Project Design Document

Kohonen Self-Organising Maps

Eklavya Sarkar

November 16, 2017

Abstract

This document is intended to be quite detailed so that any reader can simply start implementing the product efficiently based on the given design, required features, system documentation and evaluation.

Contents

1 Summary of Proposal **3**
- 1.1 Background . . . 3
- 1.2 Aims . . . 4
- 1.3 Objectives . . . 5
  - 1.3.1 Essential Features . . . 5
  - 1.3.2 Desirable Features . . . 5
- 1.4 Changes to Specifications . . . 6
- 1.5 Summary of Research and Analysis . . . 6

2 Software Design **7**
- 2.1 Description . . . 7
- 2.2 Data dictionary . . . 7
- 2.3 Software Technologies . . . 7
- 2.4 Data Structures . . . 8
  - 2.4.1 Logical Sequence . . . 8
  - 2.4.2 Physical Structure and Implementation . . . 9
- 2.5 System Design . . . 12
  - 2.5.1 UML Class Diagram . . . 12
  - 2.5.2 Use-case diagram . . . 13
  - 2.5.3 Use-case descriptions . . . 13
  - 2.5.4 System boundary diagram . . . 17
  - 2.5.5 Sequence Diagram . . . 18
- 2.6 Algorithm Design . . . 21
  - 2.6.1 Self-Organising Map . . . 22
  - 2.6.2 Canvas . . . 24
  - 2.6.3 Flask . . . 26
- 2.7 Interface design . . . 29
  - 2.7.1 User Interface . . . 29
  - 2.7.2 User Experience . . . 32

3 Evaluation Design **33**
3.1 Evaluation Criteria 33
3.2 Criteria Assessment 34
3.3 Participants 34
3.4 Testing 35
3.4.1 Hardware 35
3.4.2 Software 36
3.4.3 Strategy 36
3.5 Expected Conclusions 36

4 Data **37**
4.1 Ethical use of Data 37
4.2 Ethical use of Human Participants 37

5 Review Against Plan **38**
5.1 Progress 38
5.2 Changes 38
5.3 Gantt Chart 39

Chapter 1

Summary of Proposal

1.1 Background

Humans and animals have always been fundamentally proficient at pattern recognition, having been trained since birth to be able to innately identify and respond to patterns, allowing us to communicate and interact in different biological ways, thanks to the brain intricate ability to constantly learn.

Computers, although very competent at following large sets of linear, logical and arithmetic rules, have historically not been as capable as humans at discerning both visual and audible patterns. Sub-fields of Computer Science involved in facial and speech recognition, handwriting classification, natural language processing have not seen a software implementation with high accurate results that solve these problems until very recently.

Artificial Neural Networks (ANNs) are essentially biologically inspired algorithms, in the field of Artificial Intelligence, in an attempt to enable computers to seemingly *learn* from observational data, by approaching pattern recognition in a way more conducive for Machine Learning.

Instead of specifically programming the software to do tasks following certain rules, neural networks are instead given training input data and a learning algorithm, which allows the net to improve itself by adjusting it's connection's weights. These eventually become remarkably capable of doing certain tasks that conventional programs cannot, and have, in certain cases, shown to surpass human abilities as well, as seen with DeepMind's AlphaGo. One such type of neural network, Self-Organising Maps (`SOM`), will be the focus of this study.

Pioneered by Finnish professor and researcher Dr. Teuvo Kohonen, a SOM is an unsupervised learning model intended for applications in which maintaining a topology between input and output spaces is of importance. This means the vectors that are close in high dimensional space are also mapped to close nodes in the 2D space. It is in essence a method for dimensionality reduction, as it maps a high-dimension input to a low (typically two) dimensional discretized representation and preserves the topology of its input space.

Self-Organising Maps have typically been used for visualisation of high dimensional data, after having it sorted and clustered into a low, typically two-dimensional, field to see if the given unlabelled data has any structure to it. They differ from standard ANNs in several manners, such as by having 2D grid of neurons instead of a series of layers, and having competitive learning instead of error-correction learning. This means that at every time an input is presented to the neural network, only a single node is activated after they compete for the right to respond to the input. It has practical applications in meteorology and oceanography.

1.2 Aims

The aim of this project is to build a Kohonen network that is capable of clustering letters of different input handwritings, and then to use it as an interactive teaching tool in an web application that would also allow a user to test their own handwritten alphabet on the network.

The implementation of such a project would require the following steps and tools:

1. Implementing an initial functional Kohonen SOM mathematical model.
2. Training the network with a large quantity of relevant synthetic data until it reaches a high success rate of clustering letters
3. Implementing a web application to host and interact with the model
4. Using the network with user input letters via the website to test the clustering
5. Displaying the topological map of the letters on the website based on the trained network

1.3 Objectives

1.3.1 Essential Features

- The web application should communicate with the computational back-end model and retrieve the clusterisation data sent back
- The website should have an interactive 'Draw' page where users can draw their letter on a Graphical User Interface(`GUI`) canvas and have the website process and display which letter it is, by interacting with the `ANN` model.
- The website should display the neural network's topological map of alphabets to the user based on training data
- The website should highlight where your input would be placed on the displayed topological map
- The website should have a 'Learn' page which displays animations or clickable diagrams of neural networks and `SOM`s, to show how weights are adjusted and converged, and how the network is trained over time.
- The website should have a 'Database' page which contains information on the dataset used to train and test the neural network, such as the number of images used, the size of the entire database, links to the source files.

1.3.2 Desirable Features

- The users should have an 'in-depth' option of seeing the steps the network goes through, such as re-centring, cropping and down-sampling of the input, probabilities numbers or graphs of which letter the input corresponds to.
- Allow users to input more than one single input ie enter a whole 'training set'.
- The 'database' page which shows a sample training data letter for each alphabet from A to Z, and after clicking on one of the letters, the entire training dataset images of different handwriting for that alphabet should be shown. This is to give a visual representation and sense of scale of how many different handwritten letters were used to train the neural network for each alphabet.
- Some of the instructions sentences on the website could be written using the synthetic training data images.

1.4 Changes to Specifications

1.5 Summary of Research and Analysis

Most of the research and analysis so far has gone into understanding the various components of following elements:

- Substantial researching and conducting an extensive literature review on a wholly new topic to understand the nature of Self-Organising Maps: their topological mapping, competitive process, sample usages, general applications and actual implementation. Also substantial work done on reviewing Dr Irina V. Biktasheva's `COMP305: Biocomputation` module and Stanford's excellent `Introduction to Machine Learning` course by Prof. Andrew Ng. Full references can be found in the appendix.
- The system design on making the `SOM` interactive for human users and all the extensive technologies being to host it on a web application, especially for front-end graphics visualisation and back-end algorithmic modelling.
- Examining publicly available datasets on handwritten input and existing similar applications in order to make a distinguished novel project. In this case, there are many existing live-time applications that use `ANN`s to classify hand-drawn **digits** using the `MNIST` dataset, but almost none that use a `SOM` with competitive learning in real time to cluster handwritten **letters** and displaying the topological map with data sorted according to the similarities between the letters using the `EMNIST` dataset for example.

Chapter 2

Software Design

2.1 Description

This chapter describes how the overall software and system will work and interact with all of it's moving components by giving an in depth explanation about the flow of data.

2.2 Data dictionary

A data dictionary was maintained throughout this project so far, to ensure all the terms and fully understood by the reader and are referred to in the correct context. The full glossary and acronyms list can by found in the appendix.

2.3 Software Technologies

The following table lists out the required technologies, languages, libraries and frameworks to implement this project in its entirety. If additional unplanned technology is required at a later stage, it will be added to the updated version of this table in the final dissertation.

Task	Technology	Library
Implementing Kohonen SOM	-Python	-NumPy -Pandas -Matplotlib
Training the network with synthetic data	-Python	-Scikit-Learn -EMNIST database
Implementing web application to host SOM	-HTML -CSS -JavaScript -Flask	-jQuery -AJAX -Bootstrap
Displaying topological map and general model response	-JavaScript	-D3.js
Hosting and backing up source code	-Github	-Git
Hosting the website and model	-Web server	-University server -`github.io` page

2.4 Data Structures

2.4.1 Logical Sequence

The following is the sequence of events to **train** the neural net:

1. Input image
2. Feature Extraction and Preprocessing
3. Learning and Recognition using `SOM`:
   (a) Initialise network
   (b) Present input
   (c) Best node wins via competitive learning
   (d) Update weights accordingly
   (e) Return winning node
4. Output result
5. Match output with labelled letter
6. Output letter
7. Repeat with different input

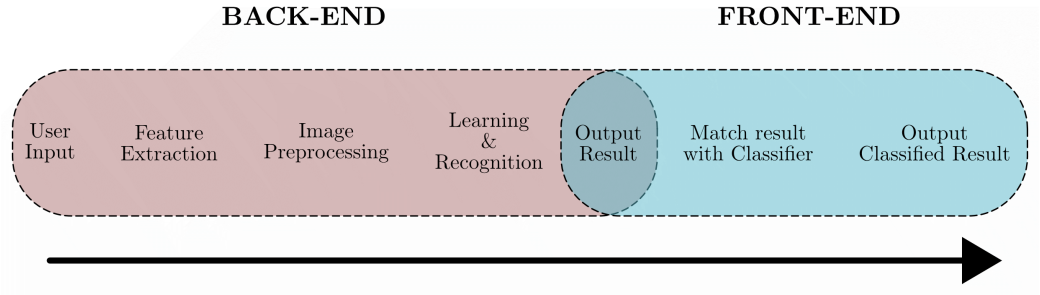


Figure 2.1:

The following is the sequence of events to **test** the neural net:

1. Input image
2. Feature Extraction and Preprocessing
3. Recognition using `SOM`:
   (a) Initialise network
   (b) Present input
   (c) Return winning node
4. Output result
5. Match output with labelled letter
6. Output letter

2.4.2 Physical Structure and Implementation

The `EMNIST` dataset contains images of handwritten letters (A-Z), along with labels for which alphabets they actually are corresponding to (1-26), for both the training and testing set. Each image is 28x28 pixels.

