Types of Neural Networks in Python



**Naqi Ahmad**

B.Sc. (Hons) in Software Development

Advised by Gerard Harrison

Contents

[Abstract 8](#_Toc7056209)

[Introduction 9](#_Toc7056210)

[Idea Raised 9](#_Toc7056211)

[The Solution 11](#_Toc7056212)

[Objectives 12](#_Toc7056213)

[Project Links 12](#_Toc7056214)

[Overview 13](#_Toc7056215)

[Methodology 13](#_Toc7056216)

[Technology Review 13](#_Toc7056217)

[System Design 13](#_Toc7056218)

[System Evaluation 13](#_Toc7056219)

[Conclusion 14](#_Toc7056220)

[Methodology 15](#_Toc7056221)

[Planning Phase 15](#_Toc7056222)

[Requirements Analysis 16](#_Toc7056223)

[Meetings 16](#_Toc7056224)

[Development 17](#_Toc7056225)

[Testing and Validation 17](#_Toc7056226)

[Problems 18](#_Toc7056227)

[Agile 18](#_Toc7056228)

[Version Control Manager 19](#_Toc7056229)

[Project Management 20](#_Toc7056230)

[Technology Review 21](#_Toc7056231)

[Python 21](#_Toc7056232)

[Important Features of Python 22](#_Toc7056233)

[Prerequisites 22](#_Toc7056234)

[Anaconda 23](#_Toc7056235)

[Python Packages 24](#_Toc7056236)

[Numpy 24](#_Toc7056237)

[Python Imaging Library 25](#_Toc7056238)

[Matplotlib 26](#_Toc7056239)

[Tkinter 27](#_Toc7056240)

[OpenCV 28](#_Toc7056241)

[URLLIB 29](#_Toc7056242)

[Sha256 31](#_Toc7056243)

[Sha256 Libraries 31](#_Toc7056244)

[GitHub 32](#_Toc7056245)

[Git Commands 32](#_Toc7056246)

[Visual Studio Code 33](#_Toc7056247)

[Keras 34](#_Toc7056248)

[Neural Networks 35](#_Toc7056249)

[Perceptron 36](#_Toc7056250)

[Convolutional Neural Networks 37](#_Toc7056251)

[K-Nearest Neighbours 37](#_Toc7056252)

[Database – MongoDB 38](#_Toc7056253)

[Prerequisites 38](#_Toc7056254)

[What is MongoDB Atlas? 39](#_Toc7056255)

[Why MongoDB Atlas? 39](#_Toc7056256)

[MongoDB Libraries 40](#_Toc7056257)

[GridFS 40](#_Toc7056258)

[System Design 42](#_Toc7056259)

[Architecture 42](#_Toc7056260)

[Perceptron 43](#_Toc7056261)

[Menu for Perceptron 43](#_Toc7056262)

[Convolutional Neural Network 47](#_Toc7056263)

[Training Convolutional Neural Network 47](#_Toc7056264)

[Testing Convolutional Neural Network 49](#_Toc7056265)

[K-Nearest Neighbour 51](#_Toc7056266)

[Training K-Nearest Neighbour 51](#_Toc7056267)

[Testing K-Nearest Neighbour 53](#_Toc7056268)

List of Figures

1.1 CNN. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.2 Multilayer Perceptron. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.3 KNN. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.1 Black Box Testing. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

2.2 Agile. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

2.3 GitHub. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.1 Python. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

3.2 Anaconda . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

3.3 PIL. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

3.4 MongoDB . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

3.5 MongoDB Atlas. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

Acknowledgements

I would like to acknowledge my supervisor Gerard Harrison for all the support.

# 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 better comprehend the different types of neural networks and the 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

In the beginning of Year four, we were given the opportunity to work in a team or individually to develop a final year project. In the past three years I have only done group projects and this time I wanted to challenge myself and try something new. Initially, the idea of the project was 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 work on the neural networks. Even though the initial idea was very interesting it became very clear that I did not have enough content covered. So, I scrapped the original plan and decided to build three different types of neural networks. The intention of this project is solely research based with a purpose of better understanding the differences between Neural Networks.

