Imperial College London

Department of Computing

BEng Individual Project Final Report

Postr

The Poster Competition Voting System

Author:
Steven Kingaby

Supervisor:
Dr. Marc Deisenroth

Second Marker:
Dr. Aldo Faisal

Abstract

The objective of this project is to create a solution for determining the most appreciated poster at a poster event. Poster events are used in academic conferences for presenting contributions. However this is not limited to academic applications. At a given event which features poster sessions, there could be hundreds of poster presentations that one could attend. Due to the large number of possible posters it may not be feasible to attend every presentation. From the perspective of an author a poster session gives the opportunity to disseminate his work.

The solution provided by this project allows conference attendees with a camera equipped mobile phone to vote for their favourite posters by capturing an image of the poster and submitting it to a server-side leaderboard.

The chosen solution implements a mobile application client and a server-side administration controller. Through the use of geolocation, the nearest events are presented to a user. The mobile client captures the poster image and uploads it to the server where an OCR service is used to capture the poster title and author. For each poster that is uploaded to the server, its vote count is incremented. Furthermore the project considers the use of QR codes or NFC as an alternative to OCR.

In order to justify this project as a usable poster competition voting system, a thorough evaluation of the accuracy of the OCR capture and interpretation at differing distances and angles of incidence was performed. For instance at an angle of incidence of 60 degrees, an accuracy rate reduction of 12.5% was reported, relative to a head on capture.

Acknowledgements

I would first like to thank my supervisor Dr Marc Deisenroth for proposing this project and his continued support and motivation throughout. I would also like to thank Dr Mark Wheelhouse for allowing me to demonstrate the project at the Department of Computing's 2nd Year Poster Session. Thanks also to my family and friends for their consistent support not only throughout the scope of this project, but my three years at Imperial College.

Contents

Chapter 1

Introduction

1.1 Motivation

Around the world thousands of people travel to assemble at various conferences and workshops in order to broaden their knowledge about a certain topic and to network amongst peers in the same field. To give a sense of the grandeur of the sheer number of such events available, at the month of writing there were 51 IEEE conferences running [1]. Not only are these conferences restricted to being sponsored by one single association, but they also only fall under the theme of technology. Factoring in other fields such as science, finance, and considering how many more topics fall under their scope, one can imagine how much the above number of conferences running worldwide will swell.

At these conferences a popular means of presenting information is in the form of the poster. This popularity extends to how often, in scientific conferences a separate room or area is set aside to host a poster session. In such sessions the authors stand beside their posters and interact with any audience members, answering any questions they have about their display. A poster benefits from being able to display information concisely and in an engaging graphical manner. How well presented the information in a given poster, i.e. the layout of it, fonts used, etc, hinges on the preferences of the audience.

The use of captivating graphics allows for a better exchange of informa-

[1] https://www.ieee.org/conferences_events/index.html

tion to the audience. The better that information can communicated to an audience, the more information that is retained. Posters as we know today first appeared in the 19th century. Following advancements in lithography, a printing process, posters benefited from a new range of colours and textures. Since that era posters have been used for their eye-capturing appeal to push political agendas, advertise products, and much more. An example of the type of posters relevent to this project is shown in Figure 1.1.

What concerns us in this project is the ability of posters' to communicate data. How well the data is communicated, and the quality of the data itself, influences an audience's opinion of the poster. Often in conference poster sessions, competitions take place to determine the most popular poster, which leads us to our problem statement.

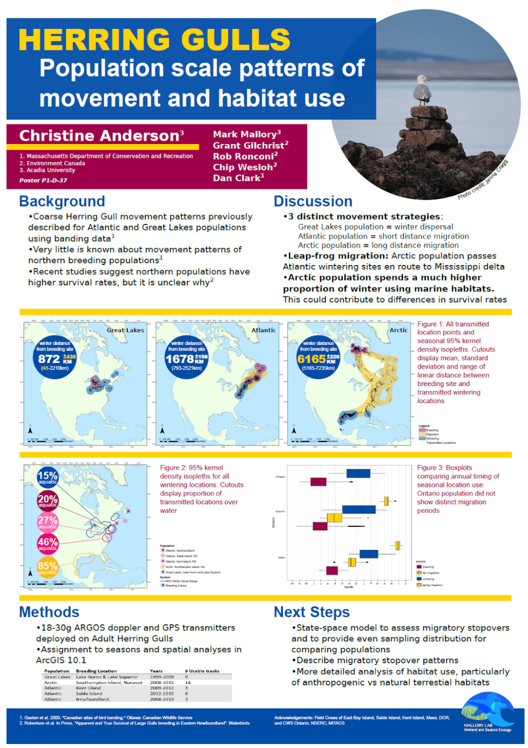

Figure 1.1: Example of a poster presenting academic research [2]

[2]http://betterposters.blogspot.co.uk/

1.2 Problem Statement

The problem for this project is create a system to find the most popular poster at a given event. The solution to this could potentially be applied to a number of different situations, but for the purpose of this project we assume the following scenario. A conference is being held for a non-specific field of interest. At the conference contributors are invited to post posters as summary statements of their research in predefined locations throughout the venue. Contributors can present their posters as described in the previous section. A competition for the most popular poster will be held and the winner will be announced at the event.

It is a requirement that posters are preregistered for this competition by the competition's organisers. Over the course of the event each delegate can vote for one or more posters. The delegate may vote for as many posters as they want but each vote carries the same weight. In order to vote in the competition, delegates are required to be registered with the Postr system. Any votes for non-registered posters will be ignored when determining the most popular poster at the event.

1.3 Objectives

The main objectives of this project is to create a system, Postr, that will be able to:

- Determine the most popular poster in the competition.
- Capture images of and vote for posters.
  - Extract the poster title and author from an image.
- Present the leaderboard for the competition.
- Provide a server-side database solution to capture poster images and calculate the leaderboard and results.
- Ensure there are no duplicate entries in database.
- Ensure that there is no double voting.
- Operate on iPhone 5 and above, and on iOS 7.0 and above.

1.4 Contributions

This project has provided a platform on which poster voting competitions can be hosted. Users are able to use its voting functionality to take a picture of a poster, thereby voting for it. A UI approach for the client was taken so that the poster capture process was as simple as possible. The details of this feature is covered in the “ Voting for a Poster” section of the Implementation Chapter.

Postr supports the means to notify of users of nearby poster session events, through the use of geolocation services. This gives a simpler approach for the user to having to search for a particular event. Furthermore this project supports user registration and maintanence. The details of both these features can be found in the respective “Retrieving Nearby Events” and “Login and Registration” sections of the Implementation Chapter.

In the Evaluations Chapter the report provides empirical data regarding the use of OCR capture from a mobile phone camera. This empirical data demonstrate how accurate a system using OCR can be. Furthermore this project provides an evaluation of the 3 possible data capture techniques: OCR, QR codes and NFC. This can be found in the case study section of the Background Chapter.

Figure 1.2: A Poster Session [3]

[3]http://www.fotoglz.com/poster-sessions-at-conferences.html

Chapter 2

Background

Firstly we will explore an existing solution to the problem statement given in the Introduction Chapter. By analysing an existing solution, the merits and flaws in this implementation have been identified and used to improve the presented system in this project. Afterwards we will discuss the relevant technologies and theories that Postr is built upon.

2.1 Existing Solutions

2.1.1 Case Study: Poster in my Pocket

In order to develop an effective system that meets the objectives outlined in the introduction, we will consider the following analysis of the iPhone app *Poster in my pocket*. I chose this app to analyse as it generally follows the same themes as in this project's problem statement. Namely that is designed to be used in conference environment like in the the assumed scenario in section 1.2, and that it involves the processing of information displayed on a poster.

What distinguishes it from this project's app is that it involves no notion of a competition or voting for posters. The objective of *Poster in my pocket* appears to be to allow users to download their favourite posters from a conference. It is worth mentioning that following an exhaustive search on Apple's App Store, an app specifically for a poster competition at a conference or workshop could not be found.

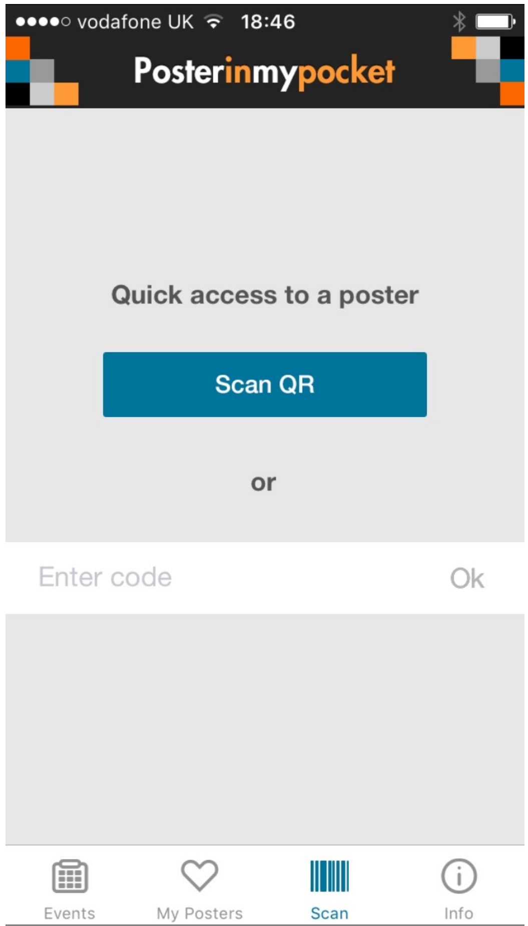

Figure 2.1: *Poster in my pocket* UI

In figure 2.1 we see the UI open on the "Scan" tab which leads us to our first point of discussion. In order to download a poster using *Poster in my pocket*, users scan a QR code that is expected to be present on a given poster. While barcodes are one-dimensional numeric codes, QR codes are two-dimensional one. This difference allows a QR code to hold up to 3Kb of data, which is more than enough to store information such as the title and author of the poster. Which leads us to the question, should this project system make use of QR codes?

However the main drawback of using QR codes in this project is that users will only be able to vote for posters that have QR codes on them. One could apply this argument in the debate to use say NFC technology; it would add further complication and cost for poster authors to attach the NFC chip to their poster and if they don't, people wouldn't be able to vote for it. This would defeat the purpose of having voting system to elicit users' poster preferences and potentially discourage participation in the competition. If they couldn't even vote for their preferred posters, why bother taking part in the competition? Furthermore if the QR code is visually corrupted or damaged, again a user wouldn't be able to vote for that poster. Ergo despite its merits, QR codes are not suitable for this project's system. Instead this project's solution will adopt an OCR approach to process the relevant information in a poster.

What one can appreciate in *Poster in my pocket* is its minimalist design. It's very easy to navigate through the different views of the app and know how to get from one point in the app's storyboard (map of all the different views) to another. It directly solves it's objectives of allowing user's to download posters they find interesting at conferences, without having excess features that crowd the app's core functionality. Elements of this app that I would incorporate in this project's solution include the smooth transitions it has when navigating from one view to another. Furthermore upon selecting the "My Posters" tab, a list of poster titles comes up on display. Each entry in the list also has a thumbnail with an image of the relevant poster, which is a nice aesthetic touch and something that could easily be incorporated into this system.

One final note will be made of the tab bar used in *Poster in my pocket*. The same tab bar interface is used in other prominent apps such as *Whatsapp* or *Facebook Messenger*. Due to it's popularity amongst iOS apps it would be an idea to find another approach to allow users to navigate through this project's app. One alternative could be a single menu icon which when selected will display on the view possible navigation choices to make.

2.2 Optical Character Recognition

An ongoing theme in this project is the idea of data capture. Data exists in many ways and can be present in a form that is readable to humans or readable to machines. For example in the context of this project the title and author on a poster is clearly readable by humans but the data stored in, say, a bar code would not be directly readable to humans. One way to acquire and then interpret the data in a bar code, would be to illuminate it with a low intensity laser beam and the reflected light would be detected and measured. An analog signal would be generated with varying voltage that is dependent on the intensity of the reflected light. This signal would be digitised and then interpreted to extract the stored data in the bar code. This process is an example of how a machine would acquire the data in a bar code, which is not readable to humans.

As laid out in the objectives sections in the introduction, the title and author from an image of a poster will be extracted. For a human this data would be easily acquired through the interaction of the complex components of the human eye and brain. However for this project extracting the title and author of data must be done by a machine. To replicate human vision perception, Optical Character Recognition has been incorporated in this project. Optical Character Recognition, or OCR, is the process of electronically extracting text from images and it falls under scope of Computer Vision, Pattern Recognition and Machine Learning [1].

OCR first appeared commercially in the 1950s [2]. Commercial computers took the form of mainframes and due to their expensive price tag, only large government agencies and corporations possessed. Data was entered in the mainframe using punched cards and depending on the amount of data, processing time could be lengthy. Naturally to make the most out of these expensive machines, methods to reduce the turnaround time were sought after [3]. OCR was one such cost effective way to deal with the large amounts of data to be processed.

The late 1960s saw the introduction of fonts such as OCR-A and OCR-B were developed specifically to be read by OCR systems [1]. While still read-

[1]https://sourceforge.net/projects/ocr-a-font/files/OCR-A/1.0/

able to humans, these fonts were developed to improve the accuracy rates of OCR systems. For instance fixed spacing and fixed character widths were introduced, as well as thick strokes being used to form characters. OCR-A was a standard American font and OCR-B was a standard European font. As can be seen in the figure below, the OCR-B font has a much more natural appearance to OCR-A.

Figure 2.2: Top: OCR-A, Bottom: OCR-B [2]

Since then OCR has been introduced to a wide range of applications from automating postal systems, number-plate readers, to being used as a means of signature verification. The level of sophistication of an OCR system can measured by the speed and accuracy of its text extraction. It is expected that a very basic OCR system is able to recognize characters given in a fixed font whereas a more advanced one is able to correctly process handwriting. However the accuracy is also highly dependent on the quality of the given image. Difficulties that might be encountered in an image of a poster include the variations in fonts and changes in the serifs of characters, the mixture of text and graphics, and any obstructions to the characters such as someone standing in the way of the poster at the time of image capture. We will now discuss in order the steps that lie behind OCR. A diagram depicting the steps of OCR is given on the next page.

[2] https://www.nr.no/ẽikvil/OCR.pdf

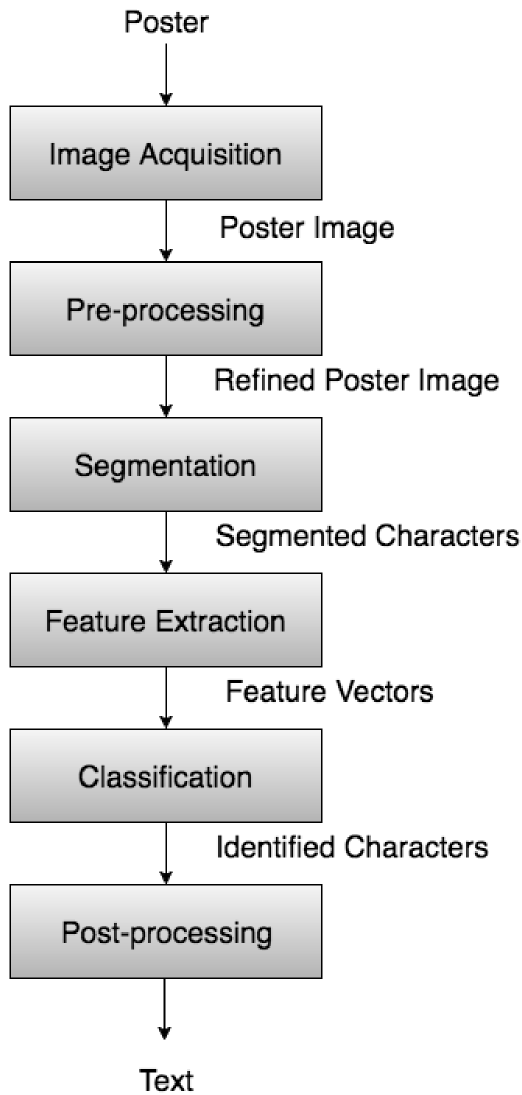

Figure 2.3: Steps of the OCR process

2.2.1 Image Acquisition

Initially a digital, RGB, image is constructed from a document, i.e. a poster. This image is then converted to a gray-scale image which consists of pixel values between 0 and 255. The gray-scale image is then converted to a black and white image (pixel values are either 1 or 0) during a process called thresholding. During this process a threshold value is set. If a gray level is above the threshold level it is identified as white and if it is below it is identified as black. In the event that it is white the pixel value is set to 0 and if black it is set to 1.

A straight forward approach to this binarization process is to set a constant threshold value. However when taking into account properties such as contrast and brightness, textual information in the document can be lost. This is can be seen in the bottom image of Figure 2.4.

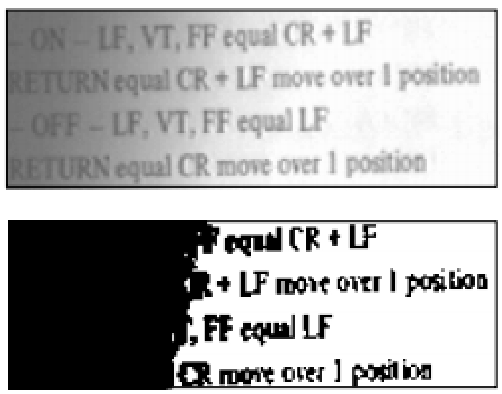

Figure 2.4: *TOP:* original image, *BOTTOM:* image after thresholding [3]

A constant threshold value could only really be used if it can be assumed that the inputted poster image has a uniform background and that there is a high level of contrast between the text and is surroundings. This is not an assumption that can be made with the poster images considered in this project. A more advanced method would be to vary the threshold value in response to the brightness and contrast of the digital image [4]. The results of this more advanced method is demonstrated in Figure 2.6, which shows that far less textual information has been lost.

[3]https://www.nr.no/ eikvil/OCR.pdf

ON - LF, VT, FF equal CR + LF
RETURN equal CR + LF move over 1 position
OFF – LF, VT, FF equal LF
RETURN equal CR move over 1 position

Figure 2.5: Image after using a variable threshold level [4]

An adaptive threshold could be introduced by partitioning the image into non-overlapping sections. Within each section an individual threshold value would then be set. Despite its merits, an adaptive threshold method would demand more computational power and more memory. This drawback would not be suitable if the OCR system was local to a hand held device such as a mobile phone, as was initially considered early on in this project. Hence it would have to be considered whether the improved accuracy in text recognition outweighs the computational costs.

2.2.2 Pre-processing

At this stage image pre-processing techniques are applied to the image in order to improve the accuracy of the final classification. Essentially the aim of this step is to remove what defects or noise might be present in the image. For instance if the image was not properly aligned in the image acquisition step, the lines of characters may not be perfectly vertical or horizontal. In the early years of OCR image acquisition would have been achieved with the use of an optical scanner. However in recent years a more popular means of capturing images would be by using a digital camera. This form of image acquisition is more relevant to this project, as users of Postr would use the camera on their iPhone to take a picture of there posters.

An image taken by a phone camera is all the more prone to having skewed text. This could occur in the case when the photo was taking, the camera's line of sight was not orthogonal to the poster. The natural solution would be to rotate the capture image. To do this quadrilaterals that circumscribe bodies of text are identified. After evaluating the dimensions of the newly rotated quadrilateral, coordinates within the original quadrilaterals would be

[4] https://www.nr.no/ eikvil/OCR.pdf

transformed to their new positions, rotating the text as required [6].

In some cases the described method of rotation would not suffice. For example in Figure 2.6 the lines of text are curved towards a point. Furthermore the width of characters within each line progressively gets smaller as it approaches the centre of the image. This effect could be observed in images taken by Postr users if the poster was mounted on a curved surface. To rotate this a non-linear transformation would have to be employed, the details of which are unnecessary for the scope of this report.

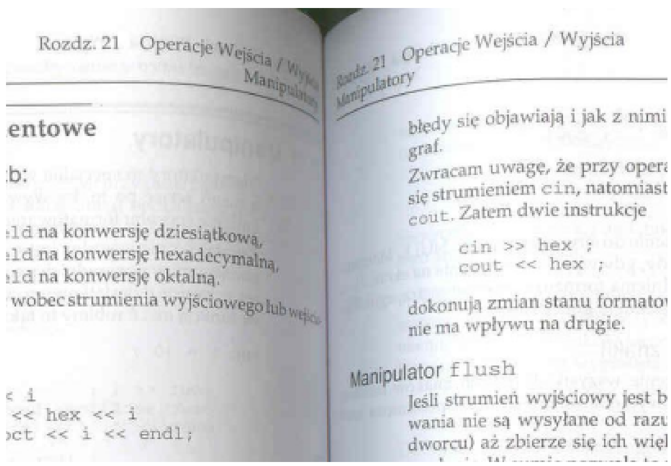

Figure 2.6: Lines of text curved towards the centre [5]

Noise in a poster image can take place in the form of broken or blurred characters. Such noise can be eliminated by "smoothing" out these characters. This smoothing effect can be achieved through filtering techniques such as applying a local average filter. A window in which, e.g. a filter, is applied would be translated across the image, smoothing out defects present.

2.2.3 Segmentation

Segmentation is the process in which individual characters in text are isolated. Firstly regions of text are distinguished from graphics. Then lines of text identified. Given that in the first step the image is thresholded, the

[5] http://wbieniec.kis.p.lodz.pl/research/files/07_memstech_ocr.pdf

characters should appear in black and everything else is white. Textual identification is loosely based on the principle that lines of text are separated from each other by white space. So a natural solution would be to locate horizontal strips with a high density of black pixels. Some segmentation methods make use of the Hough Transform, which incorporates mathematical functions that describe boundary curves (e.g. curves that circumscribe text lines) [7]. The presence of drop letters, overlapping, skewed text can interfere with the segmentation [8]. As mentioned in the pre-processing section, skewed text can be tackled with rotation. However overcoming drop letters and overlapping text lines is more problematic and only advanced OCR systems would tackle them. Fortunately the appearance of drop letters in the posters this system will be dealing with are uncommon.

Next lines of text are segmented into words. This part of segmentation is similarly based on the principle that words are separated from each other by white space. Word segmentation is more straightforward to line segmentation given that words are unlikely to be touching one another. The final stage of segmentation is to segment the words into characters. The space between each character in a word is defined as “interletter space” [8]. A fixed font would have characters of the same width and there would be the same interletter space between characters. However for a proportional font neither of these properties would hold [6]. A typical way to determine whether a font is fixed or proportional is to compare two lines of text with the same number of characters (and white spaces). If they have the same length the font is fixed, otherwise it is proportional.

Character segmentation is much easier when a fixed font is used instead of a proportional font one, because the body in which each character resides varies. The OCR fonts in Figure 2.4 are examples of fixed fonts. Given that we have no control of the fonts used in posters registered in a competition, the OCR system used in Postr must be able to deal with proportional fonts effectively.

We will briefly discuss problems that might occur in segmentation. Firstly accents on a character might be falsely considered as noise. Moreover if a low threshold level was applied to the captured image, joints can occur between

[6] http://practicaltypography.com/monospaced-fonts.html

characters. Conversely if a high threshold level was applied, characters may appear to be fragmented when they are not supposed to be [4].

2.2.4 Feature Extraction

In this step the main attributes of the segmented characters are extracted and all unimportant information is stripped out. Once the relevant features have been extracted image features are compared against known character features to find the nearest match. One such algorithm that can be used to establish a match is k-nearest-neighbours. There are a number of different approaches that can be taken to achieve feature extraction, of which we will discuss a few:

Matrix Matching

One of the simplest methods to implement, here a matrix containing a character from the image is compared against a library of character matrices. The distance between the library matrix and the image matrix is compared, e.g. euclidean distance. If the two matrices fall within a pre-defined level of similarity, the image character is labelled as the corresponding character of the library matrix. This notion of establishing similarity is discussed further in the next section. While this method benefits from simplicity, it lacks the ability to handle rotated characters and is sensitive to any noise present in the image. It should also be noted that in this method no characteristics, or features, are extracted, ergo its simplicity.

Distribution of points

With this approach the statistical distribution of points are analysed to extract features from the character shape. For instance the frequency of how often, in certain directions, vectors cross the character. It should be noted that this particular technique requires low computation effort and is thus ideal for a mobile device such as a phone (assuming the OCR takes on the device).

Structural Analysis

Here geometric features. e.g. intersections between lines and loops, of the character shape are extracted. Such features are affected by rotation and translations, but overall as an approach structural analysis is very tolerant of noise [2].

2.2.5 Classification

Using the features extracted from the previous steps, the characters found in an image are "recognised" to belong to a character class. Generally the character features that are extracted in the previous step are represented numerically in the form of vector, which is known as a "feature vector". Consider the following three approaches that could be used in the classification stage:

Statistical Classifiers

With statistical classifiers we aim to use a scheme that minimizes the probability of the classification error; i.e. the probability that a character is misclassified [10]. One such classifier is Bayes' classifier, which works on the rule of conditional probability laid out by "Bayes Theorem".

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes' Theorem

Training data is used to calculate preliminary probabilities which will later be combined with Bayes' Theorem. For a feature vector of an image character and for every character class that exists, the probability that the character (represented by the feature vector) belongs to a certain class is calculated. The character associated with the feature vector is assigned to the class which has the highest probability [11]. The set of character classes includes an individual class for each alphabetical letter and also other symbols that the OCR system is capable of recognising. This is also case for the other classification methods explored in this section.

To conclude this classification method we will demonstrate how Bayes' Theorem is used [7]. Suppose that the feature vector is made up of the values (fea-

[7] http://www-inst.eecs.berkeley.edu/ cs61a/su10/proj/4-naive-bayes/proj4.pdf

tures) $F_1 ... F_N$. We want to calculate the following probability $P(character|F_1, ... F_N)$ for each character, and then select the one with the highest probability. By Bayes' theorem,

$$P(character|F_1, ... F_N) = \frac{P(character) \prod_{i=1}^{N} P(F_i|character)}{\sum_{j=1}^{M} \prod_{i=1}^{N} P(F_i|character_j)}$$

Calculating probabilities

In the denominator M represents the number of characters that the OCR system is capable of recognising. $P(character)$ is calculated by running the Bayesian classifier before hand on training sets of hand-classified text and dividing the frequency of *character* by the sum of the frequencies of all characters [12]. $P(F_i|character_j)$ can be directly derived knowing what feature values are associated with a $character_j$ class. The character represented by the feature vector is classified into the character class that yields the highest probability.

Neural Networks

A neural network is a structure that has weighted connections between its units or neurons. These structures appear in biology in the form of complex interconnections of neurons in the central nervous system of animals. The use of (artificial) neural networks within OCR classification is data-driven and is a form of supervised learning. By data-driven we mean that training data is provided to alter the structure of the neural network so that when a feature vector is inputted the correct character class is assigned to the corresponding character in the image. Furthermore by supervised learning we mean that the training data provided consists of examples of input feature vectors along with their corresponding target vectors (i.e. character classes). With unsupervised learning the corresponding target vectors would not be provided in the training data [13].

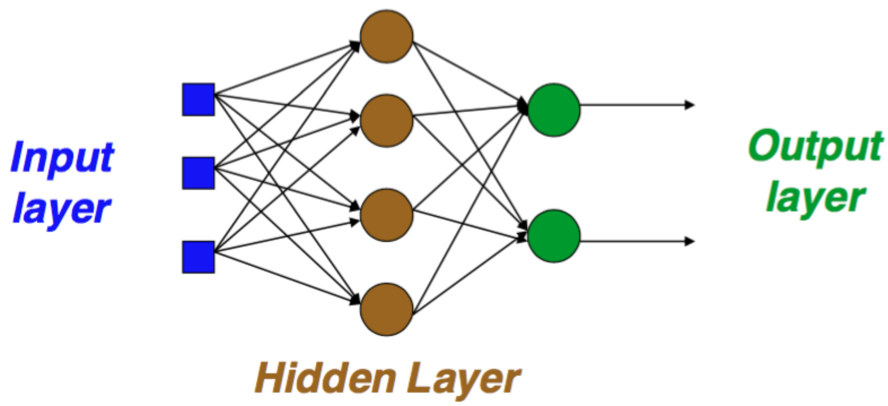

Figure 2.7: An example structure of a neural network [14]

Before and after the training process the number of neurons present and how they are connected remains the same. What changes is the weights of the edges between neurons. Note that weights are numerical values assigned to the edges. During training the weights within the network are adjusted until the desired output is given. A brief description will now be given to illustrate how a neural network would process a feature vector outputted from the previous feature detection OCR step.

Suppose an input vector $\mathbf{x}$ is made up of the values $x_1 \ldots x_n$. The values in this input vector come from the previous layer of neurons in the neural network. The weights that connect the neurons from the previous layer, that gave rise to the values in the input vector, are defined as $w_1 \ldots w_n$. The following net value is defined as $\sum_{i=1}^{n} w_i x_i$. This value is then passed through an "activation function". Examples of activation functions include the step function, sigmoid function and the sign function. The output of this activation function is passed onto the next layer of neurons.

Advantages to neural networks are that they are very tolerant of noise [15]. Furthermore because neural network computations are carried out in parallel, high computational rates can be achieved. Note that the neural network should not be trained while OCR is taking place but instead before. The time taken in the training phase is proportional to training iterations are used. The more iterations used, the smaller the classification error in assigning class labels to inputs from the training data. The same cannot be said of data from outside the training set. In fact after a certain point, an increase in iterations will yield an increase in classification error. This observation is known as "overfitting" and essentially means that the network does not generalise well to unseen input feature vectors [16].

Matching

The family of techniques that fall under this section relies on the notion of similarity, as explained in the Nearest Neighbour Search section. Classification of a character is to be based on the similarity it has with different character classes. For instance having already represented the character features in the form of a vector, one could use the euclidean distance to establish a similarity with a character class.

2.2.6 Post-processing

In the final step measures can be taken to increase the overall accuracy. No OCR system is can guarantee a 100% character recognition rate, especially when environmental factors like occlusions can interfere with the image capture of the poster. The inability to achieve this inspired the need of a nearest neighbour search in Postr, as explored later in the Design Chapter. Knowledge of what is to be expected in the OCR output can be used to improve accuracy. For instance one can define a list of words that is allowed to be present in the OCR system output [17]. The list could consist of the words present in the dictionary.

Furthermore this step provides an opportunity for error detection. For example the grammatical rules that govern the English language could be used to detect an error [18]. Knowing that a question mark can only appear at the end of the sentence, an OCR system would be able to flag an error if it found a question mark in the middle of a sentence.

How an OCR system handles such error depends on the application that it is being used in. One possible solution would be to redo the classification step, and then the post-processing step. With regards to Postr, having a less than desirable accuracy rate is permissible after OCR, since a nearest neighbour search is employed. This nearest neighbour search should return the desired output, i.e. the extracted title and author from a captured image of a poster.

2.3 Nearest Neighbour Search

Firstly we will discuss the motivation behind using a Nearest Neighbour Search. The OCR output of the poster image will not have a 100% accuracy rate, and will also have recognised text from the main body of the poster caught in the image. Parsing out the title and author would be problematic as one cannot make assumptions such as the title being present in the first line of text from the OCR output, and the author on the second line. The solution is to perform a Nearest Neighbour Search with the OCR output and the collection of poster titles and authors constructed from Postr's database of poster entries. The following is the Nearest Neighbour Search problem:

> Given a set X of points in a space Q and a query point $q \in Q$, what is the closest point in X to q [19].

The heart of any solution to this problem is to establish the distance between two points. The "closer" two points are, the more similar they are. Ergo the solution point to the Nearest Neighbour Search problem is one that is most similar to the query point. The Nearest Neighbour Search problem arises in the fields of Computer Vision and Pattern Recognition, as is discussed in the OCR section of this chapter. Listed below are two standard methods of calculating distance between two points $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $p \in [1..n]$. Note that here $\mathbf{x}$ and $\mathbf{y}$ are being treated as feature vectors where their constituent components, i.e. $x_1..x_n \in \mathbf{x}$, represent different numerical features in their respective points [20].

Euclidean Distance:

$$\sqrt{\sum_p (x_p - y_p)^2}$$

Manhattan Distance:

$$\sum_p |x_p - y_p|$$

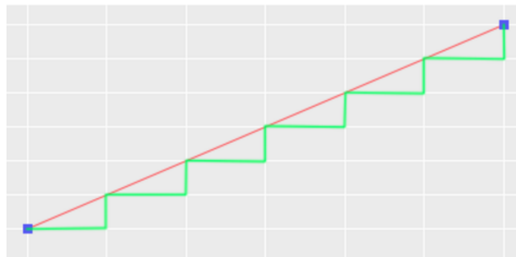

Figure 2.8: Euclidean and Manhatten distance

The Euclidean distance between two points can be thought of as the “diagonal distance” between the two, while the Manhattan distance is based upon a horizontal and/or vertical path between the two points. The differences between the two are illustrated in the Figure above.

There are numerous methods of solving the Nearest Neighbour Problem, each using different equations to measure distance. A variation of this problem, relevant to this project, is the K-Nearest Neighbour search. From here on in K-Nearest Neighbour shall be referred to as KNN. Rather than finding the most nearest neighbour to the query point, the top K nearest points are identified.

It is worth distinguishing that in the scope of Postr we are concerned with searching for the nearest points. However “KNN algorithms” are not strictly limited to search but can carry out other applications, such as classification, and does not necessarily mean it only outputs the K nearest neighbours. For instance KNN algorithms can be used to classify a query point based on the class of its K nearest neighbours. KNN algorithms can be used in the classification step of OCR, as discussed in the next section. Nevertheless in terms of direct application, Postr is concerned with a KNN search alone.

2.4 Authentication: Hashing and Salts

Postr incorporates password-based authentication in its system. Within its database a separate schema is set aside to store user credentials. A secure way to store credentials within a schema, a password file, etc is to store a hash of a password instead of the password itself. Hash functions have the useful property that in practise they are troublesome to reverse. Hence if a malicious user was able to obtain any stored credentials, it would be difficult to obtain the original password from the stored hash value.

The authentication process is as follows. A user presents his username and password to the system. Firstly a check is made to see if the username is present in wherever the system stores user credentials. Then a check is made to see if the provided password is correct. To do this a hash of the password is produced, and then this hash value is search for. If found access is granted, otherwise denied.

However storing hashed passwords is not sufficient to protect credentials from attacks such as an "offline dictionary attack". In such an attack, a database of common passwords and hash values is built up [21]. In the event that an attacker was able to obtain a stored hashed password of a user, the original password could be obtained by using the database. If the stolen hash is present in database, then the original password can be found.

Another line of credential defense is the use of salted hashes. A salt is randomly generated piece of data which is combined with a password via bitwise OR operations, before being inputted to the hash function. The salt is stored alongside the hashed passwords and usernames. Any attempted offline dictionary attack would have to construct a larger database in order to consider every possible salt. However if an attacker were able to obtain a salt unique to a particular user, an offline dictionary attack would still be feasible.

2.5 Location: Latitude and Longitude

As explained in later chapters, Postr requires a way to represent users' geographic location. To achieve this latitude and longitude will be used. Furthermore a way to determine the distance between two points, represented using latitude and longitude, is required. Note that both latitude and longitude are measured in degrees. Latitude represents how north or south a point is with respect to the equator. A point with a latitude of $0°$ will be positioned on the equator of the earth. Longitude represents how west or east a point is with respect to the Greenwich Meridian[8]. The Greenwich Meridian is the vertical equivalent of the earth's equator . Ergo a point with a longitude of 0° is positioned along the Greenwich Meridian.

When calculating the distance between two geographic points, the curvature of the earth will have to be considered. Due to Earth's spherical shape, using a euclidean method would only yield accurate results for small distances. Hence the following "haversine" formula is used [9]. Firstly the latitude and longitude points are converted from degrees to radians. Let (x_1, y_1) and (x_2, y_2) be the longitude and latidude respectively of two different points, measured in radians as opposed to degrees.

Haversine formula:

$$\text{let } \alpha = \sin^2(\frac{y_2 - y_1}{2}) + \cos y_2 \cdot \cos y_1 \cdot \sin^2(\frac{x_2 - x_1}{2})$$
$$\text{let } \beta = \arctan(\sqrt{\frac{\alpha}{1-\alpha}})$$

The distance between the two points is R $\cdot\, \beta$, where R is the earth's radius (6,371,000m) [10]. The haversine formula calculates the shortest distance between two points over the spherical surface of the earth.

2.6 Voting

To conclude this chapter, lets consider the broader theme of which our problem is a subset of. At the heart of it the general problem is to find a means to go about the capture and then processing of personal preference. The challenge that is addressed in this project is a realisation of this more general theme. The data we gather are the poster preferences of conference attendees that comes in the form of a vote. Furthermore we will process this data by evaluating the number of votes for each poster and then ranking these posters by popularity in a leaderboard.

[8]http://www.thegreenwichmeridian.org/tgm/articles.php?article=0
[9]http://www.movable-type.co.uk/scripts/latlong.html
[10]http://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html

The history of voting can be traced back to ancient Greece where the decision to expel a citizen was left to the community. Citizens would cast their vote by inscribing their preference on broken pieces of pottery [11]. In the present day the concept of capturing peoples' opinions has become an industry in itself. This industry ranges from the gathering of votes in an election campaign to social media, where "likes" for a given post are accumulated. A stream of information of peoples' preferences is just another drop in the ocean that is Big Data. The ability to capture opinion is central to the purpose of this project. Moving swiftly on, what concerns us in this section are the different types of voting systems that exist, and which ones are relevant to this project.

2.6.1 Plurality Voting

The simplest and most widespread form of voting is plurality voting. Also known as "first-past-the-post", in this voting system each voter votes for one preference. The preference that has the most votes wins [12]. The advantage of incorporating this voting system is its simplicity; it would be very easy to implement. Moreover in the given requirements of the project when one votes for a poster, its entry in the database in incremented by one. So an implementation of plurality voting in this app would follow naturally as there has to be an associated voting count for each poster.

However it is worth mentioning the disadvantages of plurality voting. In the context of voting, "majority" means more than half the votes. With plurality voting the winner is the one that has the most votes, even though it might have received less than a majority of the overall number of votes. So arguably the winner of a plurality vote might not be a representative choice of the votes. This leads us to our next voting system.

2.6.2 Majority Voting

The majority voting is distinguished from the plurality voting in that in order to win one must have a majority of votes. In the United States, a

[11] https://en.wikipedia.org/wiki/Ballot
[12] https://www.mtholyoke.edu/acad/polit/damy/BeginnningReading/plurality.htm

majority system is used as the foundation to political decision making. The immediate advantage of this voting method is that it is clear the overall vote is representative of the majority of the voters. Furthermore the majority voting system follows May's Theorem which, in the context of voting, states that there is one clear winner [13]. Moreover in the event that there is a tie let us assume that except for one voter, all other voters do not change their preference. However the one excepted voter changes his vote and whatever preference that voter changes to will win.

As majority voting followings May's theorem, it can be argued that it is a fair way to determine a consensus of a group of voters. However the disadvantages of this voting system lie in the event that there is no majority vote. If there is no majority event, actions like having to host a second round of voting would have to take place. The notion of having to have a second round of voting would not be good for this project. A second round of voting would be inconvenient for users of the system and hence is likely to suffer from a low voter-turnout. Why would a user want to revisit a poster to resubmit his vote?

2.6.3 Secret Ballot

It should be briefly mentioned the importance of anonymity when it comes to voting. The idea of voting is to capture the opinions of voters, so we want it to be as representative as possible of voters' preferences. Ergo we wish to prevent voters from being influenced by others so that their vote is truly in line with their preference. Voters might be influenced as a result of vote buying or wishing to gain favour from others. Hence in this system it will not be made public who voted for which poster in order to maintain anonymity. However there will be the unavoidable case where, given that one can only vote for and not veto a poster, seeing someone use this app for a poster at the conference is a sure sign that they are voting for it.

[13] http://study.com/academy/lesson/majority-rule-definition-examples.html

Chapter 3

Design

In this chapter we discuss the design of our system. Furthermore we layout the technologies and the principal libraries and frameworks used to develop Postr. In the design of this system the Separation of Concerns principle has applied. By doing so the code base for the system was partitioned into different parts, each implementing separate features, or "concerns" of the system [1]. This is an advantageous structure to have as in theory it prevents any implementation changes in one component of the system from other affecting other components. For instance, a change in the UI design of the login page does not affect the authentication process that takes place in the server.

To put the Separation of Concern principle in practise, a Model-View-Controller architecture has been incorporated into the system [2]. The MVC pattern is clearly reflected in the directory structure of the iPhone app, shown in Figure 3.1, in which there are separate directories for views and controllers. Incorporating this separation logic became particularly useful when expanding the codebase and including new system components. For example in the development of the iPhone app any implementation details of the web app could be ignored, such that only the Postr web API was considered.

[1] https://deviq.com/separation-of-concerns/

[2] https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

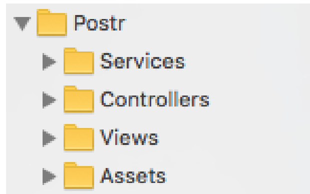

Figure 3.1: MVC application

3.1 System overview

There are a number of different components that make up this system. The diagram in Figure 3.2 illustrates these. The backend of the system consists of a SQLite database and the Postr server. The frontend includes the iPhone app and Web app, which make their HTTP requests to the same web API. As discussed in the technology overview, two Google Cloud services [3] have been used in this system. This dependency can be seen in the diagram. Furthermore, the logic for the main functional components of this system, e.g. voting for a poster, user authentication, etc, takes place in the server.

As a design choice the ability to vote for a poster is available only on the iPhone and the ability to set up an event can only be done on the web app. The iPhone is the "voting device" of the system. For similar reasons it was decided that the configuration of an event should be made available only on the web app, in order to avoid loading the iPhone app with excessive processing demands which could result in a slow device response and thus poor user experience. Note that the leadership board can be viewed on both the iPhone and web app. The winner of the competition is determined using the plurality voting system, which is described in the Background Chapter. To be precise, the winner of the competition is the poster with the most votes and joint winners are accepted.

[3] https://cloud.google.com/

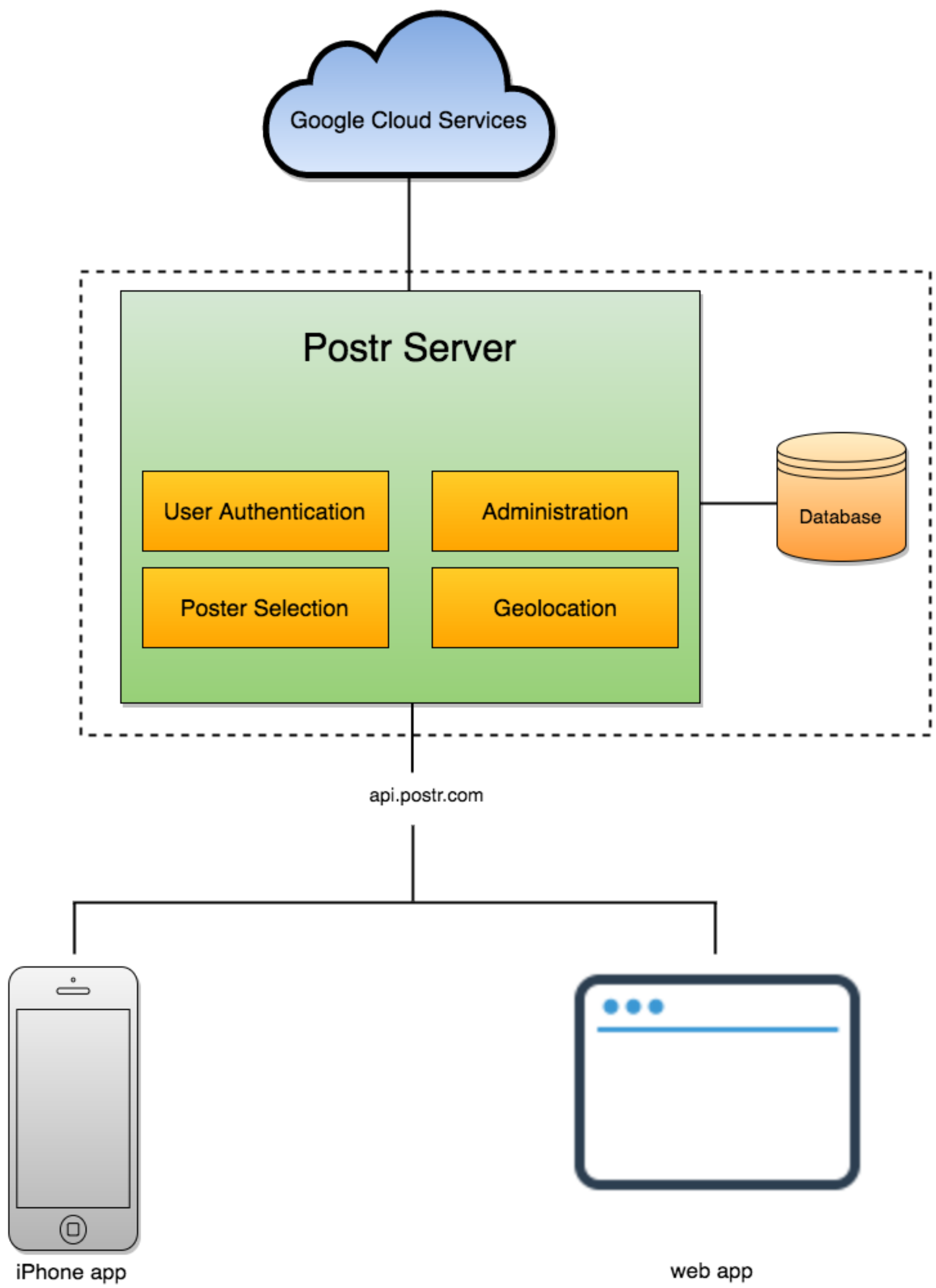

Figure 3.2: Overview of the Postr System

3.2 Design process

To further the use of the Separations of Concern principle, an API specification was created, detailing the description and action of each route. It served as a useful point of reference in the frontend implementation, and gave a structured approach when implementing the backend.

Route	Action	Description
/events/<event_id>/findPoster	GET	Uploaded file is sent to the Google Vision service to extract text present on poster image file. The event id in request body is used to retrieve posters associated with that event from the database. A query point is constructed using the OCR output. Then a KNN search is performed using the title and author attributes. The 4 most similar posters are outputted. Those with measures of similarity greater than the threshold value, are returned in the response body.

Figure 3.3: Excerpt from Postr's API specification

Next mockups for the Postr UI were generated. These mockups later influenced a structured approach when implementing application behaviour. Furthermore an intuitive UI design was desired so that Postr users could focus entirely on poster voting. Having to understand e.g. how to navigate to the voting view, would distract the user away from the voting experience. The mockups developed in this design process helped implement this intuitive UI.

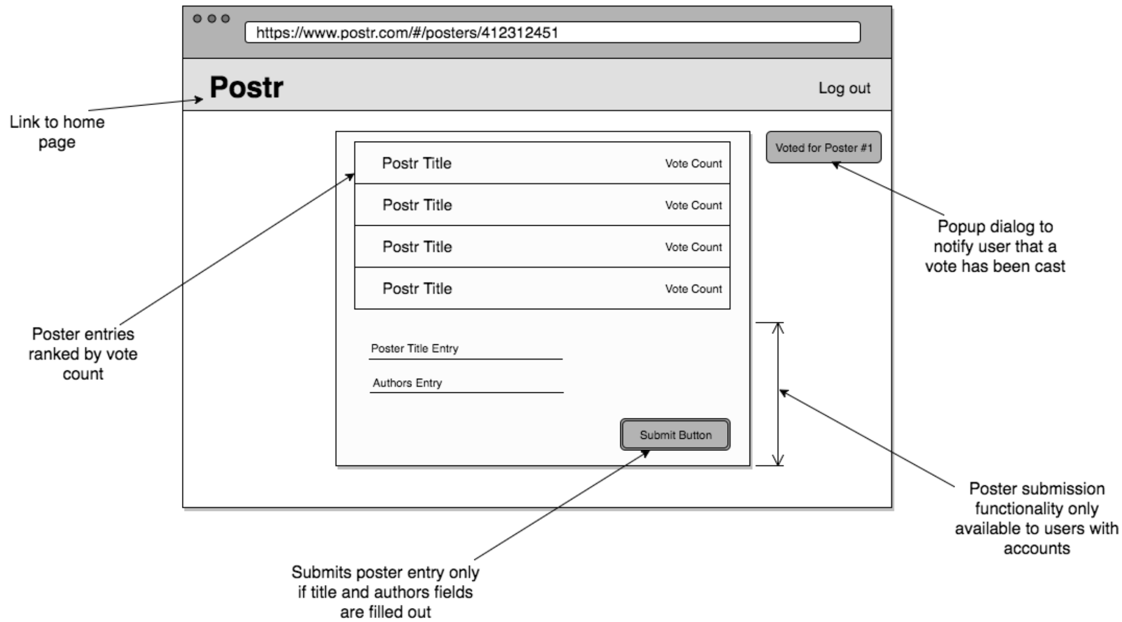

Figure 3.4: Postr Web App Mockup

3.3 Technologies Overview

3.3.1 Backend

Frameworks

We have used the framework Nodejs to build the server in this system[4]. The reasoning behind this choice is because Nodejs is a lightweight and fast framework, which would contribute to the responsiveness of the system overall. Moreover the majority of node modules are written in javascript. Since the web app side of the frontend is written in javascript, choosing Nodejs would cut down on the number of implementation languages used in this system. Given that there are 3 principle components, the server, iPhone app, and web app, there is a danger of using too many implementation languages would reduce general readability of code.

Furthermore the Nodejs framework Express has been used[5]. The features

[4]https://nodejs.org/en/
[5]http://expressjs.com/

provided by this framework greatly simplified the process of developing an interface to respond to HTTP requests. JSON (Javascript Object Notation) based data was used in the responses to the requests from the frontend [6]. This was a logical choice as Nodejs uses JSON internally and runs on javascript.

Storage

Storage in Postr is achieved using a relational SQLite database [7]. The SQLite database is embedded in the backend part of this system. In this system there are numerous inter-related components that need to be stored in the database. For instance events and their corresponding posters, users that have voted for a given poster, etc. Ergo it was concluded that the structured power of a relational database was required, with its ability to reduce data duplication. The use of a relational database enables related data to be passed into queries and the resulting output data to be quickly retrieved.

Google Cloud Services

The system makes use of two Google Cloud services. Firstly, the OCR functionality of this system uses the Google Vision service [8]. While this service is not free of charge, in order for state of the art text recognition to be performed it was decided that the OCR should be outsourced to a high-end service. As explained in the next chapter, Postr requires the ability to obtain the latitude and longitude points of a given address. To do this, the Google Maps service was used [9].

3.3.2 Frontend: Web App

Standard web programming languages were used to develop the web app; HTML, CSS and Javascript. Furthermore the Angularjs framework was incorporated into the application [10]. Angularjs simplified the AJAX calls, which are used to make requests to the server without having to reload the webpage. In addition to this, the CSS framework Materialize was used to give

[6] http://www.json.org/
[7] https://www.sqlite.org/
[8] https://cloud.google.com/vision/
[9] https://developers.google.com/maps/
[10] https://angularjs.org/

a responsive and sleek design to the UI[11]. An example of this CSS framework in action are the forms present in the Figure below.

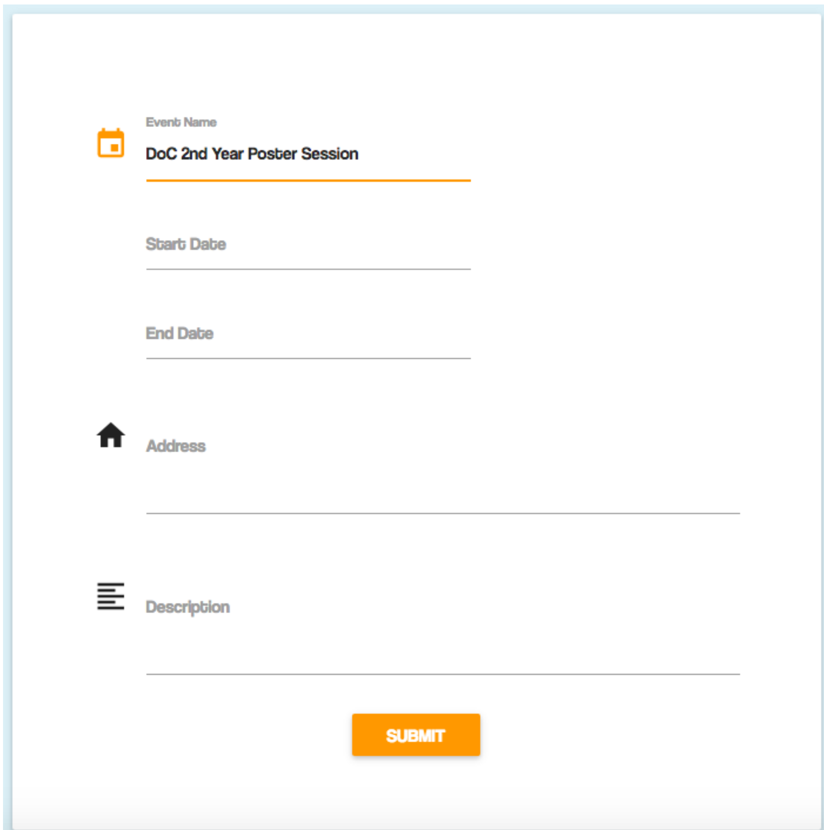

Figure 3.5: Creating an Poster Session Event UI

3.3.3 Frontend: iPhone App

There are two implementation languages available in iOS development; Objective C and Swift. While an understanding of Objective C was still required during development, as many existing frameworks are written in this language, the future of iOS development is with the recently released Swift language. Ergo in order to develop a system as up-to-date as possible with modern technologies, the iPhone app was coded in Swift.

[11]http://materializecss.com/

To aid the HTTP networking with the server, the Alamofire framework was used [12]. In particular, it greatly simplified the means in which image files are uploaded to and data is requested from the server. Furthermore the AV-Foundation framework was used to interact with user's camera devices on their iPhones [13].

It is worth mentioning the use of storyboards; a graphical layout of the of the view transitions in an iPhone app. Storyboards can be used in the iOS developer's environment, and they provided a useful way to directly map the designs laid in out in the UI mockups to the iPhone app. Rather than having to program the visual appearance of the UI, objects could be directly drawn in the development environment using the storyboards.

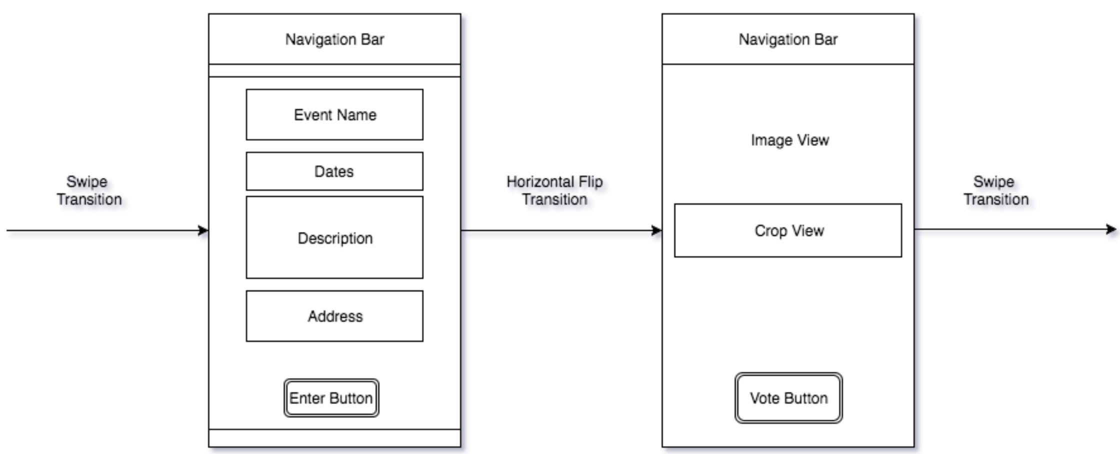

Figure 3.6: Postr iPhone App Mockup

[12]https://github.com/Alamofire/Alamofire
[13]https://developer.apple.com/av-foundation/

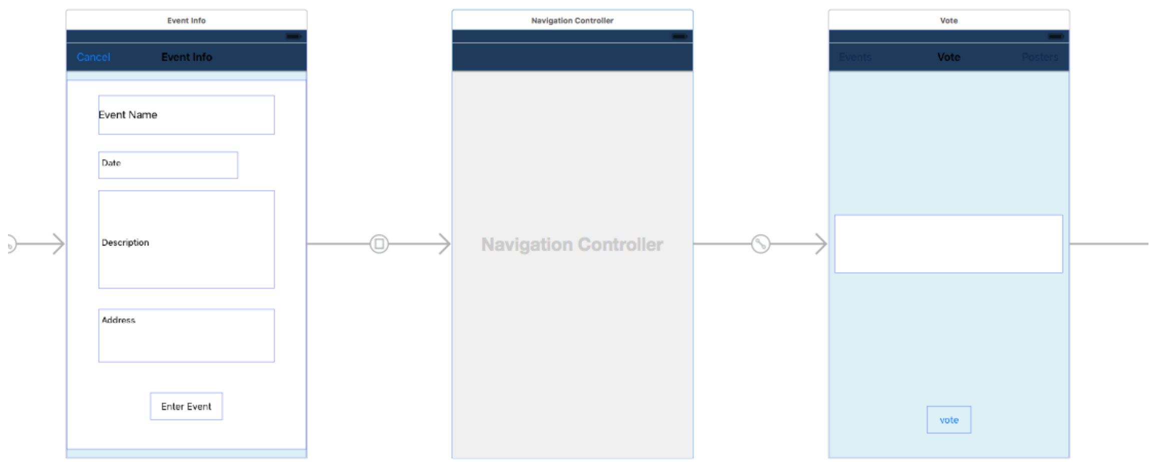

Figure 3.7: Postr Storyboard

Chapter 4

Implementation

For a given Postr user we assume that the following principal tasks will be carried out. In this chapter we will discuss the underlying implementation of each of the tasks. Naturally it is expected that only event administrators would be interested in executing the "Creating an Event" task, but for completeness this has been included.

1. Login and Registration
2. Creating an Event
3. Retrieving Nearby Events
4. Voting for a Poster

We will now discuss the implementation that underpins the login and registration feature of Postr.

4.1 Login and Registration

For a user new to the Postr system the first activity is to register his credentials to be written to the system's database. After having filling the relevant fields on the client-side UI, a HTTP post request is made to submit the data to the server. Two levels of users are required; administrators and basic users. Administrators would have full access to the server side-functionality while a basic user would not be able to create an "event" and add posters to it. The request body consists of the user's email address, desired username and password. A query is made to the database and existing email addresses and usernames are retrieved from the schema "users".

If the user's inputted username or email address is found to already exist then a response is returned to the user. The response has a status code of 402 and a JSON body consisting of a message informing the user that the respective fields already exists and asks for new ones to be re-entered. Email addresses should be unique in this schema as a malicious user could circumvent the system's attempts to prevent double voting by creating multiple accounts with the same email address. Furthermore usernames are desired to be unique to make it clear that a vote was cast by a particular individual. The choice of response code is justified by the following description of the meaning it conveys: *"The request could not be completed due to a conflict with the current state of the resource"* [1].

If the presented email address and username does not already exist, a salt is randomly generated and a hash is constructed using the presented password and salt. The motivations for using salts and and hashes were discussed in Background Chapter of this report. Finally a query is made to database and the user's email address, username, hashed password and salt is inserted to the user schema. The response body returned to the client-side consists of a signed JSON Web Token (or JWT).

JWTs enables authentication in this system. Namely we want to restrict certain features of Postr to only users with accounts. For instance in the web app the ability to create events should only be available to registered users. Furthermore every feature, e.g. voting for a poster, in the iPhone app is available only to administrative users. This decision is influenced by the fact that the iPhone app is thought of as the "voting device" of this system. Only registered voters, i.e. users, should be able to be vote in a poster competition. The expiration dates on the JWTs are set to 5 days, as it is unlikely that an event which hosts poster sessions would last longer than this.

Similarly when a user logs in on Postr, a JWT is returned provided that valid credentials are given. Specifically, the hash of the given password matches that of the relevant hashed password stored in the database. To conclude

[1] https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

this section, it should be noted that the ability to register and login is present on both the iPhone app and the web app.

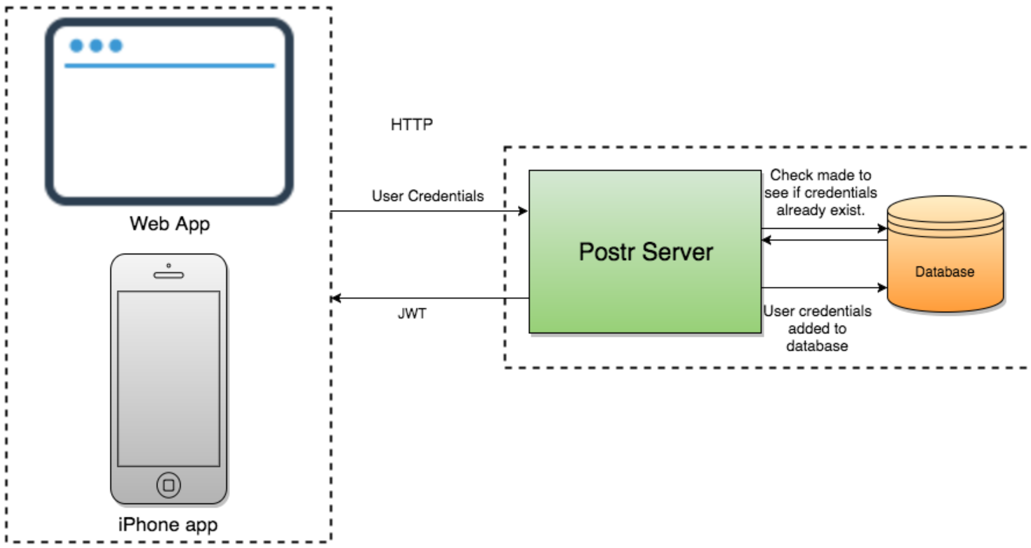

Figure 4.1: User Registration

4.2 Creating an Event

In order for a competition to be enabled in Postr, an "event" has to be set up with the relevant poster entries added to it. It would be expected that the administrators of a poster session would set up such an event. The ability to create an event and add posters to it, so that they become votable, is restricted to only administritive users. Furthermore the creation of an event can only be done on the web app of this system. To reiterate the phone app is thought of as a "voting device", so the features that it supports are strictly for the purpose of enabling user voting in an event. To overload the iPhone app with additional features such as event creation was felt to pull it too much away from its original voting purpose. Moreover it would pollute the iPhone app's user experience; Postr aims to provide a simple and minimalistic platform to vote on.

After filling out the fields in the web app UI, a HTTP post request is made

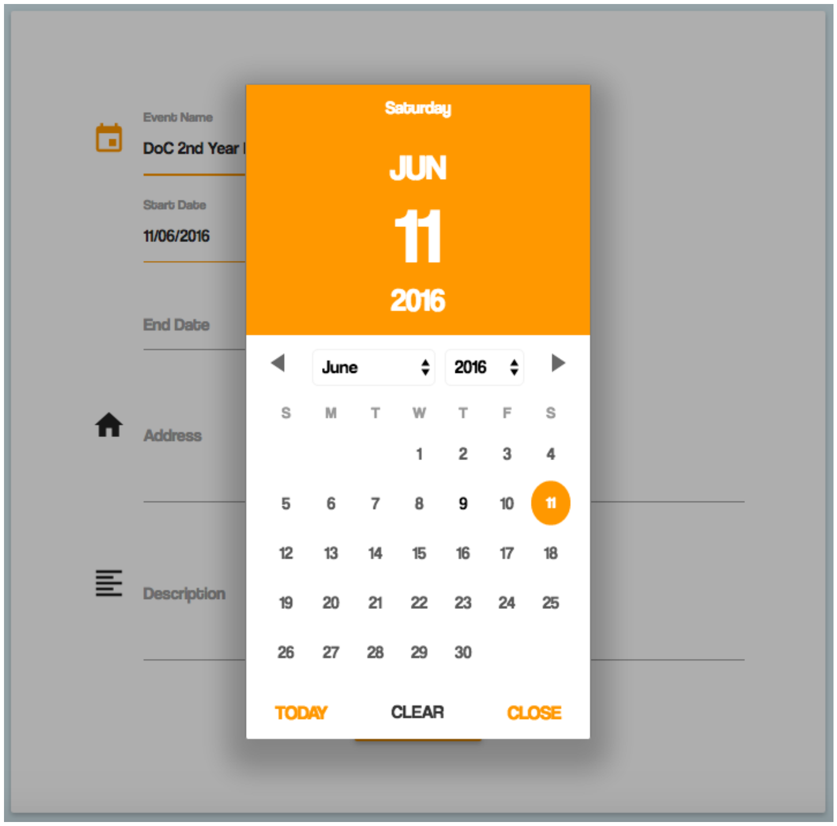

Figure 4.2: Event Configuration

with its request body consisting of the inputted field values. In this request body includes the address of the event, and using this string value we wish to apply a process of geoencoding to obtain a geographic location. Namely, we desire a latitude and longitude representation of the event's location. This is required when the system presents the nearest events to the user, as discussed in the next section.

To achieve this the Google Maps service has been used, which returns float values for both the latitude and longitude points representative of the address in the request body. These values are inserted into the "events" schema in the database, along with the parameters embedded in the original post request body. A similar process ensues when adding posters to a given event, so that they can be voted for by users in a competition. Note that poster entries, or rows, in the database are initialized with a vote count of zero. On the client-side for both the iPhone app and the web app, when the scope of an event is "entered", the leadership board is presented to the user. Posters are ranked by vote count, with the most voted for poster at the top.

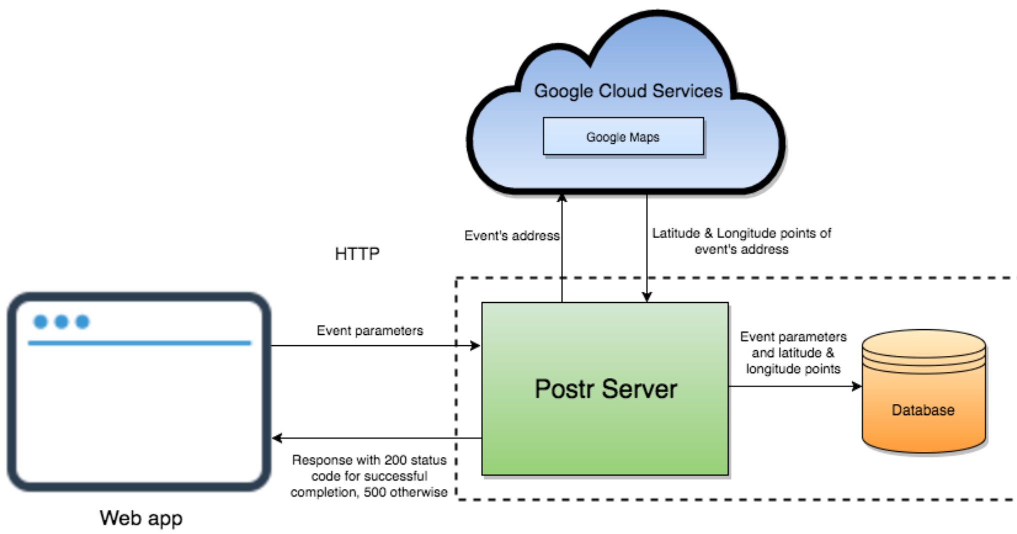

Figure 4.3: Creating an Event

4.2.1 Schema

events							
event_id	name	address	latitude	longitude	description	start_date	end_date
100	"DoC 2nd Year Post..."	"Huxley Bldg..."	51.498828	-0.17917	"2nd Year st..."	11/06/16	11/06/16

Figure 4.4: Events Schema

posters				
poster_id	event_id	title	description	votes
586	100	"APX Music"	"State of ..."	0

Figure 4.5: Posters Schema

The relevant schemas for this section are given above. In both the events and posters schema a unique id value has also been included; "event_id" and "poster_id". These values are known as primary keys and provide the means to index a particular entry in the schemas. Furthermore an event_id column has been included in the poster schema. This acts as foreign key pointing towards the event_id column in the events schema, establishing a relationship between the two schemas. For instance, consider the given row in the events schema and the given row in the posters schema which share the same value in their event_id column entry. This relationship is representative of how the poster, which corresponds to the row in the posters schema, is present at the event which corresponds to the row in the events schema.

poster_authors	
poster_id	**author**
586	"John Doe"
586	"Martin Boyd"

Figure 4.6: Poster Authors Schema

Similarly in the Figure 4.5, the poster_id column acts as a foreign key pointing towards the poster_id column in the posters schema of Figure 4.5. The assumption cannot be made that a poster has only one author; there could be multiple authors. However we cannot have multiple columns in the posters schema to account for multiple authors, especially given that the system is unaware of how many authors that could exist for a given poster. Ergo the solution is to factor out the poster authors into its own schema, as shown in figure 4.6.

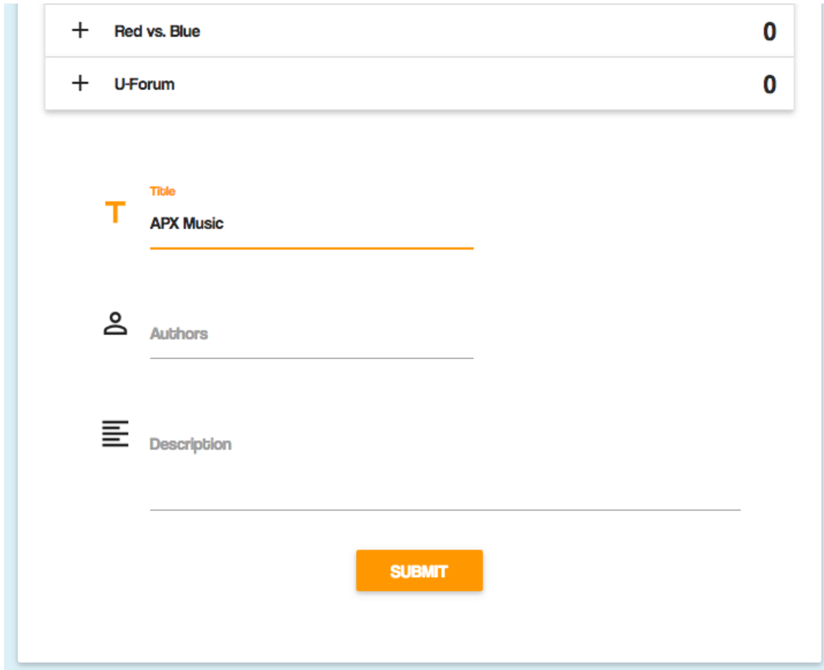

Figure 4.7: Setting up a leadership board for an event

4.3 Retrieving Nearby Events

After having logged in or registered on the iPhone app, the user is presented with the 4 nearest events. While the option to see all events exists on the iPhone app, it was considered inelegant to display all of them in one of the first views in the User Experience. For a user having to scroll through and search for a desired event is tedious and time-consuming. Instead the iPhone app presents nearby events, which are likely to be the events the user is searching for. In this feature the use of geolocation and the stored latitude and longitude points of each event in the system's database comes into play.

The iPhone app sends a HTTP post request to the server with the latitude and longitude of the user's location in the request body. Using the latitude and longitude of the stored events in the database, a KNN search is performed to return the nearest 4 events to the query point given in the HTTP post request body. With this search distance, or similarity, is established using the Haversine formula described in the Background Chapter. The nearest 4 events is returned to the iPhone app in the request body with a response status code of 200, which indicates that the initial request has succeeded.

So long as there are at least 4 events registered with the system, the 4 nearest events will be returned in the request body to the user. If there are only 3 events registered, then those 3 events are returned, and so on. Once iPhone app has obtained the nearest events, it presents these to the user. The user can then choose an event and enter its "scope"; the user can now vote for posters at that event, as described in the next section. However if no events are registered with the system, the user will not be able to proceed and use any further features of the iPhone app. This is an acceptable scenario as we consider the iPhone app to be strictly a voting device; if there are no events to "enter", there are no posters that can be voted for.

4.4 Voting for a Poster

For the sake for clarity the steps in this particular feature of the system are given below. Moreover a diagram depicting the flow of requests and responses that correspond to the outlined steps has been included.

1. Poster image is captured and cropped on the iPhone app.
2. Poster image is sent to the server from the iPhone app.
3. Poster image is sent to the Google Vision service from the server for OCR.
4. OCR is performed on the poster image by the Google Vision service.
5. Text recognised in the poster image is sent to the server from the Google Cloud service.
6. Using the text received from OCR, KNN algorithm is run on server to determine the most likely poster entries that match the poster captured in the image.
7. Server sends the most likely posters back to iPhone app.

8. User confirms which poster the vote was intended for and this is sent back to the server.

9. Check is made in poster_voters schema to see if the user has already voted for this poster

10. Vote is logged in for the relevant poster entry in database.

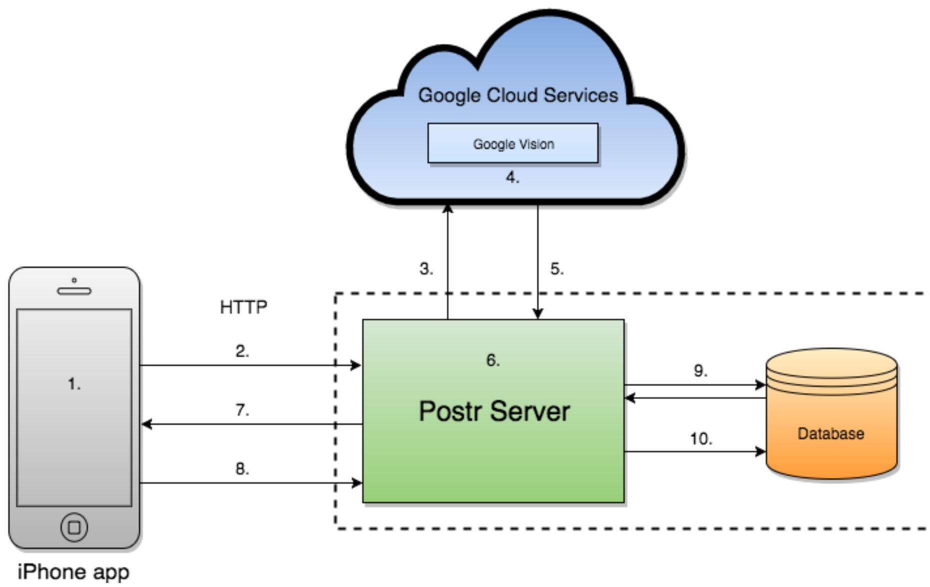

Figure 4.8: Voting for a Poster

We will first discuss the client-side implementation of this feature.

4.4.1 Voting for a Poster: Client-side

Upon entering an event the user is presented with the vote view. In the launch sequence of this view the Postr app searches amongst devices present on the iPhone. The device it searches for supports video media type and is positioned on the back of the iPhone device. This search criteria can be seen in the nested if statement of the code snippet in Figure 4.5. The function **viewDidLoad()** is called at the launch time of the view and the object **AVCaptureDevice** represents a physical iPhone (capture) device and its underlying properties [2]. This object is used to configure the properties for the camera device to enable the capture of high quality images to improve the text recognition performed by the OCR functionality in this system.

```
override func viewDidLoad() {
    super.viewDidLoad()

    let devices = AVCaptureDevice.devices()

    // Find back camera amongst phone's devices
    for device in devices {
        if (device.hasMediaType(AVMediaTypeVideo) &&
            device.position == AVCaptureDevicePosition.Back) {
            captureDevice = device as? AVCaptureDevice
            configureCamera(captureDevice!)
        }
    }
}
```

Figure 4.9: Searching for the camera device

In the Figure 4.8 we have the voting view. In the background lies the image preview provided by the iPhone's camera. In the foreground we have a button labelled "vote" and a rectangular crop view with a dashed orange border. When the image is captured at the click of the vote button it is cropped using the coordinates and dimensions of the rectangular crop view. The idea is to position the iPhone such that the crop view is over the poster title and author so that the resulting image does not include the main body of the poster. In doing so, we can avoid the computational cost of recognising any text in the poster body when the OCR is applied to the cropped image. This speeds up the voting process, providing a more responsive experience to users.

Furthermore the crop view can be resized by the user dragging his fingers across the iPhone screen. Event handlers respond to the touch gesture by keeping track of the movements and coordinates of the user fingers, and then redrawing the crop view on the screen based on these values. Once the image is captured and processed accordingly it is sent to the server in a HTTP get request. Because the request data consists of an image file, it has to be encoded using a "multipart/form-data" method rather than the de-

[2] https://developer.apple.com/library/mac/documentation/AVFoundation/Reference/AVCaptureDevice_Class/

fault "application/x-www-form-urlencoded" one [3]. An example of a cropped image sent to the server is given below.

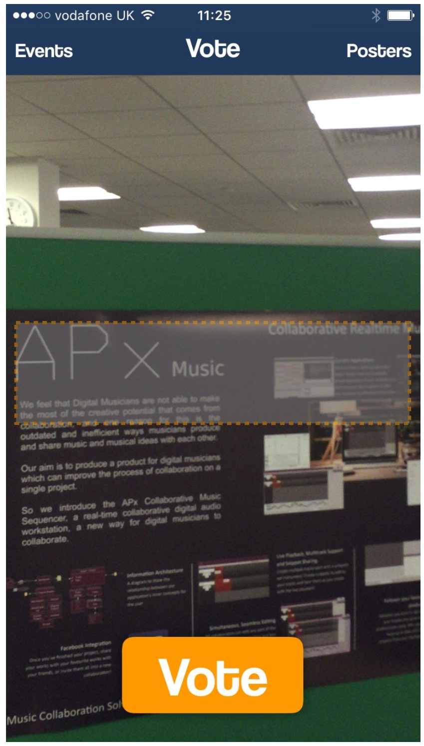

Figure 4.10: Postr Voting UI

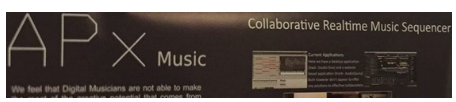

Figure 4.11: Cropped image sent to Server

4.4.2 Voting for a Poster: Server-side

KNN Search

The uploaded image file is immediately sent to the Google Vision Service, from which a json-based response is returned with the recognised text in its

[3]https://www.w3.org/TR/html5/forms.htmlmultipart/form-data-encoding-algorithm

body. A KNN algorithm is run to determine the four poster entries in the database that are most similar to the poster captured in the image. The KNN algorithm establishes similarity between the posters using the title and author attributes of posters. In the code snippet below, this configuration can be seen in the **fields** variable, in which it also sets the measure field to ensure that the title and author attributes are compared using a similarity method intended to be used on words.

```
function findNearestNeighbour(ocrOutput, event_id, res) {
    // Nearest neighbour base of comparision.
    var posterText = ocrOutput[0].desc
    var query = { title: posterText, author: posterText };

    // fields variable used in nearest neighbour search
    var fields = [{ name: "title", measure: nn.comparisonMethods.word },
                  { name: "author", measure: nn.comparisonMethods.word }];

    var getPostersForEventQuery = "SELECT title, author \
                                   FROM posters \
                                   WHERE event_id=" + event_id + ";";

    db.all(getPostersForEventQuery, function(err, rows) {
        if (err) {
            return res.status(500).json({
                error: err
            });
        } else {
            knnSearch(posterText, query, rows, fields, res);
        }
    });
}
```

Figure 4.12: Prepping for the KNN search

Next the SQL query **getPostersForEventQuery** is run and the response is stored in the array **rows**. Note that the **event_id** variable is used to retrieve only the posters that belong to that corresponding event when querying the database. **rows** consists of the title and author pairings, each unique to and thus representative of a poster. For the query point **query** in this KNN search, the OCR text output is passed as the values for both the title and author attributes. The relevant variables are passed into the **knnSearch** function in the callback of **db.all(...)**, where the KNN search is then carried out.

The KNN search outputs the four nearest posters, each with a given measure of similarity. This measure of similarity can be used as a threshold value to deal with false positives. For example, if at some given event a user were to take a picture of a leaflet with the Postr app, we want to avoid the system accidentally voting for some poster. We will now explore the justification of the threshold value that has been chosen for Postr.

Threshold Value

An experiment was set up in which a test event was created and there existed two groups of poster; registered and non-registered posters. Each group contained twenty posters. The posters that belonged to the registered group were added to the event. So if a test user viewed the leadership board for the event these posters would have appeared. The test user then attempted to vote for these posters, both registered and non-registered, using the Postr iPhone app. To standardize the tests the vote attempt was made at a distance of 0.5m from the mounted poster, and the image was captured using only the default zoom factor on the device's camera.

Ignoring whether a vote had been cast or not, the measure of similarity for the most similar poster outputted by the KNN search was recorded. For the attempted votes made to the posters in the registered group, the most similar poster outputted by the KNN search matched that of the poster in the captured image. Naturally for the attempted votes made to posters in the non-registered group, no matching posters were outputted from the KNN search. Nevertheless the measure of similarity outputted for the most similar poster was recorded.

For each group the minimum and maximum similarity measure was noted. Furthermore the lower quartile, median and upper quartile values were calculated. If the similarity values of each poster group were arranged in ascending order, the quartiles are the values that divide the data set into quarters. So all the ordered values less that the lower quartile, or q1, would make up a quarter of the data set. In the case of the median it would make up a half of the data set and for the upper quartile, or q3, three quarters of the data set. The results of the experiment are given below.

Statistic	Registered	Not Registered
min	0.175438596	0.0342834
q1	0.2638495	0.068181818
median	0.295454545	0.073170732
q3	0.3274921	0.083333333
max	0.380952381	0.092592593

Figure 4.13: Threshold Data

Using these experimental results a box and whisker plot was made for each group. This can be seen in the figure below, where the experimental statistics used are labelled on the diagram. A box and whisker plot visualised depicts the distribution of our experiment's results. The plot on the right hand side represents similarity values that correspond to the registered posters group, while the left one represents those that correspond to the non-registered poster group.

A threshold value was chosen such that it lies in between the values denoted by the right whisker of the left box plot, and the left whisker of the right box plot. Using these experimental results it can be claimed that if the similarity values output by the KNN algorithm were less than the threshold value, it is likely that the captured poster is not registered in the event. This is because the entire range of similarity values of the tested posters in both groups can be partitioned from each other by a single (threshold) value. The current threshold value is set to 0.15 and its effectiveness in dealing with false positives is evaluated in the next chapter.

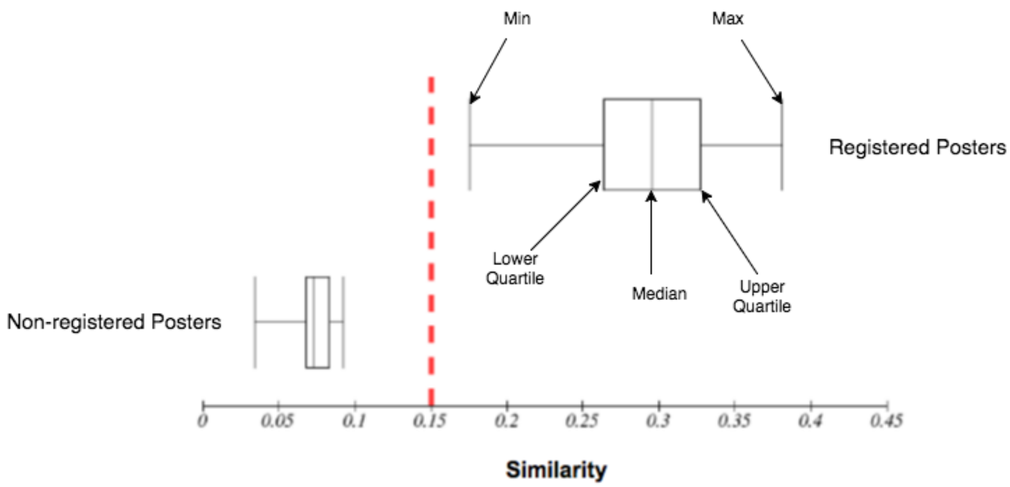

Figure 4.14: Threshold Boxplot

Preventing Double Voting

For each of the four outputted posters from the KNN search, the measure of similarity associated with it is compared to the threshold value. If it exceeds the threshold value, then that poster is included in the response body that is sent to the iPhone app. Otherwise it is discarded from further consideration.

The user on the iPhone app is presented with the most similar poster found. If it doesn't match the poster in the captured image that the user intends to vote for, then the other posters in the response body are presented. Again if these don't match, then a popup dialog is displayed on the iPhone app requested that the user attempts vote for the poster again. Otherwise if any of the given posters matches the one the user intends to vote for, the iPhone app sends a HTTP put request to the server. The body of the request consists of the parameters of the poster, e.g. poster_id, title, etc, that the user selected, as well as the user's username.

The poster_voters schema serves as record of which users have voted for which posters. A query is made to the database query checking if rows with the given poster_id and username exist. If they do, the user has already voted for this poster in the past. As this system prevents double voting, a response is returned to the iPhone app with a status code of 500 and an error message detailing the reasons for the rejected vote, which is passed onto the user in the form of a popup alert.

If no such votes exist, an entry is made to the poster_voters schema with the poster_id and username. Then the vote count of the row in the posters schema with a matching poster_id is incremented. This concludes the system's method of preventing double voting.

poster_voters	
poster_id	username
586	123
586	214

Figure 4.15: Poster Voters Schema

Chapter 5

Evaluation

5.1 Quantitative Analysis

The first step in evaluating the effectiveness of Postr was to conduct a series of experiments to determine various measures of accuracy. In each experiment an event was set up on the system and two groups of test posters were defined. The first test group consisted of posters registered for the event. In a real life scenario, users should be able to vote for posters in this group. The second test group consisted of posters that were not registered for the event. Users should not be able to vote for posters in this latter group.

We want to assess the system's ability to correctly recognise registered posters and hence that users' votes are being cast to the intended posters. It should be noted that in a real-life poster session it is unlikely that there will be a collection of posters which are not registered for a competition. However in the conducted experiments the non-registered poster test group is intended to represent objects that users may accidentally or intentionally try to vote for using Postr. For instance, in the event where a user takes a photo of a book cover with Postr, we want to avoid a vote being cast to some poster in the system's database.

Each group consisted of 20 posters each. Arguably for the experiment results to be considered more representative of posters in general in any poster session, more posters should have been used. However in the interest of saving time and printing costs, 20 posters were chosen for each group. Actual poster templates were used in these examples so that the experimental results obtained could be more applicable to actual posters used in poster sessions. An example of one such poster template is shown in the figure 5.1.

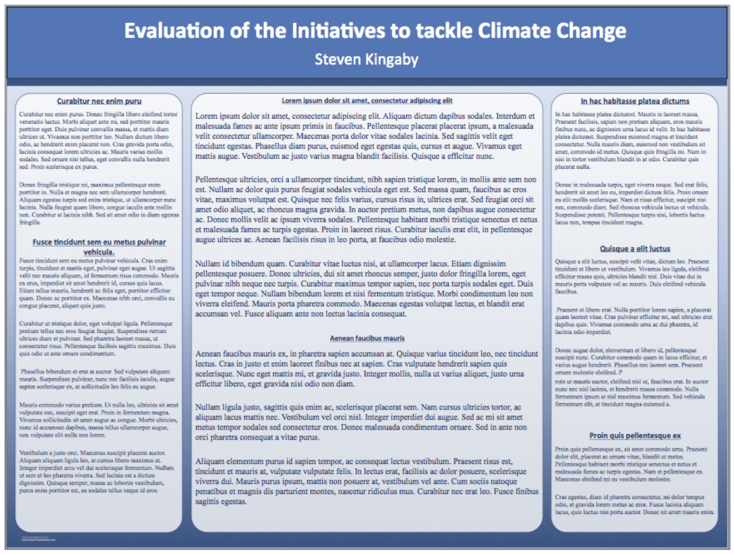

Figure 5.1: Mock posters used in the evaluation of Postr

Note that the content of the main body of poster is irrelevant, as this is cropped out of the image by the system before OCR is applied to it. Ergo dummy text has been used in the body content. In order to standardise these tests the authors of the poster has been positioned below the title. Furthermore font-sizes have been chosen such that for each poster, the height of the text in the title is 3cm and the height of the text in the author is 2cm. This approach is required so that any conclusions drawn from the experiment's results can be strictly associated with the effects of varying some factor rather than the features of the poster itself. An example of a factor that is varied is the distance of the phone's camera from the poster. But before we explore the details of the conducted experiments, we will cover some required definitions.

5.1.1 Classification

For the sake of these experiments, we have defined this system as a classifier which when presented with a poster assigns one of two possible labels.

A poster can be labelled as either belonging to the class of posters that are registered for a given event, or the class where they are not registered. We require this definition so that we can apply the following classification measures. But before we explore these, we first discuss the use of confusion matrices. A confusion matrix is a useful structure adopted in the calculations of the performance measures used in the evaluation of Postr.

	Registered Poster: Predicted	Non Registered Poster: Predicted
Registered Poster: Actual	TP	FN
Non Registered Poster: Actual	FP	TN

Figure 5.2: Confusion Matrix

A confusion matrix illustrates the correct and incorrect classification of posters in a visual manner [1]. A description of the terms used in the confusion matrix is given below, written in the context of this section's experimental poster session. The total number of posters that fall under a a certain category is inserted into the relevant cell in the confusion matrix. E.g. the cell labelled as TP represents the total number of posters which can be described as TPs.

- TP (or true positive) represents a poster that has been correctly classified as being registered.
- TN (or true negative) represents a poster that has been correctly classified as being not registered.
- FP (or false positive) represents a posters that has been incorrectly classified as being registered.
- FN (or false negative) represents a poster that has been incorrectly classified as being not registered.

The posters which attribute to the values that make up TP and FP will be voted for by our system. Naturally we want to keep the value of FP as low as possible as it represents the number of times poster entries in the database have been mistakenly voted for. Similarly we want to minimise the value of FN as it represents posters which users have tried to vote for but have been

[1] https://en.wikipedia.org/wiki/Confusion_matrix

prevented by Postr. The values that make up the confusion matrix have been used in performance measure calculations which we will now discuss.

The classification rate is defined as below. As may be expected it is desired that we keep this value as small possible, particularly because it is directly proportional to the classification error. To be precise, the classification error = 1−classification rate.

$$\text{Classification Rate} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

The classification rate gives a broad picture of the success of our system. However it is susceptible to bias in the event of imbalanced datasets where the number of elements in one class exceeds that of another [22]. While this is not a problem in the evaluation of Postr, where each class (i.e. registered and non-registered posters) both have 20 posters, other performance measures have been used to thoroughly evaluate this system.

The next performance measure used is recall. Recall can be thought to represent a classifier's completeness [23]. A high recall value is indicative of a small number of FNs. Furthermore it shows that the threshold value, as defined in the Implementation Chapter, is small enough to avoid incorrectly classifying posters as being non-registered.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The final performance measure used is precision. Precision can be thought to represent a classifier's exactness [23]. A high precision value is indicative of a small number of FPs. In particular, it shows that the threshold value is large enough to avoid incorrectly classifying posters as being registered. The original purpose of applying the threshold value was to eliminate FPs, so this measure can directly assess its effectiveness.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

We will now investigate the experiments themselves used to evaluate Postr, and the consequent findings.

5.1.2 Experiments

Evaluating the effect of varying distance

Putting the metrics defined in the previous section into practise, the first experiment was to evaluate the effect of the distance between the test user's iPhone camera and the mounted poster. To further standardise these tests all posters were mounted on a flat surface, rather than a curved one, and the zoom on the camera was left to the default value as set in the iPhone app's launch sequence. However, the test user was permitted to tap the iPhone screen in order to focus the device's camera when attempting to vote for a poster.

The system was used to "classify" the individual test posters at varying distances. To reiterate, for the purpose of these experiments we are treating Postr as a classifier. However the overall aim of this system is to capture user preferences and to determine the most popular poster in a competition, not to be a classifier of posters. The system was used at 4 different distances; 0.5, 1, 1.5 and 2 metres.

Needless to say that it was unnecessary to test the effectiveness of the system at a distance of 0 metres from the poster, as nothing distinguishable would have appeared in the captured image. Moreover it was felt that varying distances so that they exceed 2 metres was unnecessary as without the ability to zoom, no distinguishable text could be made out from the capture image. Furthermore it is unlikely that a user would attempt to vote for a poster more than 2 metres away without adjusting the zoom settings of the camera.

	Registered Poster: Predicted	Non Registered Poster: Predicted
Registered Poster: Actual	16	4
Non Registered Poster: Actual	2	18

Figure 5.3: An example of a confusion matrix used in the evaluation of Postr

Using the results obtained for each distance, a confusion matrix was constructed. One of which is displayed above, which displays the results relating to the classifications conducted at a distance of 2m from the poster. Next the performance measures described in the previous section were calculated using the values in the confusion matrix. The final results are illustrated in the bar chart below:

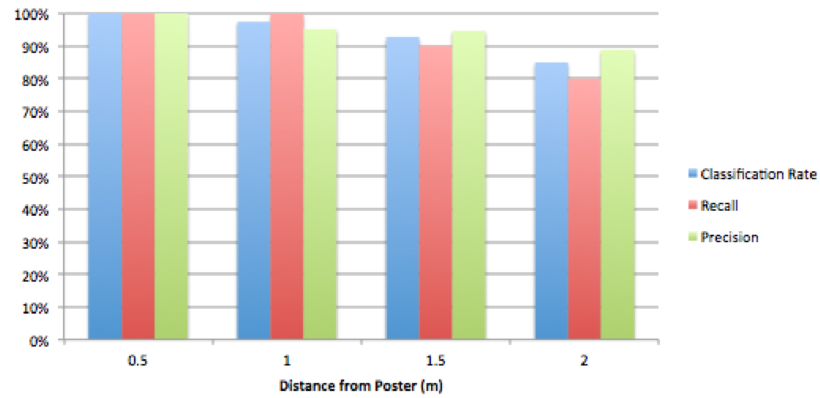

Figure 5.4: Diagram depicting the effects of varying distance

As can be seen the classifications taken at distance of 0.5 metres achieved 100% rate across all performance measures. This means that at this distance the system had no errors in correctly voting for or rejecting votes for posters when appropriate. In the next graduation of 1 metre we see no change in recall, but a slight decrease in both the classification rate and precision. Arguably this could be seen as acceptable as the classification rate is still above 95%. Ergo in proportion to the total number of posters, only a few posters would lead to a voting errors. However the fall in precision indicates that FPs having crept in, suggesting that the system's threshold value in dealing with such errors is too low.

In the next two graduations we see a global decrease across all performance measures. In particular at a distance of 2 metres recall falls to a dangerous low of 80%. This means that at such a distance Postr is rejecting the test user's attempts to vote for posters that are registered in the event. This demonstrates that the current threshold value is too high. The balance in determining the threshold value is a delicate one as setting it too high could result in user attempts to vote for registered posters, and too low it could allow users to vote for non-registered.

In spite of this delicate balance, in this system allowing users to vote for registered posters is prioritised over preventing users to vote for non-registered posters. Ergo following this evaluation, a decision could be made to decrease the threshold value. It is worth mentioning that these low rates were obtained for the performance measures at a distance of 2 metres, while the test user was restricted from using the zoom feature on his camera.

Evaluating the effects of varying the incident angle

The next experiment taken was to investigate the effects of varying the incident angle on Postr's ability to distinguish between registered and non-registered posters. The origin is defined to be the camera's line of sight, which meets the poster, when perpendicular to the mounted poster's surface. The incident angle is the angle between the camera's line of sight and the origin. The diagram below illustrates these definitions.

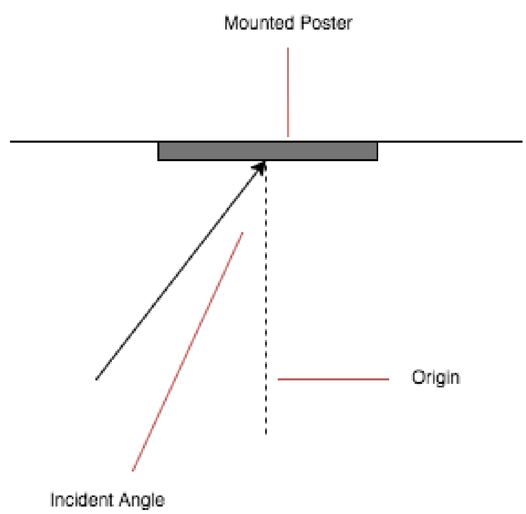

Figure 5.5: Varying the incident angle experiment

To eliminate the effects of varying distance in this experiment, the test user kept a constant distance of 0.5 metres from the mounted poster. This value was chosen as in the previous experiment it was discovered that at such a distance, 100% rates were achieved across all performance measures. Ergo it would serve as a suitable value to eliminate the possibility of the distance of the camera from the mounted poster from affecting this experiment's results.

Similarly, the test user was restricted from zooming in on the iPhone camera but was permitted to tap the screen to focus the camera.

Just as with the previous experiment, a confusion matrix was constructed and the values within it were used to calculate the performance measures. The results of this experiment are shown below. The graduations chosen were 20, 40 and 60 degrees. Experimenting with an incident angle of 0 degrees was avoided as this would just be repeating the previous experiment at the 0.5 metre graduation. Furthermore it was felt that using an angle of at least 80 degrees was unnecessary as no text was outputted by the OCR functionality of this system when images were captured at this angle. Moreover in a poster event it is unlikely that users would attempt to capture images at such an incident angle when no text is distinguishable on the camera view.

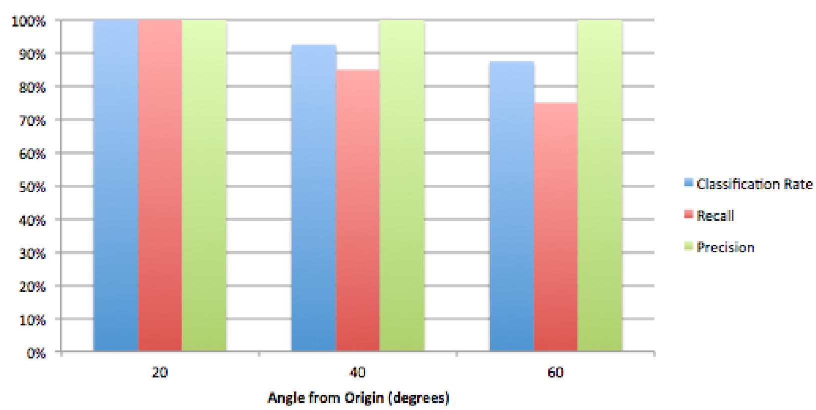

Figure 5.6: Diagram depicting the effects of varying the incident angle

At an incident angle of 20 degrees a 100% rate was achieved across all performance measures. This result is promising as it portrays that small deviations from the origin will not affect the systems ability to classify posters. Looking across all categories the precision stayed constant at a 100% rate. This an interesting observations as it shows that the system has no difficulties in dealing with FPs. So within the scope of varying the incident angle, threshold value can be seen as sufficient.

However there is a significant decrease in recall across all categories, as the incident angle is increased. This degradation is not desired as explained in the analysis of the results of the previous experiments. The system's ability to vote for registered posters is prioritized over its ability to reject user votes for non-registered posters. A low value of recall indicates a high number of FNs have occurred, i.e. the system has rejected the test user's attempt to vote for a registered poster. Reaching a similar conclusion to the previous experiment, the threshold value should be decreased in order to prevent a significant number of FNs from occurring. With additional time these tests could have been extended to include low-end mobile phones with a lower resolution camera. The intention being to determine the lower limits of mobile device configuration that are acceptable. An iPhone 6 device was used in the experiments covered in this Evaluation Chapter.

5.2 Qualitative Analysis

The second step in evaluating Postr was to deploy the system at a poster session event. In particular Postr was deployed at Imperial College's Department of Computing 2nd year poster session. Students, or rather poster authors, presented their group project work in the form of a poster to examiners. The nearby, immediate, location of the session provided the perfect opportunity to test the system in a real life scenario. Furthermore a list of all posters and their respective authors were kindly made available by the poster session's coordinators. This enabled an event to be created in the system with its associated posters as described in the Implementation Chapter.

Five testers assisted in this evaluation of Postr. Each of them were given access to installing Postr on their iPhones and each completed a survey before they left the event. The survey was intended to evaluate their experience with using Postr and hence, evaluate the overall system's usability to instigate and support a poster voting competition. The following survey questions were asked:

1. Were you able to successfully install the app?

2. How was the User Experience?

3. How easy was it to register and login?

4. How easy was it to find the event?

5. How easy was it to vote for a poster?
6. How clear was it to you who won the competition?
7. How responsive did you find the app?
8. Do you have any comments or recommendations?

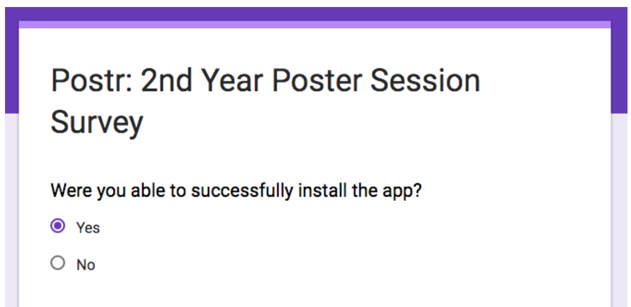

Figure 5.7: Google Forms Survey[2]

The first question gave two possible answers; yes or no. Naturally, an answer of yes is the only acceptable answer here and all five test surveys responded yes. Question 8 accepted a written response as its answer, and it was useful to receive such feedback to identify the weak points of this system.

Questions 2 to 7 gave a ranking of 1 to 5 as possible answers. An answer of 5 indicating the most positive response and a 1 indicated the least positive response. Acceptable answers to this particular range of questions is considered to be at least a 3 as this value corresponds to a "satisfactory" assessment level. The majority of the questions in this range were given positive, acceptable, responses with the main exception of question 5: *"How easy was it to vote for a poster?"*

Of the 5 test users in this poster session, one of them gave a response of 4, three of them gave a response of 3 while the final test user gave a response of 2. This was an initially worrying response as one of the objectives of Postr is to provide a platform on which users can vote for their favourite posters. A further inspection into the responses to the question 8, *"Do you have any*

[2] https://www.google.co.uk/forms/

comments or recommendations?", elaborated on these low ratings.

The posters used so far in the testing and evaluation process of this system have all featured both a title and an author. This characteristic follows from the academic posters researched in this project, where it is common practise to position the names of authors below the poster title. However many of the posters encountered in the poster session where this system was demoed did not follow this style. In fact many of them didn't include the authors' names in the posters. Rather than using a poster to display the authors' research or findings, many of these posters were instead advertising a product or system developed by students. An example of such a poster found at the event is given below, which is advertising a system called "APX Music".

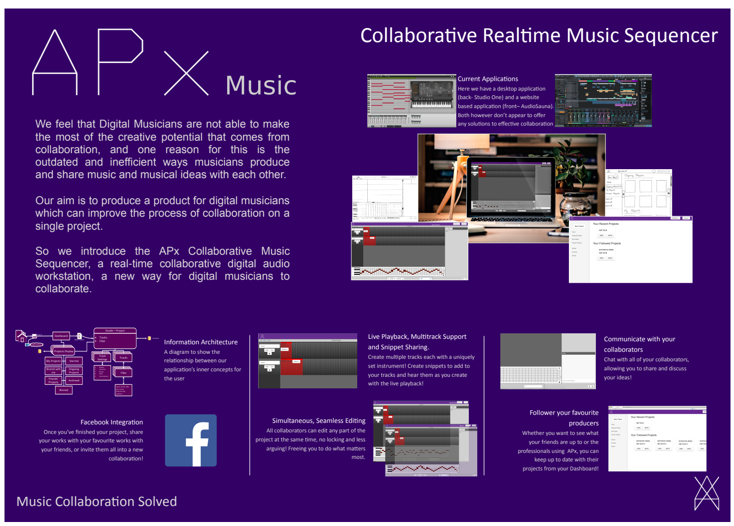

Figure 5.8: APX Poster

The absence of the authors' name from the poster appears to have adversely affected the voting process of this system, preventing some of the test users from voting for their favourite posters. In the quantitative analysis section of this chapter as a solution to the problems encountered, it was suggested to adjust the threshold value in order to reduce the number of FNs and FPs. However it would be difficult to apply a global threshold value that could be applicable to both posters with a title and authors label, and posters with just a title. As it stands the KNN algorithm is run by calculating similarities between the text captured in the poster image by the OCR process, and the title and author attributes of poster entries in the database.

A possible solution could be to calculate the similarities using just the title attributes of poster entries in the database. In which case a global threshold value could be adjusted so that it can cover both posters with a title and author label, and posters with just a title and no author label. However there is the possibility that two posters may have the same title name, and the only way to distinguish them would be by using the posters' authors names. An alternative solution could be to restrict this system to support only academic poster sessions, so that we can make the assumption that on each posters a title exists and its authors' names are positioned below it.

Other feedback given included positive feed of the UI design and its consistency of style between the web app and the iPhone app. Furthermore it was pointed that one particular transition in the iPhone app, between the event info view and the vote view, was abnormally slow. In response to this feedback, measures can be taken to remedy this latency. If the app were to be deployed at a larger sized event, additional questions would have been added to the survey. For instance questions inquiring about the age of users or the field that they work. Such information can be used to correlate user opinions of Postr with certain demographic groups.

Figure 5.9: Department of Computing Poster Session

Chapter 6

Conclusion

6.1 Future Work

To conclude this report, we will explore the various extensions and improvements that could be made on Postr. Furthermore we discuss what changes that would have to be made if this system were to be employed at large-scale event. Naturally if there had been more time some of these could have been implemented during the scope of this project.

6.1.1 Data Capture Methods

The data capture method employed in Postr is OCR, which extracts out the title and author of an image of a poster. However the voting process that ensues involves a KNN search, and also requires confirmation from the user that the intended poster is amongst the calculated most similar posters. Furthermore due to the nature of OCR, the poster recognition process can not guarantee a 100% accuracy rate in returning the correct poster to the user. This is simply because even the best OCR system cannot guarantee that the text recognised matches exactly the text in the poster image.

An expansion on this system would be to employ a data capture method which directly indicates which poster is being voted. In which case the voting process would be considerably faster. For instance the use of QR codes or NFC technology could serve as an alternative data capture method. The voting process would be much more immediate, as the processing time of OCR and a KNN search can be avoided. However a drawback of using these technologies would be that it would have to be a pre-requisite that posters had a QR code or NFC chip installed on them. Otherwise users wouldn't be able to vote for posters using these data capture methods.

6.1.2 Scalability

In order to deploy Postr a change in database would be required. The main advantage of using a simple SQLite database was that the set up required very little configuration. However SQLite databases are not well suited to scalability and are generally used in small scale applications such as low to medium traffic websites, embedded devices, etc [1]. If Postr were to be deployed, and assuming the system is popular, it would have to use a production grade database. During the evaluation of Postr, the system encountered no problems when dealing with multiple test-users. However performances issues in the current database are to be expected when dealing with hundreds or thousands of voting users.

Under ideal conditions, separate APIs would be required for the iPhone and web app. This is due to the fact that mobile apps have lower processing power, smaller screens, less memory, different user behaviours and expectations, and are more likely to be disconnected than desktop browser. Additionally in a production environment separate APIs would be required so that during maintenance or upgrades, both web and mobile services wouldn't be lost.

6.1.3 Additional Features

The underlying objective of Postr is to provide a platform on which a poster voting competition can be hosted. This objective has been met accordingly in this project. However to expand the user experience of Postr, additional features could be added. For example we could integrate this system with social media sites. When a user votes for a poster, they would be given the option of posting/tweeting their preference on Facebook or Twitter.

In the problem statement at the start of the report it is assumed that the user of the app is the rightful owner of the app. What if the current user

[1] https://www.sqlite.org/whentouse.html

had stolen the phone and is abusing the votes cast by the original user? One extension could be to introduce security measures, e.g. Touch ID security, when casting votes to avoid this. Another could be to implement augmented reality, which would provide an enhanced user experience by giving over-lay information about the poster when the user is viewing a poster. Finally, this project could be continued by porting this system's iPhone app to other iOS devices, such as the iPad, and also to other platforms such as Android.

Bibliography

[1] Schantz, H. F. *The history of OCR: optical character recognition, Recognition Technologies Users Association*. 1982.

[2] Line Eikvil. *Optical Character Recognition*. 1993.
Available: `https://www.nr.no/ eikvil/OCR.pdf`

[3] Andrew S. Tanenbaum and Herbert Bos. *Modern Operating Systems (4th Edition)*. 10th March 2014.

[4] Ravina Mithe, Supriya Indalkar and Nilam Divekar. 'Optical Character Recognition'. In: *International Journal of Recent Technology and Engineering (IJRTE)*. 2013.
Available: `http://www.ijrte.org/attachments/File/v2i1/A0504032113.pdf`

[5] Ray Smith, Chris Newton and Phil Cheatle. *Adaptive Thresholding for OCR: A Significant Test*. 1993.
Available: `http://www.hpl.hp.com/techreports/93/HPL-93-22.pdf`

[6] Wojciech Bieniecki, Szymon Grabowski and Wojciech Rozenberg *Image Preprocessing for Improving OCR Accuracy*. 2007.
Available: `http://wbieniec.kis.p.lodz.pl/research/files/07_memstech_ocr.pdf`

[7] R. Fisher, S. Perkins, A. Walker and E. Wolfart. Hough Transform. 2003 (visited on 23/05/2016).
url: `http://planning.cs.uiuc.edu/node444.html`

[8] Ivo Vynckier. How OCR Works: A Close Look at Optical Character Recognition. 2016. (visited on 14/05/2016).
url: `http://www.how-ocr-works.com/`

[9] Yasser Alginahi. *Preprocessing Techniques in Character Recognition* . 17th August 2010.
Available: `http://cdn.intechopen.com/pdfs/11405.pdf`

[10] Steven M LaValle. Planning Algorithms: Optimal Strategies. 20th April 2012 (visited on 16/05/2016).
url: `http://planning.cs.uiuc.edu/node444.html`

[11] Jason Brownlee. How To Implement Naive Bayes From Scratch in Python. 8th December 2014 (visited on 18/05/2016).
url: `http://machinelearningmastery.com/naive-bayes-classifier-scratch-python/`

[12] Steven M LaValle. Planning Algorithms: A Bayesian Classifier. 20th April 2012 (visited on 22/05/2016).
url: `http://planning.cs.uiuc.edu/node444.html`

[13] Tom M. Mitchell. *Machine Learning* 1st October 1997.

[14] Maja Pantic. 'Lecture 7-8: Artificial Neural Networks I'. *Machine Learning C395*. 2016.

[15] Maja Pantic. 'Lecture 9-10: Artificial Neural Networks II'. *Machine Learning C395*. 2016.

[16] Prateek Joshi. Overfitting In Machine Learning. 9th June 2013 (visited on 24/05/2016).
url: `https://prateekvjoshi.com/2013/06/09/overfitting-in-machine-learning/`

[17] Ning Li. *An Implementation of OCR System Based on Skeleton Matching*. August 1991.
Available: `https://kar.kent.ac.uk/21129/1/OCRNing.pdf`

[18] Ivan Dervisevic. *Machine Learning Methods for Optical Character Recognition*. 18th December 2006.
Available: `http://perun.pmf.uns.ac.rs/radovanovic/dmsem/completed/2006/OCR.pdf`

[19] Donald Knuth. *The Art of Computer Programming Vol 3: Sorting and Searching (2nd Edition)*. 1998.

[20] Mohammad Reza Abbasifard, Bijan Ghahremani and Hassan Naderi. *A Survey on Nearest Neighbor Search Methods*. June 2014 (visited on

10/05/2016).
Available: http://research.ijcaonline.org/volume95/number25/pxc3897073.pdf

[21] Ding Wang and Ping Wang. *Offline Dictionary Attack on Password Authentication Schemes using Smart Cards*. 2013) (visited on 10/06/2016).
Available: https://eprint.iacr.org/2014/208.pdf

[22] Maja Pantic. 'Lecture 5-6: Evaluating Hypotheses'. *Machine Learning C395*. 2016.

[23] Jason Brownlee. Machine Learning Process: Classification Accuracy is Not Enough. 21 March 2014 (visited on 10/06/2016).
url: http://machinelearningmastery.com/classification-accuracy-is-not-enough-mor