Types of Neural Networks in Python

GMIT

INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Naqi Ahmad

B.Sc. (Hons) in Software Development

Advised by Gerard Harrison

Contents

List of Figures

Acknowledgements

I would like to acknowledge my supervisor Gerard Harrison for all the support and GMIT staff for their help throughout the last four years.

Abstract

Technology of all kinds play a significant role in society and it can be considered an important, almost essential part of modern-day life. The improvements that have been made in the last few years are phenomenal. In this project, I will build three different types of neural networks using the high-level programming language Python. Neural Networks have gained popularity in recent years even though the technology has been in development since the mid nineteen fifties. Platforms such as TensorFlow and Keras are end-to-end source platforms that allow for easier machine learning and enable many developers to divulge further into neural network technology. A decade ago, we thought that getting a computer to tell the distinction between one object and another would be almost impossible. Now we have trained neural networks that can tell the difference with a precision of greater than eighty percent accuracy. This project is a research project which would allow me to have a better comprehension of the different types of neural networks and the overall goal of this project is to compare several different types of neural networks and to highlight the similarities and differences between them. I will be assessing the tasks of planning and developing, as well as design and implementation of all the components that make up my project. This project supports further expansion and insertion of new components such as more varieties of neural networks. The proposed solution will be comprised of a command line application which will give the user the ability to test and train the neural network. The project will also implement the CRUD functionalities which allows the user to login and save their details.

Chapter one

Introduction

At the beginning of my final year, our class were given the opportunity to work as a team or individually to develop a project over the course of the year across two semesters. In my past three years of education I have been given the opportunity to develop both solo and group projects. I feel that for my main project in fourth year I wanted to challenge myself individually and showcase my talent while also challenging myself to learn new skills. Initially, my first idea for the project had been to create a Convolutional Neural Network and use the Django Framework to make a web application. However, due to time constraints and this being an individual project I decided to only focus my work on neural networks. Even though the initial idea was very interesting it became very clear early on that I did not have enough content covered. So, I scrapped the original concept and decided to build three different types of neural networks. The intention of this project is solely research based with the purpose of developing a better understanding on the differences between each Neural Network.

Developing my Design

This project is based on machine learning using Neural Networks and written in the Python programming language. The project uses Keras which runs on top of TensorFlow. Keras was created with an attention on enabling fast experimentation. The idea for undertaking this project stems from me wanting to figure out how uniquely different neural networks function. At the beginning of the project it was challenging to select what neural networks I would use for the comparisons. There are various types of neural networks that I could have utilized in my project but after researching my options I chose to work with the three most widely used neural networks. During my research the different types of neural networks I came across were Recursive neural network (RNN), Convolutional neural network (CNN), Multilayer perceptron, Recurrent neural network (RNN) and k-nearest Neighbours (KNN). The three Neural Networks I chose to work with were Multilayer perceptron, Convolutional neural network (CNN) and k-nearest Neighbours (KNN). The figures below show how each of the Neural Networks work. I will cover about Neural Networks more in dept later in my dissertation.

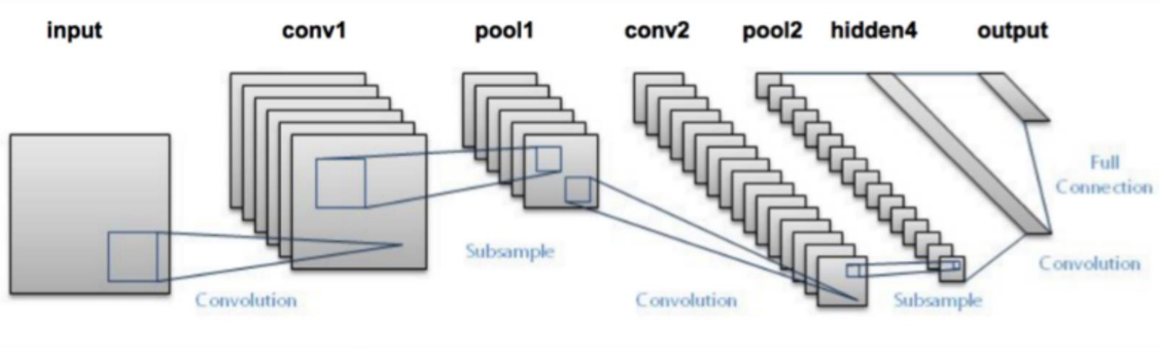


Figure 1.1: Convolutional neural network (CNN)

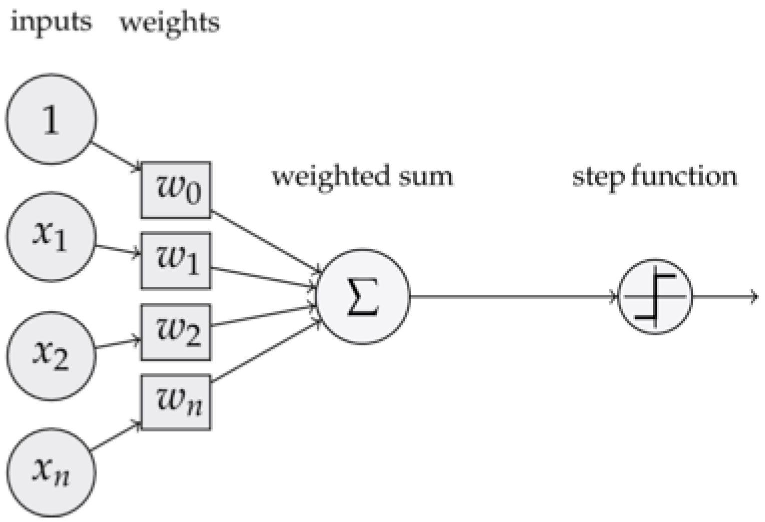


Figure 1.2: Multilayer Perceptron

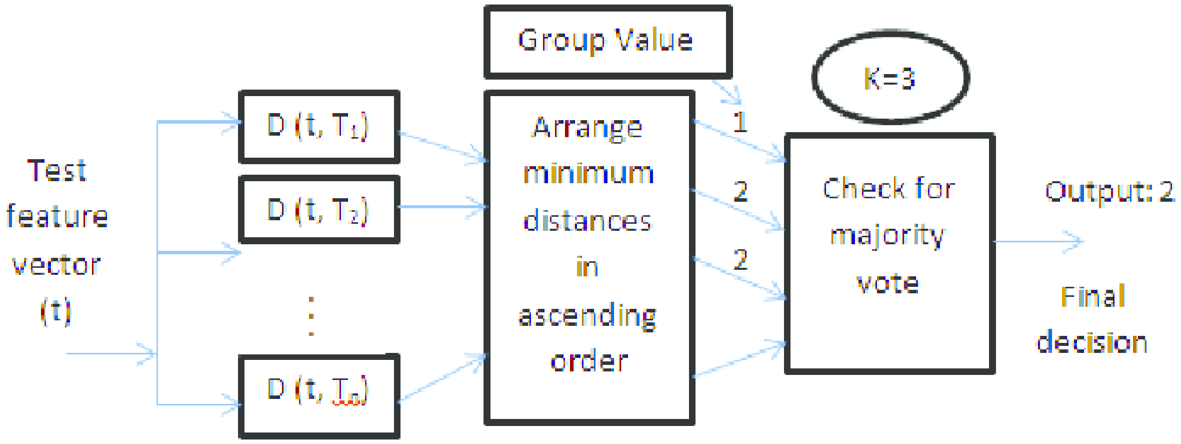


Figure 1.3: k-nearest Neighbours (KNN)

Final Design

Neural Networks in Python (the title of my project) is a project involving APIs, Python, MongoDB and cloud technology. The Implementation of the project is a command line interactive project that trains and tests the three types of neural networks with the support of a log in service. The log in service authenticates a user and allows to be able to train and test the Neural Networks. You can create a user which is saved in MongoDB in cloud. At the point when the client is logged in you can see the three diverse choices for testing the distinctive sorts of neural networks. Each option leads into a different menu. For example, if you select the multilayer perceptron then this would lead into another menu. This menu allows you to test the multilayer perceptron with your own unique set of values.

Furthermore, the context of this project revolves around the user being able to test the different types of Neural Networks and perceive how each type of Neural Network produces its own outcome.

Objectives

The undertaking of this project will require several objectives to be attained in order to provide a solution that works and is high performing.

- The first objective will be to build the Convolutional neural network (CNN). The dataset I will use is the MNIST Fashion Dataset. At the beginning of my project my first objective is to train and test the dataset. This Neural Network will be able to recognise clothes from an image.

- After building the Convolutional neural network (CNN) I will start the development of the Multilayer Perceptron. The dataset I will use to train my Multilayer Perceptron is the Iris Dataset. The user will be able to enter unique values to test the trained Neural Network.

- Finally, I will have to start the k-nearest Neighbours neural network. This will be for the colour detection in an image. I will have to make my own dataset of colours and train the Neural Network.

- MongoDB database will be used to store the images and user data i.e. their username and password. The MongoDB database must be setup on cloud so its easily accessed from anywhere.

- To store the passwords, I will be using the Hashing Algorithm SHA256. This allows for safe storing of passwords. The hashed string of the password will be stored in the MongoDB. When the user enters the password the it will be hashed and compared with the string from the database.

- Lastly, I will make the command line application more interactive by adding more GUI.

Project Links

Link to Repository

- https://github.com/nakster/FinalYearProject

Overview

With this project being a key aspect of my final year, I acknowledged that I was required to challenge myself with a project that would be intricate and allow me to learn at each stage of its development. The layout of my dissertation is structured into chapters, each one containing different aspects of the project. The following is a breakdown of what is examined in every section.

Methodology

This section describes the steps that were taken in order to secure an effectively successful project. In this chapter I will diagram the distinctive Methodologies that were considered in the development and implementation of my project's final design. It will mainly discuss the Agile Methodology, version control and testing. I will also cover why I chose to use python and different technologies and the problems I encountered during the development of the project.

Technology Review

This section will be about the technologies that I came across during my research. I will plot and evaluate the different technologies such as MongoDB and SHA256. I will talk about how to set up and use the technologies.

System Design

This section of the dissertation focusses on how each component of the project is developed. I will also review each component with detailed discussion on how each part contributes to the whole system of the project. Alongside the detailed review, I will also provide code snippets for my project as well as working screenshots.

System Evaluation

This section of the dissertation will outline the system performance, robustness and scalability. I will also discuss the advantages and limitations I encountered when creating this project.

Conclusion

This section I will conclude the goals and objectives that were set. I will discuss the outcome of the project and the issues that were experienced. I will likewise talk about how I came up with the solutions for any of the problems encountered.

Chapter Two

Methodology

In this section I discuss the methodologies I used in my project to prevent contingencies. Before a project is started, it is very important to plan and control the development process of a piece of software. The reason for this is it refrains us from rushing the start of the coding phase in the project which is a tempting predicament. A poorly planned outline of development will lead to possible contingencies being used such as deadlines or milestones not being met on time, scaling down the overall design mid development and ultimately final goals of the project not being achieved.

Planning Phase

During the initial planning phase I set out the scope of my project and decided to use an Agile [1] style approach to the research, design and implementation of my project. I investigated the different types of methodologies such as the waterfall model but in the end decided the best approach was to use an Agile methodology. The reason I chose Agile over Waterfall is due to the fact the waterfall model lacks flexibility. Whereas Agile offers better flexibility, incremental delivery and continual development of the software we are developing. I assigned myself the responsibilities to completing a final year project, the most important being carried out first. Before commencing any development of the project, I carried out some research about technologies that I was going to use. After evaluation the different technologies, I chose to use the programming language Python, MongoDB and Visual Studio Code. The reason I selected python is because of python's easy to learn syntax. Python looks like a human readable language that is easy to learn. During the planning phase I drew up the architecture of my project and presented it to my supervisor, Gerard Harrison. After help from my supervisor I decided to make a command line application and if I have the time, I will make it into a Django application.

Requirements Analysis

When designing the final year project determining the requirements is a key factor. These steps allow for the requirements to be analysed, understood and broken down into stories. These stories are then used to resolve the priority of each task. Therefore, at the start of the research I did a project requirements analysis based on what I knew. I had to use my own personal experience to gather a set of requirements that would help me design the implementation of the project.

These requirements include the following.

- First to have a software application that works and returns a correct outcome to the input entered.
- Secondly, having a complete login registration system implemented. I will need to use a database that can store the user information effectively.
- Thirdly I need to allow the user to test their own set of values, this could be their own input or from online URL.
- Each of the Neural networks must have a menu that tests and trains each type of Neural Network.
- Lastly if I have enough time, I will use the Django framework to make my application a web app.

Meetings

On accounts of me being a single person developing the final year project I did not have group meetings. As I started my project, I met up with my supervisor during college time to keep in check that I was developing a software product with the quality expected from a fourth-year student. I presented my supervisor with weekly progress and achievements and asked for guidance for the future development of my project. During the meeting the suggestion given by my supervisor were implemented during the successive week. The meeting normally took place every Tuesday or Thursday of each week.

Development

I divided the attainable tasks into sprints. So, when I started the development progress the issues that become apparent were either resolved or taken in account in the next iteration. Continuous Integration is achieved in this project by completing the sprints and testing if they work and continuously adding more.

Testing and Validation

Testing is an important part of a software project and during the life cycle it provides benefits to the overall quality and reliability of the code in a project. I used the terminal command line interface to perform testing and verification of the project during the development. I also performed Black Box Testing; this is where functionality of a piece of software is tested without the need to check the internal workings of the project [2].

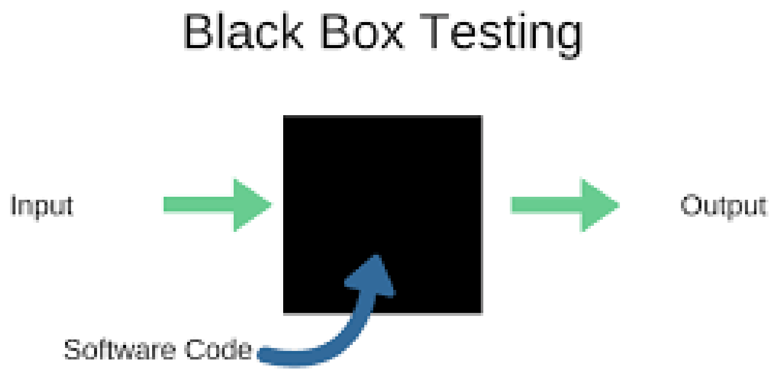


Figure 2.1: Black Box Testing

An example of this would be when the user enters the username and password to login into the application. The Black Box Testing checks if the user is logged in when the correct input is entered. This type of testing is very useful to check if the piece of software is performing and returning what is expected.

Problems

During the development phase I encountered only a handful number of problems or minor bugs. Most of my problems were from taking in the input from the user and making it compatible to test against the trained Neural Networks. These problems were dealt with during the development by means of research and getting to the root of the problem.

Agile

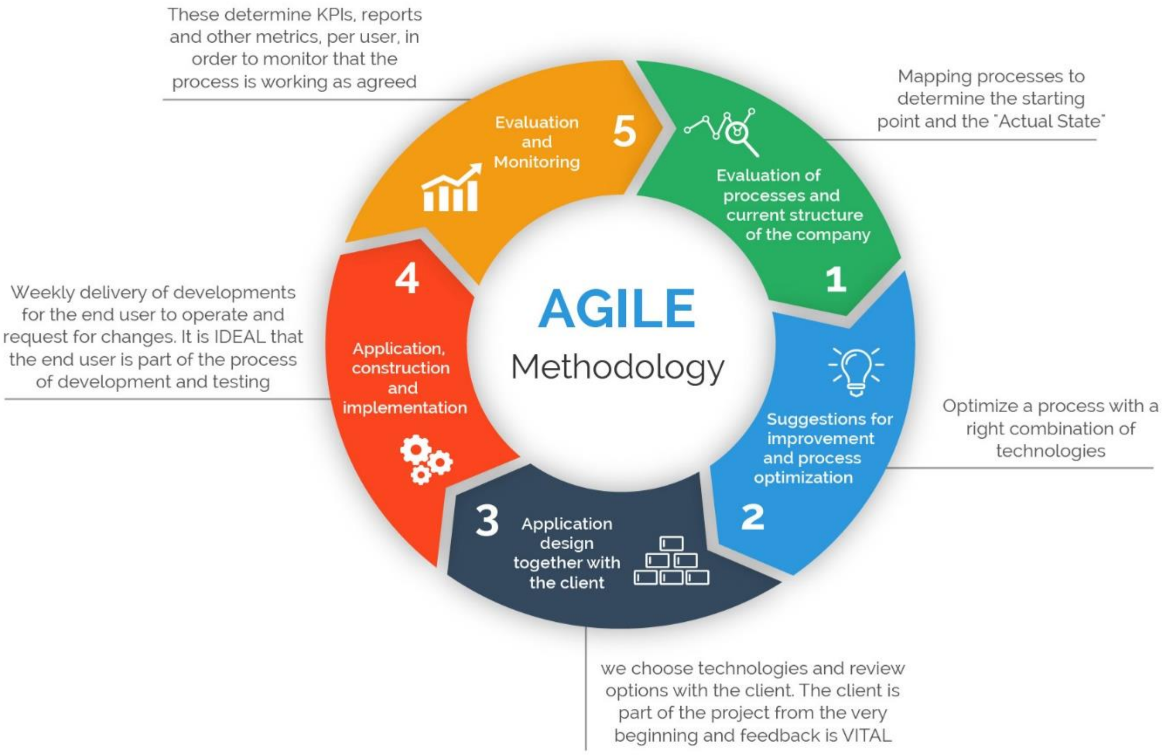
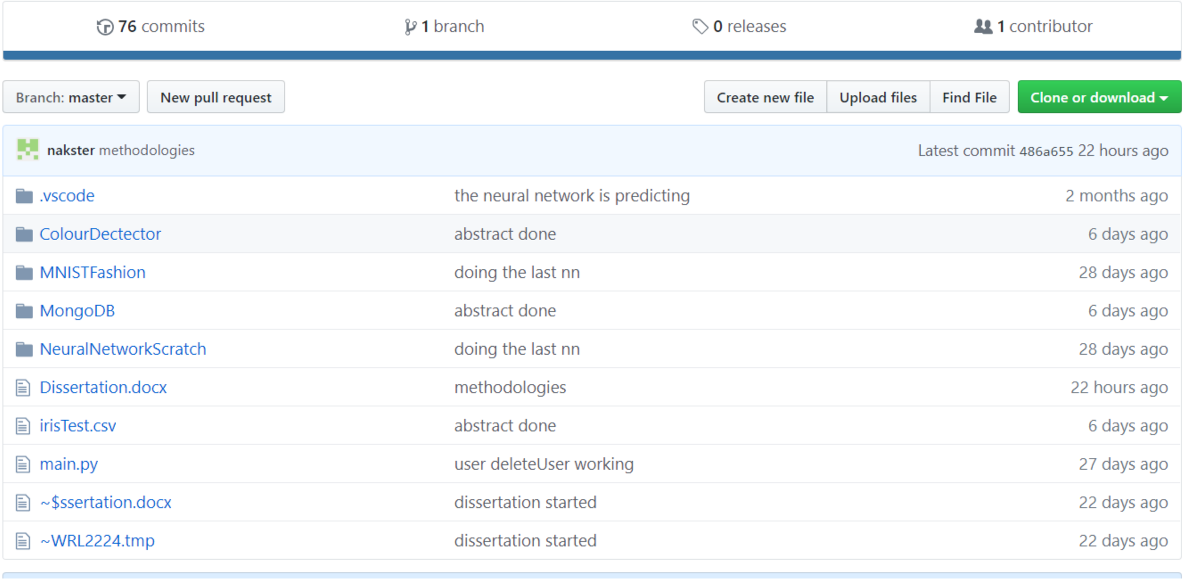

These determine KPIs, reports and other metrics, per user, in order to monitor that the process is working as agreed
Mapping processes to determine the starting point and the "Actual State"
5
Evaluation and Monitoring
Evaluation of processes and current structure of the company
1
AGILE
Methodology
4
Application, construction and implementation
Weekly delivery of developments for the end user to operate and request for changes. It is IDEAL that the end user is part of the process of development and testing
Suggestions for improvement and process optimization
2
Optimize a process with a right combination of technologies
3
Application design together with the client
we choose technologies and review options with the client. The client is part of the project from the very beginning and feedback is VITAL

Figure 2.2: Agile

Agile is a development methodology which acknowledges that requirements can evolve and change over time.

Version Control Manager

When I started the coding part of my project, I needed a sort of a project version control manager. GitHub is a web-based hosting service that allows for version control with Git. For this reason, I chose to use GitHub [3]. I have been using GitHub for my personal and college work, so it made perfect sense to use it for my final year project. Some of the features GitHub provides include wikis, bug tracking, feature request and task management. GitHub allows you to go back to your previous commits and see the changes that were made. GitHub as a development tool encourages the committing of small pieces of code rather than everything all at once. This allowed me to keep pushing the changes up and if I were to develop a bug all I had to do was to see the changes that I made and roll back. To take full advantage of the version control tool I personally tried to incrementally commit small pieces of code. One of the features you can use when creating a GitHub project is creating a README file to coincide with the project. This is a document that is used to write the stages of development and information about the software with running instructions. GitHub was the perfect tool to use for my project as it allowed for version control along with other main features.



nakster methodologies — Latest commit 486a655 22 hours ago

.vscode	the neural network is predicting	2 months ago
ColourDectector	abstract done	6 days ago
MNISTFashion	doing the last nn	28 days ago
MongoDB	abstract done	6 days ago
NeuralNetworkScratch	doing the last nn	28 days ago
Dissertation.docx	methodologies	22 hours ago
irisTest.csv	abstract done	6 days ago
main.py	user deleteUser working	27 days ago
~$ssertation.docx	dissertation started	22 days ago
~WRL2224.tmp	dissertation started	22 days ago

Figure 2.3: GitHub

Project Management

Project Management [4] is an important almost essential element in any software development project. Managing a project allows for shaping of the structure, as poor management can impact the quality and delivery time of the project. To build a robust application I must use my knowledge of project management to carefully develop an efficient plan. As poor management can have a detrimental impact on a project, I put an emphasis on having a clear plan with a strong strategy to manage the lifecycle of my project. A planning phase before I begin any coding allowed me to decide on what features I would carry out first. This made things easier as an initial planned design was set for me to achieve a working piece of software.

Chapter Three

Technology Review

This section of the dissertation involves me discussing the various types of technologies I used to develop my application.

Python

Figure 3.1: Python

The main coding language I am using for my project is Python [5]. Python language is an interpreted, high level and object-oriented language. Python has been around for the last 25 years and is still rising in popularity due to the simplicity of the code which makes for easier understanding or comprehension of the language. Python is easy to learn, easy to read and still is a very powerful language. Python is managed by the non-profit Python Software Foundation and is an open source programming language that has volunteers who are constantly trying to improve and develop the language going forward. This is most likely one of the main reasons that Python has remained fresh and current with the newest trends. Python features many libraries for almost any application, and I am using some of them in my project. Python automatically compiles code into byte code which is then executed. Python can be used to carry out extensive tasks because python provides the needed speed.

Important Features of Python

Python has a lot of the same features as most programming languages i.e. if, if-else, while and for loops. The multiple levels of organisational structure include packages, classes and functions. These are all included inside Python's standard library. Python possess the ability to compile code to byte code without the need for another step.

One unique feature of python is the indentation of the code. When coding in Python we indent one level to indicate the beginning of a block. So, for example

```
if b>a:
  print("b is greater than a")
```

If the code wasn't properly indented it would not work. If we were to add another if statement, we would have to indent one level again to make the code work.

```
# this works because the first if statement has one indentation
# Also, the second if statement also has one indentation
# if there were no indentations the code would not work

if b>a:
  print("b is greater than a")
  # second if statement
  if c>d:
    print("c is greater than d")
```

What can Python be used for?

Python is very flexible and can be used for various kinds of projects. In my project I am using python for machine learning to train and test Neural Networks. Other uses for Python can be in the fields of computer vision, data analysis, web scraping and more.

Prerequisites

To be able to run a Python programme we must download and install Python. There are a couple of ways to install Python, but I decided to use Anaconda.

Anaconda

Figure 3.2: Anaconda

Anaconda [6] is a free open-source distribution of Python. The reason I chose Anaconda is that it simplifies the installation process. Anaconda comes with more than 1400 preinstalled packages. Whereas if we were to do this with a traditional Python installer it would take a lot of time for installation and configuring all those packages. Another reason that I chose to use Anaconda is that it comes with Jupyter Notebooks.

To install Anaconda

- I went to Anaconda's website [6] and then to their downloads section.
- In the downloads section, I decided to download the Python 3.7 version.
- This downloaded the Anaconda 64bit version.
- After the download completion, I ran the executable file and followed the setup guide.
- After this, I was able to use the Python programming language.

There are other perhaps less straightforward ways to install python, for example, it can be installed from the Python Software Foundation website at python.org. This requires you to download the installer that is appropriate for your system. After running the installer, you must follow the setup guide. The guide has an option to add python to the path, what this does is it adds Python to the environment.

Python Packages

In this heading I will be talking about the python packages I have used for this project.

Numpy

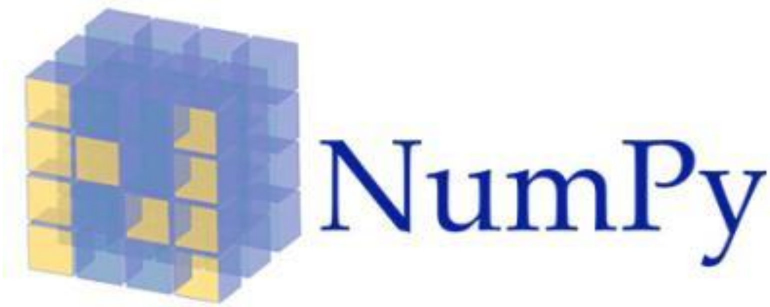


Figure 3.3: Numpy

I am using the NumPy package [7] as it provides an alternative to the regular python list, which has limitations. Numpy library provides a high-performance multidimensional array. Numpy comes with the basic tools to compute and manipulate these arrays. Numpy allows us to analyse the data with a more considerable amount of efficiency than using regular lists. One of the biggest advantages of Numpy is that it performs calculations over the entire array without the need the need to call each value inside the array.

In order to use Numpy it must be imported first.

```
# import the library
import numpy as np
```

This is an example of how we would declare a Numpy array. If this was a regular list, it is not possible to print the contents of the array the way we can for a Numpy array.

```
# How to declare a Numpy Arrray
List_A = np.array([1,2,3,4,5])
# How to print the array
print(List_A)
```

Python Imaging Library

Figure 3.4: PIL

Image Module [8] is part of the pythons imaging library, which allows us to load, create, modify and convert image files to different types of formats. This library is perfect for this project as I am working with Neural Networks. To be able to test against the trained Neural Network I need to be able to take an input from the user. The input would be taken and converted into a Numpy array, which I will then use to test the trained neural network. Some of the Python Imaging Library's capabilities include per-pixel manipulation, masking and transparency handling, image filtering and more.

```
# To install the library but should already be included in Anaconda
Run the PIP INSTALL PIL

# import the library
from PIL import Image

# This is how we load an image in python using the pythons imaging
# library
img = Image.open("file.png")

# Display the image to the user
img.show()
```

Matplotlib

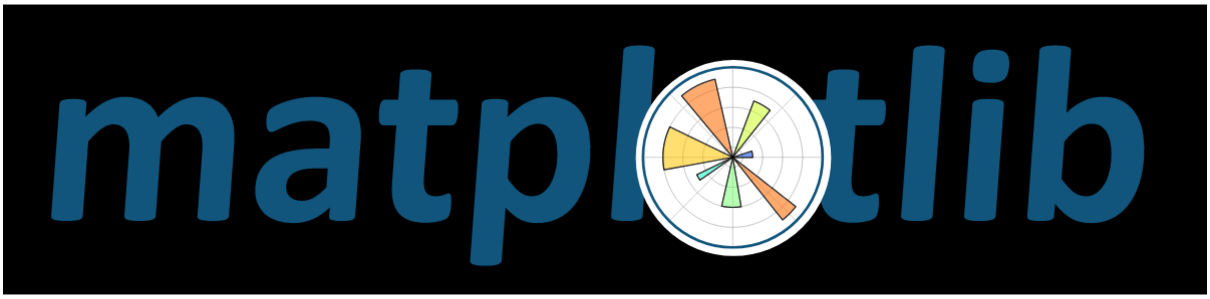


Figure 3.5: Matplotlib

Matplotlib [9] is one of the most popular modules to use for data visualisation in Python. Matplotlib is an object-oriented API that is used for embedding plots and it extends the Numpy module.

Matplotlib is pythons most popular plotting library:

- With just a few lines of code, it produces dozens of different types of plots and charts.
- Matplotlib allows plotting Numpy arrays python lists and Panda data frames.
- Matplotlib gives us full control of styles, properties and axes.

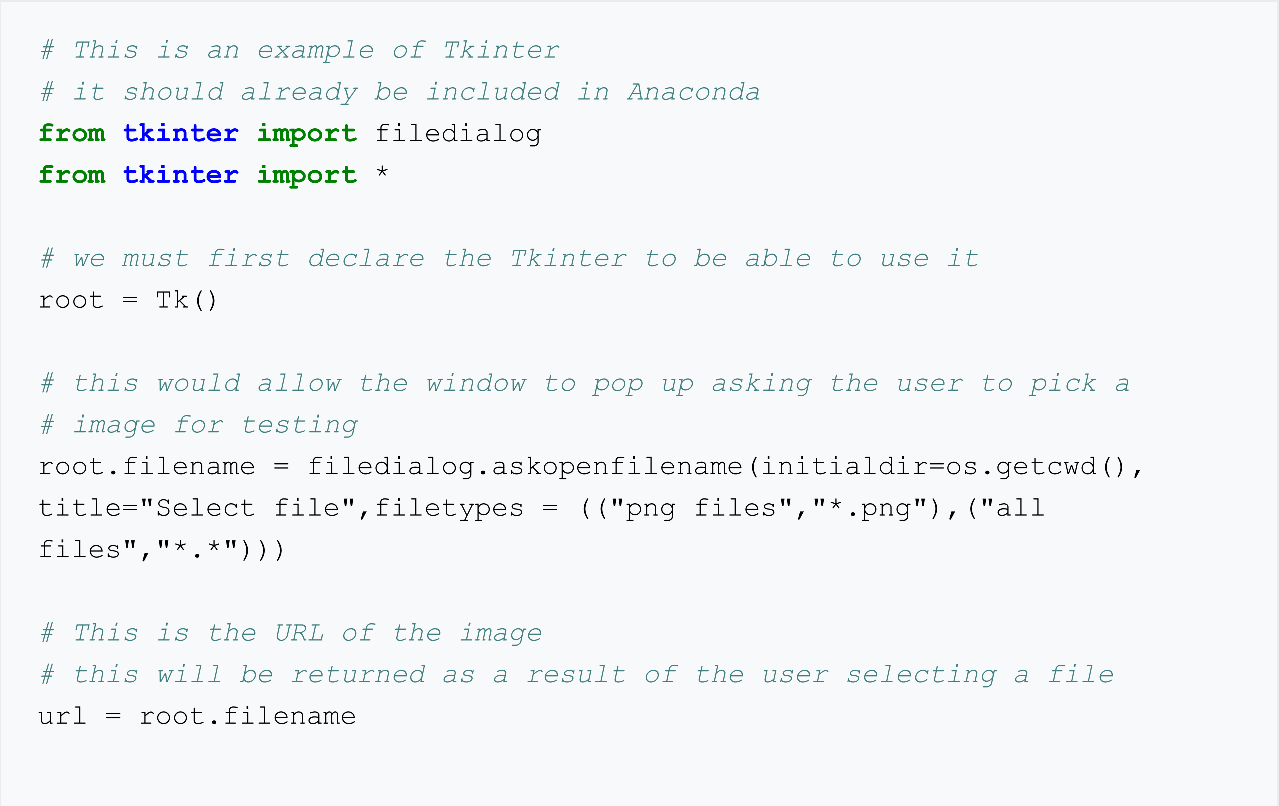
We must have python installed to be able to use the library. The first thing you need to do when using the Matplotlib library is to import it at the top of the program.

```
# This is an example of Matplotlib
# To install the library but it should already be included in Anaconda
Run the PIP INSTALL Matplotlib

# import the library
import matplotlib.pyplot as plt
# This is to give the title to the figure
plt.figure("The Image to be Tested ")
# This piece of code prints out the image which is being passed
# inside the parameter
plt.imshow(image[..., ::-1])
plt.show()
```

Tkinter

Figure 3.6: Tkinter

Tkinter [10] is free software that comes as part of the core Python language. Therefore, we don't need to install anything new. Tkinter is the standard interface for the Tk GUI toolkit. Tkinter allows us to create a GUI interface for our applications. Tkinter is a Python wrapper around a complete Tcl interpreter that is embedded in the Python interpreter. The reason it is possible to combine Python and Tcl is that Tkinter calls are translated into Tcl commands.



```
# This is an example of Tkinter
# it should already be included in Anaconda
from tkinter import filedialog
from tkinter import *

# we must first declare the Tkinter to be able to use it
root = Tk()

# this would allow the window to pop up asking the user to pick a
# image for testing
root.filename = filedialog.askopenfilename(initialdir=os.getcwd(),
title="Select file",filetypes = (("png files","*.png"),("all
files","*.*")))

# This is the URL of the image
# this will be returned as a result of the user selecting a file
url = root.filename
```

OpenCV

Figure 3.7: OpenCV

OpenCV [11] is a library that is made up of programming functions. These functions are mainly aimed at real-time computing vision. OpenCV first developed in the year 1999 by Intel by Gary Bradski and was released in the year 2000. OpenCV is a free, cross-platform library under the open source BSD licence. OpenCV originally was written in C++, today there are bindings for other languages such as Python and Java. OpenCV is nothing but a python wrapper of the original C++ implementation. In OpenCV, all the images are converted to or from a Numpy array. This makes it easier for images to be tested against the trained neural network.

```python
# This is an example of OpenCV being used
# it should already be included in Anaconda
import cv2

# we going to read in the image that is going to be tested
print("Enter image name!!")
userInput = input()

# this here reads the image in
# image for testing
image = cv2.imread('location of image' + userInput + '.jpg')
```

URLLIB

Figure 3.8: URLLIB

The urllib library [12] allows you to access the internet via Python. The urllib.request is the primary module that is used for fetching and opening URLs. This library can retrieve data from the internet in different formats such as JSON, HTML and XML. The goal of the urllib library is to make HTTP requests simpler and more human-friendly.

In the diagram below clearly shows how we access web data through python library urllib.

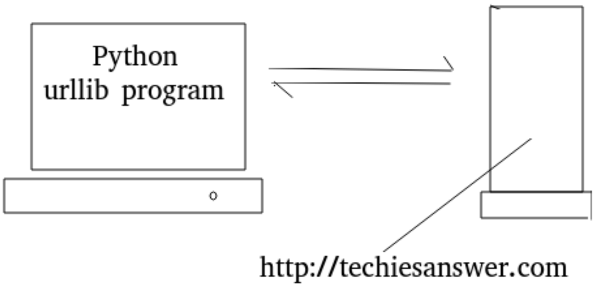


Figure 3.9: URLLIB Explained

Example of how urllib works

```python
# This is an example of urllib being used
# it should already be included in Anaconda
# first thing import the library
import urllib

# Here we make a request to the url that we want to connect to
# in this case the url is the address of an image
request = urllib.request.urlopen(url)

# Then I read the returned value from the request I made
# I save this as a image
image = np.asarray(bytearray(request.read()), dtype="uint8")
image = cv2.imdecode(image, cv2.IMREAD_COLOR)
```

Sha256

Figure 3.10: Sha256

Sha stands for Secure Hash Algorithm [13], SHA-256 is a cryptographic hash function that has a digest length of 256 bits. It is a keyless hash function. A hash function is one that takes in an input and outputs a value deterministic of the input value. For example, if I input the value ABC, I will always get the same output for the value ABC when the hash is run. So, in a hash function, every input has a determined output.

Sha256 Properties

- Sha256 makes sure that you will never get same result for different values.
- It is computationally unfeasible to extract information
- Even a small change in the input value will produce a completely different output.

I am using the Sha256 to hash and save the passwords in MongoDB.

Sha256 Libraries

Hashlib

- Module implements many different types of secure hash algorithms.
- These include SHA1, SHA256, SHA512 and more.
- I use this library to implement the hashing algorithm SHA256 in my project.

GitHub

Figure 3.11: GitHub Logo

GitHub is great for hosting, sharing and collaborating on projects together. GitHub allows hosting of code in a variety of different languages such as Python, Java and C. GitHub is a git repository hosting service and is used as version control manager as discussed earlier in my dissertation. GitHub provides a GUI interface for command-line based Git.

In addition to GitHub being a hosting service, it is a great social networking site. You can follow other GitHub users, browse public projects, show your skills as a developer, subscribe to project updates and much more. GitHub has been described as a portfolio for programmers and is a great tool to show potential employees your skillset. One of the main features of GitHub is forking a repository this allows a project to be copied to your own account. We do this because we do not have the right to make push requests to other user accounts. Forking allows us to make as many changes as we want.

Git Commands

```
# To initialise an empty repository
Git init

# To add all the modified and untracked files
# In the current working directory
```

```
Git add .


# Commit changes
Git commit -m "First commit"

# push the changes made to the repository
Git push origin master
```

Visual Studio Code

Figure 3.12: Visual Studio Code

There are many different types of development tools, but I prefer to design and write my project using Visual Studio Code. This is a source code editor developed by Microsoft, this editor is compatible with Windows, Linux and MacOS. One of the main features of this editor is that it is very programmer friendly. Visual Studio Code is not bulky at all unlike Visual Studio which is both slow and of considerable size. The editor can download extensions depending on your needs.

It is possible to code in many different languages such as C sharp, TypeScript and C. Visual Studio Code is highly customisable, as users can change their theme, add shortcuts and preferences. Visual Studio Code has an excellent intelliSence, refactoring of code, navigation and git support. I believe Visual Studio Code is more convenient than the full version of Visual Studio.

Keras

Figure 3.13: Keras

Keras [14] is a free open-source neural network library written in python. Keras is an interface that allows us to easily access and customise different types of machine learning frameworks. These frameworks include TensorFlow CNTK and Microsoft cognitive. These work in the background and do all the heavy lifting. You can use these frameworks on their own however they are quite complex. As a result, people prefer using a high-level API such as Keras to run on top of them. Keras is a great library for an easier implementation and fast experimentation of deep neural networks. Keras is user-friendly modular and extensible. I will be using Keras to build my own CNN for this project.

```python
# how to import Keras
import keras

# These are the modules from the keras library
# I will use to build my neural network
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.optimizers import Adam
from keras.utils import np_utils
```

Neural Networks

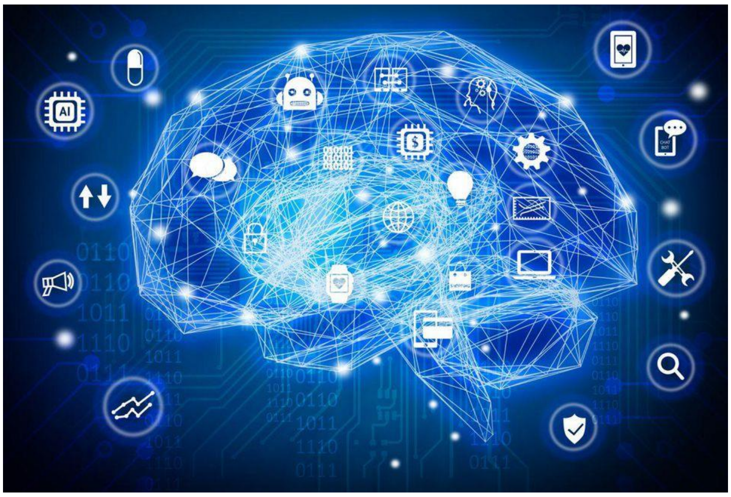


Figure 3.14: Neural Networks

Machine learning is a method of teaching machines to make decisions based on input provided. Machine learning is a branch of artificial intelligence and is based on the idea that machines can learn from data, be able to identify a pattern and make decisions for itself without the need for human intervention. This allows machines to make decisions based on their own analysis, observation, self-training, and experience. Machine learning allows for the continuous progress of computing by exposing it to new scenarios. Then testing and adapting according to that scenario, while; subsequently, employing pattern and trend detection to make improved decisions with similar situations. Machine learning can be confused with the concept of data mining, which is a process of discovering patterns.

Neural Networks are the most popular machine learning algorithms, and these algorithms outperform other means in accuracy and speed. The neural networks are fast becoming a choice for Computer vision scientists. Therefore, learning about neural networks has become imperative.

Perceptron

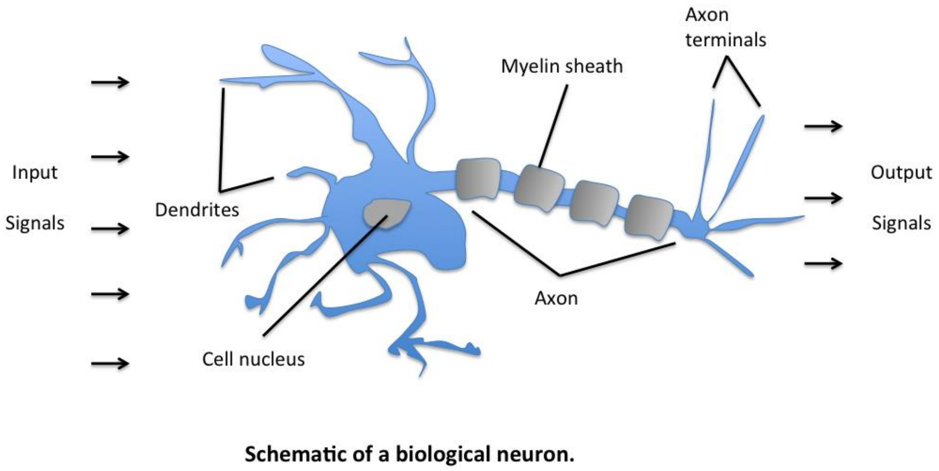


Figure 3.15: Biological Neuron

Perceptron was one of the first biological [15] inspired units used in programming of artificial intelligence and machine learning. I will discuss how a perceptron works and how it inspired further development in artificial neural networks.

In the past a perceptron was used as the foundation of an investigation when attempting to research machine learning and artificial intelligence. Therefore, it is important for my project to understand how it operates. Perceptron is an algorithm for supervised learning of linear classifier. It is a linear classifier meaning the predictions are based on linear predictor functions combining a set of weights with a feature vector. A linear classifier is an algorithm that returns either 0 or 1 for the input given.

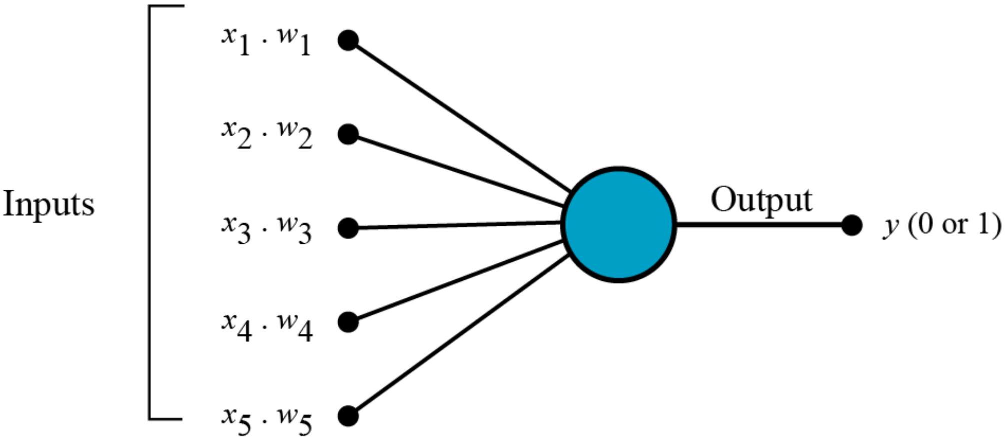


Figure 3.16: Perceptron

Convolutional Neural Networks

Convolutional Neural Networks are made of neurons that self-enhance through learning [16]. It is a class of deep neural networks and is commonly used in Computer Vision. CNNs use perceptron's which mean they require minimal pre-processing. In, computer vision Convolutional Neural Networks receive a raw image or video input vectors, and it performs an operation on each of them. CNNs are primary used for image detection and pattern detection. It is also one of the main categories for image classification and object detection. Convolutional Neural Networks learn to recognise objects very slowly just like we do as kids. We must show the neural algorithm millions of images before it can able to generalize and make predictions for new input. The way we see the world is different to how machines do. So, we must convert the raw image or video data into 2D arrays of number [17].

Convolutional Neural Networks image classifications take a raw input, processes it and classifies it depending on the category of the image i.e. human and animals. Classification is the task of assigning labels to what the input data might be. Whereas localisation is the process of finding where the objected detected might be. Which is usually output with some sort bounding box around the object. We use neural networks to train our models and then classify the raw input data.

K-Nearest Neighbours

K-Nearest Neighbours or KNN neural network [18] is a basic but important classification algorithm in machine learning and can be used for pattern recognition and object detection. The way KNN works is that the algorithm is given the training dataset which then classifies coordinates into groups. These groups are then identified by an attribute. When, we test the algorithm the KNN plots the sample in the same n-dimensional space as the training dataset. Which then looks for its K-Nearest Neighbours based on the training dataset. The input is matched with all the images in the training data and top K with minimum distances are selected. The algorithm goes through the whole dataset and therefore, it is called a lazy learning technique. One of the drawbacks of KNN is object localisation. Two of the same images with different object location would result in KNN giving a highly none-zero distance.

Database – MongoDB

Figure 3.17: MongoDB

MongoDB [19] is published under the GNU public licence and is supplied as a free open source, cross-platform program. MongoDB is a document-oriented program which means is that the program is intended to store, recover and oversee document-oriented data. This is otherwise called semi-structured data. MongoDB is a NoSQL database that uses JSON to structure its documents. MongoDB is a document database that grasps the performance, flexibility and scalability of a NoSQL and is designed for ease of development. MongoDB allows for real-time aggregation in other words up-to-the-minute view of your data, indexing i.e. the way MongoDB associates a key with the location of relating data record and ad hoc query. All these characteristics allow MongoDB to be a very powerful program that allows for access and analysation of our data.

MongoDB stores all the data in the JSON format, this allows for adjustment over time. MongoDB provides a lot of other worthwhile features these include horizontal scaling i.e. the distribution of data across several machines, geographical distribution and high availability. MongoDB also allows for automated management and provisioning for continuous delivery and integration. For storing large documents or files MongoDB makes it very easy and flexible. MongoDB's indexing allows for no compromise of data access, complex aggregations and schema governance controls.

Prerequisites

To get started with MongoDB I implement it locally. I downloaded the MongoDB installer, after running the installer I had to make a folder to store the data from MongoDB. The folder looked like this C:\data\db. This process of setting up nd running MongoDB was tiresome. For example, every-time I needed to use MongoDB I had to first run it locally and then use it. I had to run the run mongod and run mongo in separate command line terminals for it to be able to work.

So, after a while I decided to use MongoDB Atlas, this is a cloud-based MongoDB database with MongoDB atlas I did not need any local implementation and it made things easier compared to do it locally.

What is MongoDB Atlas?

Figure 3.18: MongoDB Atlas

MongoDB Atlas [20] makes it easy to set up, use and scale MongoDB deployment in the cloud. Atlas allows for the same functionalities MongoDB does, such as scalability, security, and disaster recovery but with the advantage of accessibility from anywhere in the world. The service provides 500MBs for free when you sign up, after that you will be required to pay for more storage.

Why MongoDB Atlas?

I decided to use MongoDB Atlas for my project due to the reason that I wasn't dealing with any complex transactions in my project. By using MongoDB Atlas in my project, it allowed me the opportunity to enhance my NoSQL skills and help me learn a new technology. I could have used CouchDB but chose MongoDB instead because of the easier learning curve for MongoDB.

MongoDB Libraries

The following are the python modules I needed to import to make the MongoDB work. I will discuss both briefly.

GridFS

Figure 3.19: GridFS

GridFS [21] is an API provided by MongoDB. It makes storing large files such as video, audio and images in a mango database easier. It is a npm package that can be plugged into any python application. GridFS allows a way to store large files in a database instead of using the filesystem. You cannot store large files in MongoDB. However, it will not allow any document or file that is larger than 16MB as MongoDB will fail and throw an error. To overcome this limitation, we use GridFS. The way GridFS solves this problem is it breaks up the file into smaller more manageable chunks. GridFS stores chunks of data in a collection called fs.chunks. The information of the file is stored in another collection called fs.files.

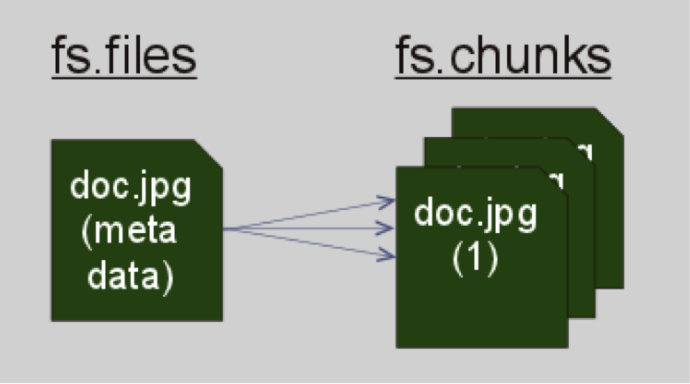


Figure 3.20: fs.files and fs.chunks

Each chunk stores 255KB of data and the number of chunks created depends on the size of the file. The chunks store the actual file data and are linked to the information about the file by a property called files_id. This files_id points to a document that is stored in the fs.files.

```
# import the following modules
from pymongo import MongoClient
import gridfs

# here we establish a connection MongoDB
# The "dnspython" module must be installed to use mongodb+srv:// URIs
client = MongoClient("mongodb+srv://root:root@cluster0-
xyrvy.mongodb.net/test?retryWrites=true")

# here we get the collection we want to use
db = client.MongoProject
testCollection = db.myImageCollection
fs = gridfs.GridFS(db)

# store the image
imageID = fs.put(imageString, encoding='utf-8')

# create our image meta data
meta = {
  'name': userInput,
   'images': [
       {
          'imageID': imageID,
          'shape': image.shape,
          'dtype': str(image.dtype)
        }
     ]
}

# insert the meta data
testCollection.insert_one(meta)
```

Chapter Four

System Design

In this section of my dissertation I will cover the overall design and implementation of my project. I will do this by providing code snippets and visual aid.

Architecture

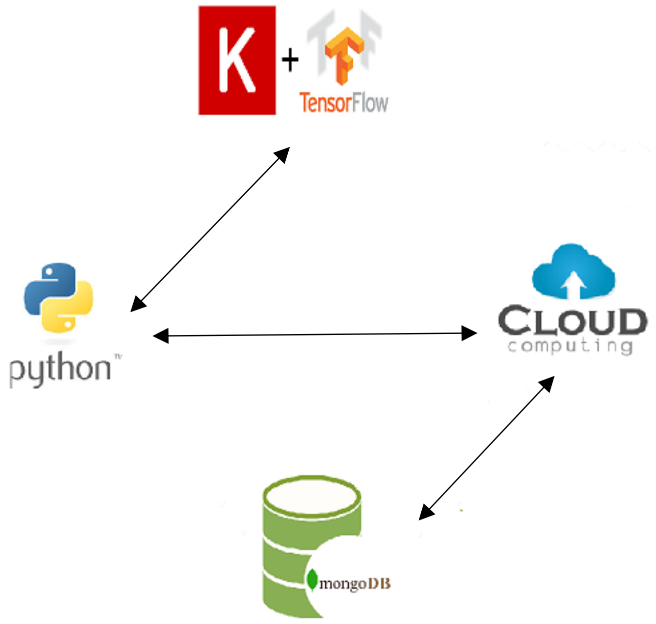
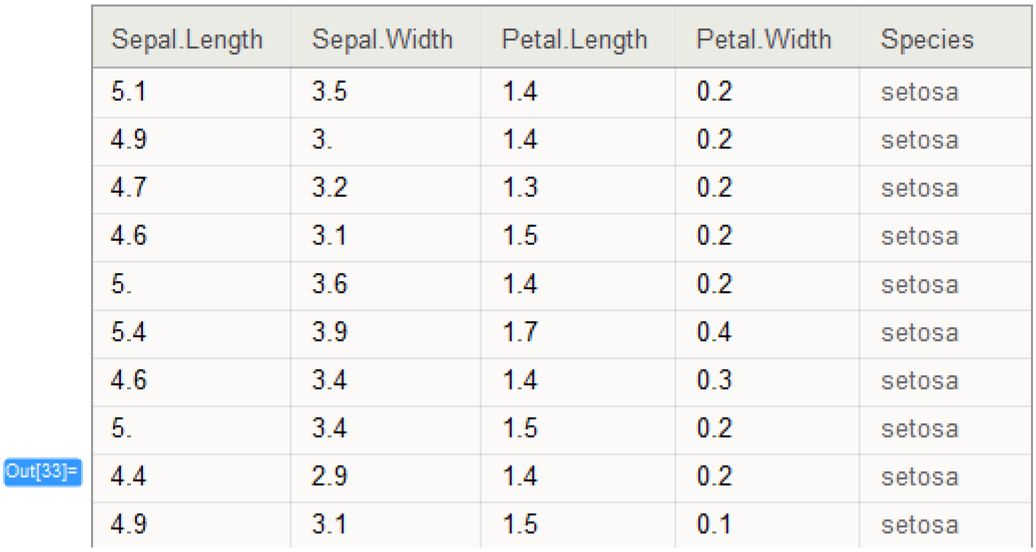


Figure 4.1: Architecture

The above image outlines my projects architecture. I have developed my application from scratch after studying the requirements and careful planning. In the project I am using Keras with TensorFlow to train my main neural network. I have two other neural networks that are not using any libraries for training of neural networks. My project operates input through a command-line basis. My project interacts with MongoDB Atlas for logging and sign-up.

Perceptron

For my first neural network I decided to train the Iris data set, Fisher's Iris data set is a multivariate data set introduced by Ronald Fisher. This data set consist of 50 samples from each of the three species of the flower Iris. The species include Iris setosa, Iris virginica and Iris versicolor.



Out[33]=

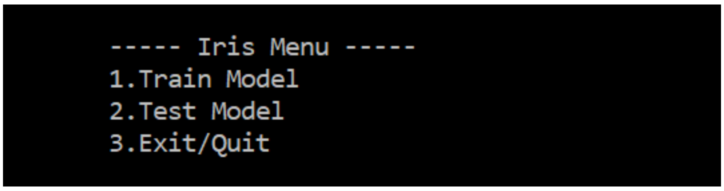
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

Figure 4.2: Iris Data Set

The data was collected by a man called Edgar Anderson, the reason behind the collection of data was to quantify the morphologic variation of three related species of Iris. The features that were measured included the sepal length, sepal width, petal length and petal width. The measurements were taken in centimetres.

Menu for Perceptron

This is my menu for the iris perceptron, I will discuss each of the options given below.



```
----- Iris Menu -----
1.Train Model
2.Test Model
3.Exit/Quit
```

Figure 4.3: Iris Menu

Train Model

Option one in the menu allows the user to train the neural network model. This model will then be used to test against the users input. I will now discuss how I train my first neural network model.

Firstly, I will need to import all the needed modules for the training of this neural network. I start with loading the iris dataset into python memory, I do this by using a python library called panda.

```
# load Iris Flower dataset
IrisData = pd.read_csv('https://raw.githubusercontent.com/uiuc-cse/data-
fa14/gh-pages/data/iris.csv')
```

I normalise the data that is saved in variable called IrisData. After normalisation I convert the name of the species setosa, versicolor and virginica into numerical values. This is done to ensure that it can be used for the training of the neural network.

After separating the training data from testing data, I make a variable with the expected result of the iris data.

```
# This is the training data
X = train.values[:,:4]
targets = [[1,0,0],[0,1,0],[0,0,1]]
# The target is basically the expected result of the iris data set
# Y is a variable is what the iris data should be
# For example number 1 is setosa this means the target array would look like
# [1 0 0] so 1 represents the correct position of the species
y = np.array([targets[int(x)] for x in train.values[:,4:5]])
```

My neural network takes in four inputs i.e. sepal length, sepal width, petal length and petal width. The perceptron also has five hidden layers and three output layers.

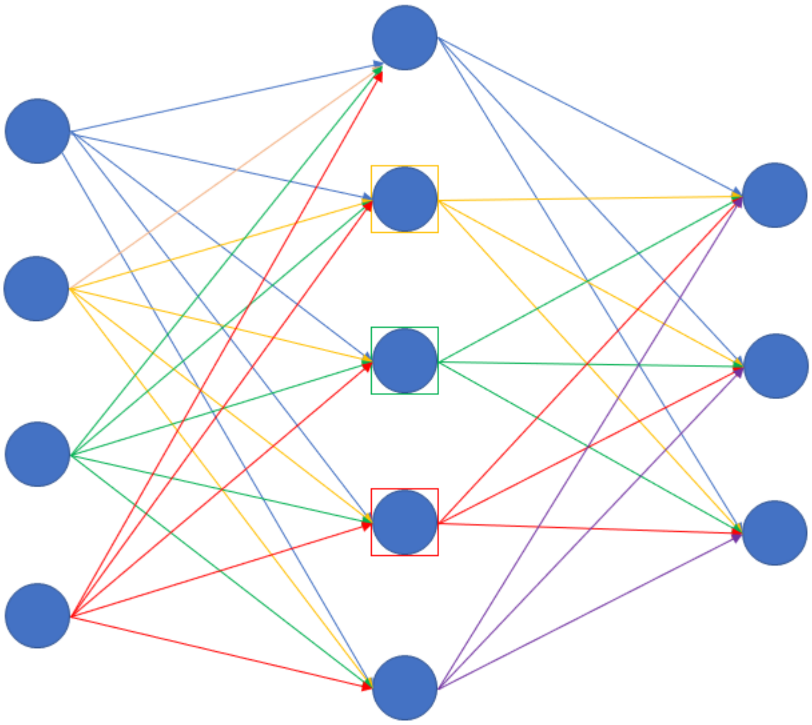
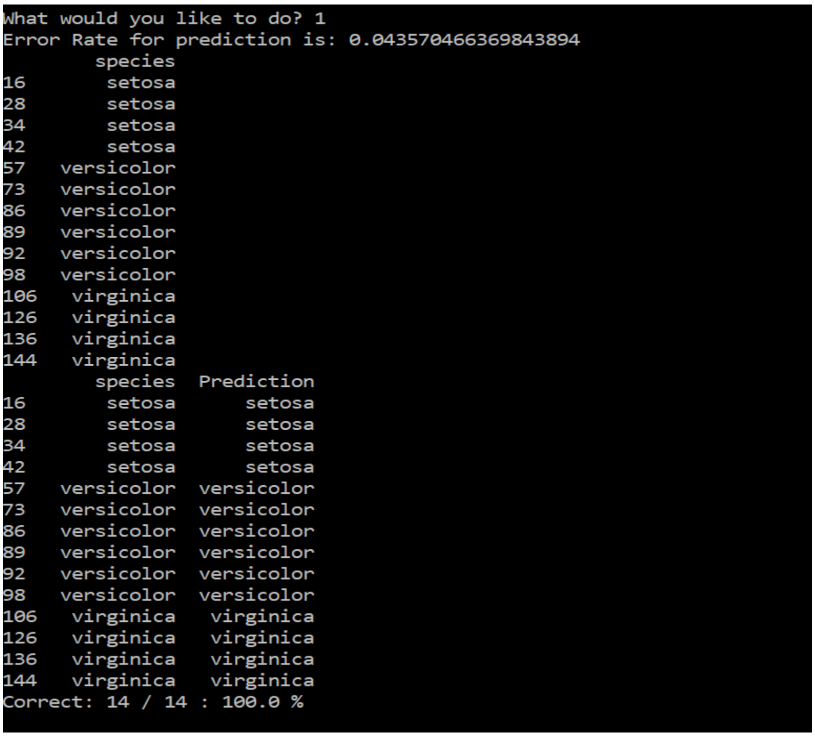


Figure 4.4: Iris Neural Network

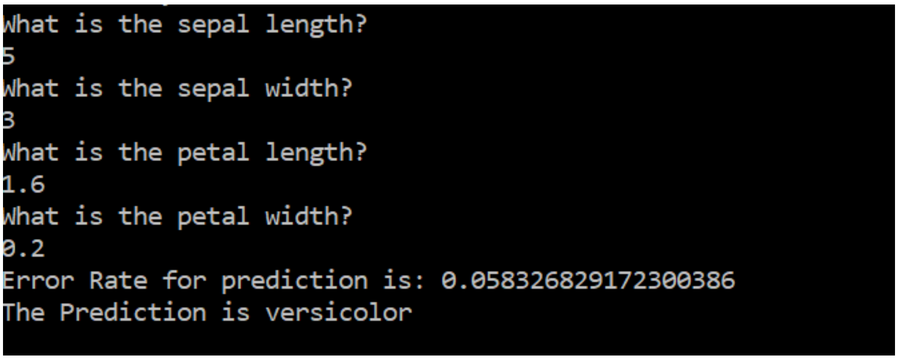
When the user trains the model



```
What would you like to do? 1
Error Rate for prediction is: 0.043570466369843894
        species
16       setosa
28       setosa
34       setosa
42       setosa
57   versicolor
73   versicolor
86   versicolor
89   versicolor
92   versicolor
98   versicolor
106   virginica
126   virginica
136   virginica
144   virginica
        species  Prediction
16       setosa      setosa
28       setosa      setosa
34       setosa      setosa
42       setosa      setosa
57   versicolor  versicolor
73   versicolor  versicolor
86   versicolor  versicolor
89   versicolor  versicolor
92   versicolor  versicolor
98   versicolor  versicolor
106   virginica   virginica
126   virginica   virginica
136   virginica   virginica
144   virginica   virginica
Correct: 14 / 14 : 100.0 %
```

Figure 4.5: Training Iris

The user can train the model and see how accurately the programme predicts. I allow the user to test their own inputs. The user enters the following details about the iris specie. An example of the output is given below.

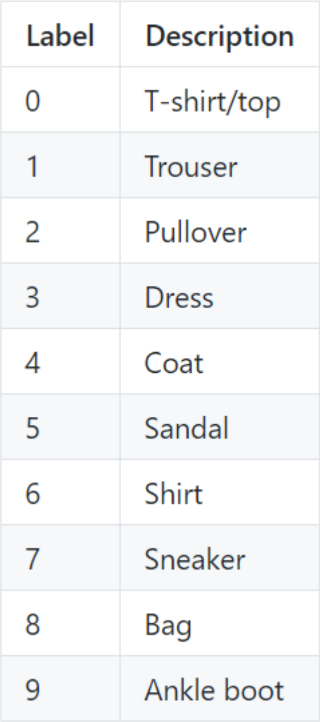


```
What is the sepal length?
5
What is the sepal width?
3
What is the petal length?
1.6
What is the petal width?
0.2
Error Rate for prediction is: 0.058326829172300386
The Prediction is versicolor
```

Figure 4.6: Testing Iris

Convolutional Neural Network

The Convolutional Neural Network I decided to train is the Fashion MNIST Dataset. This dataset consists of sixty thousand examples with ten thousand test images. All the images are 28x28 in grayscale and has ten label classes. These classes are shown below.

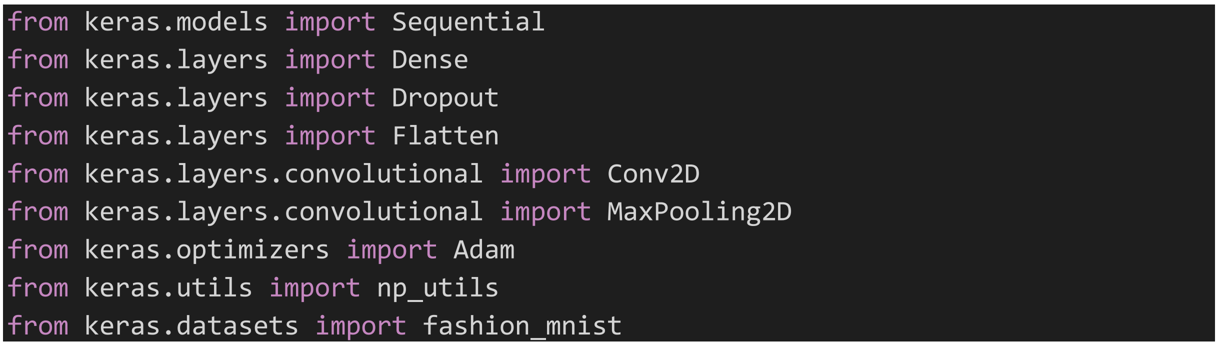


Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Figure 4.7: Labels

Training Convolutional Neural Network

These are all the imports needed for training the neural network. Keras does not come with Anaconda so this must be installed separately.



```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.optimizers import Adam
from keras.utils import np_utils
from keras.datasets import fashion_mnist
```

In my training method I first load data from the keras datasets. This saves me a lot of time because I don’t have to convert images into numpy array. This loads all the needed data into these variables.

```
# Load the fashion-mnist pre-shuffled train data and test data
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

I had to reshape the format to make it work.

```
# Reshaping to format which CNN expects (batch, height, width, channels)
x_train = x_train.reshape(x_train.shape[0], x_train.shape[1],x_train.shape[2],
1).astype('float32')
# Reshaping test data
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], x_test.shape[2],1).
astype('float32')
```

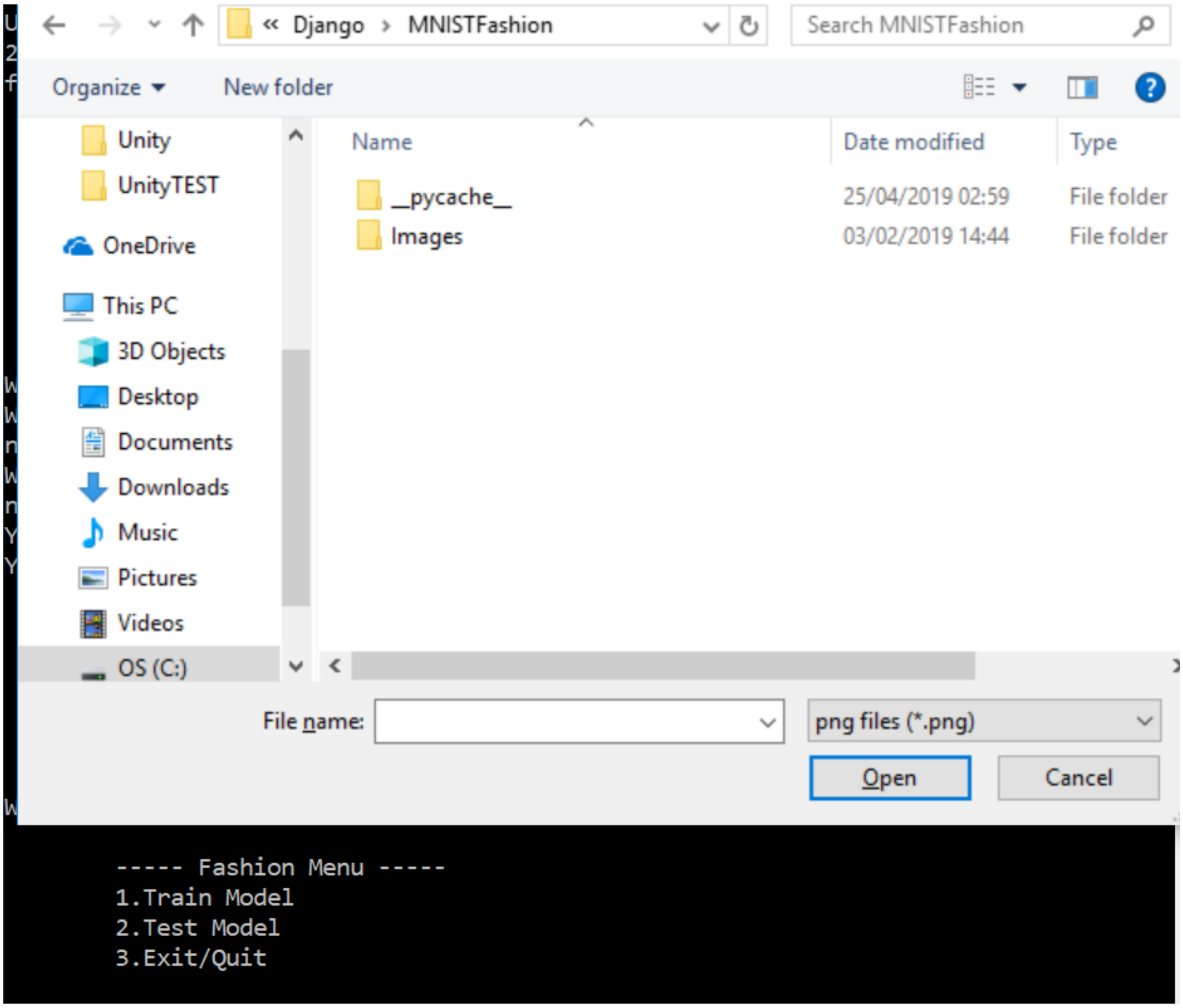
Data is normalised after reshaping; we normalise the data dimensions so that they are of approximately the same scale. This is the complete model of the Convolutional Neural network

```
# Define the model
model = Sequential()
# Must define the input shape in the first layer of the neural network
model.add(Conv2D(filters=64, kernel_size=2, padding='same', activation='relu',
input_shape=(28,28,1)))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.3))

model.add(Conv2D(filters=32, kernel_size=2, padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(classes, activation='softmax'))

# Take a look at the model summary
model.summary()

# Compile the model Fit the model
model.compile(loss='categorical_crossentropy',optimizer='adam',
              metrics=['accuracy'])
# Fit the model
model.fit(x_train,y_train, batch_size=64, epochs=10,validation_data=(x_test,
y_test))
```

We can save the trained model with keras to do all you have to do is call the save function. You can choose the model save type i.e. JSON or h5 file.

Testing Convolutional Neural Network

I have a second class which tests the saved model. When the test option is selected from the Fashion MNIST Menu the user is prompted to pick a picture from the folder. The user can select any image for the purposes of testing.



Name	Date modified	Type
__pycache__	25/04/2019 02:59	File folder
Images	03/02/2019 14:44	File folder

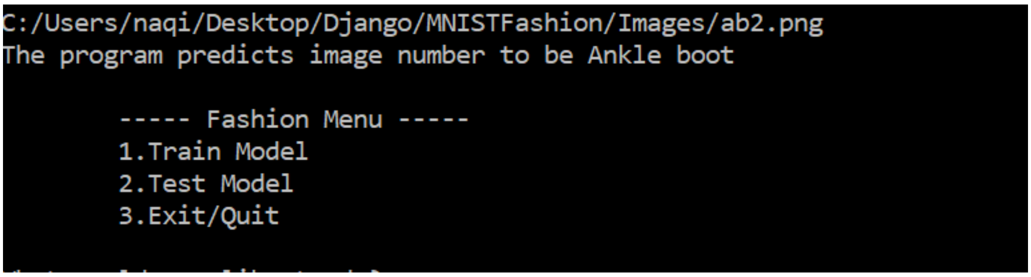
File name:

png files (*.png)

```
        ----- Fashion Menu -----
        1.Train Model
        2.Test Model
        3.Exit/Quit
```

Figure 4.8: Testing CNN

When user selects an image, the programme uses the predict function to recognise the image. The programme returns what it thinks the image might contain.



```
C:/Users/naqi/Desktop/Django/MNISTFashion/Images/ab2.png
The program predicts image number to be Ankle boot

        ----- Fashion Menu -----
        1.Train Model
        2.Test Model
        3.Exit/Quit
```

Figure 4.9: CNN Predict

K-Nearest Neighbour

For my last neural network, I decided to detect the majority colour in an image. I have trained the neural network to detect the most in an image. For example, the image below should return the value green.

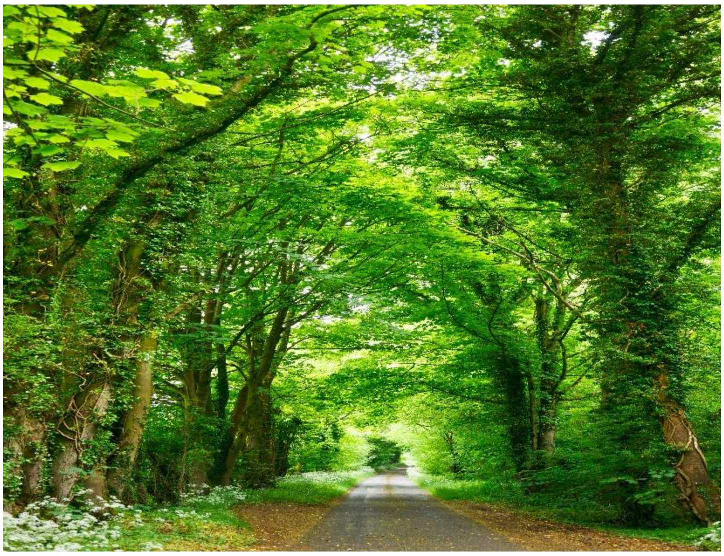


Figure 4.10: KNN Image Example

Training K-Nearest Neighbour

I had to create my own training data for this neural network. I am storing the training data in a file called training.data. This neural network should be able to recognise at least seven colours as I have included them in my training file. Here is a sample of the training data.

```
255,250,250,red
244,194,194,red
255,105,97,red
9,0,0,black
28,29,33,black
27,32,35,black
72,209,204,blue
64,224,208,blue
0,255,255,blue
102,205,170,green
0,100,0,green
85,107,47,green
253,251,251,white
```

```
242,233,228,white
242,233,228,white
254,101,33,orange
255,153,0,orange
255,103,0,orange
254,242,0,yellow
246,191,39,yellow
255,215,12,yellow
```

After creating the training data, I also created a file to store the testing data. At the start of the program I make sure that these files exist.

```
# Return True if path is an existing regular file.
# This follows symbolic links, so both islink() and isfile() can be true for
# the same path.
# https://docs.python.org/2/library/os.path.html
if os.path.isfile(dataPath) and os.access(dataPath, os.R_OK):
   print ('The Data File Exists!')
else:
   print('Data File Does Not Exist!')
```

I have a method in my API folder that takes in an image and calculates the RGB values of an image. When the method is done with calculating the RGB value is then stored into the test.data file. I will use this value to test for the colour of the image.

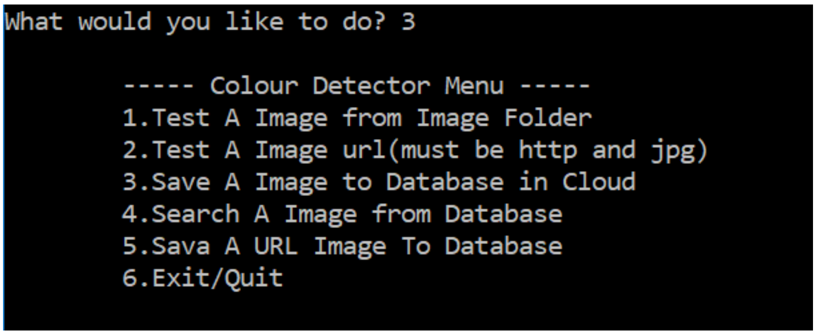
```

# here open the test.data file and right the rgb value which will be tested
with the knn
with open('ColourDectector/Data/test.data', 'w') as myfile:
        myfile.write(feature_data)
```

The test.data and training.data files are sent to the classifier class. This class is also inside the API folder. In this class I train my neural network and I return the prediction made by the trained neural network. The code below shows the for loop which keeps going until the file is finished. Inside the for loop I first get the neighbours. These neighbours are sent to the response function which return the smallest distance value. In other words, it returns a value that is closet to one of the values inside the training.data file.

```
#run the for loop len(testDatFeatureVector) amount of times
    for i in range(len(testDatFeatureVector)):
        # https://machinelearningmastery.com/tutorial-to-implement-k-nearest-
neighbors-in-python-from-scratch/
        # Locate k most similar data instances.

        # getNeigbours
        neighbors = getNeighbors(trainDataFeatureVector,
testDatFeatureVector[i], k)
        # get response
        res = response(neighbors)
        # append the response to prediction array
        prediction.append(res)
```

Testing K-Nearest Neighbour

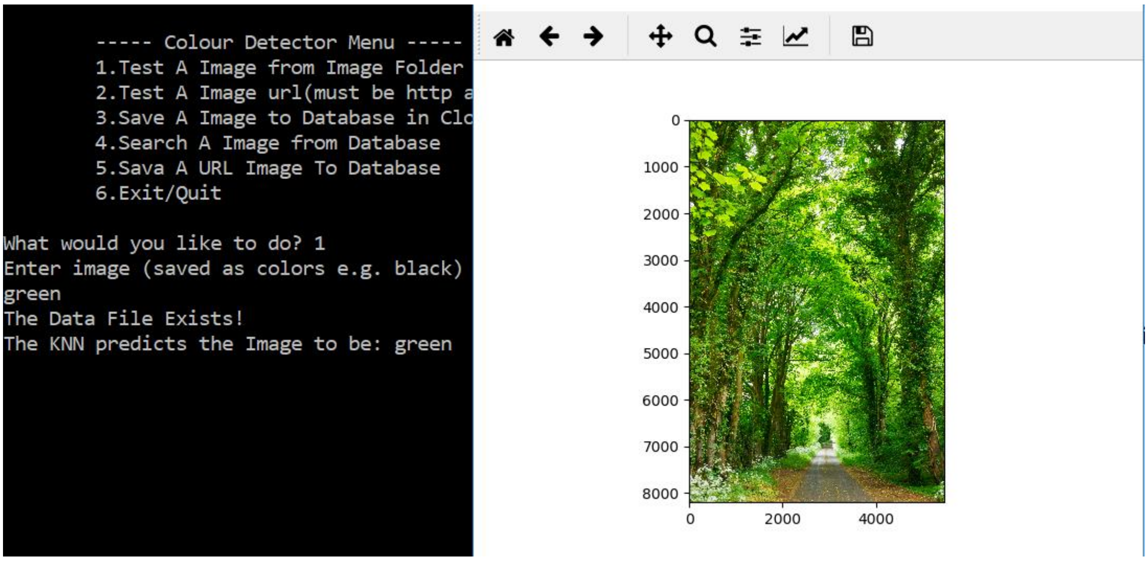
This is what the menu looks like for the K-Nearest Neighbour neural network. I will now discuss each of the menu options in greater detail.



```
What would you like to do? 3

        ----- Colour Detector Menu -----
        1.Test A Image from Image Folder
        2.Test A Image url(must be http and jpg)
        3.Save A Image to Database in Cloud
        4.Search A Image from Database
        5.Sava A URL Image To Database
        6.Exit/Quit
```

Figure 4.11: KNN Menu

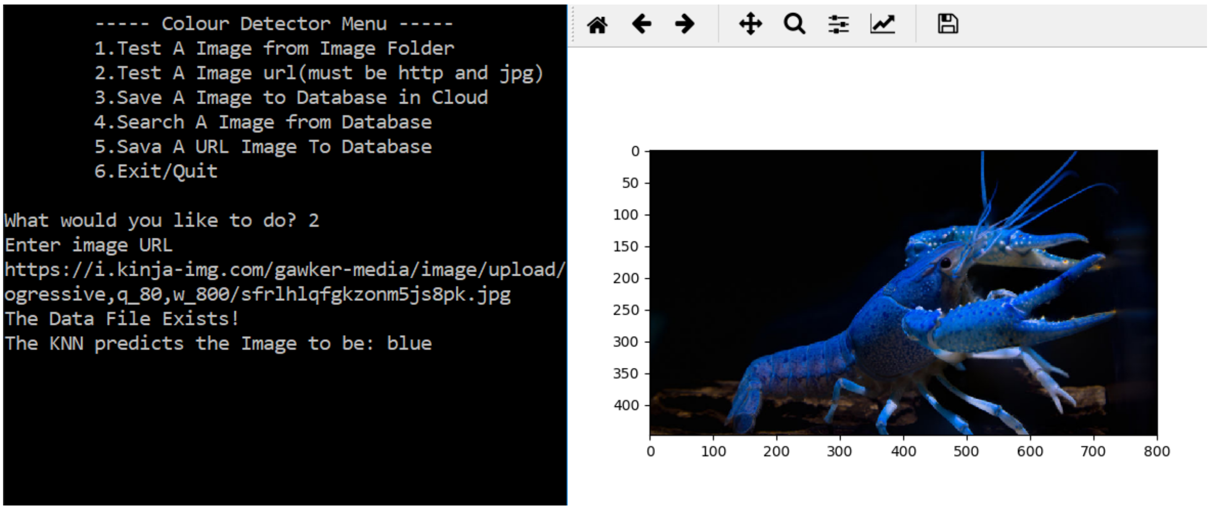
The first option allows the user to pick an image from the filesystem and test it against the neural network.



```
        ----- Colour Detector Menu -----
        1.Test A Image from Image Folder
        2.Test A Image url(must be http a
        3.Save A Image to Database in Clo
        4.Search A Image from Database
        5.Sava A URL Image To Database
        6.Exit/Quit

What would you like to do? 1
Enter image (saved as colors e.g. black)
green
The Data File Exists!
The KNN predicts the Image to be: green
```

Figure 4.12: Test KNN

The second option allows the user to enter a URL and test it against the neural network.



```
        ----- Colour Detector Menu -----
        1.Test A Image from Image Folder
        2.Test A Image url(must be http and jpg)
        3.Save A Image to Database in Cloud
        4.Search A Image from Database
        5.Sava A URL Image To Database
        6.Exit/Quit

What would you like to do? 2
Enter image URL
https://i.kinja-img.com/gawker-media/image/upload/
ogressive,q_80,w_800/sfrlhlqfgkzonm5js8pk.jpg
The Data File Exists!
The KNN predicts the Image to be: blue
```

Figure 4.13: Test URL Image

I have implemented extra functionality into my last neural network. I have included the option to add an image to the database in cloud. The other options include being able to search for the image and saving a URL image to the database.

MongoDB Features

When the user runs the program, it allows the user to either login or sign-up. I am using MongoDB to implement this feature.

Initially we must connect to the MongoDB Atlas in the cloud. To do this we must first install the PyMongo module. To access this library, we can import it at the top of the script. The sub module MongoClient is needed when connecting to MongoDB.

```
# imports the module
from pymongo import MongoClient
```

This is how we use MongoClient to connect to MongoDB. Our database is set up in the cloud, so we make a request to the following address. Then we retrieve the database we would like to use i.e. User. Finally, the last step in setting up is to give the name of the collection you are going to be using to save the user information to.

```
# connect to mongodb
client = MongoClient("mongodb+srv://root:root@cluster0-
xyrvy.mongodb.net/test?retryWrites=true")
# tell which database you would like to use
db = client.User
# name of the collection
collection = db.passwords
```

When the Mongo Client has been setup, I also have a method that allows the user to sign-up. I ask the user to enter their username and the password that would be used to login.

```
#ask the user their name
print("----- Adding a User To Database -----")
print("What is your User Name: ")
```

```
username = input()
```

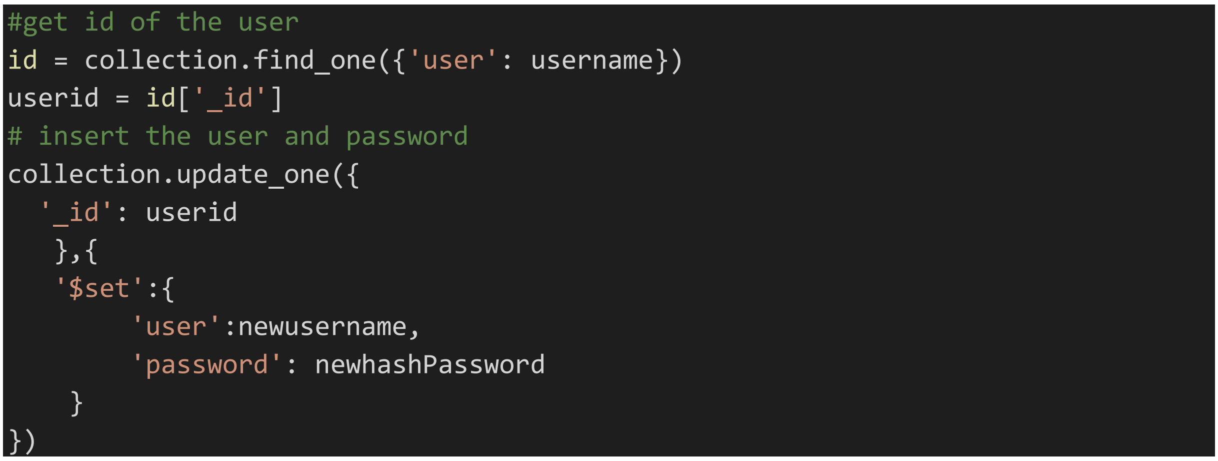
Then there will be check on the username given to see if it already exists in the database. If the user already exists, I will tell the user to try again with a new username. This is done to make sure that there are no two users who have the same username. If the username does not exist, then I allow the user to enter the password. I am also using the Hashlib module to hash the password and then passing it to the collection for storing.

```
# check if the user name doesnt already exists
if collection.find_one({'user': username}) == None:
   # if it doesnt then ask for password
   print("What is your Password: ")
   password = input()

   # save the hash of the password
   hashPassword = hash_password(password)

   # insert the user and password
   collection.insert_one({'user': username, 'password': hashPassword})

else:
    # if the user already exists then ask to add new user again
    print("Sorry User Already Exist with Same Name!!")
    print("Try Again")
```

For logging in, the user is asked to enter their username and once again I check if the username exists. If the user does exist in the cloud, I then ask for their password, I make sure that it’s the correct password by hashing it and comparing it with the users stored password.

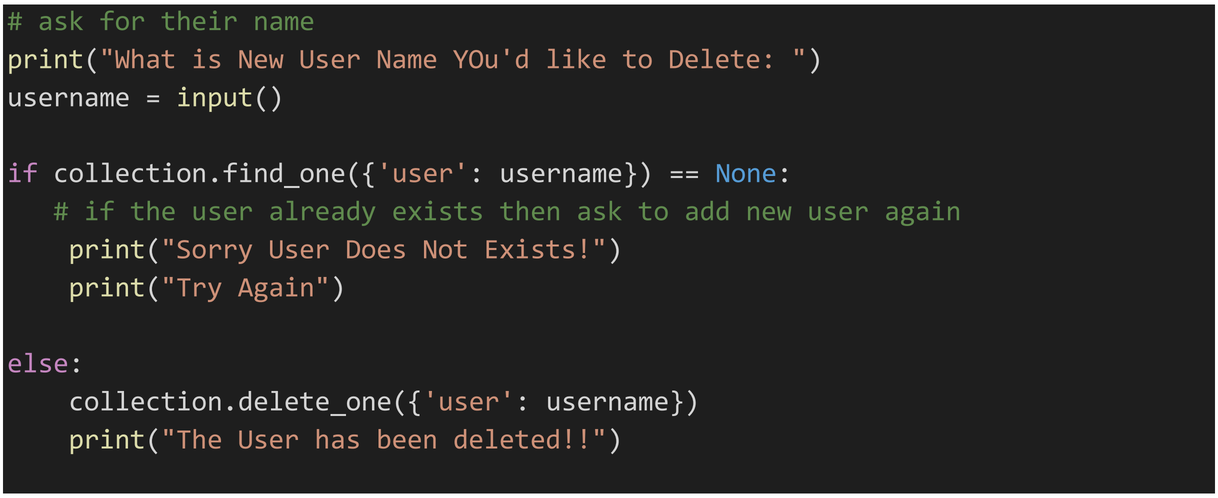
```
# get the password back from the database
# db = collection.find_one({'user': username})
db = collection.find_one({'user': username}, {'password': 1, '_id' : 0})
# save the password into a new variable
mongoHash = db['password']
#check if the password is the same
if check_password(mongoHash,password):
   print('You entered the right password')
   return True
else:
   print('I am sorry but the password does not match')
```

I have also implemented the update of the user details function. I have a method that allows the user to update their details. In the method I ask the user for the username. If the username exists, I then allow the user to make the desired changes to their name and password. When I ask for the new username, once again I make sure that there isn't a user with the same name in the database. I use the set function to update the user details.



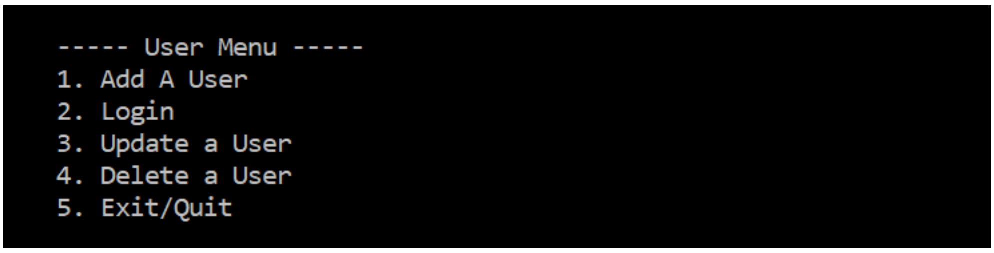
```
#get id of the user
id = collection.find_one({'user': username})
userid = id['_id']
# insert the user and password
collection.update_one({
  '_id': userid
   },{
   '$set':{
        'user':newusername,
        'password': newhashPassword
    }
})
```

The user can also delete their details if they wish to do so. Once the option has been selected, I will first ask the user for their username. I then check if the username exists on the database, if it does it will be deleted from the database.



```
# ask for their name
print("What is New User Name YOu'd like to Delete: ")
username = input()

if collection.find_one({'user': username}) == None:
   # if the user already exists then ask to add new user again
    print("Sorry User Does Not Exists!")
    print("Try Again")

else:
    collection.delete_one({'user': username})
    print("The User has been deleted!!")
```

MongoDB Menu



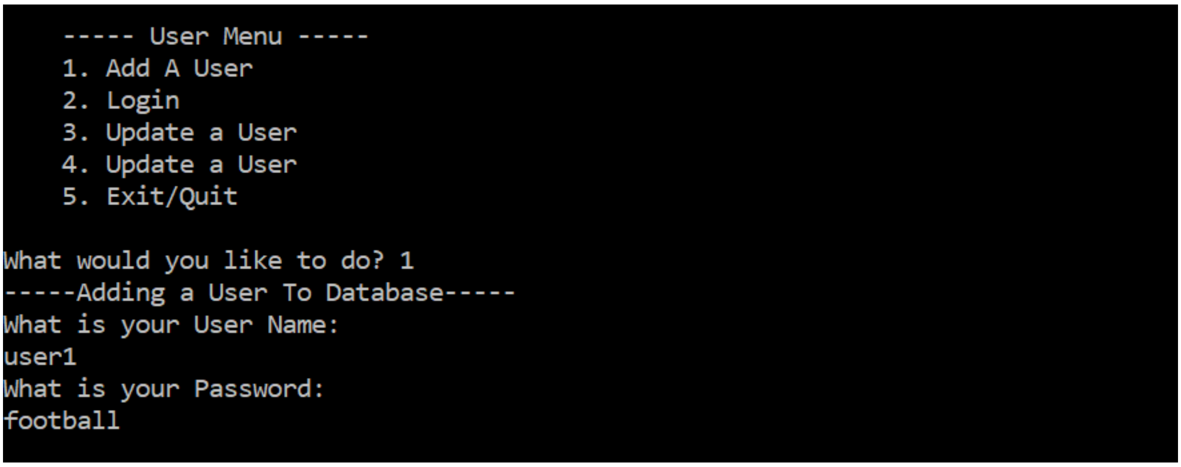
```
----- User Menu -----
1. Add A User
2. Login
3. Update a User
4. Delete a User
5. Exit/Quit
```

Figure 4.14: MongoDB Menu

This is what the user login menu looks like. I have four options which allow the user to create, read, update and delete from the database.

The Create option in the Menu.

You can see below the user can create an account to login by selecting the first option of add a user. The user will be asked to enter a unique username that does not exist on the database; then the user will be asked to enter a password. The user will not be automatically logged in once this happens; he will have to choose to login in the User Menu.

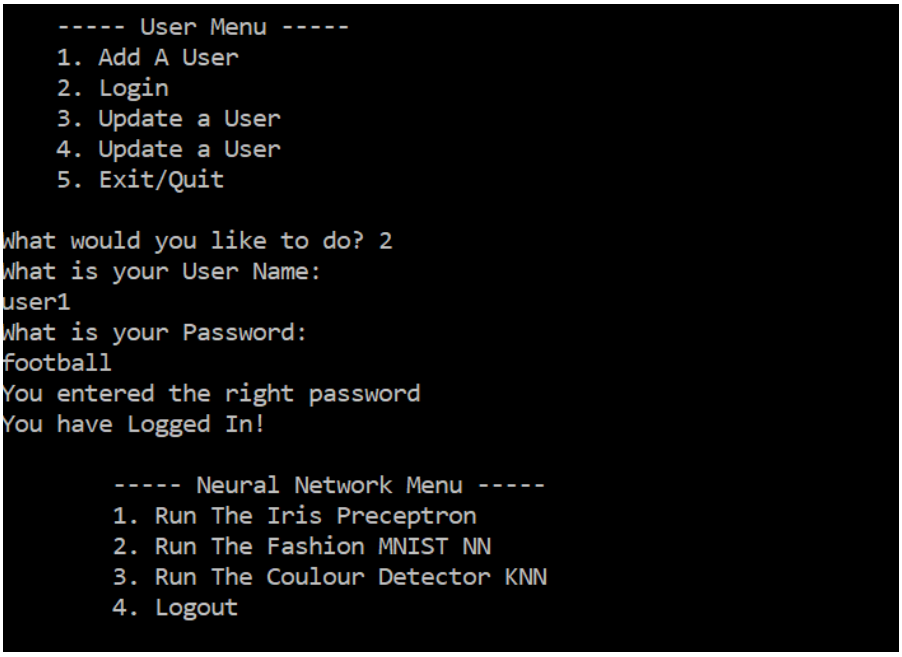


```
    ----- User Menu -----
    1. Add A User
    2. Login
    3. Update a User
    4. Update a User
    5. Exit/Quit

What would you like to do? 1
-----Adding a User To Database-----
What is your User Name:
user1
What is your Password:
football
```

Figure 4.15: Create User Option

The Login(read) option in the Menu.

Once the user logs in successfully with his own username and password then the menu of the Neural Network will be displayed. The menu will give the user the choice to test all the different types of neural networks.

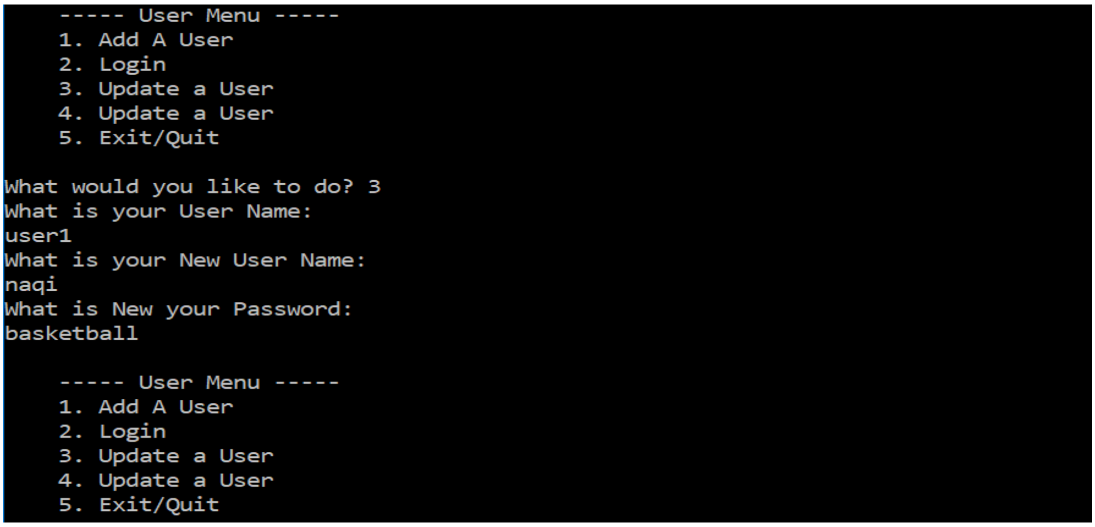


```
    ----- User Menu -----
    1. Add A User
    2. Login
    3. Update a User
    4. Update a User
    5. Exit/Quit

What would you like to do? 2
What is your User Name:
user1
What is your Password:
football
You entered the right password
You have Logged In!

        ----- Neural Network Menu -----
        1. Run The Iris Preceptron
        2. Run The Fashion MNIST NN
        3. Run The Coulour Detector KNN
        4. Logout
```

Figure 4.16: Login Option

The update option in the Menu.

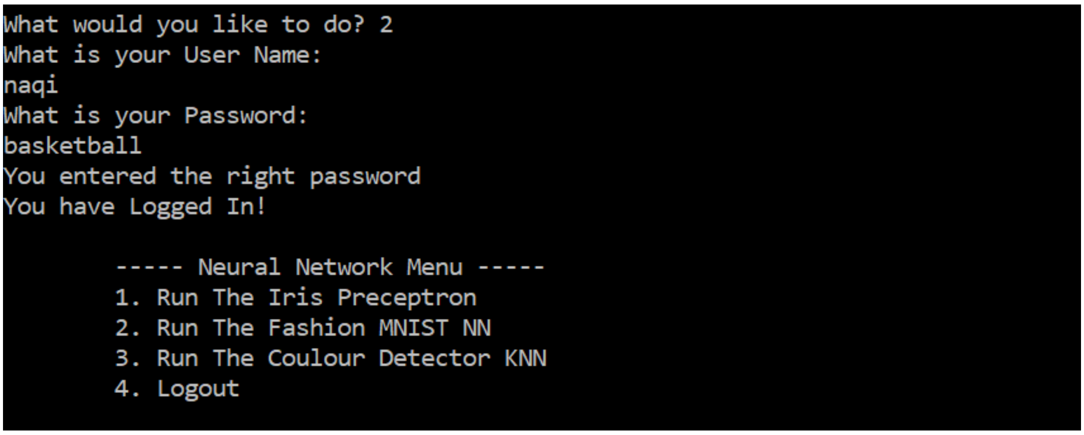
I am updating the user I created to show it properly working. The user will first be requested to give their existing username; then once entered they will be requested to enter a new username and password. Once entered the user will have to login and enter their new username and password to validate their log in.



```
    ----- User Menu -----
    1. Add A User
    2. Login
    3. Update a User
    4. Update a User
    5. Exit/Quit

What would you like to do? 3
What is your User Name:
user1
What is your New User Name:
naqi
What is New your Password:
basketball

    ----- User Menu -----
    1. Add A User
    2. Login
    3. Update a User
    4. Update a User
    5. Exit/Quit
```

Figure 4.17: update Option

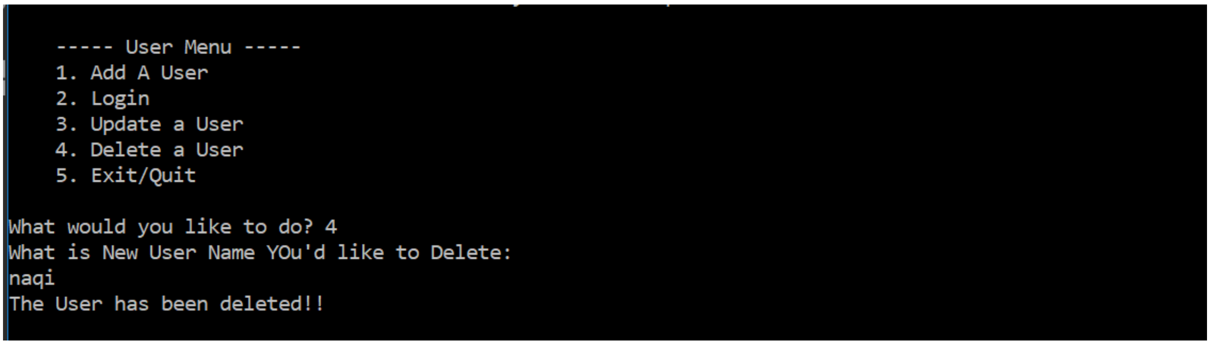
The user can now successfully login in with the updated details.



```
What would you like to do? 2
What is your User Name:
naqi
What is your Password:
basketball
You entered the right password
You have Logged In!

        ----- Neural Network Menu -----
        1. Run The Iris Preceptron
        2. Run The Fashion MNIST NN
        3. Run The Coulour Detector KNN
        4. Logout
```

Figure 4.18: Login

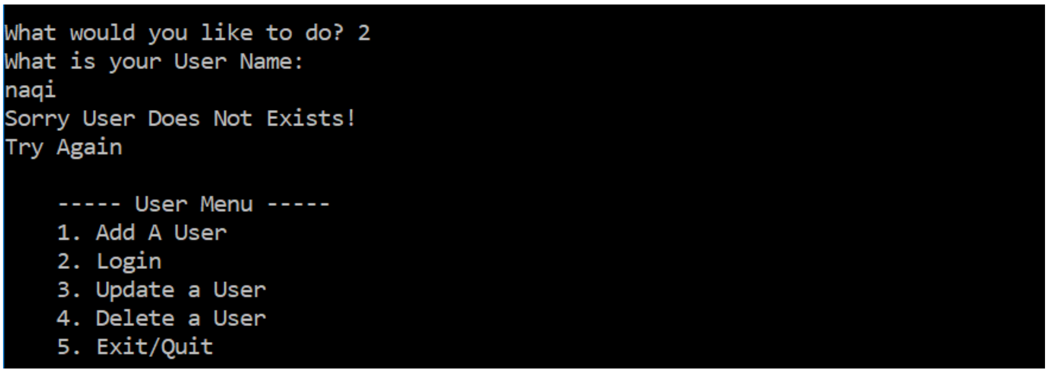
The delete option in the Menu.



```
    ----- User Menu -----
    1. Add A User
    2. Login
    3. Update a User
    4. Delete a User
    5. Exit/Quit

What would you like to do? 4
What is New User Name YOu'd like to Delete:
naqi
The User has been deleted!!
```

Figure 4.19: Delete Option

When I delete the user from the database it will not allow that user to login to their old account anymore as their details, both the username and password have been deleted. The user will have to make a new account to be able to login.



```
What would you like to do? 2
What is your User Name:
naqi
Sorry User Does Not Exists!
Try Again

    ----- User Menu -----
    1. Add A User
    2. Login
    3. Update a User
    4. Delete a User
    5. Exit/Quit
```

Figure 4.20: Delete Option

Chapter Five

System Evaluation

Chapter five of my dissertation will discuss the testing and evaluation process of the project. I will also describe the testing process and steps that were taken while developing aspects my project. My evaluation will also cover any constraints or drawbacks on the project that I had to make from the initial design. Finally, I will focus on where I could have made improvements and what could have been implemented to the final product.

Testing

Testing was carried out at each stage of development to ensure that every component of the project satisfied the predefined requirements and successfully operated with one another. Throughout the development process I was using an agile approach to my project so rather than sequential testing the project in predetermined periods of time I had to focus more on continuously testing as I made changes to the project; while it was time consuming I feel it allowed me to attempt to stay ahead of any issues before they became a larger problem. I used black box testing to ensure that each component of the project worked to its intended functionality. Another reason testing is performed was to attempt to minimise the number of bugs found in each component; as with any form of software testing it was impossible to achieve one hundred percent testing coverage. When I was working with MongoDB, I made sure that it was important to check that the data and information was being stored correctly in the cloud database. While coding and testing the different neural networks, I had to ensure that each part of the neural network was operating to its intended functionality. This was done by testing each part of the neural network in the command line. I trained my neural networks for the purposes of testing and tested each of them against data that would produce a known outcome. After carrying out these tests, I believe that this effort resulted in a sturdier system overall and I was happy to move forward with my project.

Limitations

Based on the choices of technologies and programming paradigms I made during development I encountered fewer limitations than I was expecting. Identifying possible problems occurring early on is an extremely important part of the development process; so, using Agile allowed me to attempt to try stay ahead of these issues. These can occur due to a lack of time or experience in working with new technologies. Recognising limitations isn't focussed on failure instead I view it as offering genuine insight into the creation process of the project and what could have been improved. Doing this allows me to gain valuable experience something I have an admittedly limited amount of regarding agile development. When I was working on this project, I encountered obstacles that had to be overcome to make progress. When training my Iris Neural Network, it was a genuine challenge to test the iris dataset against the input that is given by the user. I was also encountering problems normalising my data, to overcome the issue I had to use the data that I used to train the neural network and append the input to it. This way I could normalise it and test it against the neural network. After designing and training the first neural network I then realised that I needed to try foreseeing what problems could occur when training the other neural networks. This would better allow me to keep the project on track without losing focus of other requirements and to fully organise the timeline of my project as it was continuously being worked on.

One of the main features of my project was allowing the user to test a URL image against the Colour Detector Neural Network. When I was designing this the limitations were how I would take in the URL and store the image into an array to be tested. This took a lot of time, but I eventually found a solution to the issue. Another small limitation of this project is that to be able to run the user has no choice but to install a lot of modules or libraries that the project requires. Lastly one of the main limitations were the time management, as I was not able to implement the Django Framework.

Chapter Six

Conclusion

I really enjoyed the challenge of working on this project and the freedom it allowed me to choose my own approach to the design. Learning new technologies along with encountering their limitations allowed me to enhance my experience. After over three months of continuous efforts I have finally completed my project. Overall, from the initial planning of the project I can say that I have met all the objectives that I set out for myself on this project.

To conclude this project my goal was to create a program written in the python programming language that trains and tests different versions of neural networks. I have achieved this objective and more as I have implemented my other objectives. I will now explain each of the objectives in greater detail.

Objectives

The first objective was to build the Convolutional neural network (CNN). I have achieved this objective; I have a fully trained neural network that predicts with an accuracy of 80%.

My second objective was to train a Multilayer Perceptron. I have achieved this goal; I have trained the Multilayer Perceptron using the Iris dataset. The multilayer perceptron can predict the three types of iris species with an accuracy of 98%. The hardest part of this implementation was to get the input of the user working.

My third objective was to train the K-Nearest Neighbour neural network. My Colour Detector neural network is fully implemented, and this objective has been achieved. I made my own training data for this neural network which was another objective I set myself of its own. The user can now test any image against my neural network.

My fourth objective was to successfully setup MongoDB. MongoDB database has been successfully setup and now can be used to store the images and user data i.e. their username and password.

To achieve my fifth objective, I have implemented the Hashing Algorithm SHA256. The program now stores the hash value of the passwords. When the user enters the password, it will be hashed and compared with the string from the database.

Lastly, I have made the command line application more interactive by adding more GUI features. This is achieved by using Tkinter package which provides the GUI interface for python.

What I have learned

During the development process of my final year project I have learned a significant amount of skills from a large variety of subjects. From developing a python program to learning new languages, working with different types of neural networks and cloud technology. This project has allowed me to discover new technologies and working with tools that I have little experience working with before. Such as the different types of python libraries. I have used so many different python modules in my project. It was very interesting to learn about each module and what their functionality was, often when I needed them for a problem faced. For example, using the urllib module, this module makes a connection to the URL given and retrieves data back. I use this in my project to allow the user to test URL images from the internet.

Working on this project has resulted in me learning and gaining a better, in-depth knowledge and understanding of the technology I have used. Python is used as the basis to implement the three separate types of neural networks. Python is also used to connect with the database to store the user's information and images. Before the beginning of the project, I was not overly familiar with the language and had little experience coding with it. This was my first-time using python outside of writing Jupyter notebooks for a separate module. I wanted to use python for my final year project because I wanted to gain more experience coding with it to enhance my knowledge of the language.

Also working with MongoDB was a great choice, the MongoDB documentation is very extensive and easy to follow whenever I encountered any issues. I have some experience using MongoDB for a previous project written in Java but using it with Python I felt it complimented my project.

Writing a dissertation was also a new challenge for me as I have never written such an extensive paper that was to be used as a helpful aid to compliment my project. I have experience writing readme files but these are shorter instructions. Writing the dissertation allowed me to better explain my design choices and reasoning for paths I took during the developing process. I have been taught the last few years that producing satisfactory documentation is a key part of software development. Doing this even helped me gain an even better understanding of all that I have learned and experienced when developing this project.

Future Development

For the future development of the application I would recommend developing a complimentary client application that would allow the user to test each of the neural networks of my project. Further additions could be made with a different range of neural networks that I have not included. Overall, I have learned a lot from working on this project. I have really enjoyed the challenge it brought me and skills I have developing creating this project.

References

[1] (2018, July 18). Agile Testing – Principles, methods & advantages. Retrieved from Reqtest: https://reqtest.com/testing-blog/agile-testing-principles-methods-advantages/

[2] Retrieved from http://softwaretestingfundamentals.com/black-box-testing/

[3] Retrieved from https://github.com/

[4] Retrieved from https://www.pmi.org/learning/library/underlying-theory-project-management-obsolete-8971

[5] Retrieved from https://www.python.org/

[6] Retrieved from https://www.anaconda.com/

[7] Retrieved from https://docs.scipy.org/doc/numpy-1.16.1/user/quickstart.html

[8] Retrieved from https://pillow.readthedocs.io/en/stable/

[9] Retrieved from https://matplotlib.org/

[10] Retrieved from https://docs.python.org/3/library/tk.html

[11] Retrieved from https://opencv.org/

[12] Retrieved from https://docs.python.org/3/library/urllib.html

[13] Retrieved from http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf

[14] Retrieved from https://keras.io/

[15] Retrieved from https://arxiv.org/ftp/arxiv/papers/1807/1807.11819.pdf

[16] Retrieved from https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks

[17] Retrieved from https://www.sciencedirect.com/science/

[18] Retrieved from https://goo.gl/eawgbR

[19] Retrieved from https://www.mongodb.com/

[20] Retrieved from https://www.mongodb.com/cloud/atlas

[21] Retrieved from https://docs.mongodb.com/manual/core/gridfs/

Appendix

Project Source Code Link:
https://github.com/nakster/FinalYearProject.git