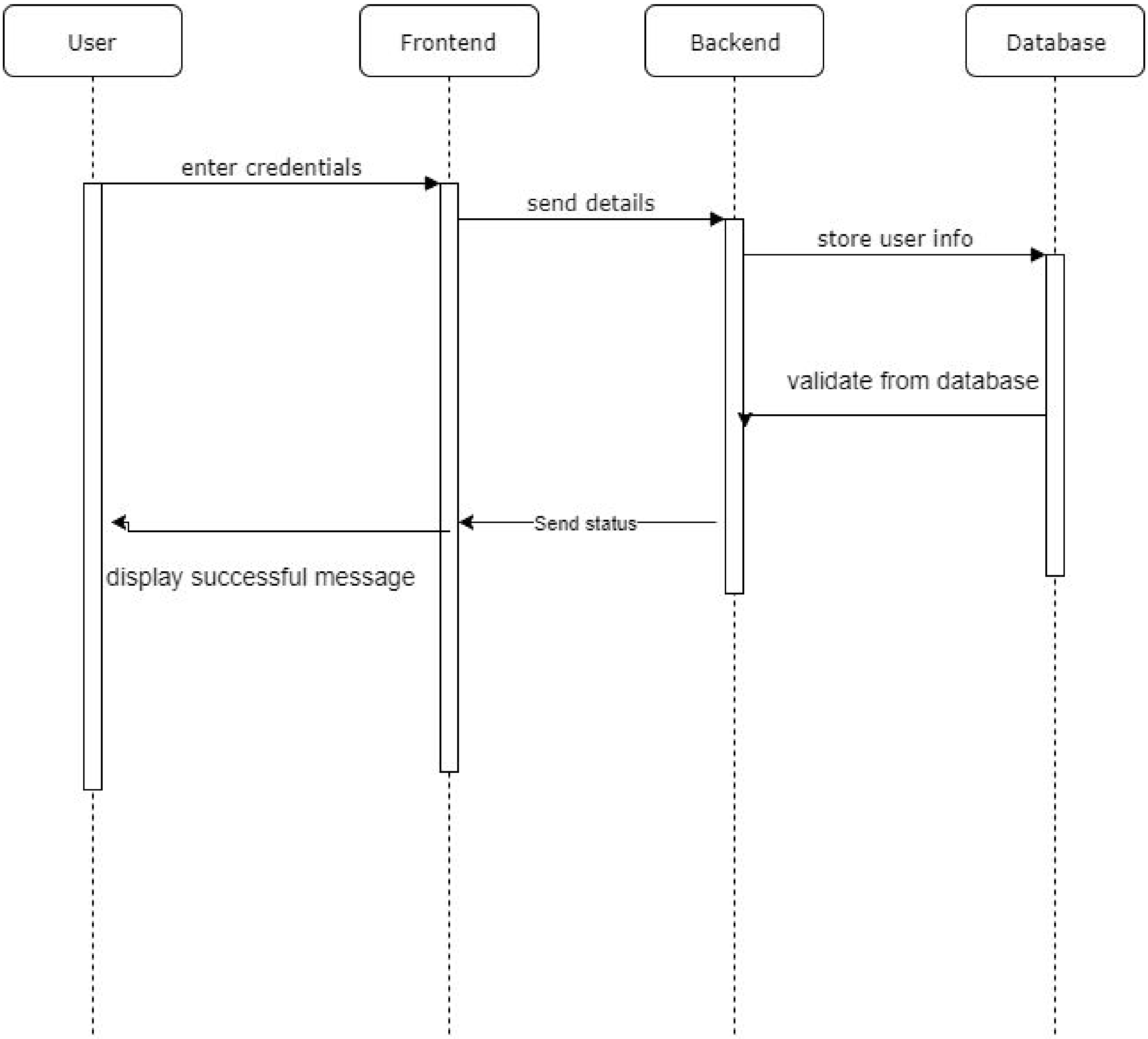
Use-case diagram



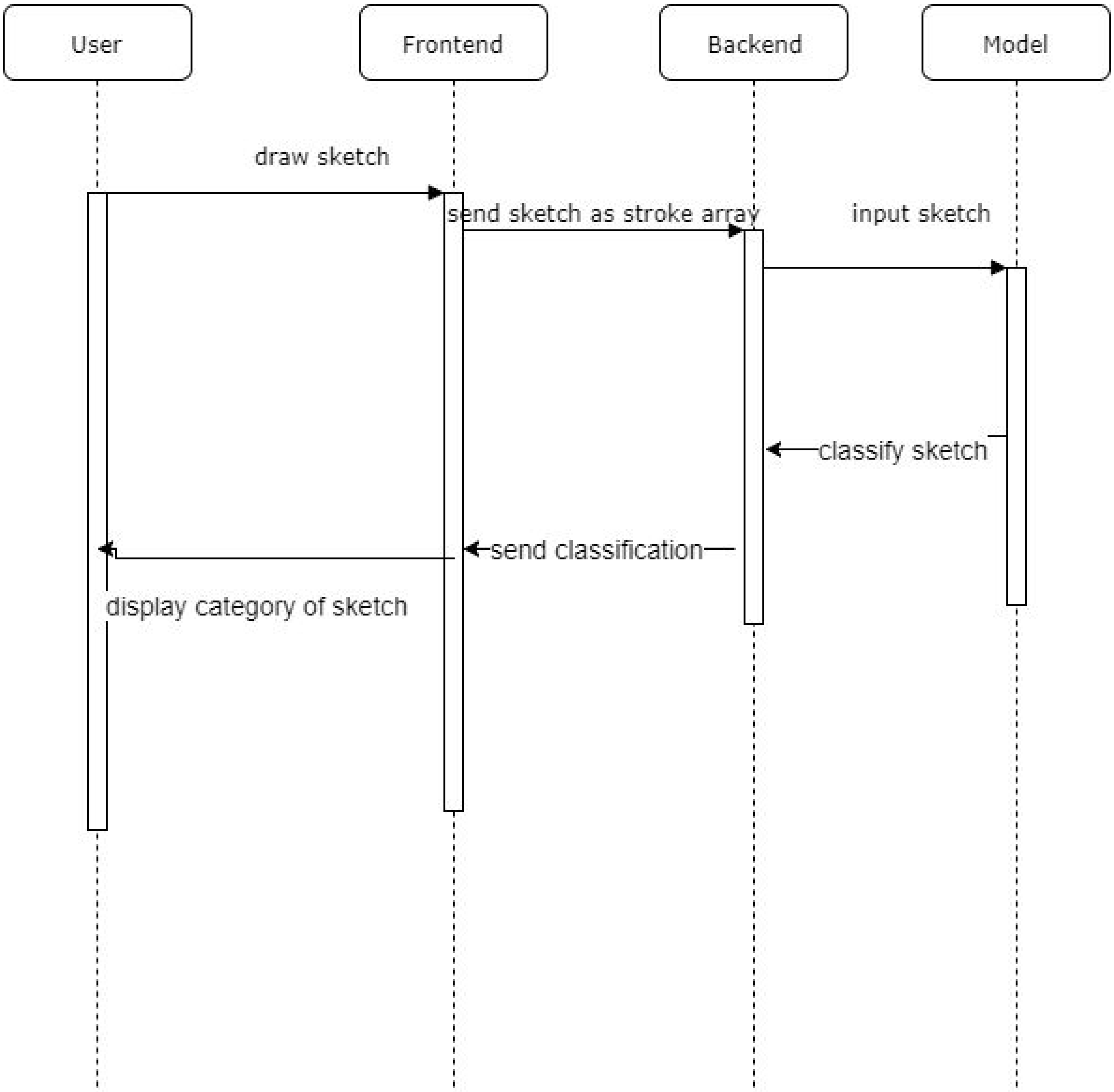
Sequence Diagram for signup



Sequence Diagram for Login



Sequence Diagram for Sketch Recognition



UNIT– 4	**METHODOLOGY & TEAM**
4.	**Introduction to Waterfall Model**

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model.