Similarly, the user input drawing will be converted to a numeric matrix based on the where the user has drawn, which will simply be used as a 1D vector of 784 numbers, thus rendering an image simply a bunch of greyscale of black and white values in a 784-dimension array.

Once the network is trained on the training data, we will evaluate the success rate of the system on test data. The following images show the sequence of image processing.

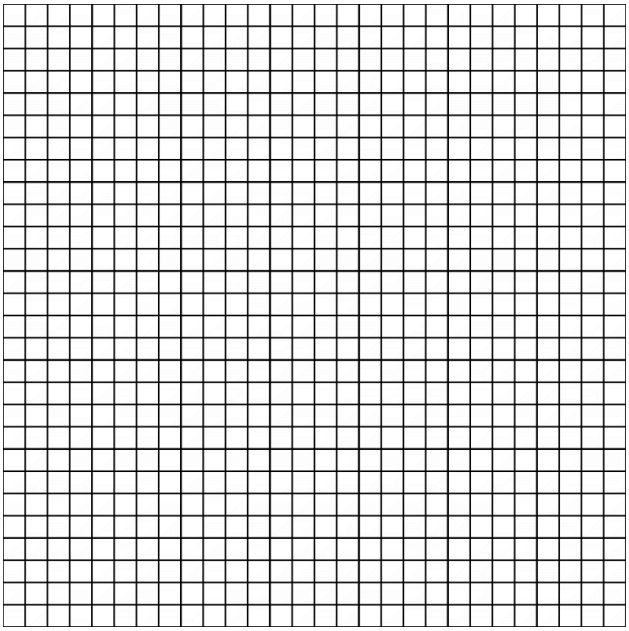


Figure 2.2: Empty 28 x 28 grid

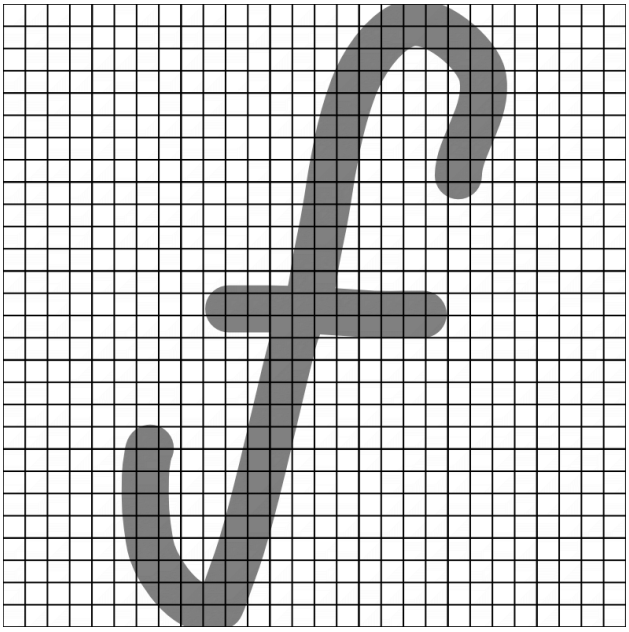


Figure 2.3: Drawn letter inside grid

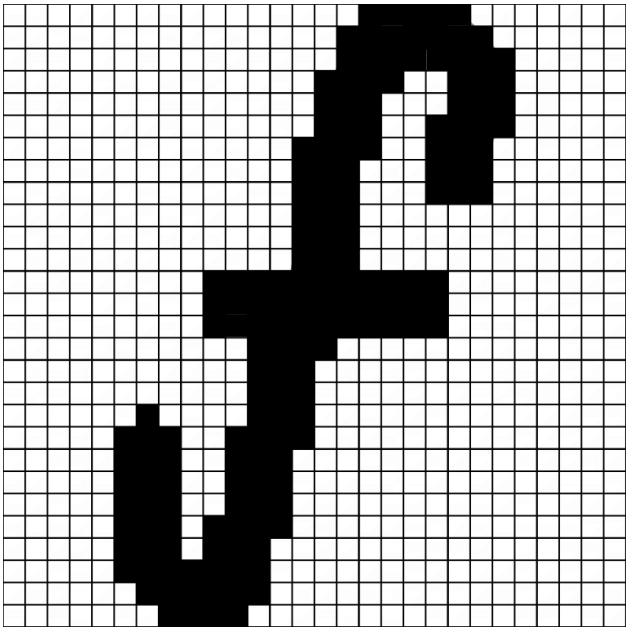


Figure 2.4: Drawn Letter with Grid with binary pixel values

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Note: in this matrix there are actually only 14x14, as 28 rows and columns were too many to show in a physical document.

Sample matrix conversion of image

2.5 System Design

2.5.1 UML Class Diagram

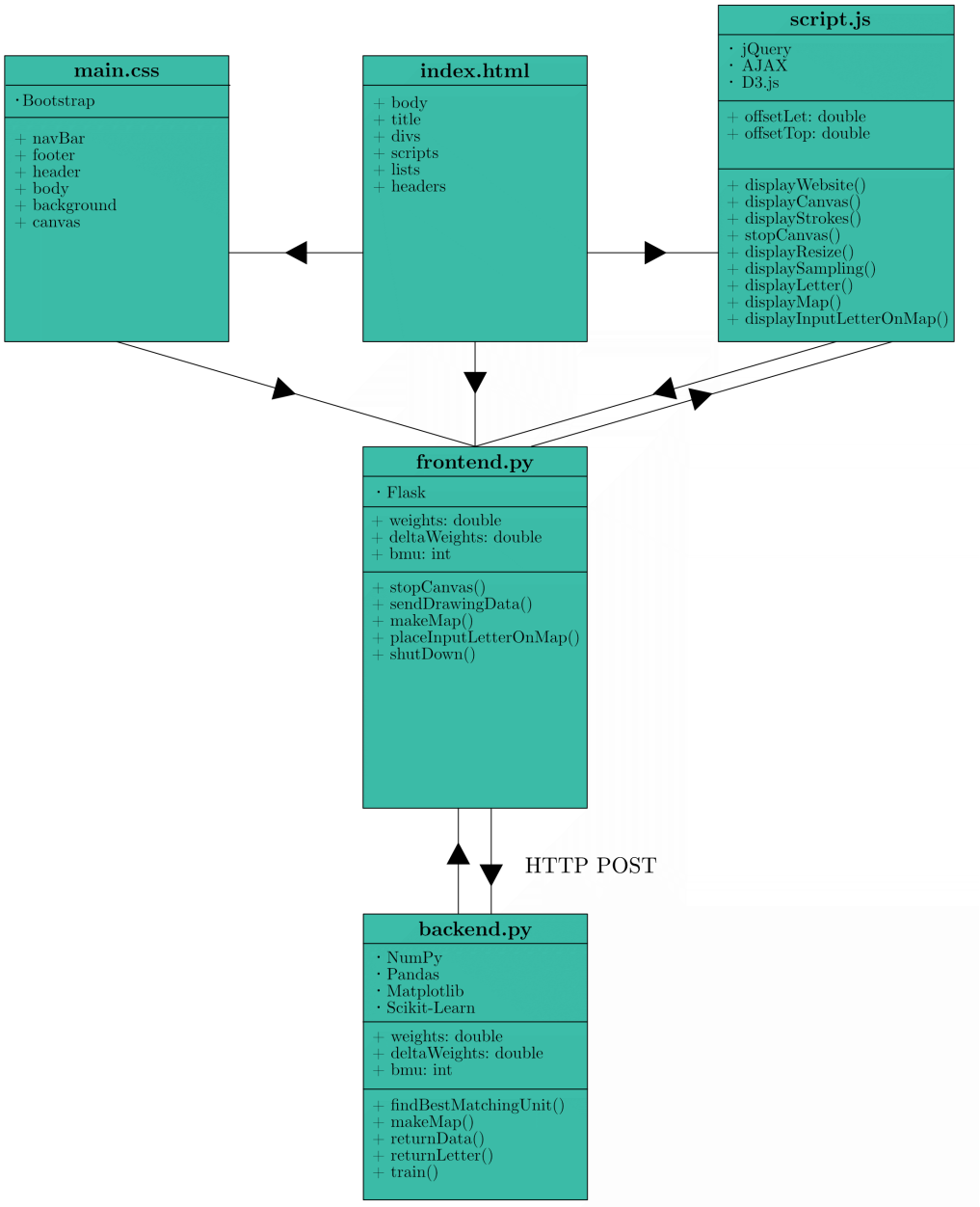


Figure 2.5: Use-Case Diagram

As shown, there are 5 main classes, 3 of which (`HTML`, `CSS` and `JavaScript`) which are 'hosted' by Flask's front-end model, which in turn also communicates with the back-end `SOM` computational back-end model.