## Idea Raised

This is a project about machine learning named Python Neural Networks. 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. Toward the beginning of the project it was extremely difficult to pick what kind of neural networks to use. 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.

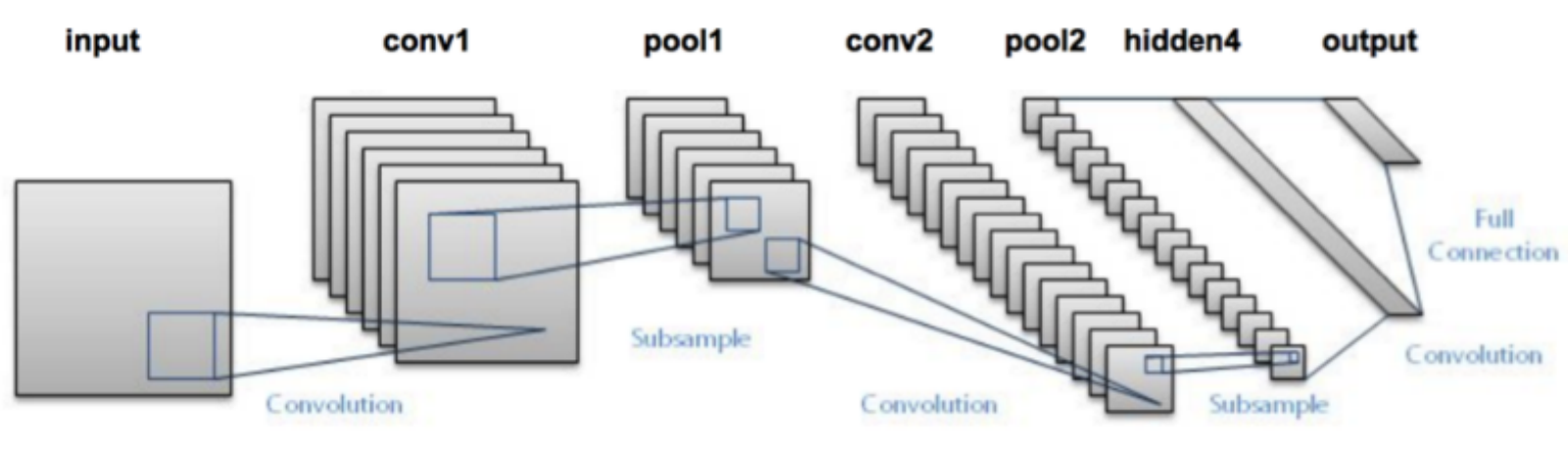


Figure 1.1: Convolutional neural network (CNN)

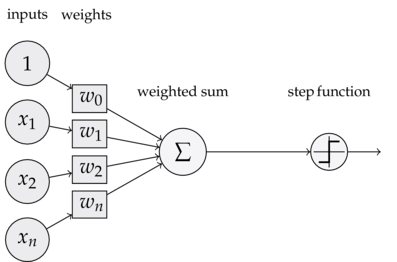


Figure 1.2: Multilayer Perceptron

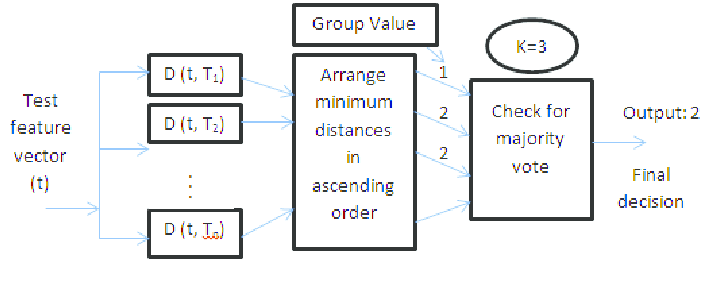


Figure 1.3: k-nearest Neighbours (KNN)

## The Solution

Neural Networks in Python 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 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 an 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. To start the project the first thing I will do 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 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/FashionNeuralNetwork.git

## Overview

As this being a final year project, it required something more intricate than anticipated, that is the reason I endeavour to undertake this project. The dissertation is structured in 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 to design and implement the solution. It will mainly discuss the Agile Methodology, version control and testing. I will cover why I chose 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 deals with how each component of the project is achieved. I will also review each component with detailed discussion on how each component contributes to the whole system in 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 discuss advantages and limitations in doing the 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 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 start coding with a poorly planned project, as this might result in goals not being achieved.

## Planning Phase

During the initial planning phase I set out the scope of my project and decided to use an Agile like approach to the research, design and implementation of my project. I investigated the different types of methodologies such as the waterfall model but at the end decided to use the Agile methodology. The reason I chose Agile over Waterfall is due to the fact waterfall model lacks flexibility. Whereas Agile offers 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 of different technologies, I decided 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. After help from my supervisor I decided to make a command line application and if I have time, I will make it into a Django application.

## Requirements Analysis

When commencing a new project determining the requirements is a key factor. These steps allow for the requirements to be analysed 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 few things. 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 overall quality and reliability of the code in a project. I used terminal command line 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.

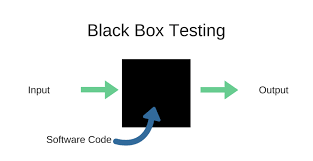


Figure 2.1: Black Box Testing

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

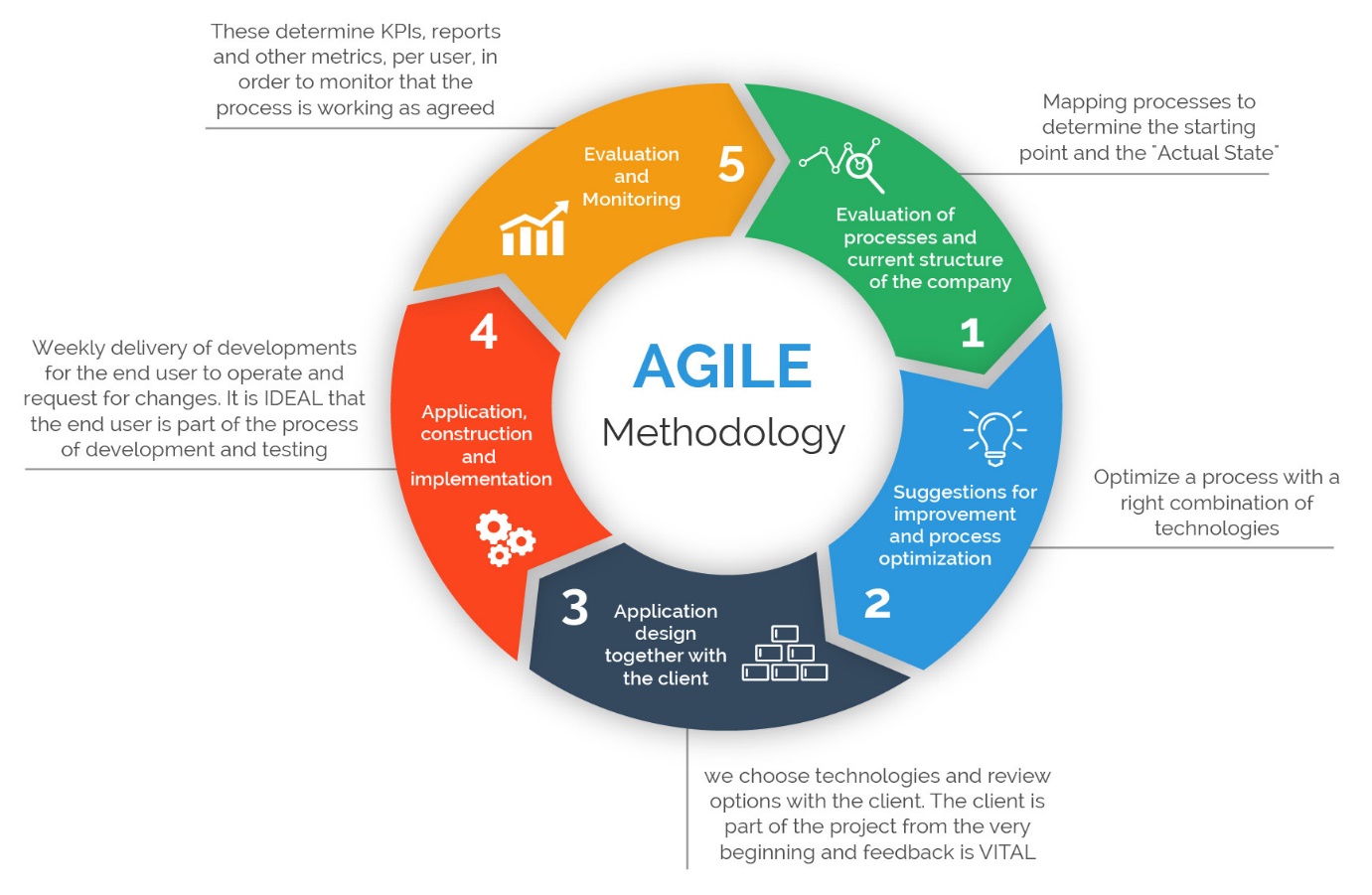


Figure 2.2: Agile

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 [5]. 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 things GitHub has is the README file. 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.

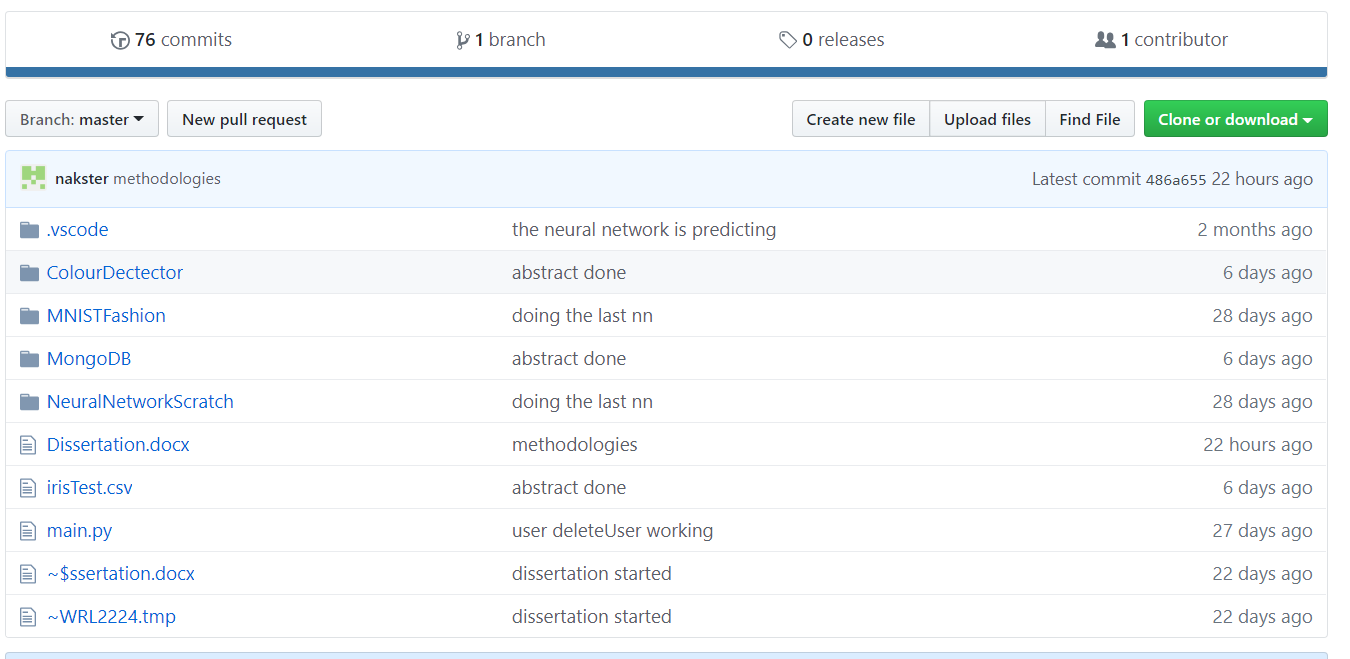


Figure 2.3: GitHub

## Project Management

Project Management 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 project management to carefully develop an efficient plan. As poor management can negatively impact a project, I put an emphasis on having a clear plan with a strong strategy to manage the lifecycle of my project. Planning before starting to code allowed me to decide what features to carry out first. This made things easier for me as a path was set 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. 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 a lot of volunteers that are constantly trying to improve Python. This is most likely one of the main reasons that Python has remained fresh and current with the newest trends. Python has a lot of libraries for almost anything 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 [4].

## 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 [5] 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 [5] 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 Python with Anaconda.

There are other 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

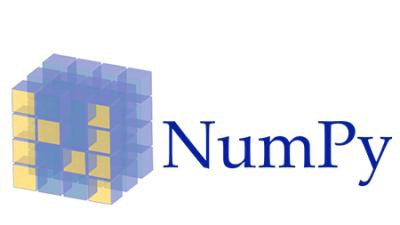


Figure 3.2: Anaconda

I am using the NumPy package 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 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 3.3: PIL

Image Module 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 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 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

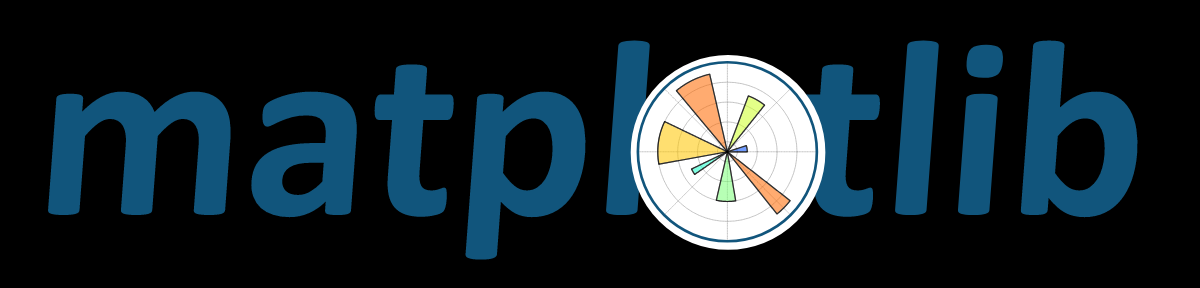


Figure 3.3: Matplotlib

Matplotlib 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 install python to be able to use the library. The first thing 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 3.4: Tkinter

Tkinter 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 mix 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 3.4: OpenCV

OpenCV 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.

*# 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 3.5: URLLIB

The urllib library 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.

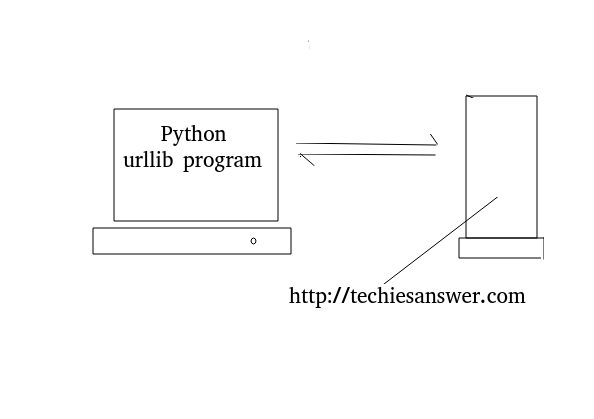


Figure 3.6: URLLIB Explained

Example of how urllib works

*# 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 3.7: Sha256

Sha256 stands for Secure Hash Algorithm, SHA-256 is a cryptographic hash function that has a digest length of 256 bits. It is a keyless hash function. What is a 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 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 3.8: 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 3.9: Visual Studio Code

There are many different types of development tools, but I prefer 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 3.10: Visual Studio Code

Keras 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 frameworks 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 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.

*# 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

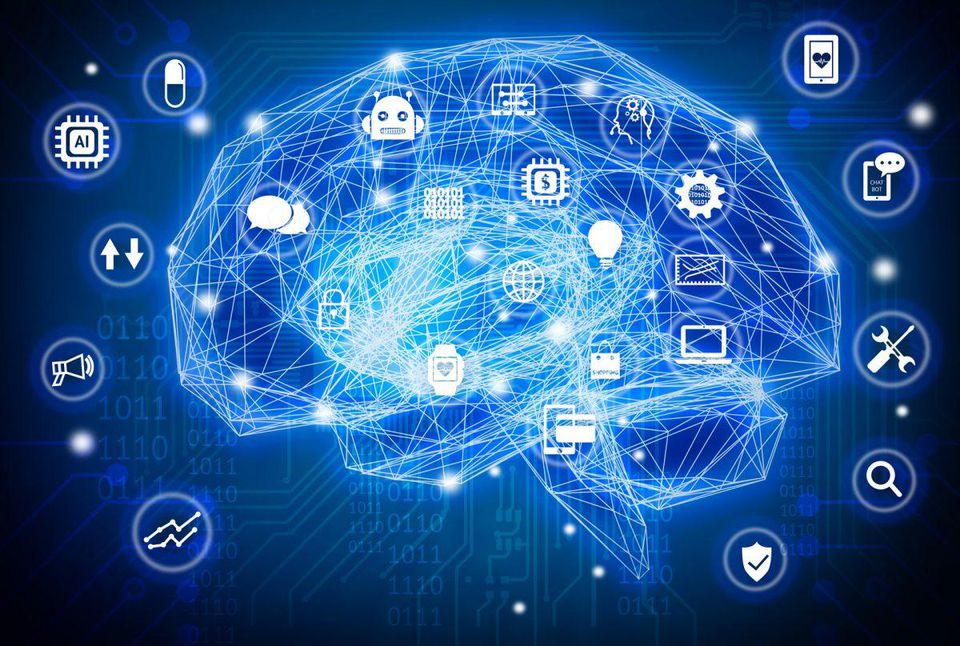


Figure 3.11: Neural Networks

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. Machine learning allows machines to make decisions based on 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

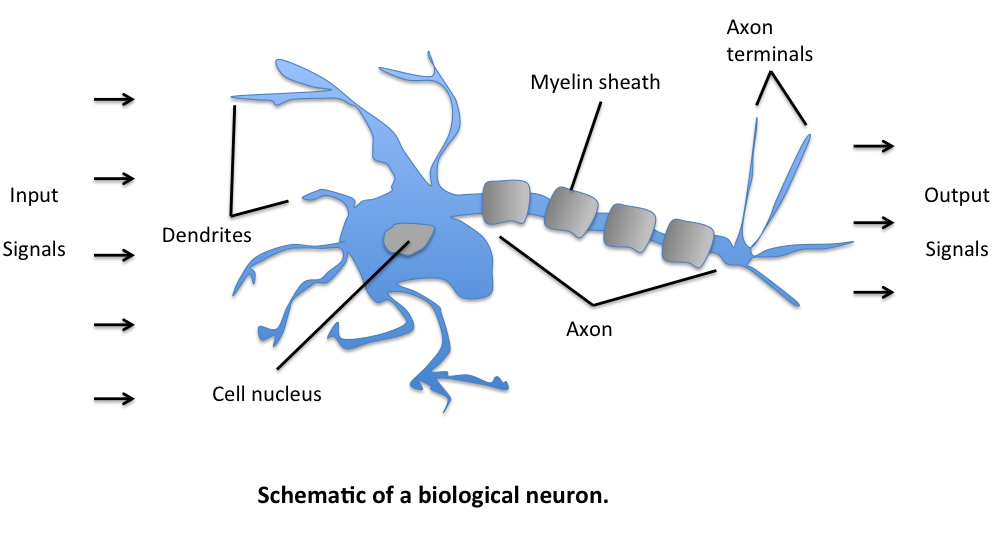


Figure 3.12: Biological Neuron

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

Back in the day perceptron used to be the bases for starting with machine learning and artificial intelligence. Therefore, it was very important for me to understand how a perceptron works. 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.

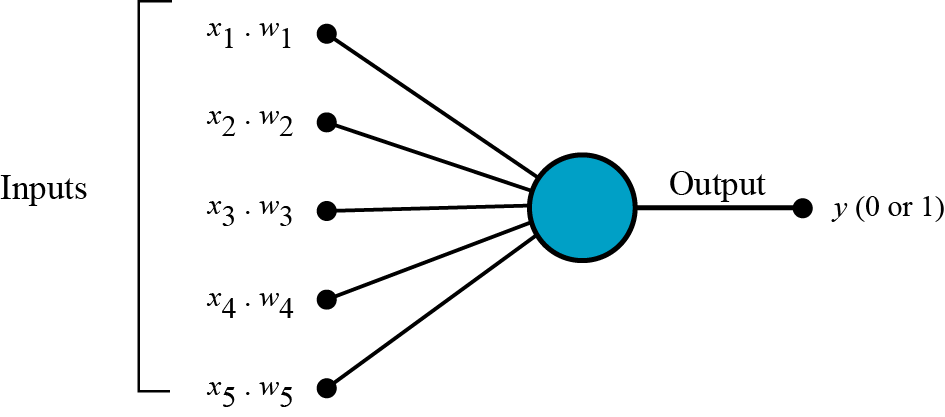


Figure 3.12: Perceptron

### Convolutional Neural Networks

Convolutional Neural Networks are made of neurons that self-enhance through learning [13]. 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 [14].

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 [15] 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 3.13: MongoDB

MongoDB [2] 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 the above 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 MongoDB was a tired-some. 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 3.14: MongoDB Atlas

MongoDB Atlas 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 have 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 to enhance my NoSQL skills and allowed me to learn a newer technology. I could have used CouchDB but chose MongoDB instead because of the easier learning curve for MongoDB.

## MongoDB Libraries

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

### GridFS



Figure 3.15: GridFS

GridFS 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. 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.

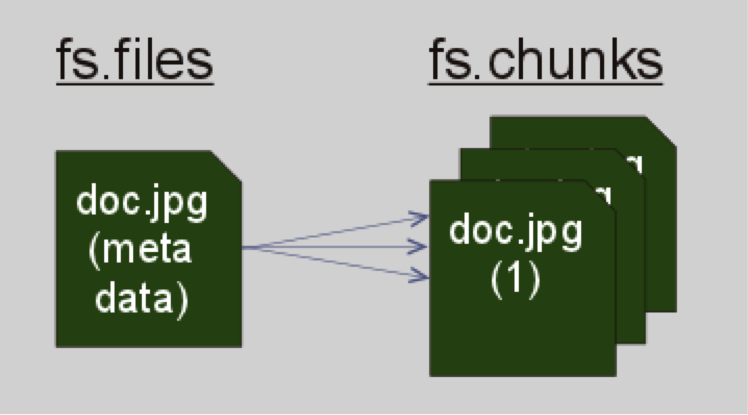


Figure 3.15: fs.files and fs.chunks

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

A screenshot of a cell phone

Description automatically generated

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 is a command-line based project. 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.

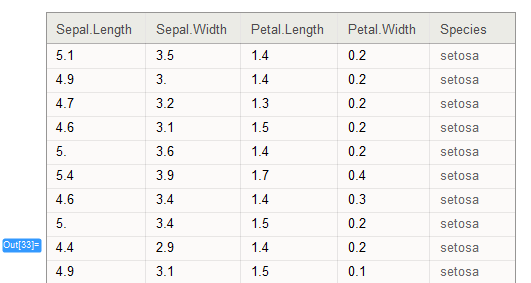


Figure 4.2: Iris Data Set

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.

A close up of a black background

Description automatically generated

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.

A picture containing sky

Description automatically generated

Figure 4.4: Iris Neural Network

When the user trains the model

A close up of a logo

Description automatically generated 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.

A close up of a black background

Description automatically generated

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.



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.

A screenshot of a cell phone

Description automatically generated

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.

A screen shot of a social media post

Description automatically generated

Figure 4.8: 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 4.8: 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