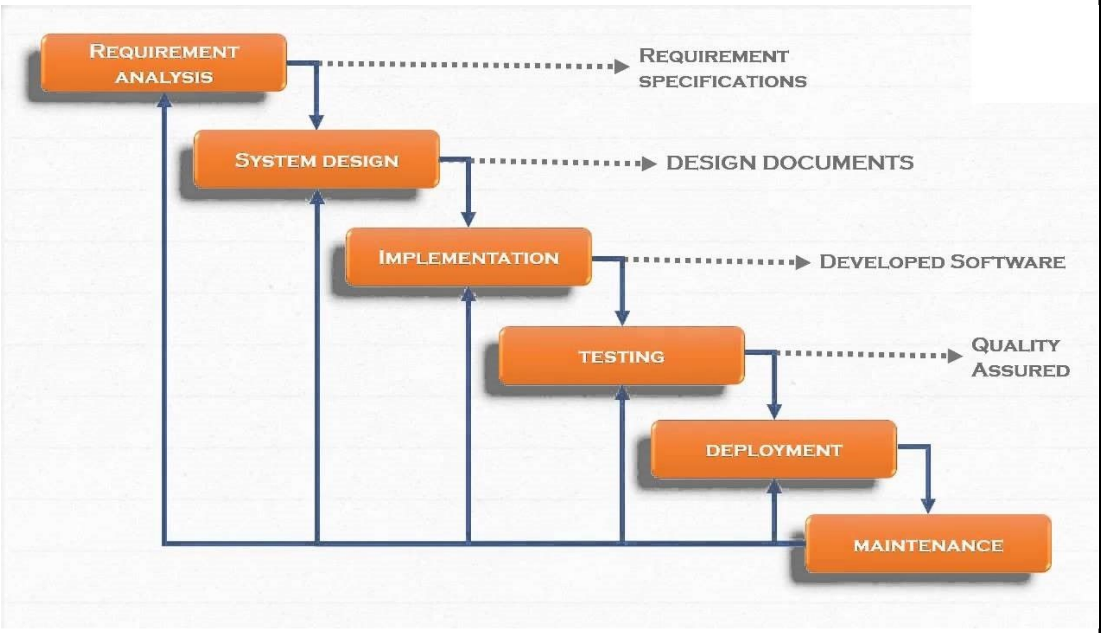


Figure 4.1 – Waterfall Model with Feedback

The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from 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.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model Pros & Cons

Advantage

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

Team members and roles

Team Member	**Project Role**
Archit Mathur	Models design
Aviral Agarwal	Backend
Charchit Sharma	Frontend

Table 4.1 – Roles and responsibilities

UNIT– 5 SYSTEM TESTING

The designed system has been testing through following test parameters.

5.1 Functionality Testing

In testing the functionality of the web sites the following features were tested:

- **Links**

a) Internal Links:

All internal links of the website were checked by clicking each link individually and providing the appropriate input to reach the other links within.

b) External Links

Till now no external links are provided in our website but for future enhancement we will provide the links to the candidate's actual profile available online and link up with the elections updates online etc.

c) Mail Links

No mail links are provided in our website till this stage but this is also a future enhancement of our website to trigger mails to people for keeping them updated about the online registration dates, the polling dates and other details.

d) Broken Links

Broken link are those links which so not divert the page to specified page or any page at all. By testing the links on our website there was no link found on clicking which we did not find any page.

- **Forms**

a) Field validation

Checks on dates have been applied

For eg. The Date of birth should be less than the current date and after that we have checked the age to be greater than or equal to 18 years (the eligible age for casting vote).

Checks have been applied on starting and ending dates, ie, the starting dates of elections or registrations should always fall before their respective ending dates.

b) Error message for wrong input

Error messages have been displayed as and when we enter the wrong details (eg. Dates), and when we do not enter any detail in the mandatory fields. For example: when we enter wrong password we get error message for acknowledging us that we have entered it wrong and when we do not enter the username and/or password we get the messages displaying the respective errors.

c) Optional and Mandatory fields

All the mandatory fields have been marked with a red asterisk (*) and apart from that there is a display of error messages when we do not enter the mandatory fields. For example: As the first name is a compulsory field in all our forms so when we do not enter that in our form and submit the form we get an error message asking for us to enter details in that particular field.

- Database

Testing is done on the database connectivity.

In the database testing we included following cases:

a) Entries in database through frontend and checked for the same in the tables.

b) Checked for the data types.

c) Checked for range of each type of data.

d) If entries made in one table are affecting other tables then we have checked those entries also.

e) We not just added details in the database by default but also did the same using the frontend.