2.5.2 Use-case diagram

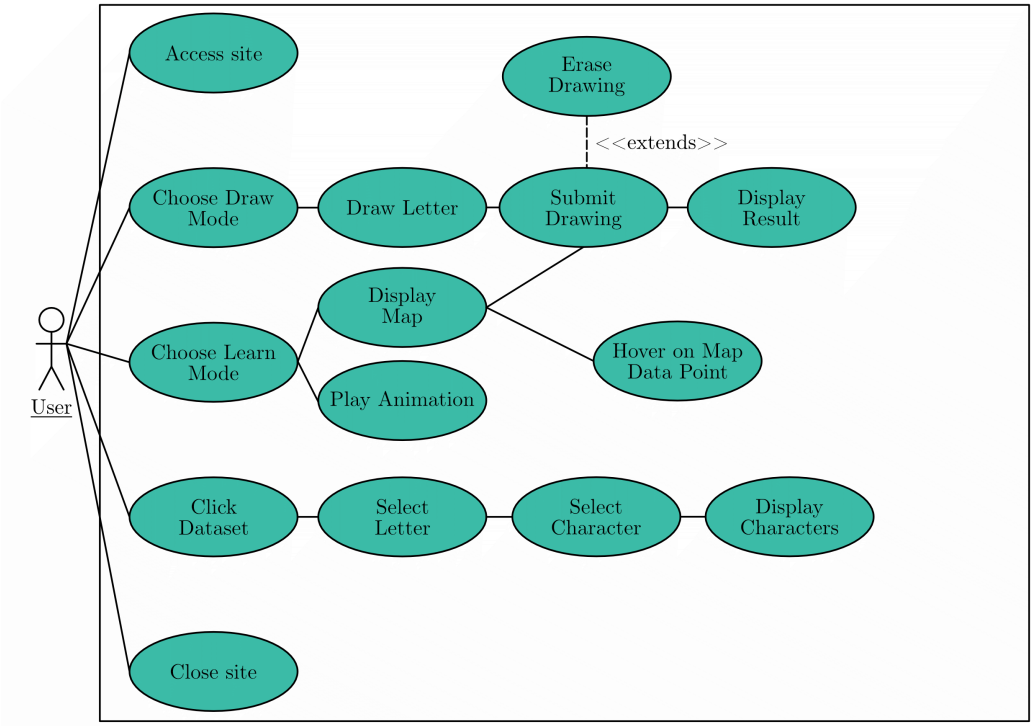


Figure 2.6: Use-Case Diagram

2.5.3 Use-case descriptions

ID	**Use Case 1**
Name	Access site
Description	The user accesses the system either via a desktop or mobile web browser
Pre-condition	System is running
Event flow	1. Open Browser on device 2. Type in website's URL

ID	**Use Case 2**
Name	Choose Draw Mode
Description	The user chooses the draw mode option
Pre-condition	System is running
Event flow	1. Click on 'Draw' button

ID	**Use Case 3**
Name	Draw Letter
Description	The user draws a letter on the canvas
Pre-condition	System is running
Event flow	1. Use mouse on desktops, fingers on touchscreen devices 2. Click/touch and drag on canvas to draw 3. Draw an alphabet

ID	**Use Case 4**
Name	Submit Drawing
Description	The user submits their input drawing to the backend
Pre-condition	System is running
Event flow	1. Press the 'submit' button
Extension points	Erase Drawing

ID	**Use Case 5**
Name	Erase Drawing
Description	The user erases all of his current drawing
Pre-condition	System is running
Event flow	1. Click on 'Erase' 2. Canvas resets to blank

ID	**Use Case 6**
Name	Display Result
Description	The website displays the returned letter corresponding to the input
Pre-condition	System is running The computational model is functional
Event flow	1. The letter with the most resemblance to the input is displayed

ID	**Use Case 7**
Name	Choose Learn Mode
Description	The user chooses the learn mode option
Pre-condition	System is running The computational model is functional
Event flow	1. Click on 'Learn' button

ID	**Use Case 8**
Name	Display Map
Description	The website displays the SOM
Pre-condition	System is running The computational model is functional
Event flow	The topological map is printed out for the user

ID	**Use Case 9**
Name	Play Animation
Description	The website plays the neural network animation
Pre-condition	System is running
Event flow	1. User clicks on play button 2. The animation is played

ID	**Use Case 10**
Name	Hover on Map Point Data
Description	The user hovers over a particular point on the SOM
Pre-condition	System is running The computatinal model is functional
Event flow	1. User brings cursor over map point data 2. Map point shows contextual values

ID	**Use Case 11**
Name	Click Dataset
Description	The user selects to view the dataset
Pre-condition	System is running
Event flow	1. Click on 'Dataset' button

ID	**Use Case 12**
Name	Select Letter
Description	The user selects a letter from all whole alphabet
Pre-condition	System is running
Event flow	1. Click on 'Dataset' button 2. Click on a letter

ID	**Use Case 13**
Name	Select Character
Description	The user selects a given character of the chosen letter
Pre-condition	System is running
Event flow	1. Click on 'Dataset' button 2. Click on a letter 3. Click on a specific letter 4. Click

ID	**Use Case 14**
Name	Display Characters
Description	The website displays the meta data on the chosen character
Pre-condition	System is running
Event flow	The meta-data on such a character is displayed as a pop-up

ID	**Use Case 15**
Name	Close Site
Description	The user shuts down the browser
Pre-condition	System is running
Event flow	
Extension points	
Triggers	
Post-condition	The user exists the browser

2.5.4 System boundary diagram

Below is the system boundary diagram with the two types of users:

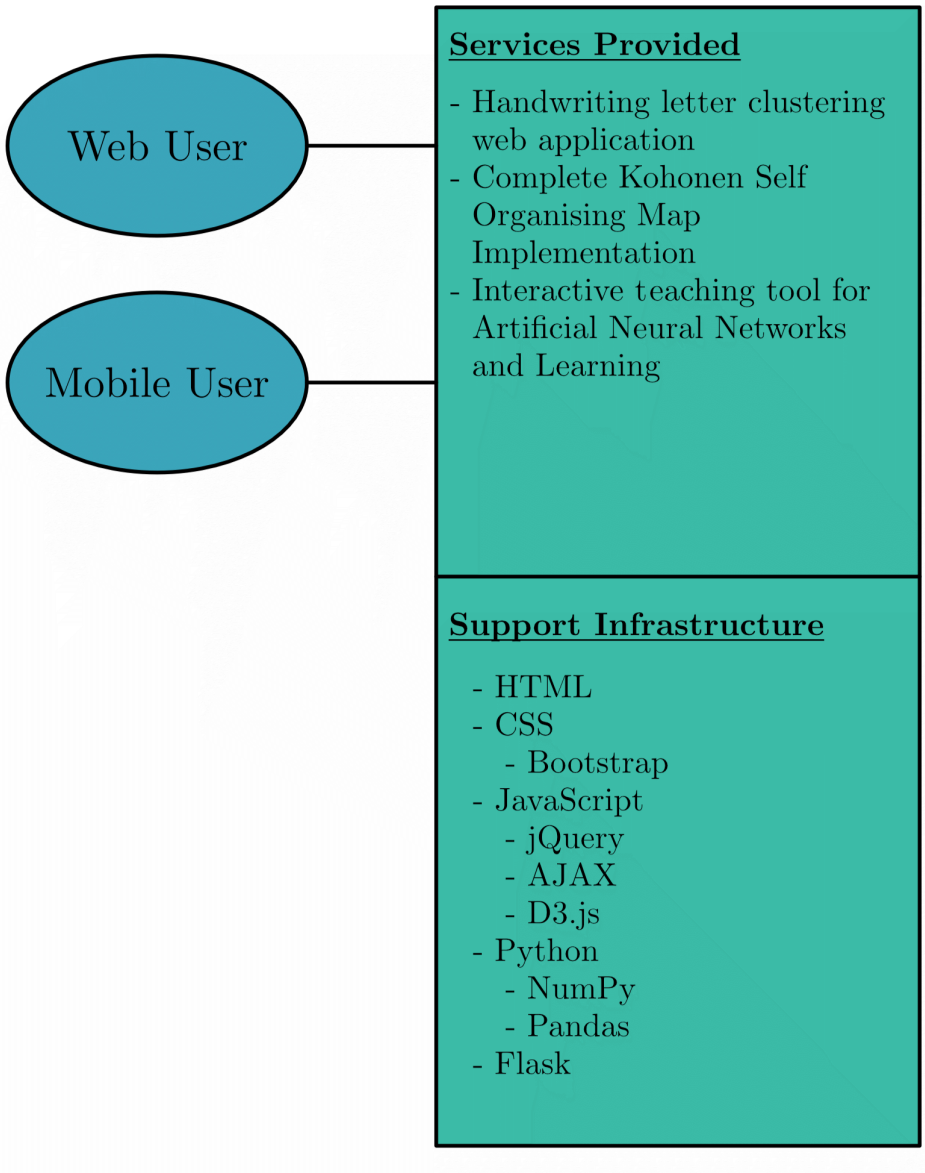


Figure 2.7: System boundary diagram

2.5.5 Sequence Diagram

The following is the sequence diagram when the user chooses the 'draw' option.

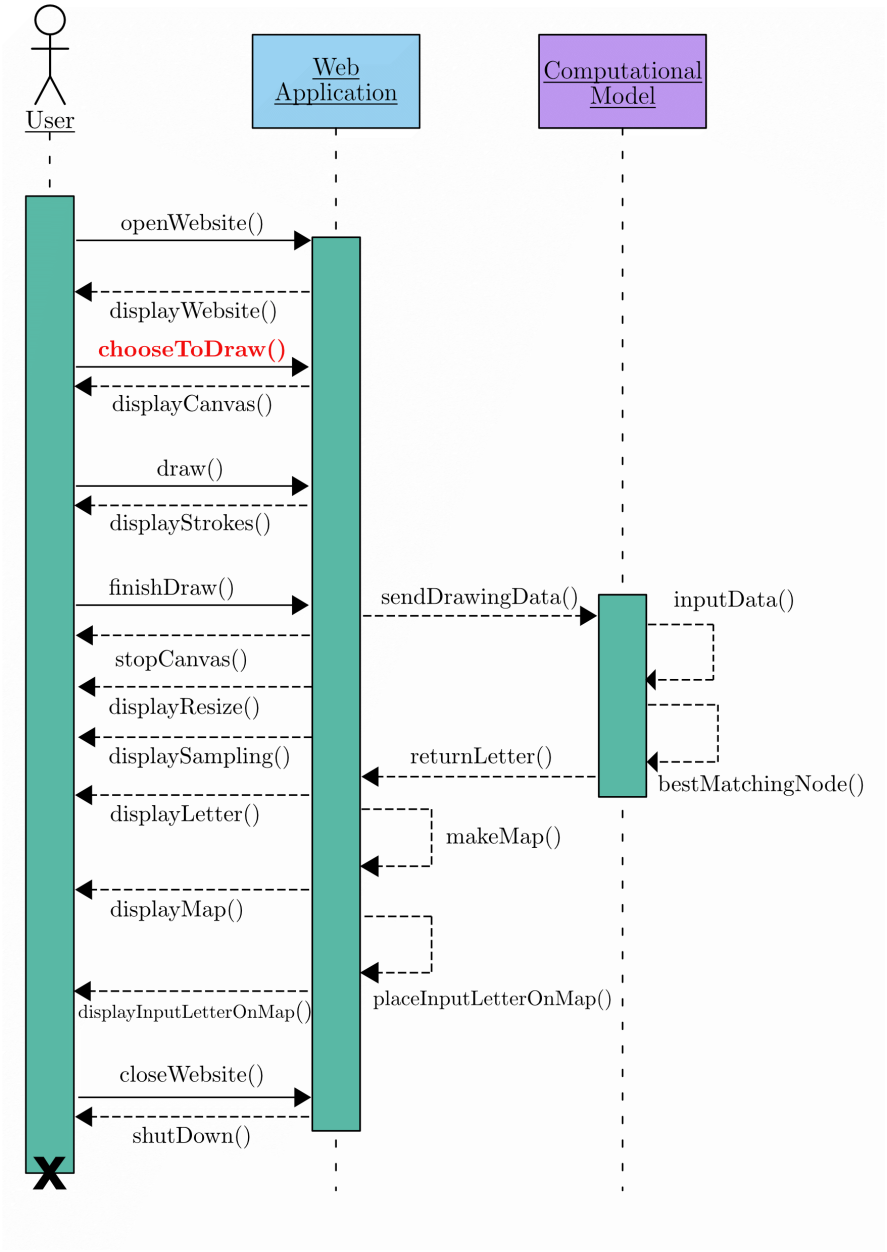


Figure 2.8: Sequence diagram for the drawing page

The sequence diagram can be broken down to the following detailed order of steps:

1. *openWebsite()*: user access the website
2. *displayWebsite()*: website content is displayed to user
3. *chooseToDraw()*: user chooses the 'draw' option button
4. *displayCanvas()*: website displays the drawable `GUI` canvas
5. *draw()*: user inputs on the canvas with his mouse or finger
6. *displayStrokes()*: website shows the strokes the user is drawing in real-time
7. *finishDraw()*: user submits his drawing
8. *sendDrawingData()*: website sends the drawing data's pixel values to the computational model as an array of integers or doubles
9. *stopCanvas()*: website stops displaying a drawable canvas to the user
10. *inputsData()*: model is fed the user's drawn data array
11. *bestMatchingUnit()*: computational model finds the best matching unit
12. *returnLetter()*: model returns the highest similarity letter's index
13. *displayResize()*: website shows the user the re-centring and re-sizing of his drawing
14. *displaySampling()*: website down-samples the user input drawing
15. *displayLetter()*: the corresponding letter with the highest similarity to the input drawing is displayed to the user
16. *makeMap()*: the topological map's data are arranged in arrays to be shown
17. *displayMap()*: the map is shown to the user using front-end graphics and the data from the array
18. *placeInputLetterOnMap()*: calculate where the user input would be placed on the map by sorting it in the array containing the other value points
19. *displayInputLetterOnMap()*: user's input letter is shown where it would belong on the map
20. *closeWebsite()*: user closes the website
21. *shutDown()*: the web application shuts down

The following is the sequence diagram when the user chooses the 'learn' option.

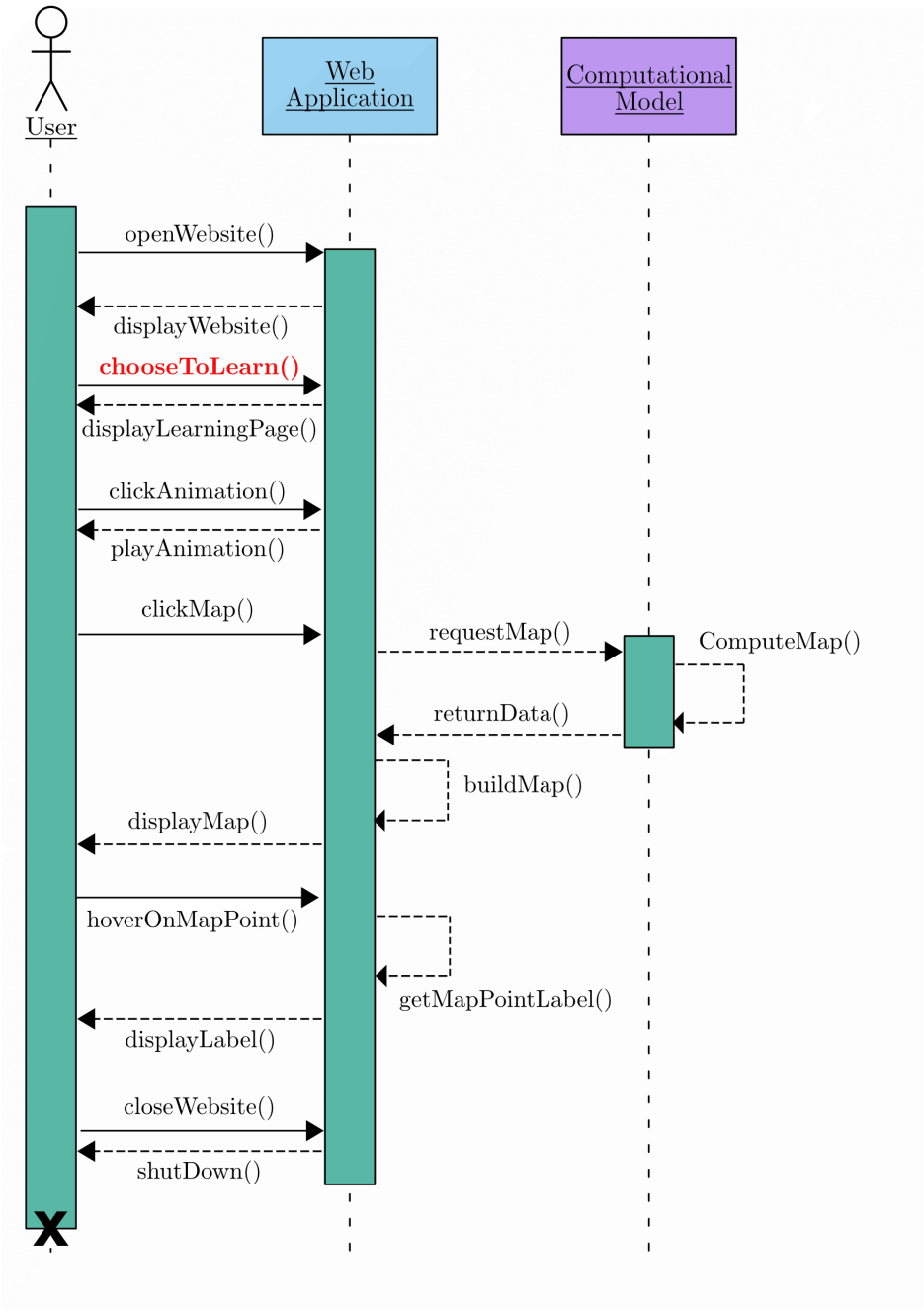


Figure 2.9: Sequence diagram for the learning page

The sequence diagram can be broken down to the detailed order of steps:

1. *openWebsite()*: user access the website
2. *displayWebsite()*: website content is displayed to user
3. *chooseToLearn()*: user chooses the 'learn' option button
4. *displayLearningPage()*: website displays 'learn' page
5. *clickAnimation()*: user presses play on an animation
6. *playAnimation()*: website plays the animation
7. *clickMap()*: user requests to open or see the topological map of the training set data
8. *requestMap()*: website requests the map from the computational model
9. *computeMap()*: computational model computes the map
10. *returnData()*: computational model returns the data of the map in a hash
11. *buildMap()*: website builds the map using the hash and its data
12. *displayMap()*: website displays the map to the user
13. *hoverOnMapPoint()*: user hovers on a specific map point
14. *getMapPointLabel()*: data for that specific point is fetched in the hash using the key
15. *displayLabel()*: data for that specific point is displayed
16. *closeWebsite()*: user closes the website
17. *shutDown()*: the web application shuts down

The sequence diagrams attempt to illustrate the interaction between user(s) and the pages via the computational model, and the flow of events as they happen.

2.6 Algorithm Design

The next few sub-sections contain key examples of pseudo-code and how the interaction between components is going to work.

2.6.1 Self-Organising Map

The Self-Organising Map is the python computational model at the back end which will adjust the network's weights during training with synthetic data and cluster similar inputs during utilisation.

1. Setup
    (a) Import necessary libraries
    (b) Create virtual environment
    (c) Create required dataframe to contain input values
    (d) Choose parameters: `SOM` size, learning parameters
    (e) Create grid
2. Normalisation
    (a) Normalise input data vectors
        - RGB: 3 vectors with values from 0 to 255.
        - Greyscale: single vector with values will be from 0 to 255.
        - Black and white: binary 0 or 1 values.
3. Learning
    (a) Initilise nodes' weights to random values
    (b) Select Random Input Vector
    (c) Repeat following for all nodes in the map:
        i. Compute Euclidian Distance between the input vector and the weight vector associated with the first node
        ii. Track the node that produces the smallest distance
    (d) Find the overall Best Matching Unit (`BMU`), ie the one with the smallest distance of all the nodes
    (e) Determine topological neighbourhood of `BMU` in the Kohonen Map
    (f) Repeat for all nodes in the `BMU` neighbourhood:
        i. Update the weight vector of the first node in the neighbourhood of the `BMU` by adding a fraction of the difference between the input vector and the weight of the neuron
    (g) Repeat this whole iteration until reaching the chosen iteration limit

4. Visualisation

   (a) Make use of `Matplotlib` for development, local testing and visualisation

   (b) Final visualisation for the user is to be done the front end with `D3.js`

2.6.2 Canvas

The canvas on the front-end is the graphical user interface the user sees as the input area in which to draw his letter using a mouse or ideally by hand on a touch screen device. To achieve this, the canvas must have 4 event listeners for the mouse and then draw black pixels continuously along where the user inputs data in the correct events.

