Flowerify
Final Project Report

DT211C

BSc in Computer Science Infrastructure

Buchita Gitchamnan

C16348651

Jane Ferris

School of Computer Science

Technological University, Dublin

11/04/2020

Abstract

Flowerify is a web application that classifies and provides the flower name from an input flower image. The application describes the flowers scientific name and botanical information which includes the maintenance and upkeep of the flower such as when and where to plant this flower, possible pests and diseases.

As part of this project one of the aims is to optimise the classifier using different methods of image processing and image manipulation. The image processing GrabCut is used for image segmentation to segment the foreground and background from the input image.

The RGB values are used to filter down the list of possible outcome flowers.

Dominant RGB values are extracted from each image of the dataset which go through the GrabCut process.

The input image RGB values are also extracted. These values are then used to identify the dominant colour of each flower type.

The ColorThief python module is then used to retrieve the dominant RGB values of an image. The classifier Random Forest algorithm is used as the classification process.

For the application to be intuitive and beneficial to users from all backgrounds, part of the project is to create a visually appealing and effective user interface.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

Buchita Gitchamnan

Buchita Gitchamnan

10/04/2020

Acknowledgements

Firstly, the author would like to sincerely thank my project supervisor Jane Ferris for her help and guidance throughout this project. The author would like to thank Diane and Mark for constructive criticism of the manuscript. Many thanks to all volunteers for participating in usability testing of the system.

Lastly, the author wishes to acknowledge the support and great love of my family, my father, Mark; my mother, Phutsarat; and my brother, Yutaphong. They kept me engaged on the project and this helped me complete the project on time and successfully.

Table of Contents

Table of Figures

Table of Tables

Table of Snippets

1. Introduction

1.1. Project Background

There are over 400,000 plant species across the world. In other words, that’s more than the number of birds, butterfly and bee species combined. Many blooming flowers can be observed in the garden, park, roadside and many other locations. The identification of the plants by their flowers is done by experienced taxonomists or botanists(1).

Almogdady, Manaseer and Hiary(2) proposed a flower recognition approach based on image processing technique and artificial neural network (ANN) algorithm. The system uses image processing techniques and ANN

The image processing techniques are used for steps of enhancements, features extraction and segmentation. The phase of features extraction is done using colour moments, GLCM and IM for colour, textures and shape respectively and partial data that include all flowers species with 20 images for each have been tested in their system and achieved 81.19% of accuracy. This inspires the author to use image processing techniques in this project.

Sripian and Yusungnern(3) built a system that was developed using Python and Random Forest Classifier method. Prior to flower identification, GrabCut is used to segment the background and the foreground from the input image. The identification of flower name from the input image is done based on RGB Histogram data. They found that the proposed system is able to classify flower images with an average accuracy of 80.67%, given a database of 10 different types of flowers. This encourage the author to test out random forest classifier and use GrabCut for image segmentation.

Most people usually look up flowers guidebooks or use the relevant websites on the internet to browse the information using keywords to find out about a particular flower. The potential techniques from the researched articles may be used throughout this project.

In summary this application Flowerify is been developed to help people identify and gain knowledge about certain flowers which includes maintenance and upkeep.

1.2. Project Description

Flowerify is designed and developed to distinguish between different types of flowers from images with the help of image processing and machine learning algorithm. This project outlines the building and deployment of a web application for identifying the subject of flower images. The web application is built around flower identification functionality.

The aim of this project is to help users to detect, recognise and understand flowers.
Flowerify will be an intuitive web app beneficial to users from all backgrounds.

Flowerify will be a web application that facilitates as a platform for users. Users can input images of certain types of flower. The system will then analyse the images and output the name and general care information, when and where you can plant this flower, possible pests, diseases and meaning of the flower.

1.3. Project Aims and Objectives

The objective of this project is to make a full stack web application with features that analyses and classify input images and deploy the application. This requires several small objectives below:

- Develop an algorithm for separating the subject of a photo from the background using image processing.
- Source a suitable dataset of flower images separated into different species.
- Build and train a machine learning model that is capable of classifying the subject of a photo into a specific flower category.
- Implement an image processing for feature extraction.
- Develop a solution to identify the dominant colours of an image
- Create a website that allows users to upload photos for classification.

1.4. Project Scope

The project is focusing on investigating different methods to optimise and increase the accuracy of the chosen machine learning algorithm to distinguish different types of flowers.

1.5. Thesis Roadmap

This section will provide a summary of each of the chapters covered in this report:

Research

This chapter explores the background research related to image classification. It examines alternative existing solutions and technologies that can be used to achieve this projects goal. Finally, it discusses the technologies and methods that will be implemented throughout this project.

Design

The design chapter examines the methodology used in the development of the project. It contains detailed use-cases and prototypes of the application. The designed technical architecture and software testing plans will be discussed.

Development

This chapter investigates into the development of the web application components regarding to the technical architecture outline in the design section.

Testing and Evaluation

The chapter covers the different methods used for testing and evaluating components of the project.

Conclusions and Future Work

This chapter will reflect on the entirety of the project and will discuss the conclusions drawn, personal reflections made and the future work planned for the project.

2. Literature Review

2.1 Introduction

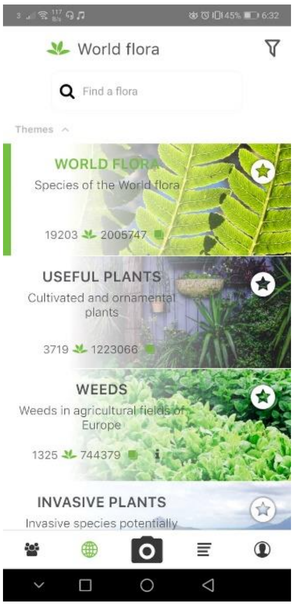
In this chapter some of the key areas of research that are important for this project will be presented. These topics include exploring some of the flower, the use of technology in helping people recognise the flower, comparing existing flower identification applications to see where they succeed, where they fail and the best ways to design a competent interactive user experience.

2.2 Alternative Existing Solutions to Your Problem

There are existing apps on the market to help a user to identify a flower:

2.2.1 PlantNet Plant Identification

PlantNet is a basic application for image sharing and retrieval the identification of plants.(4) This app helps to identify and understand all kinds of plants living in nature: flowering plants, trees, grasses, conifers, ferns, vines, wild salads or cacti from photographs. PlantNet can recognise about 20,000 species.



World flora
Find a flora
Themes
WORLD FLORA
Species of the World flora
19203 2005747
USEFUL PLANTS
Cultivated and ornamental plants
3719 1223066
WEEDS
Weeds in agricultural fields of Europe
1325 744379
INVASIVE PLANTS
Invasive species potentially

Figure 1 - PlantNet Main screen

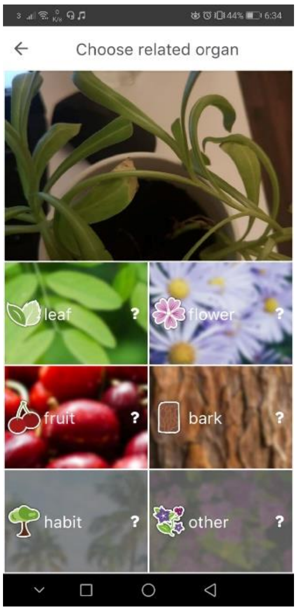
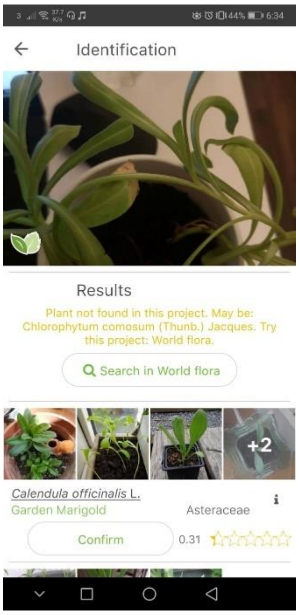


Figure 2 - PlantNet upload screen



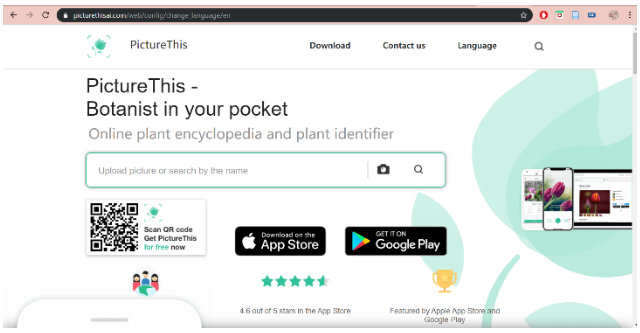
Plant not found in this project. May be: Chlorophytum comosum (Thunb.) Jacques. Try this project: World flora.

Figure 3 – PlantNet Result screen

2.2.2 PictureThis

PictureThis is available as a web application and a mobile application in Play store and Appstore. PictureThis is capable of identifying 10,000+ plant, flower and tree species with an accuracy of 98%.(5) This application allows users to learn more about plants, get suggestions and advice about the plants such as plant care tips, users can share plant photos with other users who are using this application and can view the map of plants nearby them. However, in order to use the main feature of this application a user must have enough credit in their account to identify the plant. A paid mobile application costs 19.99/year or 7 days free trial.

Web Application:


picturethisai.com/web/config/change_language/en
PictureThis
Download
Contact us
Language

PictureThis -
Botanist in your pocket

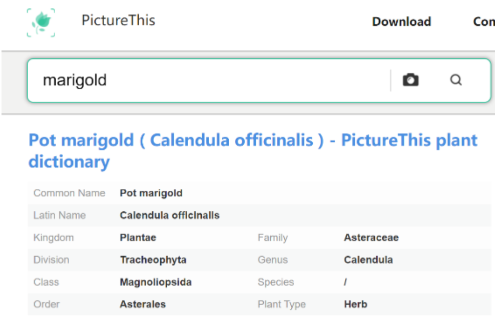
Online plant encyclopedia and plant identifier

Upload picture or search by the name
Scan QR code
Get PictureThis
for free now
Download on the App Store
GET IT ON Google Play

4.6 out of 5 stars in the App Store

Featured by Apple App Store and Google Play

Figure 4 - PictureThis web application index page

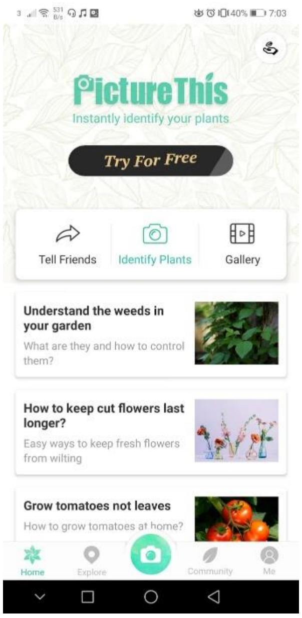


Pot marigold (Calendula officinalis) - PictureThis plant dictionary

Common Name	Pot marigold		
Latin Name	Calendula officinalis		
Kingdom	Plantae	Family	Asteraceae
Division	Tracheophyta	Genus	Calendula
Class	Magnoliopsida	Species	/
Order	Asterales	Plant Type	Herb

Figure 5 - PictureThis web application search page

Mobile Application:

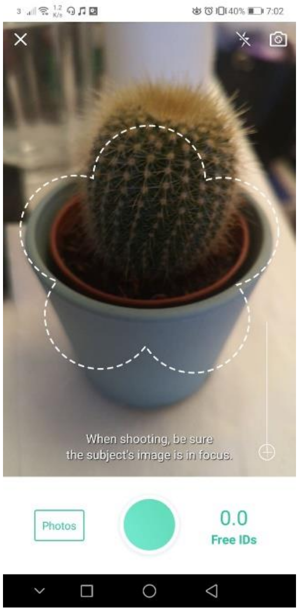


Understand the weeds in your garden
What are they and how to control them?

How to keep cut flowers last longer?
Easy ways to keep fresh flowers from wilting

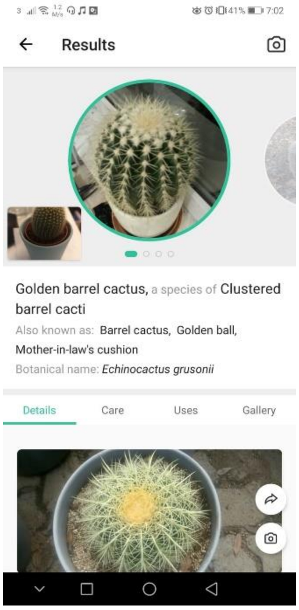
Grow tomatoes not leaves
How to grow tomatoes at home?

Figure 6 - PictureThis Main screen



When shooting, be sure the subject's image is in focus.

Figure 7 - PictureThis Upload screen



Golden barrel cactus, a species of Clustered barrel cacti
Also known as: Barrel cactus, Golden ball, Mother-in-law's cushion
Botanical name: *Echinocactus grusonii*

Figure 8 - PictureThis Result screen

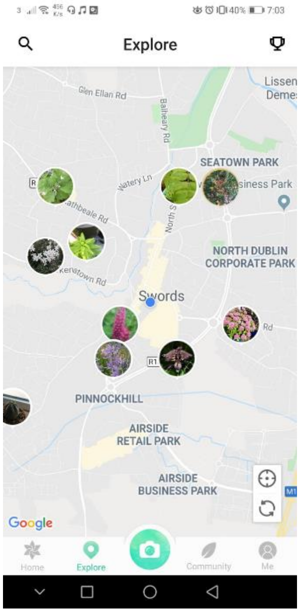


Figure 9 - PictureThis Map screen

2.2.3 PlantSnap

PlantSnap can identify 90% of all known species of plants, flowers, leaves, mushrooms, succulents, cactus and trees with 600,000+ plant species and 350 million+ images in the searchable database.(6) This application works globally and is translated into 37 languages. Mobile application built to help you identify flowers, plants & trees in a snap.

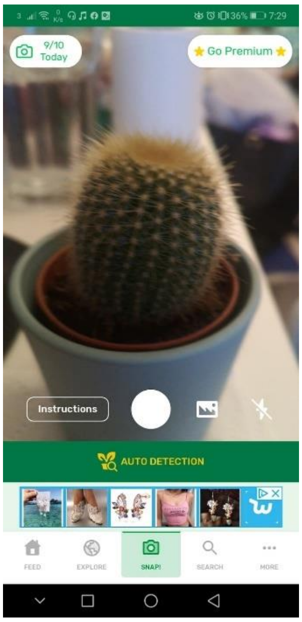
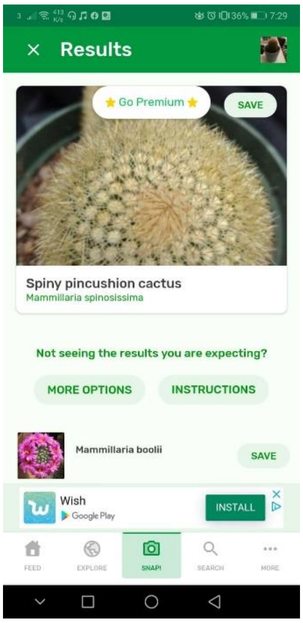


Figure 10 - PlantSnap Upload screen

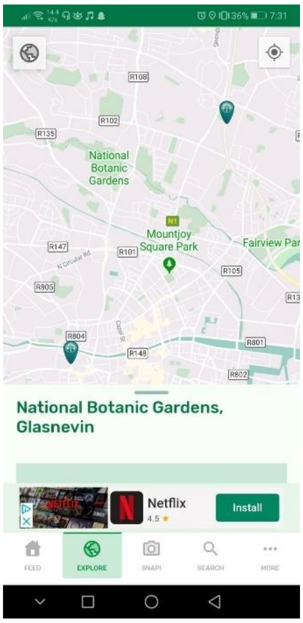


Spiny pincushion cactus
Mammillaria spinosissima

Not seeing the results you are expecting?

Mammillaria boolii

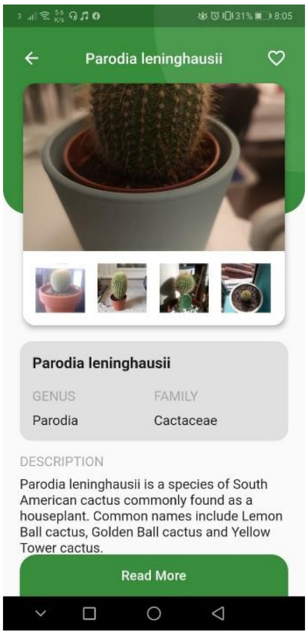
Figure 11 - PlantSnap Result screen



National Botanic Gardens, Glasnevin

Figure 12- PlantSnap Map screen

2.2.4 LeafSnap



Parodia leninghausii

GENUS	FAMILY
Parodia	Cactaceae

DESCRIPTION

Parodia leninghausii is a species of South American cactus commonly found as a houseplant. Common names include Lemon Ball cactus, Golden Ball cactus and Yellow Tower cactus.