5.2 Performance Testing

Performance testing can be applied to understand the website's scalability, or to benchmark the performance in the environment of third party products such as servers and middleware for potential purchase. This can only be done once it is put into use on the actual internet server and tested by the users.
Till now it is done using the null modem on two systems.

The system load includes:

a) **What is the number of users per time?**
b) **Checking for peak loads and how system behaves.**
c) **Amount of data accessed by user.**

This is done using only 2 systems for now so cannot be tested for load unless we deploy it on a real server machine.

5.3 Usability Testing

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

a) **Ease of learning**
b) **Navigation**
c) **Subjective user satisfaction**
d) **General appearance**

As system is not put into the real time use so it's not yet tested for usability.

5.4 Server Side Interfacing

In this we tested the server side interface. This was done by verifying that communication is done properly. Also the compatibility of server with software, hardware, network and database was tested.

5.5 Client Side Compatibility

The client side compatibility is also tested using various browsers like Google Chrome, Mozilla Firefox and Internet Explorer.

UNIT–6 TEST EXECUTION SUMMARY

Execution Test Summary Report is an overall view of Testing Process from start to end. Test Plan comes at the starting of project while Test Summary Report comes at the end of testing process. This report is given to the client for his understanding purpose.

The Test Summary Report contents are :

1. **Test Case ID generated =**
2. **Total number of resources consumed =**
3. **Passed Test Cases =**
4. **Failed Test Cases =**
5. **Status of Test Cases =**

S.No	Test Case ID	Test Case Description	Expected Outcome	Test Case Status	No. of Resources Consumed
1.	PR01	Textbox on "account.py" for PINCODE accepts numerals only	Accepts 0-9 only	PASS	Monitor, Keyboard
2.	PR02	Textbox on "account.py" for EMAIL accepts a@b.com format	Accepts a@b.com	FAILED	Monitor, Keyboard
3.	PR03				

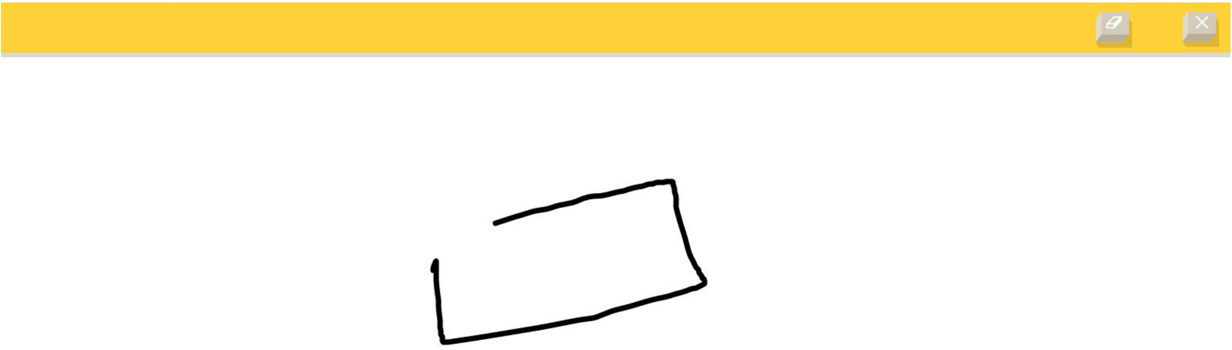
4.	**PR04**				
5.	**PR05**				
6.	**PR06**				
7.	**PR07**				
8.	**PR08**				
9.	**PR09**				
10.	**PR10**				