Note that these events should also apply for human finger input as well as for mouses. Pseudo-code for the events:

Algorithm 1 Mouse Move Event

```
if mouseMove then
    drawable ← true
    getCoordinates()
end if
```

Algorithm 2 Mouse Down Event

```
if mouseDown then
    drawable ← false
    getCoordinates()
end if
```

Algorithm 3 Mouse Up Event

```
if mouseUp then
    drawable ← false
end if
```

Algorithm 4 Mouse Out Event

```
if mouseOut then
    drawable ← false
end if
```

Algorithm 5 getCoordinates() Function

```
Previous_X ← Current_X
Previous_Y ← Current_Y
Current_X ← Event_X−canvas.offsetLeft
Current_Y ← Event_Y−canvas.offsetTop
if drawable ← true then
  draw()
end if
```

Algorithm 6 draw() Function

```
canvas.beginPath()
canvas.moveTo(Previous_X,Previous_Y)
canvas.lineTo(Current_X,Current_Y)
canvas.drawLine(Current_X,Current_Y)
canvas.stroke()
canvas.closePath()
```

Where:

- mouseDown is an `event` where the user only touches the screen, but does not yet draw, meaning only the fixed input coordinates are required.
- mouseMove is an `event` where the user draws on the screen, thereby continuously calling the draw function as the input position varies.
- mouseUp is an `event` where the user stops inputting.
- mouseOut is an `event` where the user leaves the canvas drawable area.
- getCoordinates and draw() are `methods`.
- drawable is a `boolean`
- offsetLeft is an `HTML canvas` property that returns 'the number of pixels that the upper left corner of the current element is offset to the left within the `HTMLElement.offsetParent` node'.
- offsetLeft is an `HTML canvas` property that returns 'the distance of the current element relative to the top of the offsetParent node'.
- $Previous_X$, $Previous_Y$, $Current_X$, $Current_Y$ are `int`s about the input coordinates via the mouse or finger.
- beginPath(), moveTo(), lineTo(), drawLine(), closePath() are all `HTML` methods that reference the `canvas` tag.

2.6.3 Flask

In order to fully understand how the design and interaction of the front and back-end work together, a detailed guide is required.

The `pip3` package manager is recommended to install `Flask` or any other `Python3` package:

```
$ python3 install pip3
$ pip3 install Flask
```

Virtual Environments

For `Flask`, the virtual environment in the files folder `~\myPath\myFolder` must be correctly configured to load all of our libraries, and ensure that they don't get change or mix up with the development PC's native Python installation. The following sequence shows how to create, activate, and deactivate the virtual environment in order to install packages inside the `~\myPath\myFolder` path, isolated from the global Python installation.

```
$ pip3 install virtualenv
$ cd myPath
$ virtualenv myFolder
$ source myFolder/bin/activate
$ pip3 install myPackages
$ deactivate
```

HTML Pages

To fully comprehend how `Flask` is going to work with the `HTML` and `CSS` pages, it is easiest to look at an simple sample model. First, we write the `Flask` application code `app.py`:

```
0 | from flask import Flask
1 | from flask import render_template
2 | app = Flask(__name__)
3 |
4 | @app.route('/')
5 | def homepage():
6 |  return "Welcome to Python Flask Homage!"
7 |
8 | @app.route('/helloworld')
```

```
9 | def helloworld():
10|  return render_template('helloworld.html')"
11|
12| if __name__ == '__main__':
13| app.run(use_reloader=True)
```

And then the sample `HTML` page:

```
0| <html>
1| <header>
2| <title>myTitle</title>
3| </header>
4| <body>
5| Hello world from helloworld.html page
6| </body>
7| </html>
```

Then run `app.py` on the terminal:

```
$ python3 app.py
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

And on a browser navigate to: `http://127.0.0.1:5000`

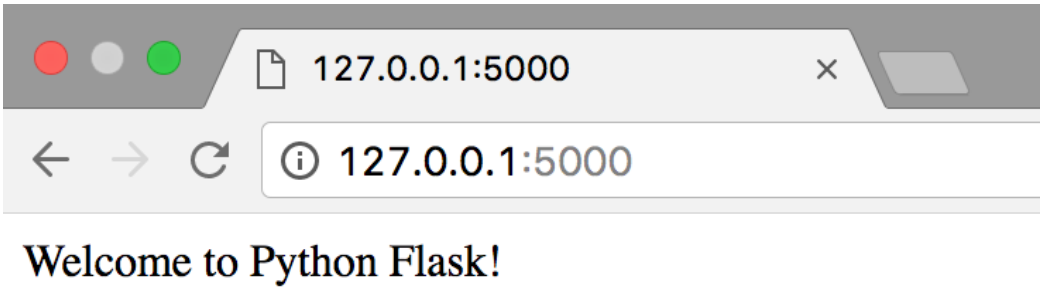


Figure 2.10: The homepage containing hard-coded output

Then navigate to: `http://127.0.0.1:5000/helloworld`

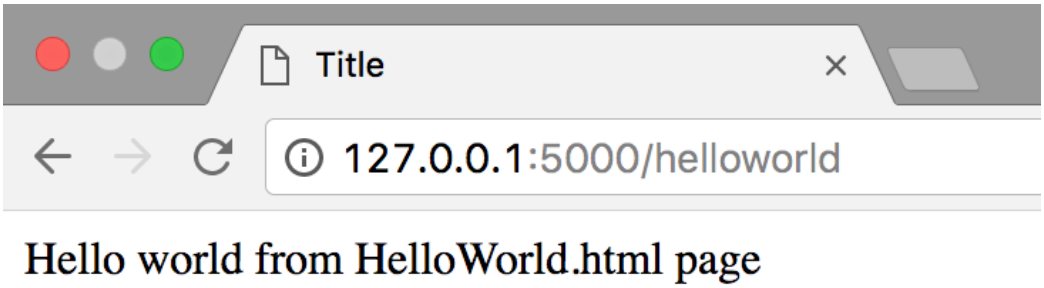


Figure 2.11: The page containing output from HTML file

The first URL simply displays the hard-coded 'Welcome to Python Flask Homepage!', but the second one shows the contents of the `helloworld.html` file: Hello world from helloworld.html page.

The `render_template` method in `Flask` is essential to host `HTML`, `CSS` and `JavaScript` files and will be used extensively for page view.

GET-POST

`GET` and `POST` methods will be used to communicate between the front-end files implemented by `Flask` and `SOM` computational back-end implemented in Python. The data will be transferred in `JSON` (or potentially `NDJSON`) format using `Flask`'s `flask.json.jsonify` method.

```
0| from flask import request
1|
2| @app.route('/nodes', methods=['POST'])
3| def get_data():
4|  if request.method == 'POST':
5|  return jsonify(winnerNode, deltaWeight, updatedWeight)
```

The returned `JSON` would look like the following inside curly brackets (here omitted):