Figure 13 - LeafSnap - result screen

LeafSnap, the free mobile app, uses visual recognition software to help identify tree species from photographs of their leaves.(7) Leafsnap has partnered with the Natural History Museum in order to create a bank of images and to help with identifying tricky trees.

2.2.5 Conclusion

To conclude Flowerify plans to learn from these apps as to what features are successful and what ones are not to implement in this system. Statistically this type of a system is implemented as a mobile application. Therefore, this project will be focusing on the development of a web application that can be used on desktops as well as mobile phones.

Author has noted different features from the examined solutions that could be useful features for this project. The main purpose of PlantNet Plant Identification and LeafSnap is to identify the plant. Another important feature from PlantSnap is a searchable database.

From examining all the existing solutions listed above the majority of the application is mobile application. PictureThis is the only solution available on mobile and web app. It has a very accuracy which is 98%. One of the main features that is in this project is the information about plants and plant care tips.

2.3 Technologies researched

The main technologies the author has researched for this project in order to develop the system.

2.3.1 Language

As for choosing language for this project, there are 2 main languages that this project could be developed on, Python and MATLAB.

2.3.1.1 Python

A high-level python supports functional, procedural and object-oriented styles of programming while having a simple syntax and being portable which can be used on Windows, Mac OS, Linux and UNIX platforms(8). Python delivers smart capacities that applicable for neural networks, identification of pictures and movement. Its compatibility with a range of libraries and tools such as open source ML library TensorFlow(9). Moreover, Python is easier and faster than Java or C++. One disadvantage of Python is that it is not suitable for mobile development.

2.3.1.2 MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation(10). The main advantage of using MATLAB for image recognition is since image recognition and matrix calculation are interconnected, MATLAB is found to be an excellent environment for deep learning and machine learning applications(11). The only disadvantage of MATLAB is the cost of license. It is very costly, users have to buy each and every module.

In conclusion, Python is the best programming language choice for development of Flowerify because of its rich collection of libraries and tools, simple syntax and portability.

2.3.2 Library

There are a few main libraries for machine learning but the top 2 libraries that will be used are OpenCV and TensorFlow.

2.3.2.1 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV is a free-to-use library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products(12). OpenCV offers access to over 2,500 algorithms to be used for deployment of various machine learning and computer vision capabilities like object identification and facial recognition(13). OpenCV provides algorithmic efficiency mainly to process real-time programs. Moreover it has been designed in a way that allows it to take advantage of hardware acceleration and multi-core systems to deploy(14).

2.3.2.2 TensorFlow

TensorFlow is a free and open source software library from Google applied for large scale machine learning processes for deep insights such as neural networks. It is used for numerical commutations with the help of data flow graphs. It has some image processing abilities but most of these are designed for formatting images to be used as training data(15). Keras is a high-level API to build and train models from TensorFlow.

It is well suited for deep learning and is perfect for neural networks with lots of layers and strange topology. TensorFlow has great tools for visualisation that assist in debugging and optimising applications. It is a widely used and freely accessible framework that is extremely well documented with hundreds of tutorials.

2.3.3 Web Framework

There are a wide range of web framework that can be used for this project. The top 2 that were picked out are Django and Flask.

2.3.3.1 Django

It is a python based free and open source web framework. Django supports Model-View-Controller (MVC) pattern. Django follows Model-Template-View (MTV) pattern. It is slightly different from MVC. The main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), then left with the template.(16) The template is a HTML file mixed with Django Template Language (DTL).(17) Django allows for scalability, re-usability and rapid development.

The main advantage of using Django is that it is “batteries included”(18). “Batteries included” means that Django comes with most of the libraries and tools required for common use cases such as authentication, URL routing, a template engine and an object relational mapping (ORM) out of the box.

2.3.3.2 Flask

Flask is a micro web framework written in python. It is said to be microframework because it does not require particular tools or libraries.(19) It has no database abstraction layer, form validation or any other components where pre-existing third-party libraries provide common functions. The benefit of using Flask is that it is easy to implement without much overhead or boilerplate.

Being a mini web serving framework, it doesn't provide enough scalability for larger projects and is comparatively slower than others. Flask is not asynchronous.

In conclusion Django is the most suitable for this project since it is widely used python web application with a “batteries included” philosophy. In comparison to Flask framework, Flask requires a separate library such as Flask login to perform user authentication. Django is a full stack web development for python whereas Flask is a lightweight and extensible python web framework.

2.3.4 Database

The top databases that can be used for this project are MySQL and SQLite.

2.3.4.1 SQLite

SQLite is a lightweight relational database included with the Python distribution. By default Django automatically creates an SQLite database for the project. It does not require a huge server running, all data is stored in a file which is easy to transfer from system to system.

SQLite is entirely self-contained and does not require a server or even any configuration. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

SQLite doesn’t support concurrency (no more than one user can be writing to the database at the same time) and it can’t scale.(20) For a simple database with only a few tables it is perfect for the task, easy setup, simple operations and no requirement to run a server process. For a large database accessible by many users simultaneously there will be a large drop in speed. This is due to the database being locked during access meaning users need to queue for access. That means there is a point at which an SQLite database base will become unusable with no way of improving performance above a specific scale.

2.3.4.2 MySQL

MySQL is an open-source high-performance and feature rich relational database management system. MySQL is for high load systems. The most common use for MySQL, however, is for the purpose of a web database.

It is quite proficient at handling multi-user interaction. It is a fast and stable server that is well suited for programs that handle large numbers of concurrent additions and edits such as order or booking systems.(21) It is also an industry standard. It remains one of the most-used database systems in the world. It is compatible with virtually every operating system

In conclusion, MySQL will be used for this project as a database because it supports concurrency and Django supports MySQL as database.

2.3.5 Web Hosting Service and Deployment

For this project, author decided to deploy the application. Web hosting is investigated for more information.

2.3.5.1 PythonAnywhere

PythonAnywhere is an online integrated development environment and web hosting service based on the Python programming language. Web applications hosted by the service can be written using any WSGI-based application framework.(22)

PythonAnywhere has great tutorials and application templates for starting a project, especially applications using the Flask or Django web framework. It also has simple configuration so apps already under development can be pulled from GitHub and setup to run publicly in a matter of minutes. The downside of this is it is a paid application however it is affordable for hosting small projects.

PythonAnywhere is the primary tool for hosting Python scripts that are required by applications we developed at our organization in other programming languages. In that way I can take benefit of python's performance efficiency with other programming languages too.

2.3.5.2 Digital Ocean

Digital Ocean is a new cloud hosting supplier. It is launched in 2011 and concentrated on developers' needs. Digital Ocean focuses on 3 key selling points: simplicity, pricing and high-performance virtual servers. It is easy and quick to set up instances which is referred as droplets. It is one of the affordable among all cloud providers with the starting price of $5/month(23) . Digital Ocean cloud is SSD-only. There are many accessible documentations for its administration. However digital ocean does not offer phone in query and response time for email can be slow. It lacks hosted databases, configuration management, analytics and load balancing among others.(23)

It is popular because of fast setup times user friendliness of the platform.

2.4 Other Research you've done

2.4.1 Dataset

Many datasets have been explored to find the suitable dataset to carry out this project.

2.4.1.1 Oxford Dataset

There are 2 sub datasets from this website. Images were gathered from various websites and some from their own photographs.(24) The first dataset is a smaller set consists of 17 different flower categories. Second dataset consists of 102 different categories of flowers.

2.4.1.2 Kaggle dataset

A dataset contains 4242 images of flowers. The data collection is based on the data flicr, google images, yandex images.(25) The pictures are divided into five classes: chamomile, tulip, rose, sunflower and dandelion. For each class there are about 800 photos. Photos are not high resolution about 320x240 pixel.

2.4.2 Classification

Artificial Neural Networks/Deep Learning

CNN is a neural network that has one or more convolutional layers and used mainly for image processing, classification, segmentation and other auto correlated data.(26) The benefits using CNN is that it helps to achieve good accuracies. The downsides of using CNN are it has high computational cost, uses a large dataset and needs a good GPU because it is quite slow in training for complex tasks.

Decision tree

Decision tree is a supervised machine learning algorithm which is used for supervised and classification problems. The benefit of using decision tree is that it is easy to understand and interpret, perfect for visual representation and requires little data pre-processing. The downside of this algorithm is that it tends to overfit.(27)

K nearest neighbour

KNN algorithm is one of the simplest of all the supervised machine learning algorithms. It simply calculates the distance of a new data point to all other training data points. The distance can be of any type for example Euclidean or Manhattan etc. It then selects the K-nearest data points, where K can be any integer. Finally, it assigns the data point to the class to which the majority of the K data points belong.(28) The advantages of using KNN are it is easy to implement, requires no training prior to making real time predictions. Therefore, KNN is faster than other algorithms that require training. The disadvantage of KNN is does not work well with high dimensional data because with large number of dimensions, it becomes difficult for algorithm to calculate distance of each dimension.

Support Vector Machine

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. The benefit for using SVM is the ability of capturing much more complex relationships between the datapoints without having to perform difficult transformations.(29) The disadvantage of SVM is that the training time is much longer that other algorithms as it is much more computationally intensive.

In conclusion, Author will be using oxford dataset with 102 classes of flowers for this project as Kaggle dataset only provides 5 classes. As for classifiers, author will examine the suitable classifiers for image classification which are Artificial Neural Networks/Deep Learning and Support Vector Machine in Prototype Development chapter.

2.4.3 Colour Dominant

2.4.3.1 Histogram in OpenCV

Histogram can be used to give an overall idea about the distribution intensity of an image(30). This distribution is plotted on a graph with pixel values (range from 0 to 255) corresponding to the number of pixels in the image.

It can be used to calculate RGB histogram features from the flower image with the work around it to get the top colour RGB.

2.4.3.2 ColorThief

ColorThief is a python module for grabbing the colour palette from an image(31). This can be used to find the palette of colours from image. It returns values of RGB of dominant colours in the image.

This is a better solution to approach to get dominant colours from an image.

2.4.4 Image processing / Image segmentation

2.4.4.1 Grab cut

GrabCut algorithm was designed by Carsten Rother, Vladimir Kolmogorov & Andrew Blake from Microsoft Research Cambridge, UK. in their paper, "GrabCut": interactive foreground extraction using iterated graph cuts(32). GrabCut is an algorithm for foreground extraction with minimal user interaction. It is considered to be used in this project for feature extraction.

2.5 Existing Final Year Projects

2.5.1 Euro Coin Classification Using Image Processing & Machine Learning - Yumin Chen

A model-based euro coin classification system has been developed in this project by collecting large datasets of images of euro coins, extracting their features and data modelling. Different approaches of building models for object recognition have been investigated and some models of euro coin denominations have been successfully developed using the visual features that are extracted from the datasets.

Different classification algorithms have been implemented and evaluated for model-based object recognition systems. Results returned in this project suggest that the Normal Bayesian classification method is 28.93% more accurate than the k-Nearest Neighbors (k-NN) method over the tested datasets.

In this project, the author mentioned that there are a few more methods that were introduced later such as the Bayesian. There are also some algorithms mentioned such as Artificial Neural Network (ANN), Support Vector Machines (SVM), that could improve the accuracy. Euro coin classification using image processing motivated this project to explore the algorithms that might improve the accuracy of the image recognition.

2.5.2 Isolate and Classify Wildlife Images - Emmet Rowe

This project outlines the building and deployment of a web application for identifying the subject of wildlife images. This functionality is available through an API so batches of images can be easily labelled, facilitating easy storage and sorting of images. The project also provides automatic background removal for uploaded images which can help digital artists and editors reduce the known problem of a bottleneck in photo processing photos for wildlife research which requires intense manual cropping.

The web application is built around these two pieces of functionality, animal identification and background removal. It is designed to encourage user interaction. Users can register for the site so their uploads are saved. Uploaded images are considered for addition to a public gallery and image processing is used to make an artistic summary of each uploaded image similar to a mosaic. The website is mobile browser friendly but a WebView based android app has also been created to allow faster access to the web application. Isolating and classifying wildlife images inspires the author to research about the technologies used in the presentation layer.

2.6 Conclusions

This chapter investigates the existing applications relating to flower identification. It also covers the various possible technologies that could be used in this project. Compare and contrast the advantages and disadvantages of each technology. The previous final year projects which are in similar domain also have been examined.

2.6.1 Requirement table

Name	Description	Priority
Image classification	Classifies the flower presents in the image	HIGH
Navbar	Allows user to navigate through the application	HIGH
Upload image	Enables user to upload image from their local device	HIGH
Accuracy display	Output result from the classification	HIGH
Gallery of flower	Enables user to browse through all the flowers from database	HIGH
Search for a specific flower	Allows user to search for a specific flower	HIGH
User login	Enables user to log into his/her account	MEDIUM
User logout	Enables user to log out from his/her account	MEDIUM
User registration	Enables user to create an account	MEDIUM
User update profile	Enables user to update his/her profile	LOW

Table 1 - requirement table

As per the requirement table above, image classification is the primary function that is vital to this project which is the reason it is in the high priority class. The UI is also needed to be constructed for this project in order to for user to interact with the system. In addition to this, recommender was planned to be added in however, it is seen as secondary function for project. It will be implemented into the system after the primary functions are finished and fully working.

3. Experiment Design

3.1 Introduction

The first section will look at the software methodology employed in this project. The next section outlines the technical architecture of the system and gives a detailed breakdown of the three tiers.

3.2. Software Methodology

In this section two software methodologies are identified to be a guideline to use for this project.

3.2.1 Waterfall methodology

It is the traditional approach for software development process. The waterfall model emphasizes that a logical progression of steps be taken throughout the software development life cycle much like the cascading steps down an incremental waterfall.(33) It is said to be a linear sequential life cycle model, each phase of the model must be completed before the next phase can begin and there is no overlapping in the phases. The whole process of software development is divided into separate phases. In this model the outcome of one phase acts as the input for the next phase sequentially.

Benefits of waterfall model are simple and easy to understand, use and manage due to the rigidity of the model.(34) Each phase has specific deliverables and a review process. It will give the developer and users a clear idea of the scope. Works well for smaller projects where requirements are very well understood.

The main disadvantage of waterfall model is that the model does not allow much reflection and revision. Once an application is in the testing stage, it is very difficult to go back to change or add in requirements that were not well documented. No working software is produced until late during the life cycle. This leads to high amounts of risk and uncertainty. It is not suitable for complex and object-oriented projects where requirements are at a moderate to high risk of changing.

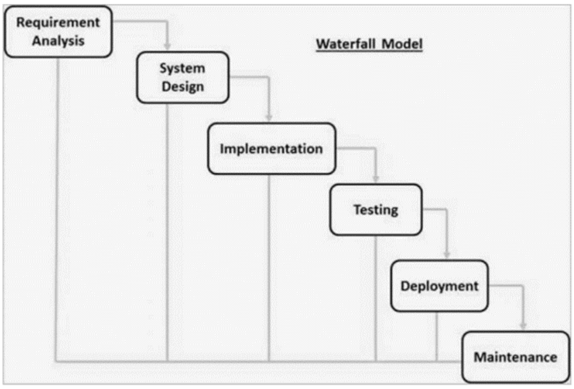


Figure 14 - Waterfall Methodology

Flowerify will have evolving needs as research and feedback will take place for the duration of the project. Therefore, Waterfall Methodology will not be suitable for the project.

3.2.2 Agile Methodology

Agile methodology allows the continuous iteration of development and testing throughout the software development life cycle of the project where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. (35)

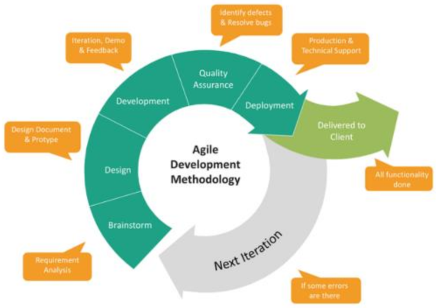


Figure 15 - Agile Methodology

Agile model is suitable for projects that have not identified the full scope requirements this means that as changes during the implementation stages are expected to occur. Testing can occur through all development stages instead of at the end of the implementation in waterfall model.

However, it is more difficult to measure progress in Waterfall because progress happens across several cycles, developers can get sidetracked into delivering new functionalities which increases the amount of unplanned work. Projects can become everlasting because there is no clear end.

Agile Methodology is more suitable for this project because Feature Driven Development approach will be followed.(36) The goal is to deliver tangible software results often and efficiently. Due to short sprints and constant feedback, it becomes easier to cope with the changes. The development of this project will be emphasising on short development cycles.