Table 6.1 – Test Case Summary

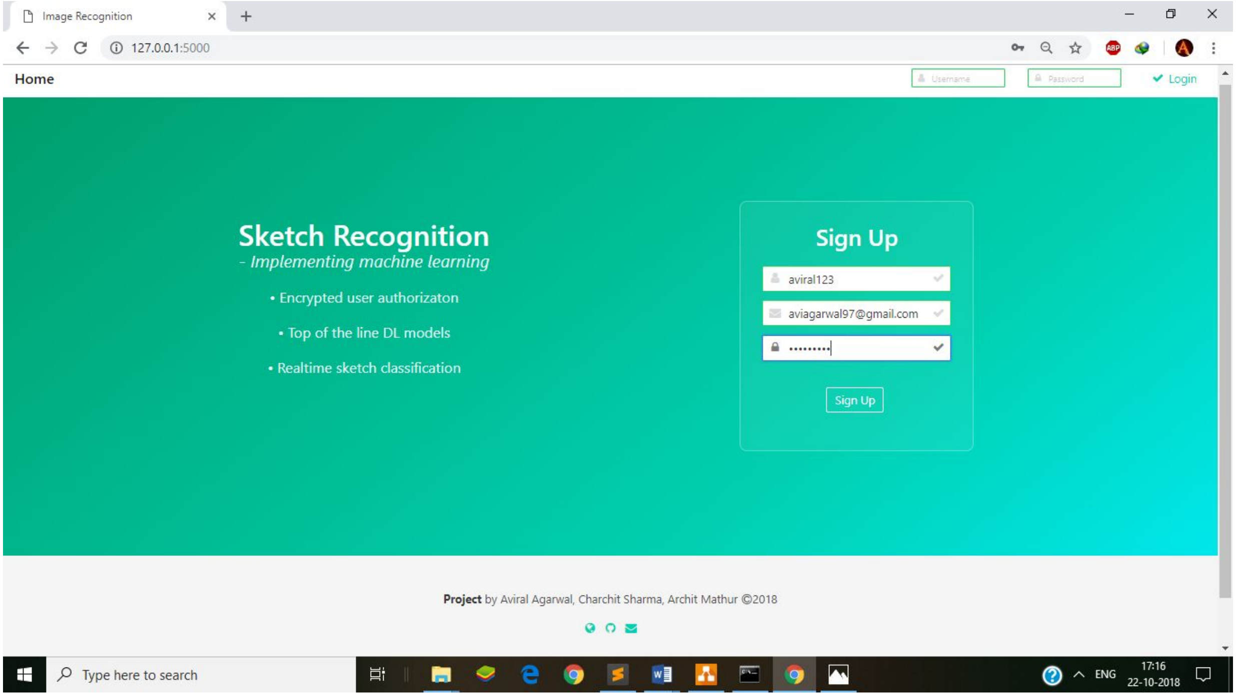
UNIT 7:

PROJECT SCREENSHOTS

-Drawing page



- Homepage



Sketch Recognition

- Implementing machine learning

- Encrypted user authorizaton
- Top of the line DL models
- Realtime sketch classification

Sign Up

aviral123

aviagarwal97@gmail.com

•••••••••

Project by Aviral Agarwal, Charchit Sharma, Archit Mathur ©2018

UNIT– 8

PROJECT SUMMARY AND CONCLUSIONS

Using sketch, various objects can be identified which can even include people, especially for the criminal recognition used by the police agencies. This can be very helpful in tracking the criminals, and help to bring rightful justice to people.

Due to the various advancements in machine learning we are able to detect to a certain degree the features of what makes an object itself. With the power of convolutions, derived from convolutional neural networks, and long short-term memory, derived from recurrent neural networks, our project aims to eliminate the difficulties and thrive on the opportunities that arise in areas such as convict detection from witness' account, teaching drawing to kids, helping doctors understand mental illness of a person by their sketching sense etcetera.

UNIT– 9 FUTURE SCOPE

Facial Recognition

This is one of the most important aspects of sketch recognition technology, as it can be used to identify people just by creating their sketches. It can be used in catching criminals as witnesses may be able to create a sketch(or get made by a specialist) using the technology which can be matched with a database of the photos of known criminals/offenders. They may be out of jails because of bail or completion of their prison sentence. Thus it will become easier to track them down and deliver justice.