```
1| "winnerNode": "3",
2| "deltaWeight": "0.2419",
3| "updatedWeight": 0.7581
```

This enables the back-end application to receive and send data, such as the user input letter's pixel coordinates and RGB (or binary black and white) values.

2.7 Interface design

2.7.1 User Interface

The home page will in principle have 3 main options for the user to choose from:

- Draw: the user can draw a letter in his own handwriting and test the Kohonen network to see if it successfully categorises the alphabet.
- Learn: the user can learn more about `ANN`s and `SOM`s, by viewing animations, graphs, and other practical applications.
- View Dataset: the user can see the data used to train the network.

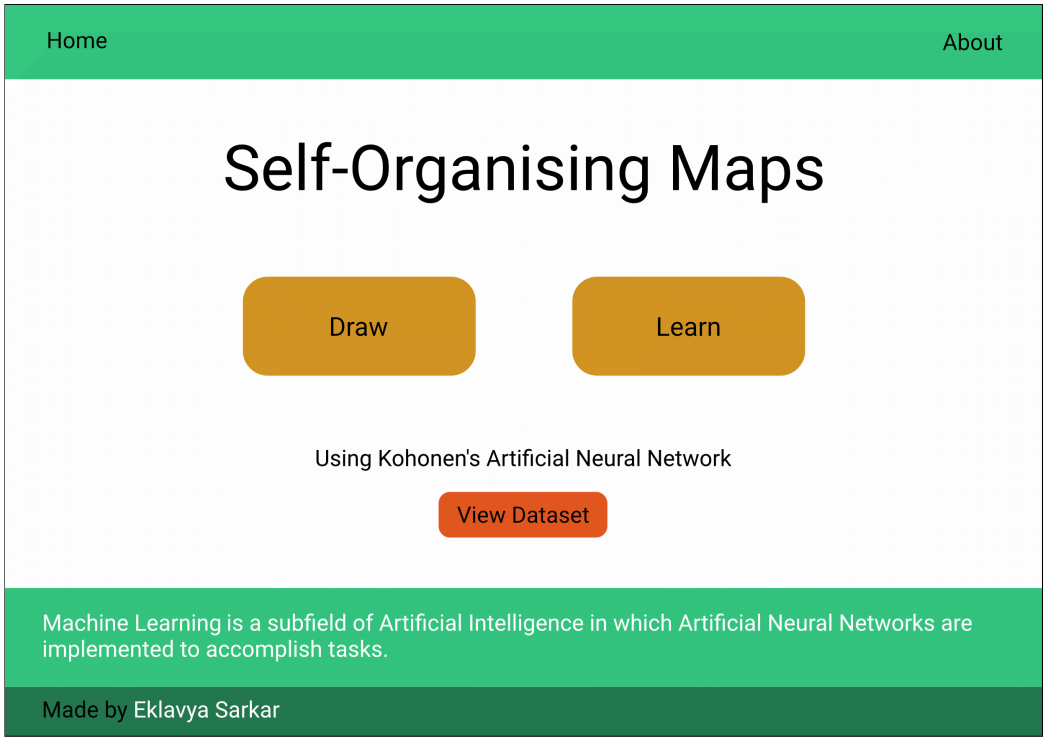


Figure 2.12: The home page

The draw page will have the main drawing canvas `GUI` in the middle of the screen, along with a 'clear' button option to erase his current drawing, and a submit button to send the data to the back-end.

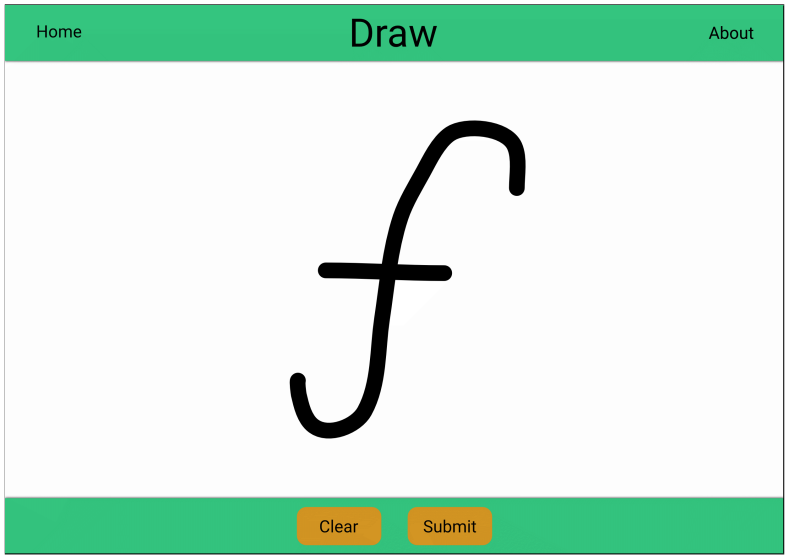


Figure 2.13: The draw page

The database page will first display the entire alphabet with only a single sample image for each letter.

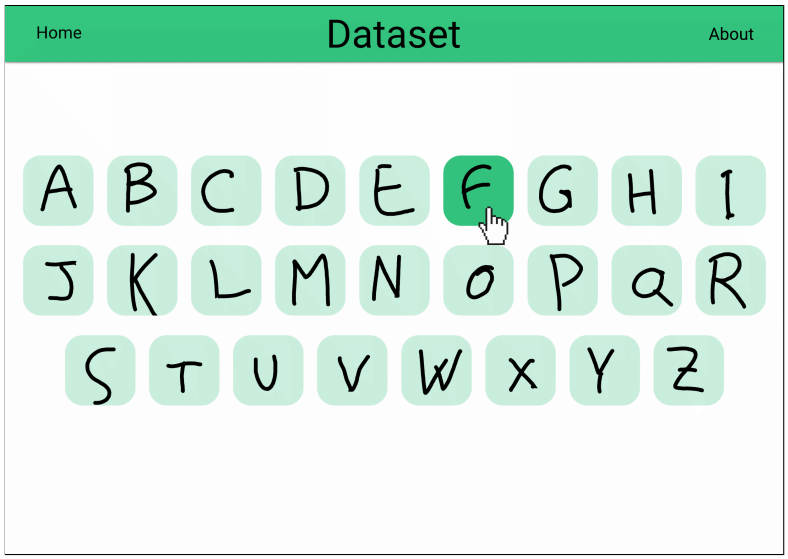


Figure 2.14: The database page

After clicking on one such letter, the user will be shown all the inputs used to train the data (or at least enough to cover up one entire 15” desktop screen).

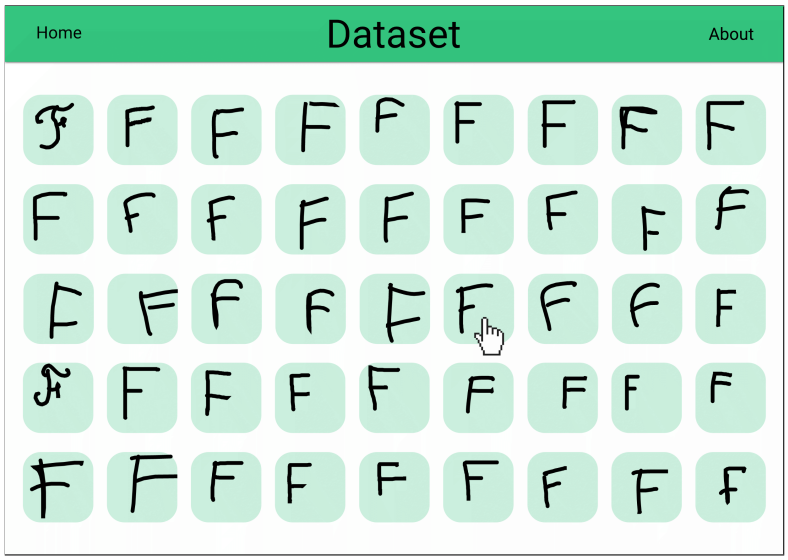


Figure 2.15: The letters page

If the data used for training contains any further relevant meta-information on each individual letter, such as input number, drawing date or location, then they could further be shown by clicking one final time on a letter.

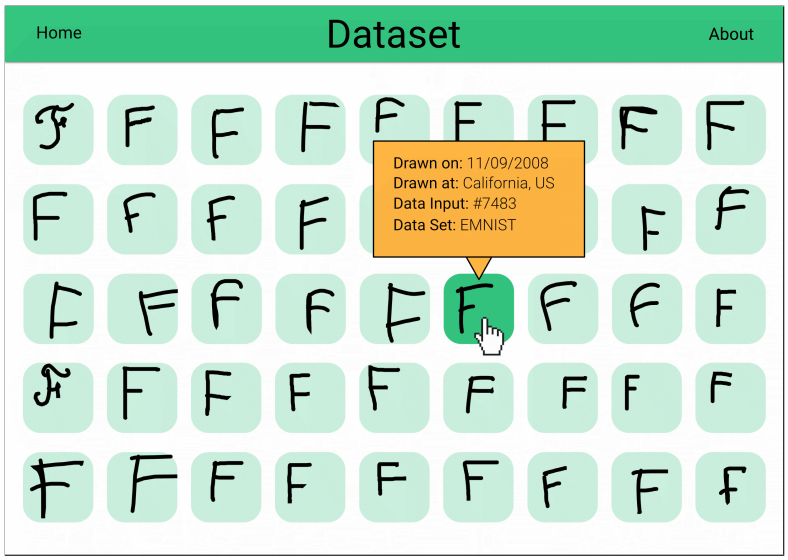


Figure 2.16: Meta-data display

2.7.2 User Experience

In terms of user experience, the goal is to have the website designed such that the user is led to interact with the website. This can be done by having the canvas stretch to the screen borders, thus giving the user an ample amount of 'open' data (as opposed to having a fixed-dimensions square). The canvas CSS should therefore be flexible and adjust to any screen size, desktop, tablet or mobile.

Furthermore, the web application should be very efficient strive to reduce the computing time at the back-end to as little as possible, in order to encourage the user to draw again (a 're-try button' should be displayed at the end of a cycle). When the program *is* loading, the front-end should have captivating visualisations that could also show information of the processes going on at the back-end to keep the user engaged. At no point should the application simply load for a period of time without any given context or information to the user.

Finally, it is important to note that the website is meant to cater to **all age groups**, and of varying level of interests. Consequently, the 'main' instructions should aim to be as simple and clear as possible, but the source code and database(s), other information sources, more in-depth analysis and further learning material should all also be provided as hyper-links to other websites (such as Github), references or textual explanations.

Chapter 3

Evaluation Design

3.1 Evaluation Criteria

First of all, to adequately evaluate the system, it first should be measured against the previously proposed essential features to ensure that the overall features are working correctly individually and collectively in a asynchronous system.

Secondly, basic user interface and experience guidelines should always work, ie clicking on a button should lead to the correct page corresponding to it, most unexpected exceptions should be caught, the website should be able to display 'error pages' in case of unforeseen crashes rather than native browser alerts.

Additionally, the loading times such as when launching the website, submitting the input, and visualising the training dataset should all be to a reasonable standard for 2017 and in the same league as other similar applications.

Furthermore, the system should be reviewed against the specifications document, and how well the application runs on clients other than the recommended ones, such Firefox or Safari instead of only Google Chrome for browsers. The front-end visualisations could likely be relatively heavy, and use a moderate amount of memory.

Finally, the website should not be invasive in any manner, and only the essential permissions should be requested. The privacy of the user should be respected, and no tracking should be done on the visitors. The data in-

put by the users should ideally be protected using a secure connection via `HTTPS` rather than standard `HTTP`. However, this can possibly be out of the developer's control as it depends on the choice of hosting website, as an external security certificate cannot be bought and implemented on the University departmental server space. Alternatives such as `github.io`, where a secure connection is possible, should be used in the eventuality of the web application being released to the general public.

3.2 Criteria Assessment

Assessing many of the criteria on efficiency will be a straightforward case of measuring response-time(s) of pages, images, animations and graphics on various devices and browsers with different memories. In terms of interface design philosophy, a record of un-matching button and page will have to be individually made, as well as any negative feedback on the user experience.

For the requested permissions, security certificates and `RAM` usage, one can check the the browser's developer console and the device's task manager for detailed information, as another way of assessing the system.

3.3 Participants

The training of the neural network and the development of the application will be accomplished at an individual level only, however for the purposes of system evaluation, 3rd party human participants **will** be involved to gather feedback for improvement and as a formal **beta-testing** assessment. It is important to note that all data will be completely anonymous and no individual tracking whatsoever will be done. The data will nonetheless be analysed and presented the final dissertation as part of the full project evaluation.

Participants will be first asked if the system works according to the given specification requirements and it if meets acceptable quality requirements. More specifically, they will be asked to rate various factors of the system, such as speed, different functionalities, reliability of outputs, general robustness, and innovation along with the UI/UX 'feel' and 'ease-of-use' of the website.

Additional questions could be on the methodology of the website as a teaching tool, how ambitious they feel the scope of the project is, and how effectively

the creator managed to convey concepts to a variety of users in innovative manners, and gave them enough content and interest in the discussed topics. General suggestions for improvements will of course be encouraged and welcome at any point.

This could be done through a written questionnaire or an online survey, subject to approval. The latter could later be permanently be part of the website as a feedback tool, if it is wholly made available to public.

I, Eklavya Sarkar, therefore hereby **confirm** having read, acknowledged and accepted all the points made in the following official University of Liverpool documents:

- Computer Science Department Ethical Procedure for Comp39X Projects 3rd Party Evaluation
- Comp39X Project 3rd Party Evaluator Consent Form
- Comp39X Project 3rd Party Evaluator Information Sheet

All of these documents can be found in full in the appendix of this document.

3.4 Testing

3.4.1 Hardware

As of the writing of this document (November 2017), the personal testing of the application is planned to be done on the following hardware devices: Macbook Pro 15" Retina (1st Gen), early 2013:

- OS: macOS Sierra
- Processor: 2.4 GHz Intel Core i7
- Memory: 8 GB 1600 MHz DDR3
- Graphics: NVIDIA GeForce GT 650M 1024 MB, Intel HD Graphics 4000 1536 MB

Samsung Galaxy S5 5.1"

- OS: Android 6.0.1
- Model: SM-G900F
- Processor: Quad-core 2.5 GHz Krait 400
- Memory: 2 GB LPDDR3

3.4.2 Software

Google Chrome's browser in developer mode also allows testing in various screen sizes which will be thoroughly used for UI testing. This allows to maintain a universal look and feel of the website across different devices, and isn't exclusively device-dependent.

The developer mode and a PC's task manager also allow to monitor any eventual memory leaks or excessive CPU usages, and will be used to optimise the web application. This is of relative importance as battery life is crucial on mobile devices, and a bad experience could deter people from using the website again. Network usage of website can also be looked at the decide whether or not to optimise or compress certain features.