3.3. Overview of System

A feature driven development (FDD) approach will be used where a feature will be planned, implemented and tested as part of a sprint. Once that feature is implemented, development of the next feature will begin.

This approach is necessary for this project as features are the main driving point.

Design and code will be delivered in stages. A feature will be through designed and researched before implementation begins. It would not suit this project to complete all the design up front followed by all coding.

The technical architecture shows how many layers there are in the application and how the layers communicate with each other. For this project a 3-Tier model will be used, presentation, logic and data layers. The front end of the design is known as presentation layer/tier. This is where the user interface is located and client-side validation occurs. It is the only layer visible to the users. The next layer is middleware/logic layer which handles the main processing and handles requests from presentation layer. The final layer of the technical architecture is the back-end design known as the data layer. This layer handles data persistence and data access.

The diagram below outlined the overview of the system

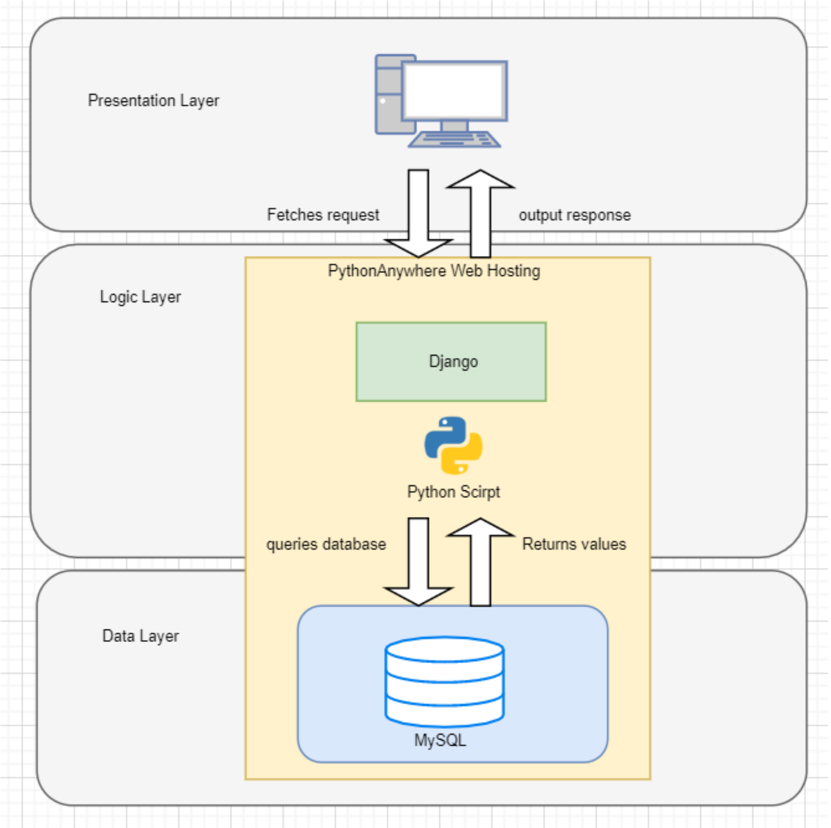


Figure 16 - Project Architecture

In the data layer, MySQL will be used for storing data such as information about the flower, the images and file storage for saving uploaded image. Logic layer is where all the computation take place, Python script and Django are the main components in this. Django follows MVT therefore HTML, CSS and bootstrap will be used in the presentation layer, for deployment the Django application which includes python scripts and database are embedded in PythonAnywhere.

3.4. Front-End

Presentation tier is the front-end layer in the 3-tier system and consists of user interface where users can interact with the application. The user interface is usually graphical interface accessible through a web browser or web-based application. Its functionality is to display content and useful information to end users. It is built on web technologies such as HTML5, JavaScript, CSS or web development frameworks. It communicates with other layers through API calls(37). This layer will allow the user to interact with the application through the web application and display the content to them, here the user will use the application to enter the image of the flower and classify the different type of flower.

3.4.1 Paper Prototype

Paper prototypes were used for the first iteration of the screen layouts. These paper prototypes were useful as they could be shown to potential users and outline the finished product.

During the design of this project’s interface, the author followed some interface design principles of Nielsen Heuristics. First the natural language is decided to be used in this application. Natural language user interface is a type of computer human interface(38) where words, phrases and concepts are similar to the user rather that system programming terms. This allows the user to understand the system, making information appear in a natural and logical order. To promote the concept of recognition rather than recall, icons and images will be used rather than big chunk of text. The use of consistent and standard structure and colour are encouraging the idea of consistency and standards principle. In order to help users recognize, diagnose, and recover from errors, some error messages will be displayed to guide users through the system.

The following images are paper prototypes outlining the interface of the system. The index page holds the information about the website and the background behind this web application. The gallery page presents all the flowers in the system with textual description.

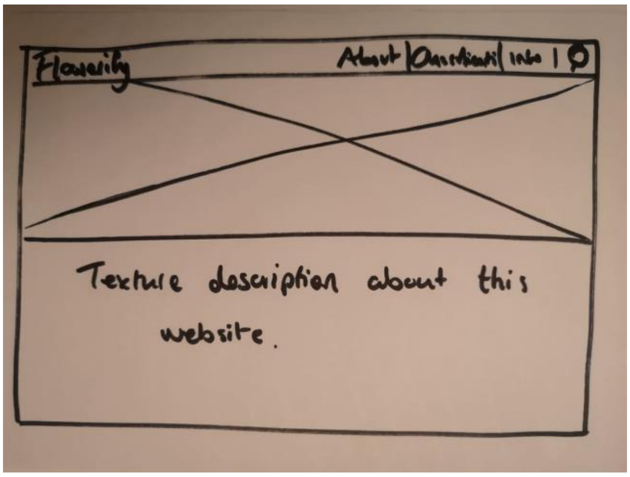

Flowerify
About | Classification | Info |
Texture description about this website.

Figure 17 - Index/Home page

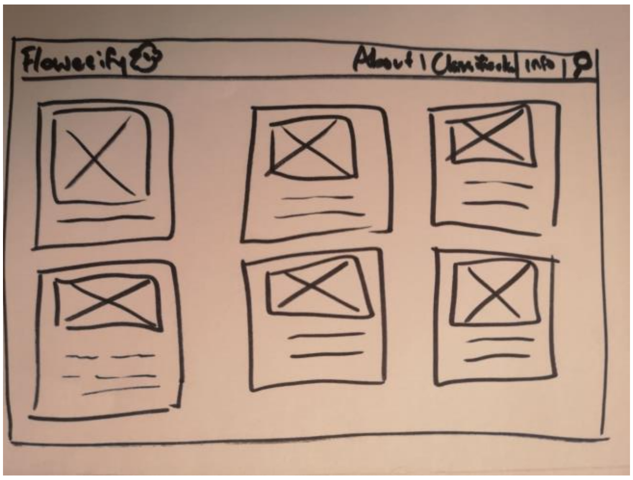


Figure 18 - Gallery page

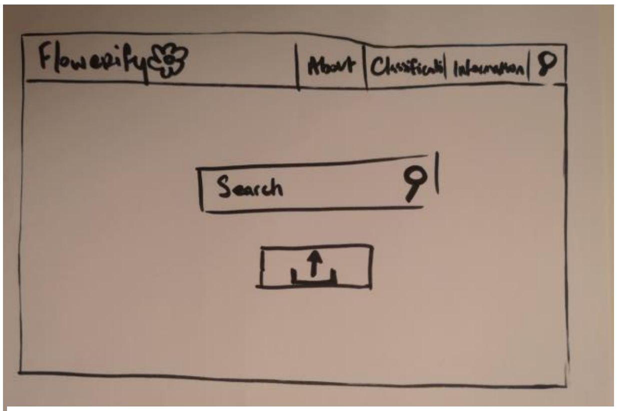


Figure 19 - Result page

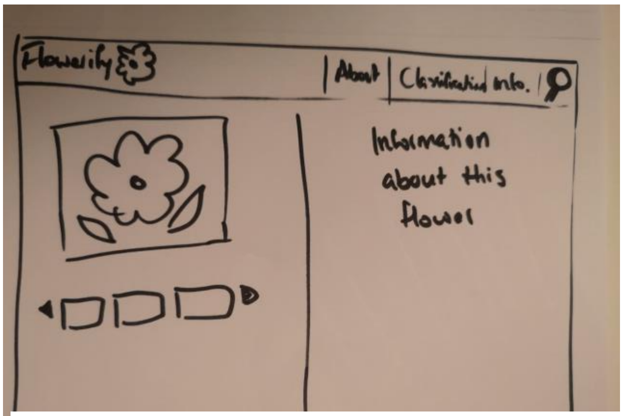

Flowerify
| About | Classification info. |
Information about this flower

Figure 20 - Upload page

The image classification page will allow the user to search, upload and take an image of the flower. The result of the classification will be display on the next screen with the information relating to that particular flower.

3.4.2 Medium Fidelity Prototype

From here, the next stage of the screen layout design phase began. For this an online application called draw.io was used. This application allowed for a high-level design for the screen layouts and interactions between the screens. This medium fidelity prototype can be seen below.

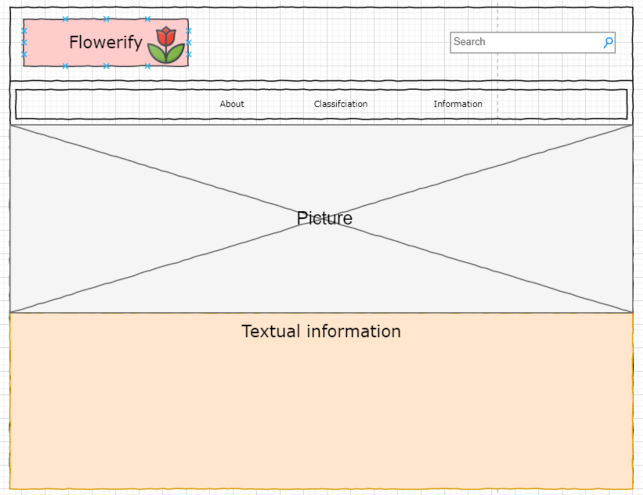


Figure 21 - Index/Home page

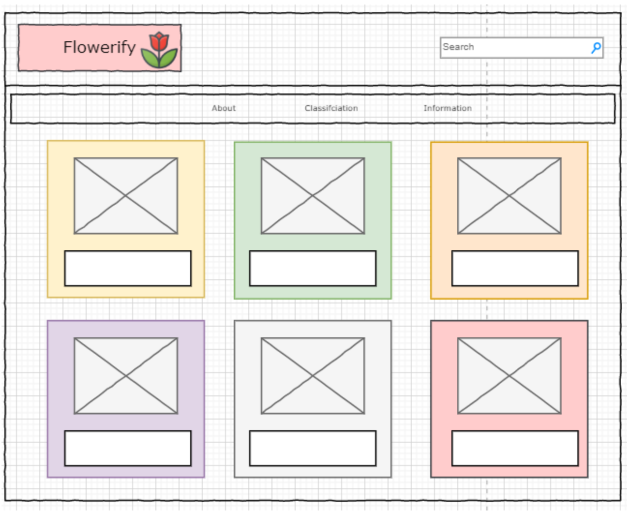


Figure 22 - Gallery page

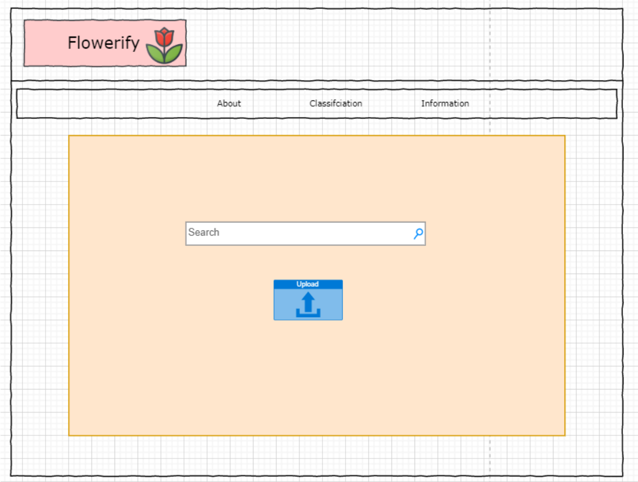


Figure 23 - Upload page

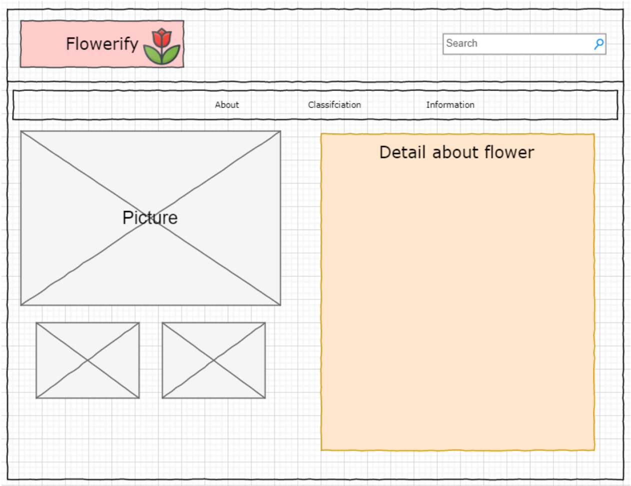


Figure 24 - Result page

All the pages will be linked together with the menu bar however the result page will only link to classification page.

3.4.3 Use Case Diagram

The user case diagrams are used to show the base functionality of the Flowerify web application.

3.4.3.1 First iteration use case diagram

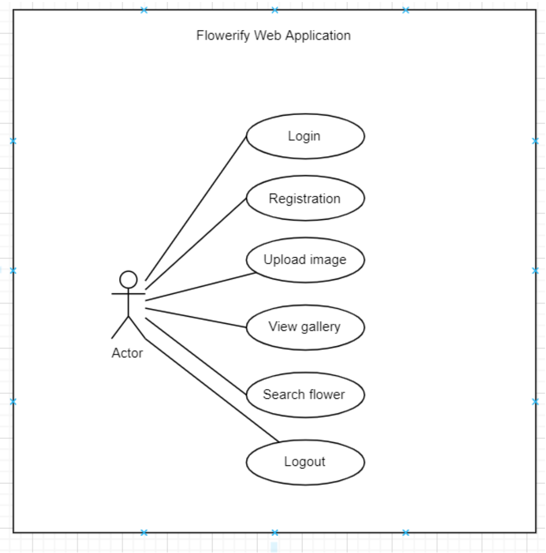


Figure 25 - initial 1st use case iteration

The initial first iteration of use case diagram for this system is as shown before. It consists of user login, logout, registration, upload, view and classification of the image. The primary functions of this system are to upload, view and classify the flower. The secondary functions are user related account.

The final first iteration of use case is as shown below. The use case includes only primary functionalities, secondary functions are removed. For this system, user account is not mandatory.

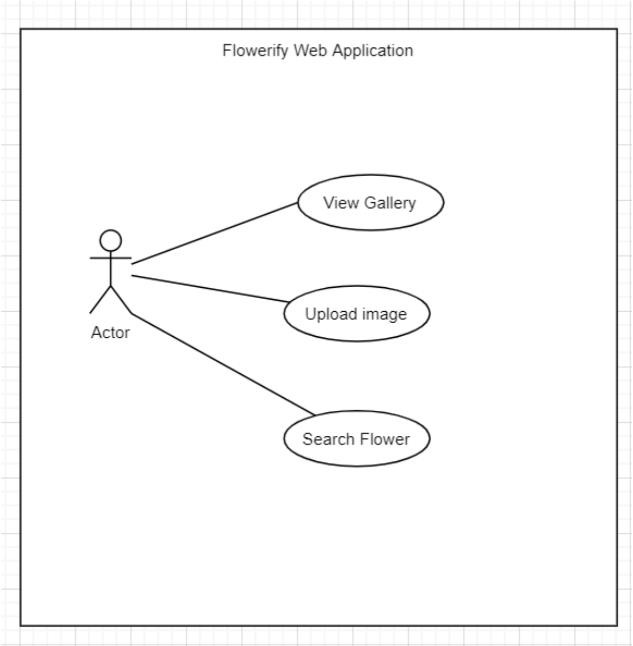
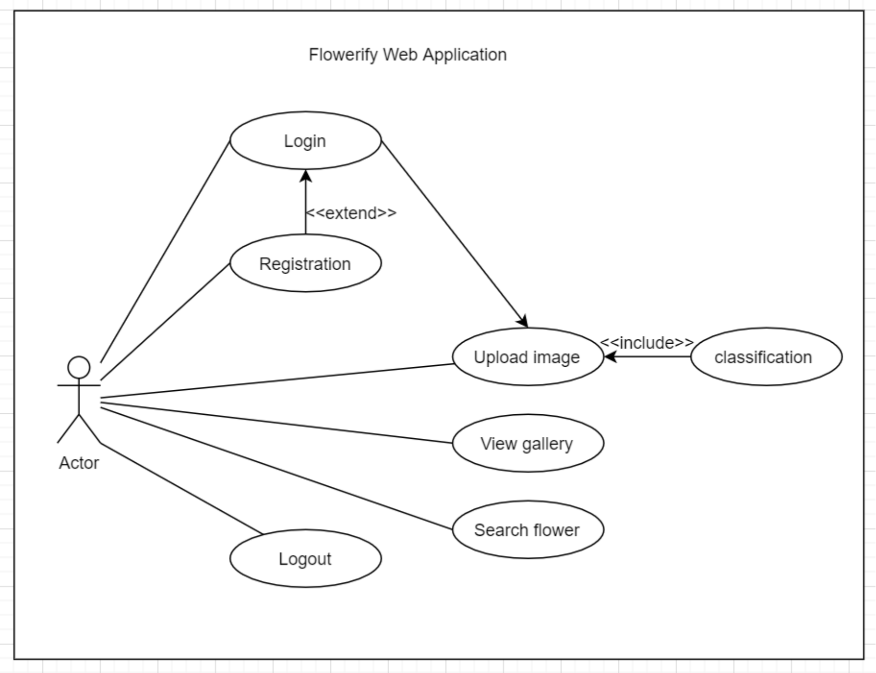


Figure 26 - final 1st use case iteration

3.4.3.2 Second iteration use case diagram



The initial second iteration use case diagram describes each action in detail. Classification use case cannot stand alone.