UNIT– 10 REFERENCES

BOOKS:

1. Herbert Schildt, Java 2: The Complete Reference, Seventh Edition, McGraw Hill.

2. Jim Keogh, J2EE: The Complete Reference, Fourth Edition, McGraw Hill.

3. Marty Hall, Web Development with Java Server Pages.

WEBSITES:

1. http://www.stackoverflow.com/
2. http://developers.google.com/
3. www.github.com
4. www.tensorflow.org
5. www.draw.io
6. docs.google.com

Sketch Recognition using CNN and LSTM

Archit Mathur
Dept of CS, SKIT Jaipur
architmathur2011@gmail.com

Aviral Agarwal
Dept of CS, SKIT Jaipur
aviagarwal1997@gmail.com

Charchit Sharma
Dept of CS, SKIT Jaipur
charchitsharma11@gmail.com

Abstract

This paper presents the architecture of a deep neural network which is able to classify objects on the basis of their drawings. The drawings are encoded in the form of strokes. We used the open-source Quick, Draw! dataset to train, test, and validate our model. The model uses a combination of convolutional layers, LSTM layers, and a softmax output layer.

1 Introduction

Image recognition is a very popular umbrella under which lots of work has been done. Many professionals and hobbyists alike benchmark their model performance on famous datasets like MNIST and CIFAR-100. However, these images, 2D/3D grids rather, don’t help much in understanding how we humans actually make out something. From a young age, we develop the ability to communicate what we see by drawing on paper with a pencil or crayon. In this way we learn to express a sequential, vector representation of an image as a short sequence of strokes.

In this paper, we will investigate how a model trained on the stroke-based representation of images performs the task of image/sketch recognition. This is useful because

2 Methodology

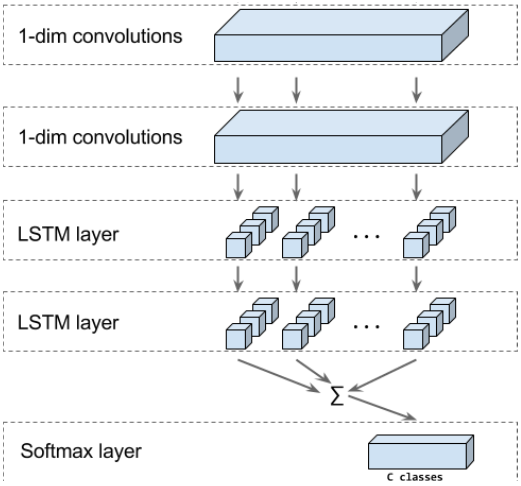
Dataset

We used the publicly available and open-source dataset Quick, Draw!. The Quick, Draw! dataset consists of stroke-based drawings of 345 common objects/animals and has more than 50 million drawings combined. We trained our DNN on 50 such categories. Each drawing is encoded in the form of a stroke array. A stroke is itself represented as an array whose elements are the coordinate positions of where the drawing pen was moved on the canvas.

We, however, don’t use the dataset as is but convert it to a different representation. The dataset is processed such that each element of the new stroke array is the offset distance, in the x and y directions, of the drawing pen from the previous point i.e (∆x, ∆y). Then we combine all the stroke arrays of a drawing into a single array and add a binary column in this array which tells if a new stroke has begun. This makes a single drawing an array of (∆x, ∆y, b). This format fares well with LSTM layers in our network.

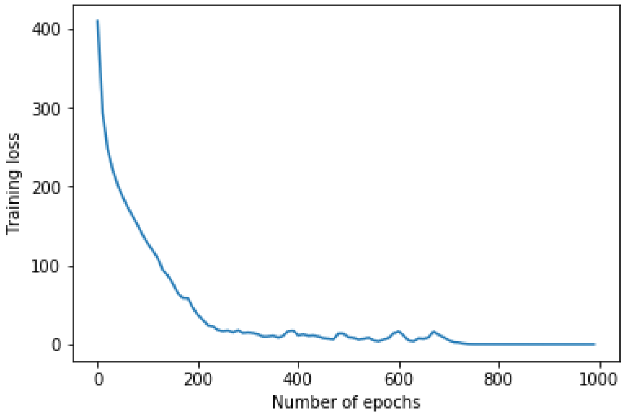
Model architecture

Our model is a combination of 1D convolutional layers and bidirectional LSTM layers. Specifically, the first convolutional layer has 48 filters with a kernel size of 5 and the second convolutional layer has 64 filters with a kernel size of 3. The output of the second convolutional layer is connected directly to the first LSTM layer. Each LSTM layer has 128 cells, both backward and forward, which models a bidirectional recurrent neural network.



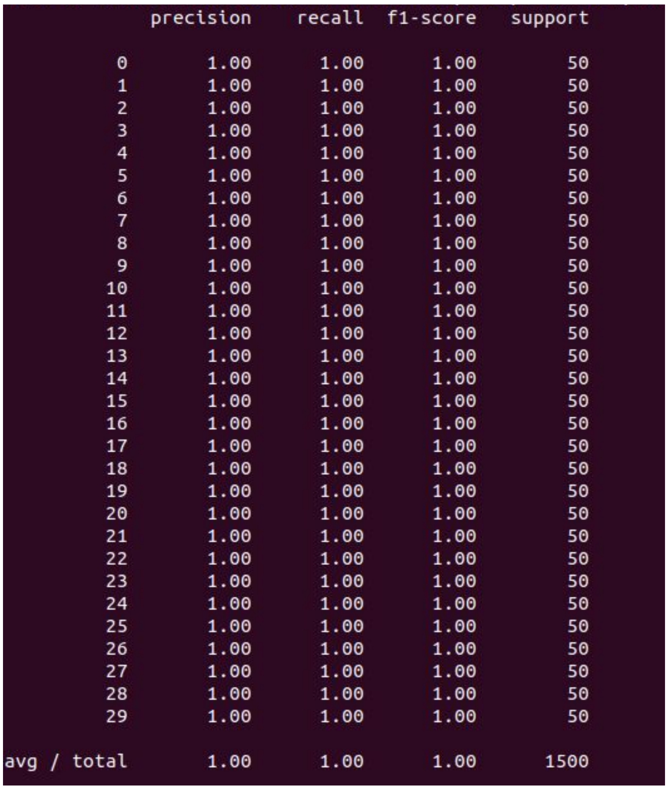
Training

We trained our model to classify 50 categories. Each of these categories had a training dataset of size 1000 and a test dataset of size 200. The model was trained for 1000 epochs using Tensorflow GPU on a laptop with NVIDIA Geforce GTX 1050 Ti (4GB), 7th generation Intel Core i7 octa-core processor, and 8GB of DDR4 RAM. Following is the training loss over time during training -



3 Experiment

We tested our model on 30 random categories, each consisting of 50 drawings, from our testing set and got the following result from Scikit-learn's classification report -



```
             precision    recall  f1-score   support

          0       1.00      1.00      1.00        50
          1       1.00      1.00      1.00        50
          2       1.00      1.00      1.00        50
          3       1.00      1.00      1.00        50
          4       1.00      1.00      1.00        50
          5       1.00      1.00      1.00        50
          6       1.00      1.00      1.00        50
          7       1.00      1.00      1.00        50
          8       1.00      1.00      1.00        50
          9       1.00      1.00      1.00        50
         10       1.00      1.00      1.00        50
         11       1.00      1.00      1.00        50
         12       1.00      1.00      1.00        50
         13       1.00      1.00      1.00        50
         14       1.00      1.00      1.00        50
         15       1.00      1.00      1.00        50
         16       1.00      1.00      1.00        50
         17       1.00      1.00      1.00        50
         18       1.00      1.00      1.00        50
         19       1.00      1.00      1.00        50
         20       1.00      1.00      1.00        50
         21       1.00      1.00      1.00        50
         22       1.00      1.00      1.00        50
         23       1.00      1.00      1.00        50
         24       1.00      1.00      1.00        50
         25       1.00      1.00      1.00        50
         26       1.00      1.00      1.00        50
         27       1.00      1.00      1.00        50
         28       1.00      1.00      1.00        50
         29       1.00      1.00      1.00        50

avg / total       1.00      1.00      1.00      1500
```

4 Conclusion

In this research paper, we developed a methodology to classify drawings using Convolutional layers and Long Short-term memory layers. The drawings were encoded as stroke arrays with each element of the stroke array consisting of offsets in the x and y directions and a boolean marking the start of a new stroke in the drawing. We demonstrated that it is indeed feasible to classify sketches on the basis of stroke-based representation i.e the way they are constructed, and intuitively understood, by a typical human.

References

1. A Neural Representation of Sketch Drawings
   https://arxiv.org/pdf/1704.03477.pdf

2. Recurrent Neural Networks for Drawing Classification
   https://www.tensorflow.org/tutorials/sequences/recurrent_quickdraw