3.4.3 Strategy

The planned strategy for testing this website would be similar to methods found in agile testing, and will be done throughout the development of this product, and then test the overall system and ensure that it works seamlessly with all components. At the end, a report with black box testing for all pages will be produced, containing the test ID (1,2,3 . . .), test data (canvas inputs), test data type (normal, erroneous, or extreme), expected result and actual results.

3.5 Expected Conclusions

As the main features of the project are definitely expected to be implemented and running, based on high-level human interaction design philosophies, good feedback on the UI/UX and main Self-Organising Map are expected.

It is likely the front-end visualisations might be too heavy for certain mobile devices working on low memory, as it will use modern technologies and frameworks such as `JavaScript`'s `D3.js`. This will not, however, be an issue if the system requirements correctly states the recommended systems for this program, as the point is in having the final product for systems capable of dealing with this type of load anyway.

Chapter 4

Data

The 'data' in this chapter only refers to the dataset(s) that will be used to train, test and adjust the weights according to the input letters of the neural network. They do not refer to the 3rd party feedback data explained in the previous evaluation section.

4.1 Ethical use of Data

For the purpose of this project, **only synthetic Data**, specifically `EMNIST` dataset, will be used. **No** human, real non-human, or any other type of data which requires approval from any Professional Body Ethical will be required.

The `EMNIST` dataset, based on the `MNIST` is freely available in the public domain which contains 28x28 pixel images of handwritten upper and lower-cases alphabets in binary format. Full reference can be found in the references at the end of this document.

4.2 Ethical use of Human Participants

For the actual realisation of this project, **no** human participants and therefore no permissions or approvals are required. The program is based on already available data, and only requires human interaction during the evaluation and usage phase, for which the consent form has been appended as mentioned previously.

Chapter 5

Review Against Plan

5.1 Progress

So far the specifications and design documents have been completed in their entirety on time. This includes almost all of the research on topic and the majority of the planning for the actual implementation. All of the Gantt Chart's internal task deadlines have been met on time, with the exception of a few system design diagrams that were slightly rushed but on schedule for the external deadlines.

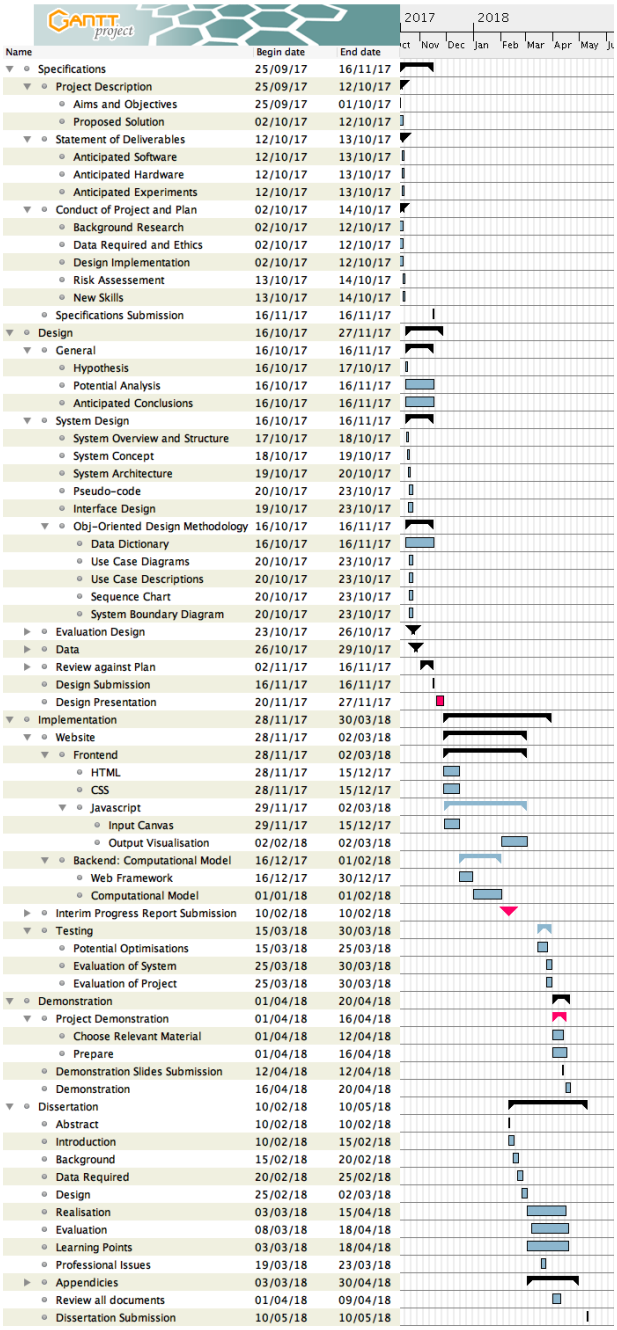
Nonetheless the time could've been better managed, as the research part of this project took longer than expected and notions of the bio-computation only fully became clear after a class test.

5.2 Changes

No significant alterations were made to the initial Gantt Chart as the project was well researched and mapped out from the start.

5.3 Gantt Chart

The complete Gantt Chart:



Name	Begin date	End date
Specifications	25/09/17	16/11/17
Project Description	25/09/17	12/10/17
Aims and Objectives	25/09/17	01/10/17
Proposed Solution	02/10/17	12/10/17
Statement of Deliverables	12/10/17	13/10/17
Anticipated Software	12/10/17	13/10/17
Anticipated Hardware	12/10/17	13/10/17
Anticipated Experiments	12/10/17	13/10/17
Conduct of Project and Plan	02/10/17	14/10/17
Background Research	02/10/17	12/10/17
Data Required and Ethics	02/10/17	12/10/17
Design Implementation	02/10/17	12/10/17
Risk Assessement	13/10/17	14/10/17
New Skills	13/10/17	14/10/17
Specifications Submission	16/11/17	16/11/17
Design	16/10/17	27/11/17
General	16/10/17	16/11/17
Hypothesis	16/10/17	17/10/17
Potential Analysis	16/10/17	16/11/17
Anticipated Conclusions	16/10/17	16/11/17
System Design	16/10/17	16/11/17
System Overview and Structure	17/10/17	18/10/17
System Concept	18/10/17	19/10/17
System Architecture	19/10/17	20/10/17
Pseudo-code	20/10/17	23/10/17
Interface Design	19/10/17	23/10/17
Obj-Oriented Design Methodology	16/10/17	16/11/17
Data Dictionary	16/10/17	16/11/17
Use Case Diagrams	20/10/17	23/10/17
Use Case Descriptions	20/10/17	23/10/17
Sequence Chart	20/10/17	23/10/17
System Boundary Diagram	20/10/17	23/10/17
Evaluation Design	23/10/17	26/10/17
Data	26/10/17	29/10/17
Review against Plan	02/11/17	16/11/17
Design Submission	16/11/17	16/11/17
Design Presentation	20/11/17	27/11/17
Implementation	28/11/17	30/03/18
Website	28/11/17	02/03/18
Frontend	28/11/17	02/03/18
HTML	28/11/17	15/12/17
CSS	28/11/17	15/12/17
Javascript	29/11/17	02/03/18
Input Canvas	29/11/17	15/12/17
Output Visualisation	02/02/18	02/03/18
Backend: Computational Model	16/12/17	01/02/18
Web Framework	16/12/17	30/12/17
Computational Model	01/01/18	01/02/18
Interim Progress Report Submission	10/02/18	10/02/18
Testing	15/03/18	30/03/18
Potential Optimisations	15/03/18	25/03/18
Evaluation of System	25/03/18	30/03/18
Evaluation of Project	25/03/18	30/03/18
Demonstration	01/04/18	20/04/18
Project Demonstration	01/04/18	16/04/18
Choose Relevant Material	01/04/18	12/04/18
Prepare	01/04/18	16/04/18
Demonstration Slides Submission	12/04/18	12/04/18
Demonstration	16/04/18	20/04/18
Dissertation	10/02/18	10/05/18
Abstract	10/02/18	10/02/18
Introduction	10/02/18	15/02/18
Background	15/02/18	20/02/18
Data Required	20/02/18	25/02/18
Design	25/02/18	02/03/18
Realisation	03/03/18	15/04/18
Evaluation	08/03/18	18/04/18
Learning Points	03/03/18	18/04/18
Professional Issues	19/03/18	23/03/18
Appendicies	03/03/18	30/04/18
Review all documents	01/04/18	09/04/18
Dissertation Submission	10/05/18	10/05/18

Acronyms

AJAX

Asynchronous JavaScript and XML.

BMU

Best Matching Unit.

D3

Data-Driven Documents.

GUI

Graphical User Interface.

ML

Machine Learning.

SOM

Self-Organising Map.

Glossary

Ajax

Ajax is a set of Web development techniques to create asynchronous Web applications that allows for Web pages and applications to change content dynamically without the need to reload the entire page.

Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications.

D3.js

D3.js is a JavaScript library for producing dynamic, interactive data visualizations in web browsers.

Flask

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine.

jQuery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.

Machine Learning

A sub-field of Artificial Intelligence.

Unsupervised Learning

.

Bibliography

[1] [1702.05373v1] emnist: an extension of mnist to handwritten letters. https://arxiv.org/abs/1702.05373v1. (Accessed on 11/16/2017).

[2] Artificial neural network - wikipedia. https://en.wikipedia.org/wiki/Artificial_neural_network. (Accessed on 11/16/2017).

[3] Cluster analysis - wikipedia. https://en.wikipedia.org/wiki/Cluster_analysis. (Accessed on 11/16/2017).

[4] The emnist dataset — nist. https://www.nist.gov/itl/iad/image-group/emnist-dataset. (Accessed on 11/16/2017).

[5] Machine learning — coursera. https://www.coursera.org/learn/machine-learning/. (Accessed on 11/16/2017).

[6] Mnist handwritten digit database, yann lecun, corinna cortes and chris burges. http://yann.lecun.com/exdb/mnist/. (Accessed on 11/16/2017).

[7] Neural net for handwritten digit recognition in javascript. http://myselph.de/neuralNet.html. (Accessed on 11/16/2017).

[8] Quick, draw! https://quickdraw.withgoogle.com/. (Accessed on 11/16/2017).

[9] Self-organizing map - wikipedia. https://en.wikipedia.org/wiki/Self-organizing_map. (Accessed on 11/16/2017).

[10] Som tutorial part 1. http://www.ai-junkie.com/ann/som/som1.html. (Accessed on 11/16/2017).

[11] Visualizing mnist: An exploration of dimensionality reduction - colah's blog. http://colah.github.io/posts/2014-10-Visualizing-MNIST/. (Accessed on 11/16/2017).

[12] Welcome — flask (a python microframework). http://flask.pocoo.org/. (Accessed on 11/16/2017).

[13] ŷhat — self-organising maps: An introduction. http://blog.yhat.com/posts/self-organizing-maps-1.html. (Accessed on 11/16/2017).

[14] Emile Fiesler and Russell Beale, editors. *Handbook of Neural Computation*. Oxford University Press, 1997.

[15] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.

[16] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[17] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[18] TensorFlow. Mnist for ml beginners. https://www.tensorflow.org/get_started/mnist/beginners. (Accessed on 10/15/2017).

[19] Yifan Wang. Artificial neural networks: Kohonen self-organizing maps (soms). Bachelor thesis, University of Liverpool, May 2015.

[20] J. Zurada. *Introduction to Artificial Neural Systems*. West Publishing Co., St. Paul, MN, USA, 1992.