Upload image use case cannot complete without the included use cases (classification and prediction). The registration use case is extending to login where it is an addition action.

Figure 27 - initial 2nd use case iteration

The final second iteration use case diagram shows the primary functionalities of the system. As mentioned above the upload image use case cannot complete without classification being done.

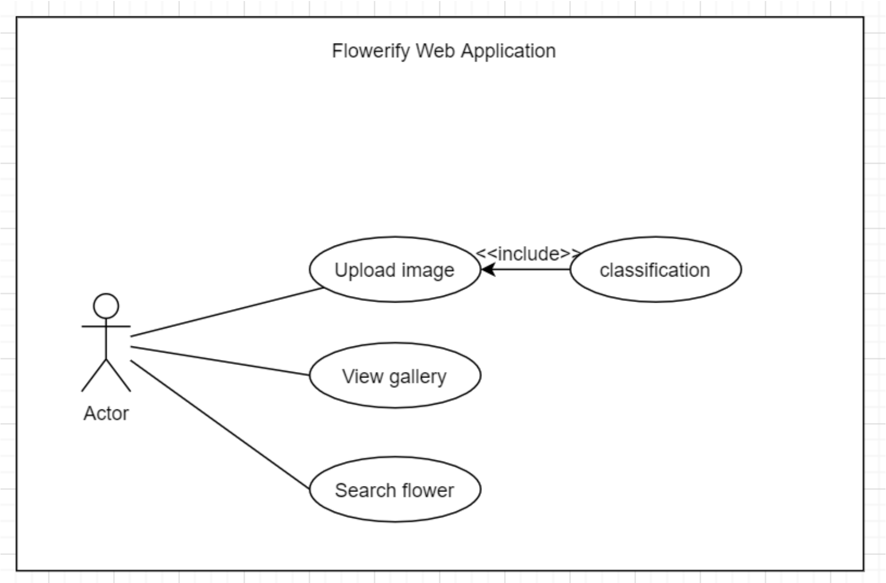


Figure 28 - final 2nd use case iteration

3.5. Middle-Tier

The middle tier or logic layer contains functional logic. It needs a code base that can fulfil all advertised features. It communicates with both the presentation and data tier. Logic tiers are designed to be durable, reliable and predictable.(37) This tier has the job of handling both image processing and image classification while also being the middleman for getting data tier results to the front end of the application. The main function for logic layer in this project is machine learning algorithm for classification. It is used for classifying different types of flower.

3.5.1 Dataset and Classification

The first approach for investigating different types of classifiers is to do quick and dirty analysis using Python generics algorithms to obtain the percentage of accuracy on the flower dataset.

The idea behind this is to narrow down the choices of the classifiers that could be used for this project. This dataset contains only 20 classes from 102 classes of oxford flower dataset. In this dataset each class has different number of images range from 50 to 200 images. The image dataset is not cleaned up not cropped and includes too much background.

The algorithm that delivered the lowest accuracy will be used in the project except the linear regression where the attempt of optimising the accuracy of the classifier is performed to get the highest accuracy percentage as possible for this project.

It is known that linear regression is not suitable for this project because it is commonly used technique in statistical data analysis. Linear regression is used for predicting output that is continuous value(39). It evaluates trends and makes estimates or forecast on it for example predicting the price of a property, where there is a linear relationship between a dependent variable and one or more independent variables(40). Therefore, the second lowest accuracy from quick and dirty analysis is chosen to be optimised for this project.

The result of quick and dirty analysis is displayed in the table below by running through the original dataset, 102 classes oxford flower dataset, consists of 10 classes.

Machine Learning Algorithm	**Accuracy**
Support Vector Machine	54%
Log Regression	51%
Neural Network	48%
Random Forest	42%
Linear Regression	-70%

Table 2 - Quick and dirty analysis of machine learning algorithms

From analysing the result, Random Forest is selected as a classifier for this project. In data science top classification algorithms are logistic regression and support vector machine naïve Bayes and decision trees(41). Random forest classifier is essentially consisting of a large number of individual decision trees.

Author applied different method of image manipulation to increase the accuracy as shown in the table using Random Forest classifier.

Method	**Accuracy**	**Image**
RGB	42%	
Normalisation	33%	
Standardisation	40%	
GrabCut	44%	

Table 3 - Image manipulating accuracy

GrabCut provided the highest accuracy percentage. Therefore, it is picked as the image processing technique to use in this project.

3.5.2 GrabCut

GrabCut was used for feature extraction(32). It is an algorithm for foreground extraction with minimal user interaction. GrabCut is decided to be running separately from machine learning process. The idea is to extract the feature from all images in the dataset. The extracted features were then analysed into dominant colours per image.

3.5.3 ColorThief

ColorThief is a python module for grabbing the colour palette from an image(31). Author used this module to find the top dominant colours from an image which features have been extracted using GrabCut. These colours were then used to narrow down the flower list.

3.5.4 Image Quality assessment

Image Quality Assessment is used in this project for detecting low quality input images. Image quality assessment is focusing on quantitatively representing the human perception of quality. These metrics are commonly used to analyse the performance of algorithms in different fields of computer vision like image compression, image transmission, and image processing(42).

The image is putting in as input and outputs a quality score as output. There are 3 types of IQAs: Full-Reference IQA, Reduced-Reference IQA and No-Reference IQA(42). The No-Reference IQA is used as IQA for this project. No-Reference is when the algorithm receives only the image as an input whose quality is being measured. It does not have anything to cross reference like Full-Reference or Reduced-Reference where only a selective amount of information is being used. BRISQUE is a model that only uses the image pixels to calculate features. BRISQUE replies on spatial Natural Scene Statistics (NSS) model(42).

3.6. Back-End

3.6.1 Database

The Data Layer comprises of the data storage system and data access layer. This is the DBMS layer and is generally accessed through the middle layer applications. A database design has been created to support the necessary elements of the project. MySQL is selected as a data storage for this project.

Initial ERD for this project is to have the user account functionality such as register, log in and log out. The picture_upload table which stores the result and path of the image that user input. The FlowerDB table is the table that stores information about the flower and an image. Lastly, result table which stores the result of classification. The structure of the tables in the database slightly changed from the initial since user account is not included in this project.

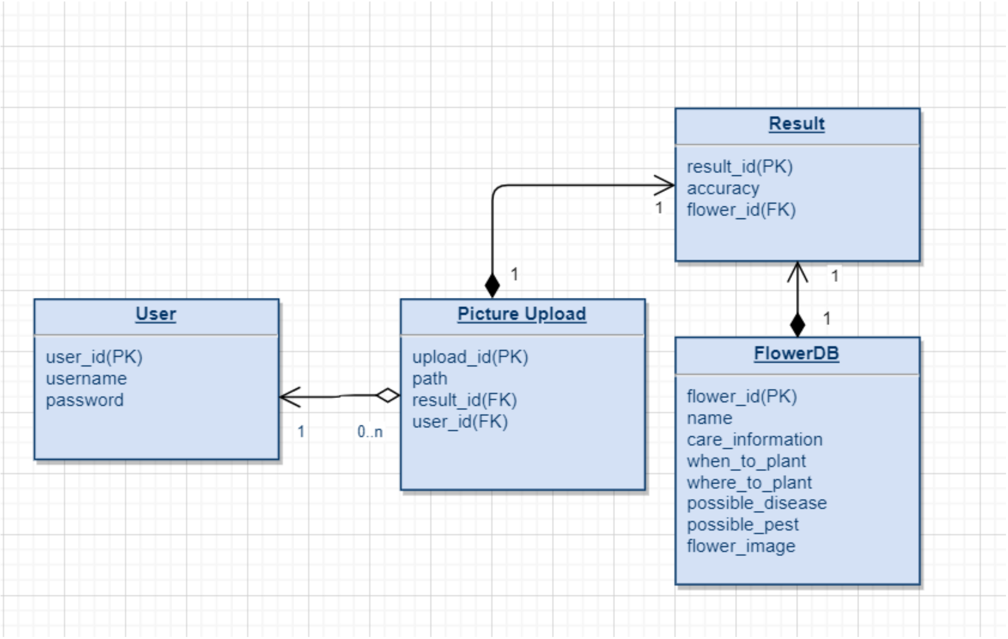


Figure 29 – initial ERD

Final ERD involves a Flower table that contains all the flower information which has id as a primary key. The other table is Flower_Dataset, this table purely storing the dataset for this project. It has flower_id as a primary key and id as a foreign key to link with Flower table. This makes this database a relational database since there is a relationship established between tables. The relationship between Flower table and Flower_Dataset is for each flower flower information; it can have more than 1 image in Flower_Dataset.

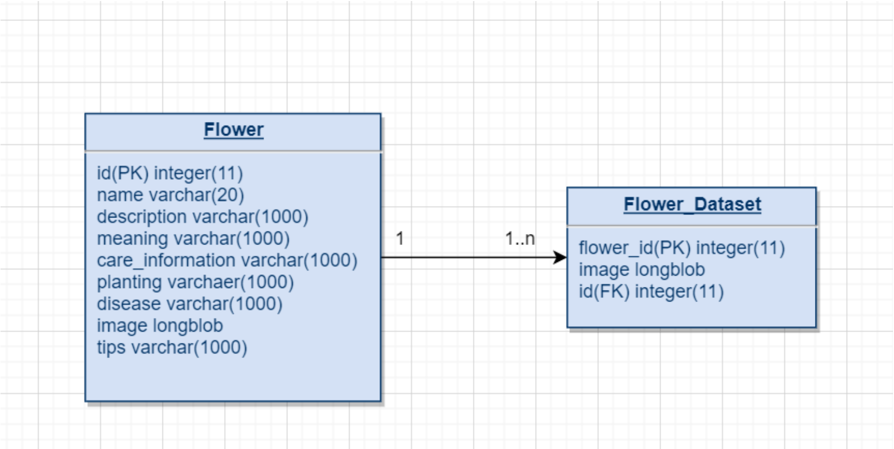


Figure 30 - final ERD

3.6.2 File Storage

The image input from user is stored in the Django FileSystemStorage class which implements basic file storage on a local filesystem(43). It is decided to store in local filesystem rather than in database because storing images in a database is more expensive than file system storage and it is slower to access an image in a database.

3.7. Conclusions

This chapter analysed the design of the system using paper prototype and fidelity prototype to outline the interface. The methodology that will be used in the development process was identified. The overview of the technical architecture was outlined where the backend layer consists of MYSQL as a database and file storage for saving uploaded image of this project. The logic layer is where all the computations happen. In this layer classification process, image processing and colour dominant analyser take place. The classifier was identified in this section which was random forest who achieved one of the lowest accuracies. GrabCut and ColorThief ideas were introduced to the project. An extra feature of image quality assessment was identified for capturing the low quality input images. The layout of the interface is decided in this section where Django is going to be used as a web framework for this project. The interface will be based in HTML, CSS and bootstrap via Django.

4. Experiment Development

4.1. Introduction

This chapter continues the subject in the previous chapter. It outlines the components created during the development process undertaken in this project. This chapter examines the processes during the creation of this system. The deployment process also discussed in this chapter.

4.2. Software Development

This first step for the development of this application was to investigate different types of classifiers for the machine learning to recognise the flower in the image. GITHUB and OneDrive are used for backing up any source code the author has implemented. GitHub is a web-based platform used for version control(44). It is also free available for TUD students. OneDrive is a file hosting service and synchronization service operated by Microsoft. It is very useful for saving files and accessing them from using different devices in other words files are not saved locally.

GitHub link to the project: https://github.com/buchita/djangoPythonAnywhere.git

Python is chosen for this project as discussed in section 2.3, PyCharm is picked as the IDE with python version 3.7.

For developing the project locally, XAMPP is used. XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl which allows author to build and develop the application on a local web server(45). XAMPP provides 2 essential components Apache and MySQL. Apache which is used to create the local server and MySQL is a database which is the database that is used in this project.

For deployment, PythonAnywhere is an online integrated development environment and web hosting service. PythonAnywhere offers an editor for writing code to make the application, any changes can be made through the interface.

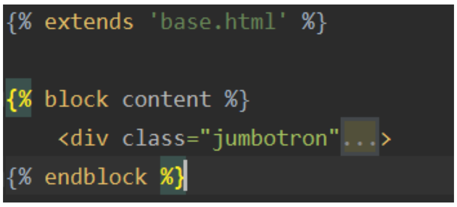
Draw.io is a free online diagram editor that can be used to create flowcharts, UML, entity relation and other diagrams(46). Author used draw.io for creating diagrams such as UML, ERD and fidelity prototype for this project.

Questionnaire was carried out using Google forms for evaluating the system. Google Form is an online survey application for creating surveys or questionnaires.

4.3. Front-End

Django is chosen for this project's front end. It is a free and open source web application framework written in python(47) which has combination of python, html and CSS. It uses MVT, model-view-template architecture pattern. Its main goals are simplicity, flexibility, reliability and scalability.

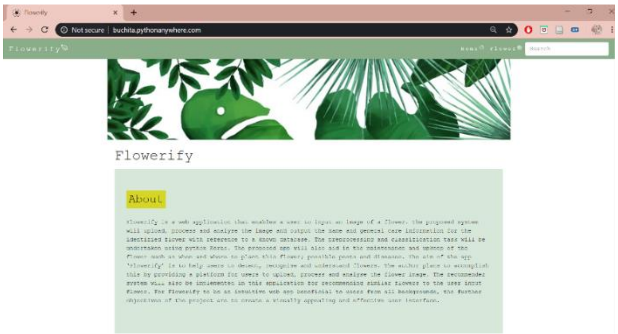
Django provides a templating structure that promotes the DRY (Don't Repeat Yourself) principle when developing the web application. It can be seen throughout the pages of the web where a page (base.html) acts as the parent page that all other pages in the project is derived from. Any headers such as CSS and CDN links are imported in the base.html so other pages can use these as well. Any components required throughout the application is placed in this base.html file such has navbar, tittle and header image of the application. Bootstrap is applied to the entire application which provides styling and responsive template. Other pages can then 'extend' from this base.html by using the keyword and add the new content block as showed in the snippet.



```
{% extends 'base.html' %}

{% block content %}
    <div class="jumbotron"...>
{% endblock %}
```

Code Snippet 1 - extends and content blocks

As designed the layout of the application, this is the index page for this application. It consists of navbar or menu bar at the top for easy accessibility to other pages, a title and information about this application.



Flowerify is a web application that enables a user to input an image of a flower. the proposed system will upload, process and analyse the image and output the name and general care information for the identified flower with reference to a known database. The preprocessing and classification task will be undertaken using python Keras. The proposed app will also aid in the maintenance and upkeep of the flower such as when and where to place this flower; possible pests and diseases. The aim of the app 'Flowerify' is to help users to detect, recognise and understand flowers. the author plans to accomplish this by providing a platform for users to upload, process and analyse the flower image. The recommender system will also be implemented in this application for recommending similar flowers to the user input flower. For Flowerify to be an intuitive web app beneficial to users from all backgrounds, the further objectives of the project are to create a visually appealing and effective user interface.

Figure 31 - Final Home page

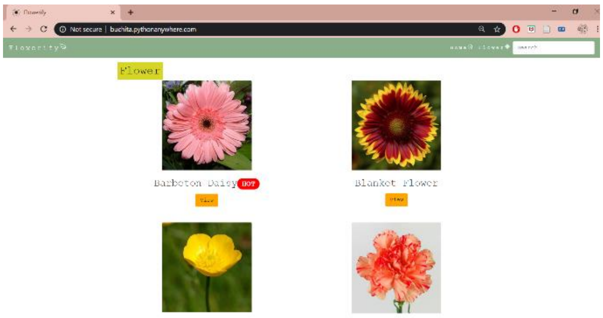


Figure 32 - Final Gallery page

The figure below is a Gallery Screen, this is where different flowers from the database are displayed. User can view each flower by clicking view button located below the image.

This is an upload screen where user can input image for classification.

The upload page linked to the result page where the prediction is presented to the user.

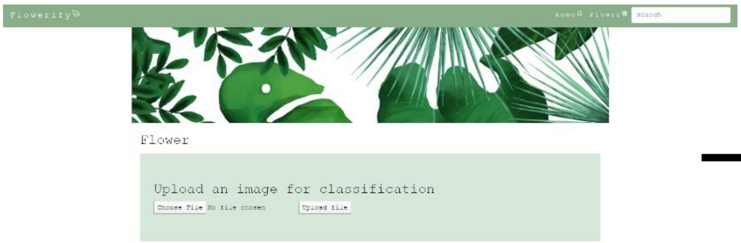
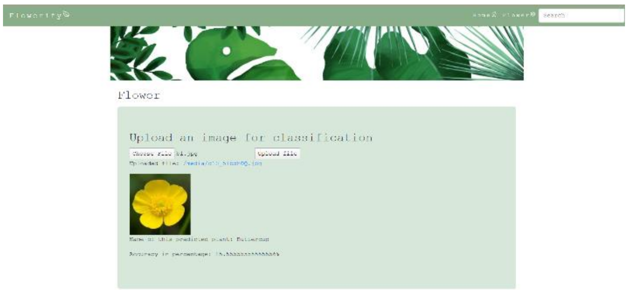


Figure 34 - Final Upload page



Uploaded files: [illegible]

Name of this predicted plant: Buttercup

Accuracy in percentage: [illegible]

Figure 33 - Final Result page

The challenge the author faced in developing the interface was Django. Django was a new web framework with unique structure where author has never learned it. It was a big learning curve, author took a while to become familiar with Django. Another difficult task while developing the interface was to convert blob into base64 for displaying images in html and convert base64 to numpy array for machine learning task.

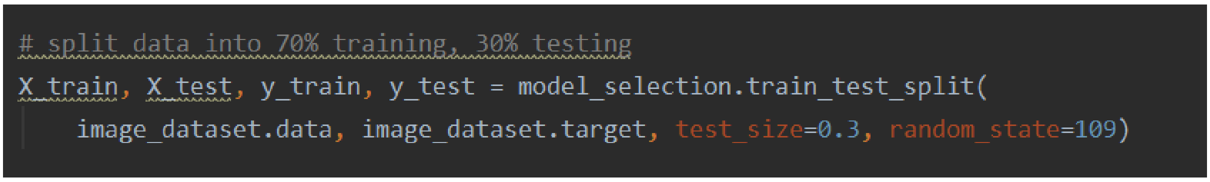
4.4. Middle-Tier

This project has many computational processes. The main process is classification process where the input image is being identified by the machine learning algorithm into classes. Here is the detail about how the classifier was being implemented and what each component was used for in the project.

4.4.1 Dataset and Classification

Firstly, the dataset went through a preparation process before it can be used. These preparations include extract 10 classes from the dataset where each class has exactly 100 images, so no bias was introduced. The images were then cleaned and tidied up where images are cropped to focus on just a flowerhead and less background.

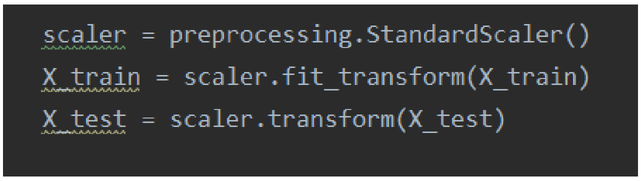
After identifying a classifier that will be used for this project the author proceeded to apply some pre-processing techniques to the images of the dataset. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing(48). The images from a dataset in MySQL database were retrieved which is in form of blob. They converted to base64 in order to convert to RGB numpy array. The label of each image was stored in an array sequentially. This information was then passed to the machine learning algorithm for classification. Before beginning any classification process, the images in dataset were resized to 200 by 200. The dataset passed to train_test_split() to split the dataset into 70% training and 30% testing randomly. X_train and X_test represent the training and testing images in numpy array respectively. Similarly, y_train, y_test are labels denoted in number representing each flower



```
# split data into 70% training, 30% testing
X_train, X_test, y_train, y_test = model_selection.train_test_split(
    image_dataset.data, image_dataset.target, test_size=0.3, random_state=109)
```

Code Snippet 2 - train test split

This was then put through a scaler which will ensure that all values fit in the same range. This is done to speed up the algorithm’s performance.



```
scaler = preprocessing.StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Code Snippet 3 - scaler

These then passed to random forest classifier to compute the result.

4.4.2 GrabCut

The image was first blurred using medianBlur resulting in smoothing edges of the image. In order to extract the feature out the rectangle measurement is defined around the foreground region which should be completely inside the rectangle(32). The algorithm considered colour distribution information inside a rectangle. The algorithm is then segmenting the region inside the rectangle iteratively to get the best result. To get rid of any error reading of flower dominant colour the image was segmented. The reading does not include the background such as leaves.

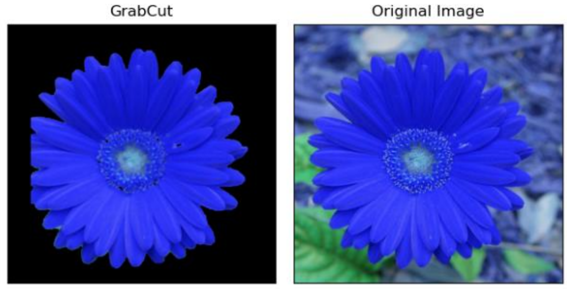


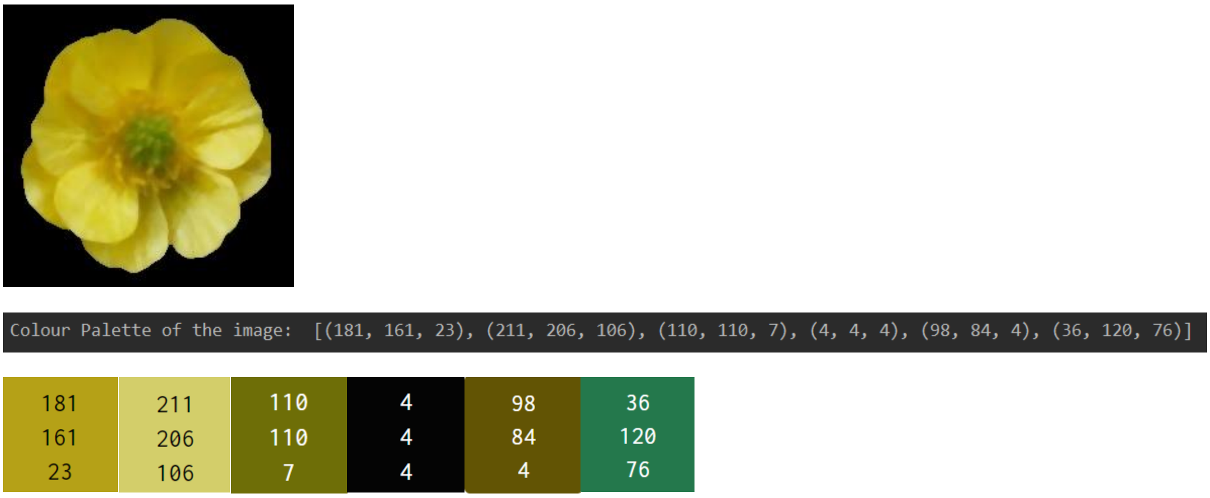
Figure 35 - GrabCut process

The challenge for this algorithm was that each image had different size, so it was hard to predict the global size for the rectangle. The author had to keep changing the size of the rectangle to suit the image.

4.4.3 ColorThief

After image segmentation using GrabCut the segmented image was then passed to ColorThief to determine the dominant colour from the image. RGB of black is removed from the ColorThief palette as it could be shown as dominant colour. This case occurred when the background is larger than the feature that was extracted. Only a top dominant colour was picked out from each image and stored in a list then compared to an input image's dominant colour to find similar range of RGB. This helps to short list the possible flower prediction. The list of certain possible flower was then passed to the classifier to classify. In the case of having no similarity of RGB the whole dataset would be passed.

Sample of the colour palette of an image of buttercup. The returned RGB values were all different colours from the image. Majority of the colour would be yellow since it is a buttercup image.

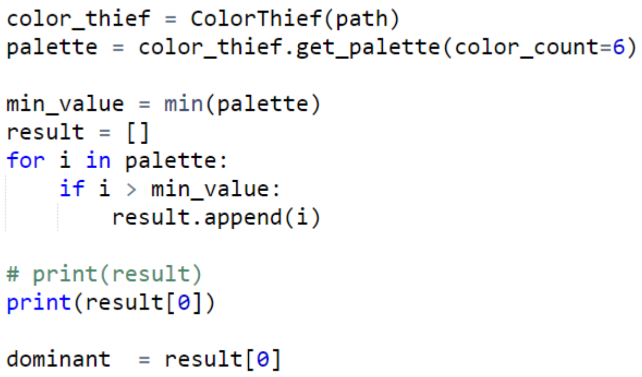


```
Colour Palette of the image:  [(181, 161, 23), (211, 206, 106), (110, 110, 7), (4, 4, 4), (98, 84, 4), (36, 120, 76)]
```

181	211	110	4	98	36
161	206	110	4	84	120
23	106	7	4	4	76

Figure 36 - ColorThief Process

The snippet below shows the process of extracting the RGB values from an image. The get_palette() retrieves the 6 highest colour repeated in the image. The lowest value of the palette, usually black which is from the background, is decided to be removed from the palette where it can be the top dominant colour.



```
color_thief = ColorThief(path)
palette = color_thief.get_palette(color_count=6)

min_value = min(palette)
result = []
for i in palette:
    if i > min_value:
        result.append(i)

# print(result)
print(result[0])

dominant  = result[0]
```

Code Snippet 4 – colorthief

4.4.5 Image Quality Assessment

As discussed in design section, BRISQUE score is used for measurement. The table below demonstrate the output values of the evaluated image.

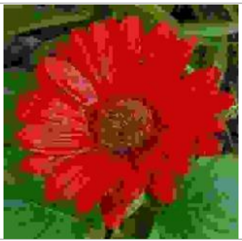
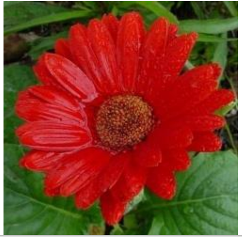
	BRISQUE SCORE	**IMAGE**
Compression quality of the image - pixelated	82.30792675255279	
High quality image	29.871755041390514	

Table 4 - sample IQA Brisque score

The lower the BRISQUE score, the higher quality of the image. The mean score of the image needs to be decided between 0 (best) and 100 (worst) in order to determine the quality of the image. This score is called the Mean Quality Score.

The input image’s brisque score is calculated every time the user uploads an image to check on the quality of the image. The mean quality score of this system is 70. If the brisque score is higher than 70, user will be asking to re-enter a clearer image again.

The challenge in this section was author did not have any previous experience in image processing and machine learning algorithms at the time of the beginning of the project. A lot of research was done by the author to get to the finish product.

4.5. Back-End

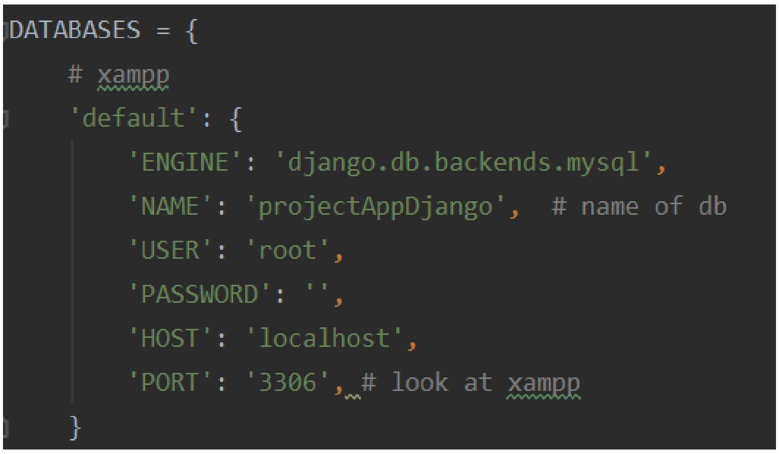
Back end is responsible for storing data which mainly is images. In this section, there were 2 types of storage that were identified in this project.

4.5.1 Database

The entity relationship diagram that was shown in the design section of this report, these tables were created in a MySQL database for the data tier of this project. Mysqlclient(49) is installed as database API driver to link between MySQL and Django. This allows Django to communicate and query the statements to and from MySQL.

Django comes with SQLite3 as a default database. Author has switched over to MySQL as a database of choice.

The connection of MYSQL database was established by providing the necessary details in setting.py.

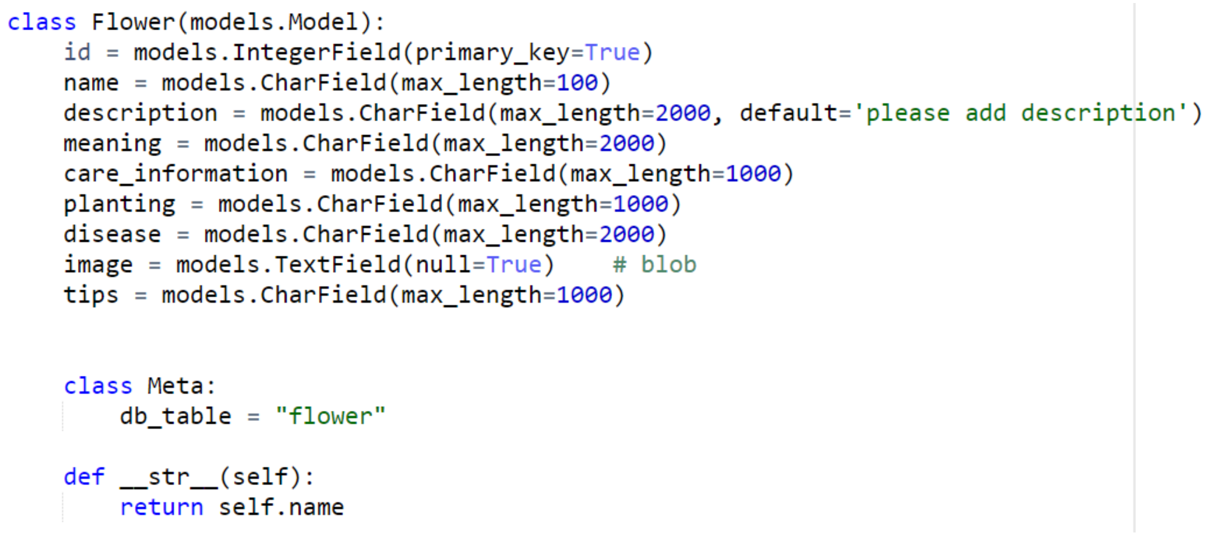


```
DATABASES = {
    # xampp
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'projectAppDjango',  # name of db
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306', # look at xampp
    }
```

Code Snippet 5 - XAMPP Database connection

Django has a unique way of conducting query to the database. By using Django's ORM (Object-relational mapping), the author can query database in Django. In order to make queries to a database in Django there are series of steps to take. It is necessary to declare a model which represents the variables in the database table. To make the application aware of this model it needs to be registered on an admin site. Finally the model can be instantiated in the view.py for making queries to the database.

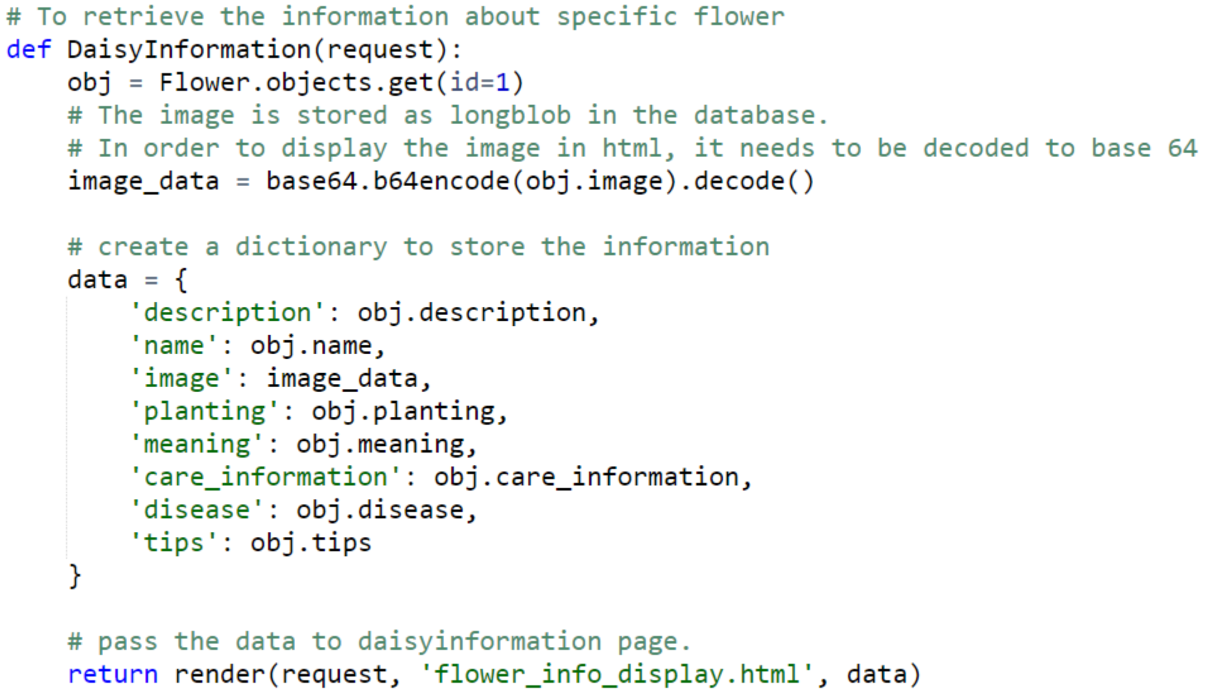
In the snippet below shows the process of querying the first object from the Flower table. This model contains the same variables with flower table in the database.



```
class Flower(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=100)
    description = models.CharField(max_length=2000, default='please add description')
    meaning = models.CharField(max_length=2000)
    care_information = models.CharField(max_length=1000)
    planting = models.CharField(max_length=1000)
    disease = models.CharField(max_length=2000)
    image = models.TextField(null=True)    # blob
    tips = models.CharField(max_length=1000)


    class Meta:
        db_table = "flower"

    def __str__(self):
        return self.name
```

Code Snippet 6 - Flower Model

The image was stored in a blob. Therefore Imagefield() cannot be used in querying and displaying the image. Blob needed to convert into base64 which returns encoded bytes then decoded bytes to a string. All the data store in a dictionary and pass it to a html file.

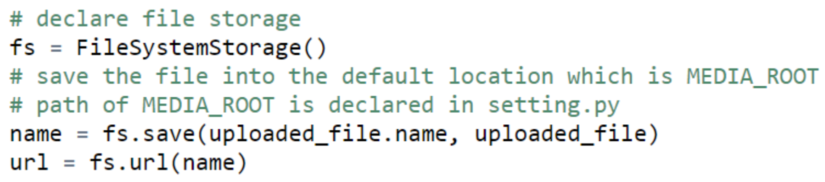


```
# To retrieve the information about specific flower
def DaisyInformation(request):
    obj = Flower.objects.get(id=1)
    # The image is stored as longblob in the database.
    # In order to display the image in html, it needs to be decoded to base 64
    image_data = base64.b64encode(obj.image).decode()

    # create a dictionary to store the information
    data = {
        'description': obj.description,
        'name': obj.name,
        'image': image_data,
        'planting': obj.planting,
        'meaning': obj.meaning,
        'care_information': obj.care_information,
        'disease': obj.disease,
        'tips': obj.tips
    }

    # pass the data to daisyinformation page.
    return render(request, 'flower_info_display.html', data)
```

Code Snippet 7 - Query to database

4.5.2 File Storage

The file system storage is used for storing temporary input images instead of the database because it is quicker in saving and retrieving than accessing database. The file system is instantiated and the image name is passed to save function to save the image in the default location of the file system which is MEDIA_ROOT(50). The path of MEDIA_ROOT is declared in setting.py which in this case is /media.



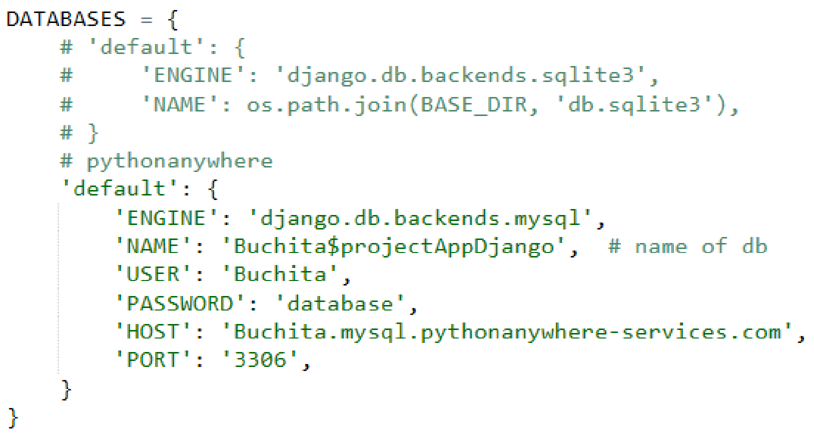
```
# declare file storage
fs = FileSystemStorage()
# save the file into the default location which is MEDIA_ROOT
# path of MEDIA_ROOT is declared in setting.py
name = fs.save(uploaded_file.name, uploaded_file)
url = fs.url(name)
```

Code Snippet 8 - file system

4.6 Deployment on PythonAnywhere

The deployment process was challenging since few changes needed to be adjusted in order for the application to work.

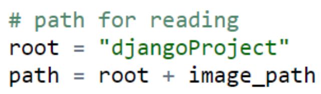
First of all, a database needed to be created on PythonAnywhere and the necessary information was then populated into this new database. The code snippet below shows the connection establishment.



```
DATABASES = {
    # 'default': {
    #     'ENGINE': 'django.db.backends.sqlite3',
    #     'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    # }
    # pythonanywhere
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'Buchita$projectAppDjango',  # name of db
        'USER': 'Buchita',
        'PASSWORD': 'database',
        'HOST': 'Buchita.mysql.pythonanywhere-services.com',
        'PORT': '3306',
    }
}
```

Code Snippet 9 - PythonAnywhere database connection

Another code that needed to be modified was the path of the uploaded images. Instead of getting the current directory the path of the media folder needed to be identified.



```
# path for reading
root = "djangoProject"
path = root + image_path
```

Code Snippet 10 - PythonAnywhere path

The challenge for deploying the project to PythonAnywhere was TensorFlow. An attempt of installing TensorFlow module was made. However, the error of console was ‘printing too much lines occurred’, redirecting the output to a temporary file was one of the solutions to this problem but it was not a success since author had ran out of disk quota.

PythonAnywhere has a preinstallation of TensorFlow on python 3.6, 3.5, 3.4 and 2.7, however the author was using python 3.7 it was not possible to just switch over python version or creating a new virtual environment to get this to work because Django has specific version that is relating to each python version on PythonAnywhere resulting in creating a new project on PythonAnywhere with python 3.6 and --system-site-packages needed to be declared while creating the virtual environment in order to use the TensorFlow preinstall package.

Overall the application was successfully deployed on PythonAnywhere.

4.7 Conclusions

In summary the tools that were used in this project were described these included PyCharm IDE which was the main element that was used to carry out this project. GITHUB and OneDrive were used for backing up the codes that the author wrote for this project XAMPP allowed author to implement the application on a local web server finally draw.io was used for creating diagrams for this project such as ERD, UML and fidelity prototype.

The front end developed in Django web framework where HTML, CSS and Bootstrap were used for this project. This section outlined the structure of how the application was following guidelines which promoted the DRY principle. A html file acts as a parent file to the other files i.e. child files, these child files can access all the headers such as JavaScript, CSS and CDN which were declared in the parent file, the child new content can be added in or replace the parent by using “extends” and “content” blocks. The final version of the application is also presented in this section.

The logic layer outlined the computational processes implementation and idea behind it. The dataset went through a preparation phase before it can be used in the classifier. The dataset was organised into 10 random classes which contains 100 images in each class limiting any bias that can be introduced. The process of transferring image from blob to useable RGB numpy array were explained in this section. After the transformation of blob to numpy array these images were then resizing to 200x200. The dataset was then passed to train/test split to divide the dataset into training and testing with the ratio of 70:30 respectively. The dataset was then passed to the classifier for the training. Another aspect that was in logic layer was GrabCut. GrabCut is a type of image processing which segments the foreground from the background it uses the RGB values to differentiate the foreground and background it was introduced for extracting the feature which is the foreground from the image, these GrabCut images would then be used to find the dominant colour of each flower type by using a python module called ColorThief. These dominant RGB values help to filter down the dataset to be passed to the classifier. IQA is an extra feature to capture low quality of input image by calculating the BRISQUE score any score that is lower than 70 is considered to be low quality.

Lastly the backend layer MYSQL connection establishment and query were discussed in detail where the model had to be made in order to query to the database. The deployment of the application on PythonAnywhere and the few challenges that were identified with a solution the author had made to overcome the challenges.

5. Testing and Evaluation

5.1. Introduction

The aim of testing is to ensure that as the application grows bigger it becomes more difficult to manually test each possible path of execution. As interactions between components become more complex so a small changes in an area can have a huge impact on the other areas(51). Therefore more refactoring or modifications will be required to make sure the applications behaviour is not effected (52). As a result of this automated testing is introduced as it is easier and more reliable to run every time changes have been made.

The testing methods that will be followed for this project is manual testing, the automated testing will take place to ensure correct behaviour, the classifier and usability evaluations were outlined in this section as well.

5.2. System Testing

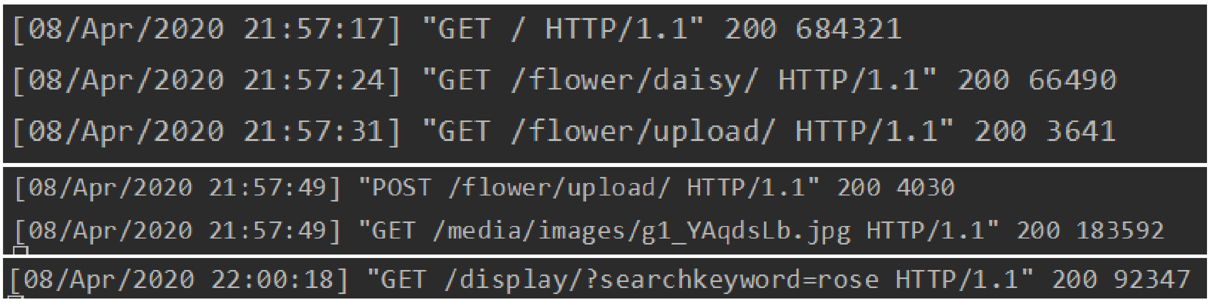
Different methods of testing have been used throughout this project to assure that the applications behaviour is correct and returns valid results.

5.2.1 Manual Testing

Manual Testing has been tested throughout the lifecycle of this project. Manual Testing is a type of software testing where testers manually execute test cases without using any automation tools(53). The main aim for manual testing is to find the bugs in the software system any new application must be manually tested before its testing can be automated. Manual testing required more effort and time. This type of testing does not require any knowledge of testing tools.

Web application testing was crucial. To ensure the project functions as designed manual testing was used as this is the interface through which the users will be interacting with the system it is important that a manual testing approach be carried out by using a manual testing approach a typical user flow can be replicated and tested for errors.

A test was performed by visiting every web page in the web application, testing all features and examining the logs.



```
[08/Apr/2020 21:57:17] "GET / HTTP/1.1" 200 684321
[08/Apr/2020 21:57:24] "GET /flower/daisy/ HTTP/1.1" 200 66490
[08/Apr/2020 21:57:31] "GET /flower/upload/ HTTP/1.1" 200 3641
[08/Apr/2020 21:57:49] "POST /flower/upload/ HTTP/1.1" 200 4030
[08/Apr/2020 21:57:49] "GET /media/images/g1_YAqdsLb.jpg HTTP/1.1" 200 183592
[08/Apr/2020 22:00:18] "GET /display/?searchkeyword=rose HTTP/1.1" 200 92347
```

Features that were tested are as follows, navigated through pages, upload the image to file system, return the uploaded image and lastly search the database for a keyword.

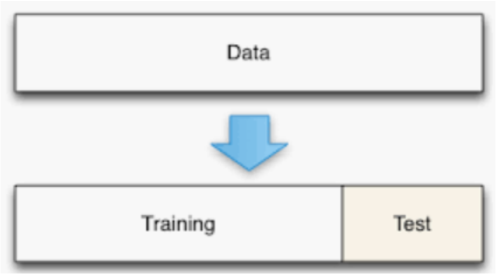
All the return code was 200 which means everything is working according to its role, the pages have been fetched without any trouble.

Along with manual testing usability testing was conducted to ensure that every task of the system can be carried out by all users regardless of their computer or IT knowledge.

5.3. System Evaluation

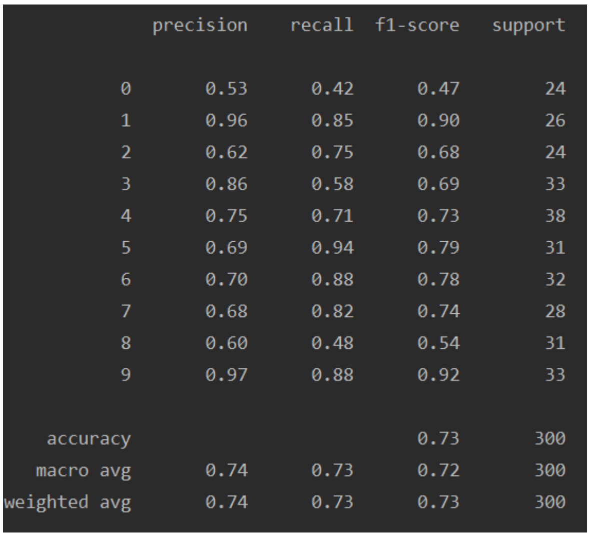
5.3.1 Image classification evaluation

Image classification evaluated using the accuracy of testing accuracy. Train/Test spilt was used to evaluate the classification. The training accuracy is when training and testing the model using the same dataset. The testing accuracy is a better estimate than training accuracy(54). Testing accuracy is involved 2 datasets (training and testing). Training dataset is used for training the model while the performance of the model is measured using the testing dataset this is called testing accuracy.



Train/Test split's ratio usually 80:20 or 70:30. For this project, the dataset is divided into 70:30 where 70% of the dataset is training dataset and 30% is for testing. The dataset is divided randomly. Train/Test split was used in this project to help preventing overfitting and underfitting

Classification Report



```
              precision    recall  f1-score   support

           0       0.53      0.42      0.47        24
           1       0.96      0.85      0.90        26
           2       0.62      0.75      0.68        24
           3       0.86      0.58      0.69        33
           4       0.75      0.71      0.73        38
           5       0.69      0.94      0.79        31
           6       0.70      0.88      0.78        32
           7       0.68      0.82      0.74        28
           8       0.60      0.48      0.54        31
           9       0.97      0.88      0.92        33

    accuracy                           0.73       300
   macro avg       0.74      0.73      0.72       300
weighted avg       0.74      0.73      0.73       300
```

A Classification report is used to measure the quality of predictions from a classification algorithm(55). Classification report outlines various attributes of algorithms performance. Number from 0 to 9 are representing different classes of flower.

Classification accuracy is the total number of correct predictions divided by the total number of predictions made for a dataset(56). The average accuracy of random forest classifier for this project is 73%.

Precision is calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes(56). In other words, it is the calculation of the sum of the actual predicted positives that were actually positive.

Recall is the number of correct positive predictions which is out of all the positive predictions that could have been made(56). This means the sum of the positives that were able to detect.

F1-Score provides a way to combine both precision and recall into a single measure that captures both properties(56). It is the weighted average of Precision and Recall e.g. the precision score of class 0 is 53 which means only 53% of them were actually Barberton daisy. The recall score of class 0 is 42 which means 42% of Barberton daisy were identified leaving 58% were missing. The f1 score is the average of precision and recall. The accuracy score is the average of f1 score.

5.3.2 Usability Evaluation

User Testing is another testing that has been performed in this project. User testing is referred to a technique to evaluate a feature or project with real users.

Black box testing is a testing technique which the application tester is testing the application without looking at the internal code structure, design, implementation and knowledge of internal paths of the software. This type of testing is based on software requirements and specifications it is focusing only at the input and output of the system without focusing at the internal code implementation. This method is often used for finding errors in the functions, interface, behaviours or performance. Application testers test the web pages by using a browser, providing inputs (clicks, keystrokes) and verifying the outputs against the expected outcome(57). The black box testing is used when the novice users tested out the application.

Grey box testing is also a testing technique which the tester has partially understanding of the internal structure, design and implementation of the software system.(58) Tester understands the purpose of the system and its core functionalities. This testing technique is very useful when testing web applications hence Flowerify will be tested under this technique for its front end. People who have knowledge about technology and understand the functionalities of this system are referred as expert and will be evaluated this system as one of the grey box testing.

The white box testing is carried out by the author throughout the development cycle as well with the finished product.

Evaluation of this project will be done by the potential users: disabled, novice and expert.

The test plan for carrying out the evaluation:

Test no.	Description	Expected output	Pass
1	Access and load website	The web application will load and bring user to the index page	🟩
2	All pages are available	Content of each page should be visible to the user	🟩
3	User can upload image	Image will be uploaded to the application	🟩
4	Image displays after uploaded	Image will show up on the output screen	🟩
5	Upload prevented without file selected	Error message will display to screen	🟩
6	Upload only jpg extension	Error message will display if other extensions are uploaded	🟩
7	Result presented after the classification process is finished	Percentage and predicted flower name display on screen	🟩
8	Classify flower	Classify flower correctly i.e. high accuracy	🟧
9	Search for certain flower	Display certain flower information	🟩

Table 5 - Test plan

5.3.2.1 Google doc survey

As mentioned in the experiment design topic under the front-end section, Nielsen Heuristics principles were considered while designing the interface. Below is the result of the questionnaire based on system by the potential users. The potential users for this application can be anyone therefore author has narrowed down to 3 types of potential users. First type is novice these users do not have a lot of experience in technology so black box testing technique can be applied to this group for testing out the system. Second type of potential user is expert this group of users have been exposed to technology and have a good amount of knowledge about that subject, last type of potential user is disabled, in this case disabled is referred as someone who is visually impaired such as wear glasses. There were 7 people who took part of this evaluation, 4 out of 7 people wear glasses and 2 out of 4 are novice. The questionnaire is divided to a few of Nielsen Heuristics principles.

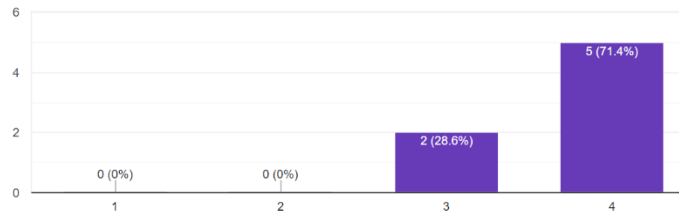
From this questionnaire, 1 represents strongly disagree and 4 is strongly agree.

Visibility System Status

It is vital that the user is aware of the system status when certain operations are performed. To provide this the web application displays the title that describes the screen content on every page to keep the user inform of which content is being presented. The consistency use of styling and icon is important since users can easily recognise the meaning. The application also displays visual feedbacks which inform users about their decision making.

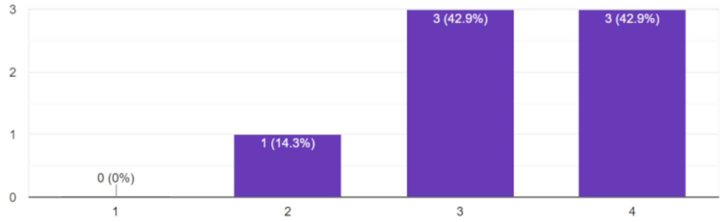
Does every display begin with a title or header that describes screen contents?

7 responses



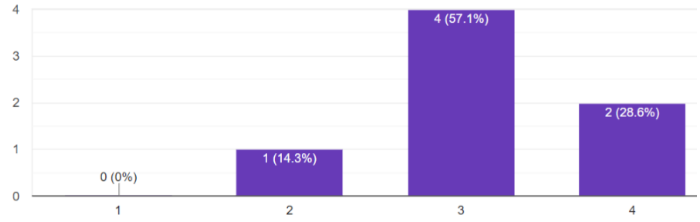
Is there visual feedback in menus or dialog boxes about which choices are selectable?

7 responses



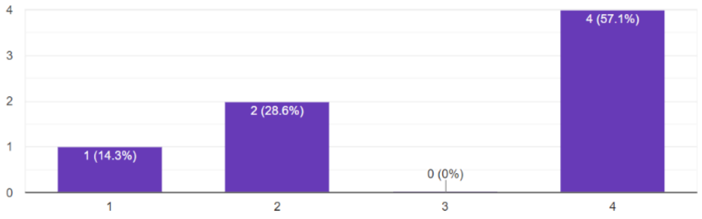
Is the menu-naming terminology consistent with the user's task domain?

7 responses



Does the system provide visibility: that is, by looking, can the user tell the state of the system and the alternatives for action?

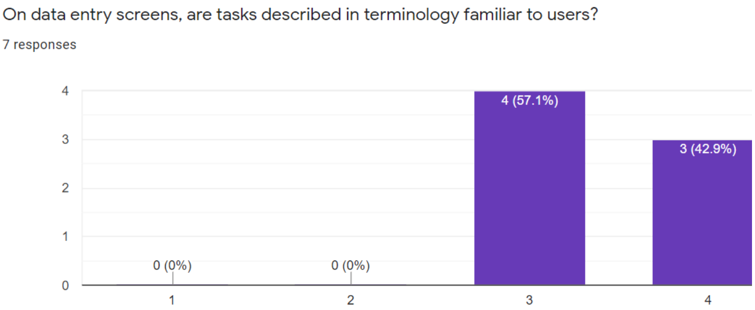
7 responses



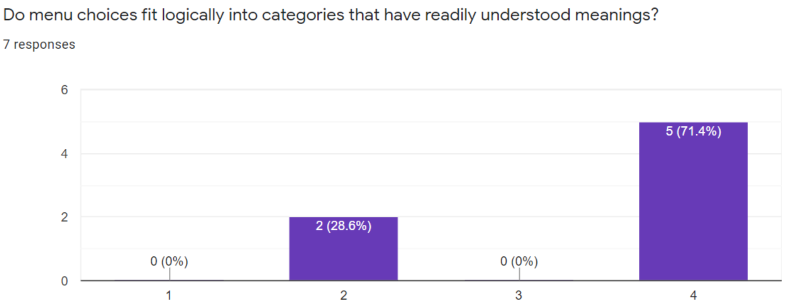
From looking at the feedback by using the titles and a menu bar notify users about the contents however the last poll from the above received low rating this is because when uploading an image the loading icon was not present

Match Between System and Real World

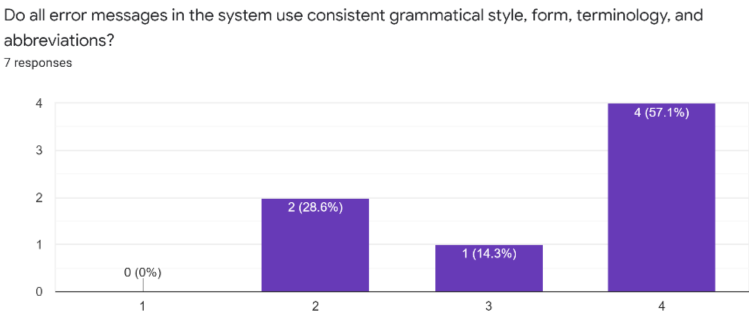
In order to ensure that the match is made between the system and the real world all the textual description presented to the users are in plain English without any technical terminologies.


On data entry screens, are tasks described in terminology familiar to users?

7 responses


Do menu choices fit logically into categories that have readily understood meanings?

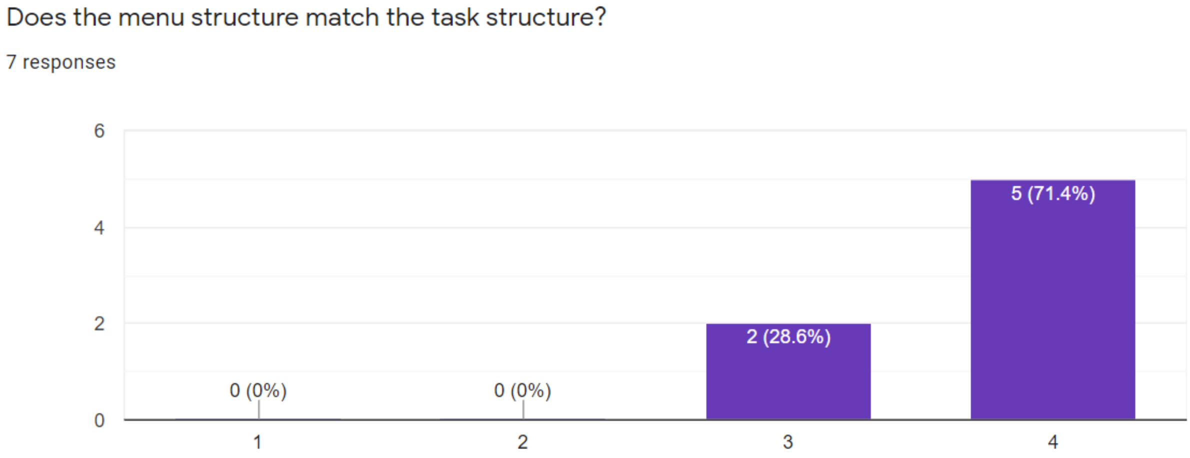
7 responses


Do all error messages in the system use consistent grammatical style, form, terminology, and abbreviations?

7 responses

Consistency and Standards

The styling of the application is kept consistent to make users familiar with the structure and layout of the application.


Does the menu structure match the task structure?

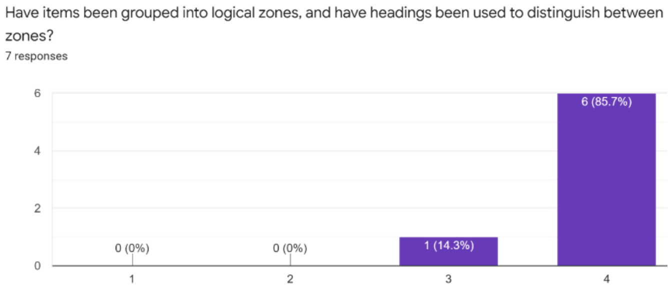
7 responses

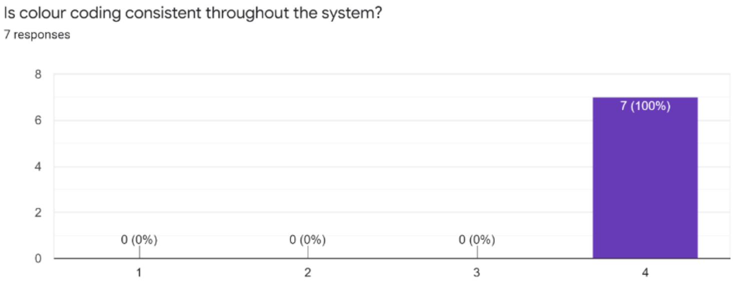
Help Users Recognise, Diagnose and Recover from Errors

It is crucial for a good system to have a good error handling to guide users to overcome the errors made by users. Error handling has been placed throughout the system such as uploading image only in jpg extension. Catch the empty input file from both upload field and search field. The errors made by users are displayed on screen.

Recognition Rather than Recall

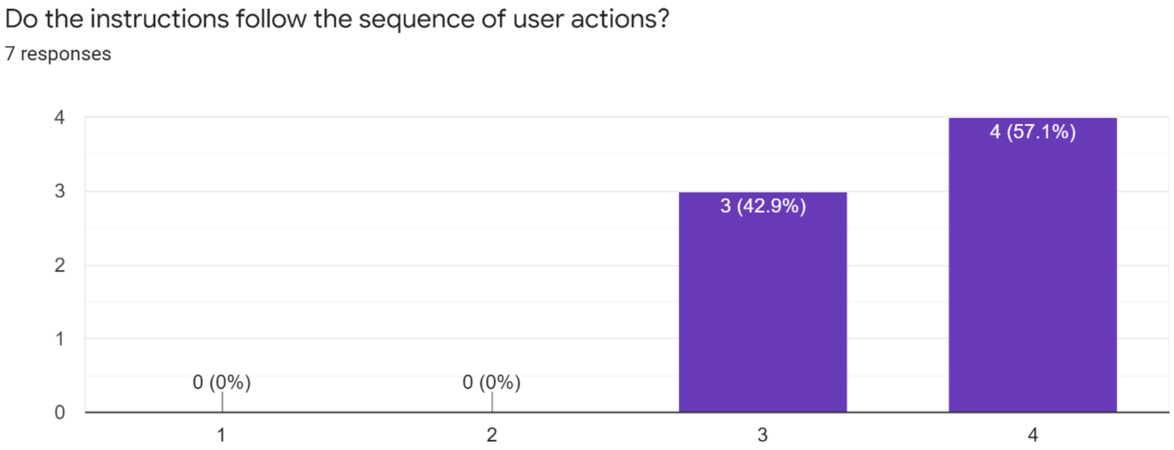
The use of logical zone and colour coding promote the recognition rather than recall principle. The system scored very high in this area due to the use of zones and colour code in the system.


Have items been grouped into logical zones, and have headings been used to distinguish between zones?
7 responses


Is colour coding consistent throughout the system?
7 responses

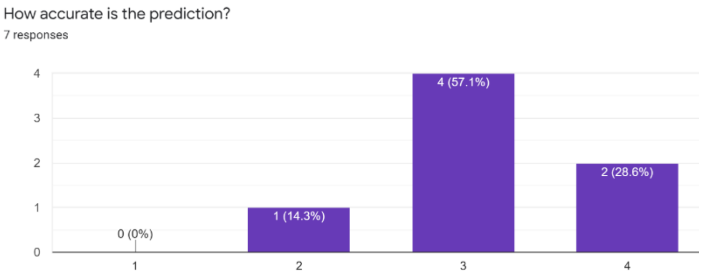
Help and Documentation

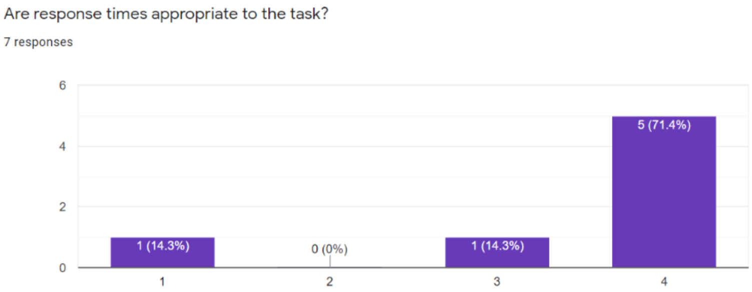
The only section that needed an instruction in order for the user to follow the correct sequence steps of process is upload an image for classification. An instruction is presented clearly on the upload screen. As a result, this section hit a high score.


Do the instructions follow the sequence of user actions?
7 responses

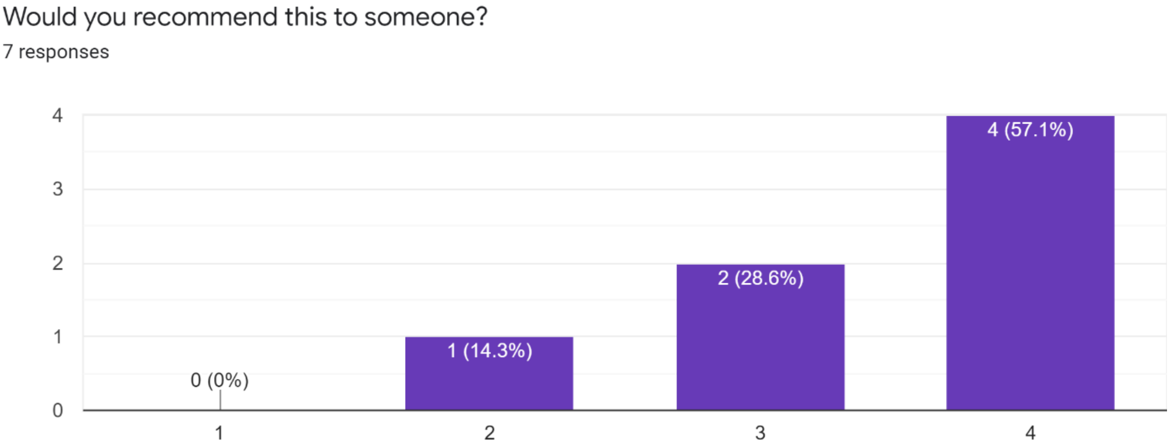
Another aspect that was asked in the questionnaire was the opinion of the overall of the system. The accuracy of the classification and its response time were questioned.

The vote varies for the prediction because a flower maybe share some similarity to other flowers in the dataset and others might have a unique appearance and colour so they do not take a lot of training time.

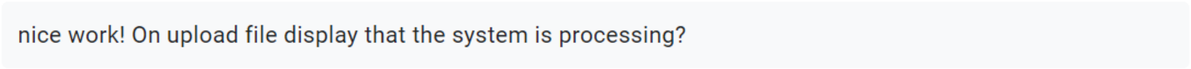




This application received a good feedback and the majority of people would recommend this application to other people.



In conclusion as stated in the visibility system status principle a loading icon needed to be implemented on the page to alert the user that the system is running still because the majority of people who participated were expert in technology, this feature was highly recommended in the comment section since the novice would not know that the process of classification was running.


nice work! On upload file display that the system is processing?

5.4. Conclusions

This chapter reviewed the different testing methods that will be followed and the software test plan. The manual testing was carried out throughout the cycle while developing the application.

The classification evaluation carried out using train/test split validation approach to ensure overfitting and underfitting were not introduced to the classifier. The classification report presented some interesting information about the classifier in which it gives a detailed evaluation of each class of the dataset such as precision, recall and F1 score. Some classes obtained high F1 score than others due to their unique colour and appearance that other classes do not have. Usability testing was carried out to evaluate the usability of this system following the Nielsen Heuristics principles. The majority of the components of the application followed the guidelines however one feature that people requested to be added in the application is the loading icon while the classification process is taking place.

6. Conclusions and Future Work

6.1. Introduction

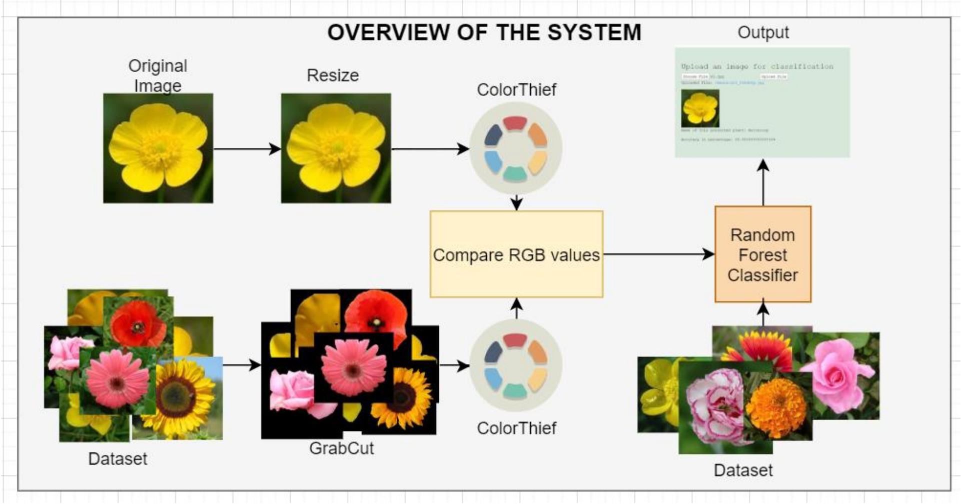
This chapter includes the conclusion of each chapter in short paragraphs. The future work that author planned to do are listed in this chapter.

6.2. Conclusions

From literature review section, Flowerify had learned from the listed apps as to what features are successful and what ones are not to implement in the system. Most people usually look up flowers guidebooks or use the relevant websites on the internet to browse the information using keywords. The potential techniques from the researched articles may be used throughout this project. The author noted different features from the examined solutions that were useful features for the project. The main features that were identified were plant identification and searchable database. Another aspect that stuck in the authors mind was that from examining all the existing solutions listed above the majority of the application is mobile application and minority of the solutions were web-based application. The author approached web-based application using bootstrap for its interface to change to mobile layout.

The design of the system was outlined in the experiment design chapter where usage of paper prototype and fidelity prototype were seen. The overview of the technical architecture was outlined where the backend layer consists of MYSQL as a database and file storage for saving uploaded image of this project. The logic layer is where all the computations happen, in this layer classification process, image processing and colour dominant analyser take place, the classifier was identified in this section which was random forest who achieved one of the lowest accuracies. An extra feature of image quality assessment was identified for capturing the low quality input images. The layout of the interface is decided in this section where Django is going to be used as a web framework for this project. The interface will be based in HTML, CSS and bootstrap via Django.

This project presented a working system of classifying the flower from input image. The author experimented different techniques of image modification to increase the accuracy of the classifier and came to a conclusion to use image segmentation to extract a feature from the background. Dominant RGB value was extracted from each image of the dataset as well as the input image. These values were then used to identify dominant colour of each flower type. The ColorThief module was used to retrieve the dominant RGB values of an image. Random Forest algorithm was used as the classification process. With 10 types of flowers, the system was able to classify flowers with 73.33% accuracy.



Many factors were identified that had an impact on the accuracy of the classifier. The classifier's accuracy started with 42% and increased to 73% as the dataset was sorted out with the image distribution equally to each class which leaves each class with 100 images. Previously when the accuracy was 42% there was an uneven number of images in each class range from 50 to 140.

The approach was to sort the dataset so there were evenly distributed images which resulted in no bias. By resizing the images in the dataset, the accuracy increased to 63%.

Dominant RGB values acted as a filter to filter possible outcome by comparing the dominant colour of input image to dominant colour of each class of the dataset. However classifier still having difficulty with similar colour and structure of the petals, this can be seen through the classification report where the F1 score of Barberton Daisies and roses are low because of their multi-colour especially Barberton daisy where its petal is very similar to sunflower and sometimes it confuses it with common dandelion.

6.3. Future Work

The first area where author plans to continue the development of this project is add in a loading icon while the classification process takes place. As suggested in the usability evaluation this additional feature would inform the user what is taking place. Another development that author would like to work on is the search bar which could be used a search criteria option when searching the name of flower that may not be in the database and providing some suggestions of similar flower that would present to the user instead.

An experiment that the author would like to investigate is to switch dataset from original dataset to GrabCut dataset where all the images have been segmented. In addition to that having input image being segmented using GrabCut to match the database might increase the accuracy of the algorithm.

Bibliography

1. 151 Types of Flowers + Images and Growing Tips | Care Guide [Internet]. ProFlowers Blog. 2017 [cited 2019 Dec 5]. Available from: https://www.proflowers.com/blog/types-of-flowers

2. Almogdady H, Manaseer DS, Hiary DH. A Flower Recognition System Based On Image Processing And Neural Networks. 2018;7(11):8.

3. (1) (PDF) Flower Identification System by Image Processing [Internet]. ResearchGate. [cited 2019 Nov 13]. Available from: https://www.researchgate.net/publication/303014817_Flower_Identification_System_by_Image_Processing

4. PlantNet Plant Identification - Apps on Google Play [Internet]. [cited 2019 Dec 5]. Available from: https://play.google.com/store/apps/details?id=org.plantnet&hl=en

5. PictureThis - Plant identifier [Internet]. [cited 2019 Nov 23]. Available from: https://www.picturethisai.com/web/config/change_language/en

6. PlantSnap - Plant Identifier App, #1 Mobile App for Plant Identification [Internet]. Plantsnap - Identify Plants, Trees, Mushrooms With An App. [cited 2019 Nov 23]. Available from: https://www.plantsnap.com/

7. Home | Leafsnap: An Electronic Field Guide [Internet]. [cited 2019 Dec 5]. Available from: http://leafsnap.com/

8. Comparing Python to Other Languages [Internet]. Python.org. [cited 2020 Apr 7]. Available from: https://www.python.org/doc/essays/comparisons/

9. Best programming languages for image recognition | SaM Solutions [Internet]. 2019 [cited 2020 Apr 7]. Available from: https://www.sam-solutions.com:443/blog/image-recognition-which-programming-language-to-choose/

10. MATLAB - MathWorks [Internet]. [cited 2019 Dec 5]. Available from: https://uk.mathworks.com/products/matlab.html

11. What Is Deep Learning? | How It Works, Techniques & Applications [Internet]. [cited 2020 Apr 7]. Available from: https://www.mathworks.com/discovery/deep-learning.html

12. About [Internet]. [cited 2019 Dec 5]. Available from: https://opencv.org/about/

13. About [Internet]. [cited 2020 Apr 7]. Available from: https://opencv.org/about/

14. OpenCV Review: Pricing, Pros, Cons & Features [Internet]. CompareCamp.com. 2019 [cited 2020 Apr 7]. Available from: http://comparecamp.com/opencv-review-pricing-pros-cons-features/

15. Why TensorFlow [Internet]. TensorFlow. [cited 2019 Dec 5]. Available from: https://www.tensorflow.org/about

16. Databases | Django documentation | Django [Internet]. [cited 2019 Dec 5]. Available from: https://docs.djangoproject.com/en/2.2/ref/databases/#mysql-notes

17. Django introduction [Internet]. MDN Web Docs. [cited 2019 Dec 5]. Available from: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction

18. When To Use Django (And When Not To) - Crowdbotics - Medium [Internet]. [cited 2019 Dec 5]. Available from: https://medium.com/crowdbotics/when-to-use-django-and-when-not-to-9f62f55f693b

19. Flask [Internet]. [cited 2019 Dec 5]. Available from: https://www.fullstackpython.com/flask.html

20. SQLite Advantages and Disadvantages - javatpoint [Internet]. [cited 2019 Dec 5]. Available from: https://www.javatpoint.com/sqlite-advantages-and-disadvantages

21. Five Advantages & Disadvantages Of MySQL [Internet]. [cited 2019 Dec 5]. Available from: https://www.datarealm.com/blog/five-advantages-disadvantages-of-mysql/

22. The PythonAnywhere help pages | PythonAnywhere help [Internet]. [cited 2019 Dec 5]. Available from: https://help.pythonanywhere.com/pages/

23. AWS vs. DigitalOcean: Which Cloud Server is Better [Internet]. [cited 2020 Mar 15]. Available from: https://hackernoon.com/aws-vs-digitalocean-which-cloud-server-is-better-1386499a6664

24. Visual Geometry Group - University of Oxford [Internet]. [cited 2019 Dec 5]. Available from: https://www.robots.ox.ac.uk/~vgg/data/flowers/

25. Flowers Recognition [Internet]. [cited 2019 Dec 5]. Available from: https://kaggle.com/alxmamaev/flowers-recognition

26. MIAP CTBsH. An introduction to Convolutional Neural Networks [Internet]. Medium. 2019 [cited 2019 Dec 9]. Available from: https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7

27. Decision tree learning pros and cons - Machine Learning with Swift [Book] [Internet]. [cited 2019 Dec 9]. Available from: https://www.oreilly.com/library/view/machine-learning-with/9781787121515/697c4c5f-1109-4058-8938-d01482389ce3.xhtml

28. K-Nearest Neighbors Algorithm in Python and Scikit-Learn [Internet]. Stack Abuse. 2018 [cited 2019 Dec 9]. Available from: https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/

29. What is a Support Vector Machine, and Why Would I Use it? [Internet]. KDnuggets. [cited 2019 Dec 9]. Available from: https://www.kdnuggets.com/what-is-a-support-vector-machine-and-why-would-i-use-it.html/

30. Histograms in OpenCV — OpenCV-Python Tutorials 1 documentation [Internet]. [cited 2020 Apr 5]. Available from: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_table_of_contents_histograms/py_table_of_contents_histograms.html#table-of-content-histograms

31. Feng S. colorthief: A module for grabbing the color palette from an image. [Internet]. [cited 2020 Apr 5]. Available from: https://github.com/fengsp/color-thief-py

32. Interactive Foreground Extraction using GrabCut Algorithm — OpenCV-Python Tutorials 1 documentation [Internet]. [cited 2020 Apr 5]. Available from: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_grabcut/py_grabcut.html#grabcut

33. SDLC - Waterfall Model - Tutorialspoint [Internet]. [cited 2019 Nov 23]. Available from: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

34. Andrew Powell-Morse. Waterfall Model: What Is It and When Should You Use It? [Internet]. Airbrake Blog. 2016 [cited 2019 Dec 5]. Available from: https://airbrake.io/blog/sdlc/waterfall-model

35. A Beginner's Guide To The Agile Method & Scrums [Internet]. Linchpin SEO. 2019 [cited 2019 Dec 5]. Available from: https://linchpinseo.com/the-agile-method/

36. What is AGILE? - What is SCRUM? - Agile FAQ's [Internet]. Cprime. [cited 2019 Dec 5]. Available from: https://www.cprime.com/resources/what-is-agile-what-is-scrum/

37. 3-Tier Architecture: A Complete Overview [Internet]. JReport. [cited 2019 Dec 5]. Available from: https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/

38. Experience WL in R-BU. 10 Heuristics for User Interface Design: Article by Jakob Nielsen [Internet]. Nielsen Norman Group. [cited 2020 Apr 1]. Available from: https://www.nngroup.com/articles/ten-usability-heuristics/

39. Why Linear Regression is not suitable for Classification [Internet]. jinglescode.github.io. 2019 [cited 2020 Mar 22]. Available from: https://jinglescode.github.io/datascience/2019/05/07/why-linear-regression-is-not-suitable-for-classification/

40. Linear Regression | Statistically Significant Consulting [Internet]. [cited 2020 Mar 22]. Available from: https://www.statisticallysignificantconsulting.com/RegressionAnalysis.htm

41. Yiu T. Understanding Random Forest [Internet]. Medium. 2019 [cited 2020 Apr 1]. Available from: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

42. Ocampo R. Automatic Image Quality Assessment in Python [Internet]. Medium. 2019 [cited 2020 Apr 2]. Available from: https://towardsdatascience.com/automatic-image-quality-assessment-in-python-391a6be52c11

43. File storage API | Django documentation | Django [Internet]. [cited 2020 Apr 1]. Available from: https://docs.djangoproject.com/en/3.0/ref/files/storage/

44. What Exactly Is GitHub Anyway? [Internet]. TechCrunch. [cited 2019 Dec 5]. Available from: http://social.techcrunch.com/2012/07/14/what-exactly-is-github-anyway/

45. XAMPP Tutorial: How to use XAMPP to set up WordPress on localhost [Internet]. Undsgn™. 2018 [cited 2020 Apr 1]. Available from: https://undsgn.com/xampp-tutorial/

46. diagrams.net - G Suite Marketplace [Internet]. [cited 2020 Apr 10]. Available from: https://gsuite.google.com/marketplace/app/diagramsnet/671128082532

47. What is Django? · Django Girls Tutorial [Internet]. [cited 2020 Mar 22]. Available from: https://tutorial.djangogirls.org/en/django/

48. Ashish Kumar. Image pre processing [Internet]. 04:14:16 UTC [cited 2019 Dec 9]. Available from: https://www.slideshare.net/ASHI14march/image-pre-processing

49. Databases | Django documentation | Django [Internet]. [cited 2020 Mar 22]. Available from: https://docs.djangoproject.com/en/3.0/ref/databases/#mysql-notes

50. File storage API | Django documentation | Django [Internet]. [cited 2020 Apr 10]. Available from: https://docs.djangoproject.com/en/3.0/ref/files/storage/

51. Django Tutorial Part 10: Testing a Django web application [Internet]. MDN Web Docs. [cited 2020 Mar 30]. Available from: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Testing

52. Testing in Django | Django documentation | Django [Internet]. [cited 2020 Mar 30]. Available from: https://docs.djangoproject.com/en/3.0/topics/testing/

53. Manual Testing Tutorial for Beginners: Concepts, Types, Tool [Internet]. [cited 2020 Apr 8]. Available from: https://www.guru99.com/manual-testing.html

54. Bronshtein A. Train/Test Split and Cross Validation in Python [Internet]. Medium. 2020 [cited 2020 Apr 2]. Available from: https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

55. Krishnan M. Understanding the Classification report through sklearn [Internet]. Muthukrishnan. 2018 [cited 2020 Apr 10]. Available from: https://muthu.co/understanding-the-classification-report-in-sklearn/

56. Brownlee J. How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification [Internet]. Machine Learning Mastery. 2020 [cited 2020 Apr 10]. Available from: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/

57. Black Box Testing [Internet]. Software Testing Fundamentals. 2017 [cited 2019 Nov 26]. Available from: http://softwaretestingfundamentals.com/black-box-testing/

58. What is Grey Box Testing? [Internet]. TestLodge Blog. 2018 [cited 2019 Dec 5]. Available from: https://blog.testlodge.com/what-is-grey-box-testing/