UNIVERSITY OF LIVERPOOL

Committee on Research Ethics

Student Projects: Honours Year Computer Science Project (Comp390), Honours Year Artificial Intelligence Project (Comp393), Honours Year Electronic Commerce Computing Project (Comp394), Honours Year Internet Computing Project (Comp395), Final Year First Semester 15 credit Project (Comp391), Final Year Second Semester 15 credit Project (Comp392), MEng Group Project (Comp591), MEng Individual Project (Comp592), MSc Project (Comp702)

Student Project 3rd Party Evaluator Information Sheet

Title of the Student Project: **Kohonen Self-Organising Maps**

Module: **COMP39X**

Student: **Eklavya Sarkar**

1st Supervisor: **Irina V. Biktasheva**

You are being invited to participate in a Computer Science Department student project 3rd party evaluation. Before you decide whether to participate, it is important for you to understand why the 3rd party evaluation of the student project is being done and what it will involve. Please take time to read the following information carefully and feel free to ask us if you would like more information or if there is anything that you do not understand. Please also feel free to discuss this with your friends, relatives and GP if you wish. We would like to stress that you do not have to accept this invitation and should only agree to take part if you want to.

Thank you for reading this.

1. **What is the purpose of the student project?**

The aim of the Honours year, MEng, and MSc student projects at the Department of Computer Science is to give the students an opportunity to work in a guided but independent fashion to explore a substantial computing problem in depth, making practical use of principles, techniques and methodologies acquired elsewhere in the course. Among the learning outcomes of the projects is the students' ability to evaluate in a critical fashion the work they have done, and to place it in the context of related work.

A computer system 3rd party evaluation/testing is among the main evaluation techniques in a Computing project, important for the students in order to demonstrate their project learning outcomes and skills.

2. Why have I been chosen to take part?

The students usually approach and ask their client(s), potential user(s), and/or friends and family who are either most close to the project or simply capable to provide the feedback on the computing system. The 3rd party evaluators will be told what for and how their feedback will be used. If the client(s)/potential user(s)/friends and family agree, the students will obtain a signed consent form from them.

3. Do I have to take part?

Your participation in the project 3rd party evaluation is voluntary and you are free to withdraw at anytime without explanation and without incurring a disadvantage.

4. What will happen if I take part?

In the end of the project, the student will ask you to try the computing system s/he delivered during the project and to provide your feedback by filling in a 3rd Party Evaluation Questionnaire. This does not involve any audio/visual recording.

5. Expenses and / or payments

No expenses/payments are needed or available for the participants.

6. Are there any risks in taking part?

The 3rd party evaluation of the project computer systems happens routinely at the Department and/or by the 3rd party evaluators at their own safe sites. We do not foresee any risks or hazards created specifically by the 3rd party evaluation process. No potential physical or psychological adverse effects are anticipated.

7. Are there any benefits in taking part?

There are no intended benefits in taking part in the student project 3rd party evaluation.

8. What if I am unhappy or if there is a problem?

If you are unhappy, or if there is a problem, please feel free to let us know by contacting Dr Irina Biktasheva (for all projects other than MSc Projects, ivb@liv.ac.uk) or Dr Russell Martin (for MSc Projects, Russell.Martin@liverpool.ac.uk) and we will try to help. If you remain unhappy or have a complaint which you feel you cannot come to

us with then you should contact the Research Governance Officer at ethics@liv.ac.uk. When contacting the Research Governance Officer, please provide title of the project (so that it can be identified), the student involved, and the details of the complaint you wish to make.

9. Will my participation be kept confidential?

Yes, your participation in the computer system 3rd party evaluation/testing will be kept confidential. The 3rd party evaluation data will be anonymised so that no individual can be recognised, and stored responsibly as part of the project report on the university's servers.

10. What will happen to the results of the student project 3rd party evaluation?

The anonymised feedback data provided by the 3rd party evaluators will be used as a reference for the student project report Evaluation chapter. The project report will be assessed solely according to academic marking guidelines, and stored responsibly on the university's secure servers.

11. What will happen if I want to stop taking part?

You can withdraw at anytime, without explanation. Your 3rd party evaluation results up to the period of withdrawal may be used, if you are happy for this to be done. Otherwise you may request that they are destroyed and no further use is made of them. Anonymised 3rd party project evaluation results may only be withdrawn prior to anonymisation.

12. Who can I contact if I have further questions?

Final Year and MEng Student Project Coordinator
Dr Irina Biktasheva
Room 3.04, Ashton Building
University of Liverpool
L69 3BX
Phone: 0151 795 4267

MSc Student Project Coordinator
Dr Russell Martin
Room 3.19, Ashton Building
University of Liverpool
L69 3BX
Phone: 0151 795 4256

UNIVERSITY OF LIVERPOOL

Committee on Research Ethics

PARTICIPANT CONSENT FORM

Title of Research Project: **Kohonen Self-Organising Maps**

Student: **Eklavya Sarkar**

1st Supervisor **Irina V. Biktasheva**

		Please initial box
1.	I confirm that I have read and have understood the information sheet on the above student project computer system 3rd party evaluation. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.	
2.	I understand that my participation is voluntary and that I am free to withdraw at any time without giving any reason, without my rights being affected. In addition, should I not wish to answer any particular question or questions, I am free to decline.	
3.	I understand and agree that once I submit my 3rd party evaluation questionnaire it will become anonymised and I will therefore no longer be able to withdraw my data.	
4.	I understand that my responses will be kept strictly confidential. I give permission for members of the project team to have access to my anonymised responses. I understand that my name will not be linked with the project materials, and I will not be identified or identifiable in the report or reports that result from the project.	
5.	"Intellectual Property" shall mean any idea, invention, method, discovery, secret process, design, trade or service mark, copyright work (including computer software and all data and other information relating thereto), database rights, trade secret, confidential information, or any similar process, right or information. I agree that the Intellectual Property that arises from the student project should vest in the University of Liverpool.	
6.	I agree to take part as a computer system 3rd party evaluator in the above student project.	

Participant Name	Date	Signature
Eklavya Sarkar Name of Student taking consent	Date	Signature
Irina V. Biktasheva Project 1st Supervisor	Date	Signature

Principal Investigator:
Name
Work Address
Work Telephone
Work Email

Student Researcher:
Name **Eklavya Sarkar**
Work Address **A325 Dover Court, 15 Great Newton St, Liverpool,**
Work Telephone **07561044354**
Work Email **sgesarka@liv.ac.uk**

UNIVERSITY OF LIVERPOOL

Participant Name:

Evaluator Name: Eklavya Sarkar

1st Supervisor Name: Irina V. Biktasheva

You have been invited to participate in the evaluation of this website. For all following questions, please rate them from a scale of 0 to 10, where 0 is the least likeable and 10 the most likeable.

1.	How well does the website work as described in the specifications document ?	
2.	How well does it meet acceptable quality requirements ?	
3.	How well do you feel it correctly classifies your input letter ?	
4.	Could you draw without issues from both finger and mouse ?	
5.	How much would you rate the UI/UX feel and ease-of-use ?	
6.	How ambitious do you feel is the scope of this project ?	
7.	How well is the content taught and conveyed ?	
8.	How much of an interest on the subject did give you ?	

Do you have any other general comment for feedback ?

Participant

Date:

Signature:

Evaluator

Date:

Signature: