Concussion Detection Application

Darragh Lally

Anthony Moore

Michael Mulholland

B.Sc.(Hons) in Software Development

May 10, 2021

Final Year Project

Advised by: Damien Costello
Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

Chapter 1

Abstract

"Ask three neurologists if someone has a concussion and you'll get four different opinions." - Dr. Rosina Samadani.

Detecting concussion in amateur and professional sport is trusted to the experience of on site medical staff. Yet studies show that it is often very difficult to diagnose due to lack of immediate or visible symptoms. The aim of this project is to create an application implementing readily available hardware, which can be used to detect if an individual is suffering from a concussive injury. Two major benefits in the development of such an application are; an IT solution introduced into this process could help eliminate human error and, the data collected through the applications use could be leveraged to an advantage in current and future research.

Current research shows that ocular motor function tracking is a viable method when detecting concussion. Using this as a backbone to the development process, five key objectives were identified. First was to capture the eyes movement, then process, analyse, store and report the data.

By targeting these five objectives throughout the development process the resulting application shows strong evidence to suggest its viability for more comprehensive testing.

Without access to real world concussive data it is impossible to prove the applications robustness. Yet by providing simulated data, the application continues to successfully match expected results. To increase its performance there are several areas which could benefit from access to more resources.

Authors Darragh Lally, Anthony Moore and Michael Mulholland are 4th year Software Development students at Galway Mayo Institute of Technology, Dublin Road, Galway.

Chapter 2

Introduction

This project falls into the domain of experimental technology for the sports science and medical fields. We have developed an application that will determine if an individual is suffering from concussion.

Our application comes at a time when concussive injuries, particularly in Rugby, are under major scrutiny. One example of this is the legal action been taken by nine former professional players against the Rugby Football Union, Welsh Rugby Union and World Rugby [1]. They argue that these bodies failed to protect them from the risks caused by concussion.

During a discussion with our industry partner, who has experience at pro level rugby, we were informed that players can be trained to pass the current Head Injury Assessment (HIA). Often pressure from coach's and the individual themselves, to stay in play will mean this culture will continue as long as there is a means to cheat the HIA.

According to a study by the University of Limerick and the Irish Rugby Football Union, during the 18/19 rugby season, concussion was the most frequently recorded injury coming in with 11% of reports [2]. This number is some cause for concern as studies are linking degenerative brain illness's to concussion [3] [4].

Currently no IT solution is available for pitch side assessment of concussion. The use of verbal assessments is no longer viable given that players are been pressured into cheating them. The goal was to provide an affordable solution that could be adapted into the current HIA. To achieve this our intent was to develop software that will map and evaluate an injured parties oculomotor function.

With the majority of concussions not having symptoms such as loss of consciousness or neurological signs [5] it is a very difficult diagnosis to come to, therefor this application could have a major impact in the diagnosis process.

A decision was made early in the process on five key objectives that needed to be worked towards to achieve an acceptable outcome see fig 2.1 below.

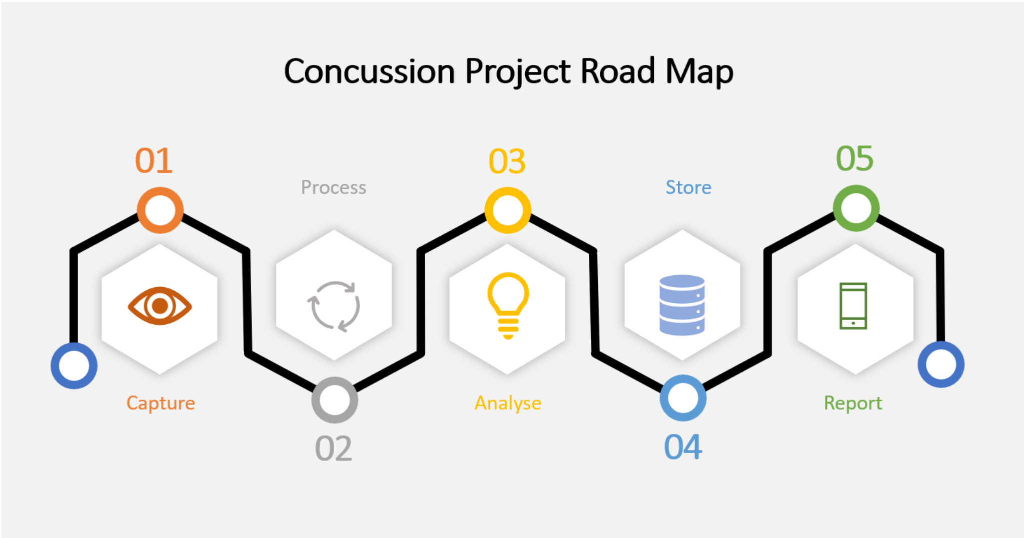

Figure 2.1: Project Objectives

- **Capture** Using a video feed, detect an individuals eyes and then map the pupils movement. This is done twice, the first capture (baseline) is taken when an individual is in a non-injured state. The second capture will occur after the individual has had a suspected concussive injury.

- **Process** Using the data captured the intent is to create graphical representations using the x,y coordinates from both eyes as points in the plane.

- **Analyse** The processed data is what will be used to make the determination. By comparing baseline data with post incident data we should see a deviation in patterns as the eyes movement will be altered while in a concussive state.

- **Store** In order for the application to be effective, the data we capture and process will need to be stored. Baseline data could be taken weeks or months before a concussive injury occurs. It would also be beneficial to store other data points such as results, incident information and bio metrics. This type of stored data could be used in further research in this field.

- **Report** As this application should be operated by a party other than the 'patient', it should provide visual feedback of the results of the assessment. This can be achieved through the use of a GUI.

GitHub

Due to the dynamic of working as part of a team, under Covid restrictions, sharing our code in an efficient manner was essential to a smooth work flow throughout the development process. We used GitHub to host our repository. Using the organisation format we were each able to work on a fork and update the one repository. Our GitHub Repository can be found through the link below:

Github.com/Concussion-Research-Project

The rest of this paper has the following structure, Chapter 3 documents our findings on the research completed which we found relevant to the domain area, Chapter 4 contains the methodology we incorporated in the development of the application, Chapter 5 holds an overview and review of the technology we used to achieve our goal, Chapter 6 has the system design, in Chapter 7 we evaluate our application and finally in Chapter 8 we close this paper with our conclusions.

Chapter 3

Research

3.1 Concussion

The definition of concussion, according to the International Consensus Conference on Concussion in Sport is ”a complex pathophysiological process affecting the brain, induced by biomechanical forces.”[6], similarly defined by the Center for Disease Control (CDC) [7]. Common symptoms been described by the Health Service Executive (HSE) as headache, nausea, confusion and loss of balance [8]. This type of injury, if untreated or mismanaged, can cause serious implications for the patient in the short and long term. It has been found that maturing brains are more susceptible to sustaining this type of injury [9]. Also, sustaining a second concussive injury takes less force and takes longer to heal. Concussion can lead to whats referred to as post-concussion syndrome, which can have a number of physical, psychological and cognitive issues that can last weeks or even months [8].

3.2 Concussion Assessment

3.2.1 Head Injury Assessment (HIA)

World Rugby have set out guidelines both for elite [10] and non-elite [11] teams. These guidelines are widely accepted and should be followed in the suspected case of a head injury or concussion.

For non-elite teams they list a number of signs and symptoms of concussion to look out for in the event of a possible concussive hit. These range from highly probable indicators such as seizures, down to the response given to a general question.

At elite level, there is a 3 stage system in place. This system is known as the Head Injury Assessment (HIA). Stage 1 also known as HIA1 consists of 4 parts;

- **Criteria 1** A list of indications calling for the immediate removal from the field.
- **Off field assessment** Made up of symptom checks, balance evaluation, medical assessment and cognitive tests.
- **Incident review** Via video recordings.
- **Clinical Evaluation**

HIA 2 is a repeat medical assessment conducted 3 hours after the injury. And finally HIA 3, which is yet further medical evaluation 48-72 hours after injury. These stages were updated in 2017 to include further cognitive tests to each stage [10].

3.2.2 Vestibular/Ocular Motor Screening (VOMS)

Another method for testing is called Vestibular/Ocular Motor Screening (VOMS) Assessment. This method calls for having baseline test to be completed in a resting state. After an injury the 5 domains of the VOMS assessment should be again tested. These 5 domains are unique and test different oculomotor functions [12].

1. **Smooth Pursuit** Test the ability of an individual following a slowly moving target.
2. **Horizontal and Vertical Saccades** Test ability of eyes to move quickly between targets.
3. **Near Point Convergence Distance (NPC)** Test ability to measure near target without double vision.
4. **Horizontal Vestibular Ocular Reflex (VOR)** Test ability to stabilise vision as the head moves.
5. **Visual Motion Sensitivity (VMS)** Test visual motion sensitivity and ability to inhibit vestibular-induced eye movement.

We intend for our application to be incorporated in HIA1, while using VOMS Smooth Pursuit domain method to aid in the detection of concussion. We shall achieve this be having an object on screen which the individual can follow in a cross motion.

3.3 Existing Technology

There are some existing technologies advertised as concussion detection systems. These systems use ocular motor screening to make a determination on the concussive state of an individual.

3.3.1 EyeGuide

Eyeguide is an American software company that specialize in health technology and bioinformatics, which was established in 2010 [13]. Their technology for concussion detection relies on a very similar method to what we hope to achieve. That is they track the ocular motor function and using that data can make a prediction on the concussive state of the individual. In fig 3.1 you can see the EyeGuides Hardware.

Figure 3.1: EyeGuide Hardware Package

EyeGuides test, tracks the eyes movement in a figure of 8 pattern. A baseline sample is taken and used to compare against results post incident. In fig 3.2 we can see tests 1 and 2, as previously mentioned the first test is a baseline example and shows a near perfect plot around the target track. Test two, in red, is layered over test one and shows significant irregularities in the plotted line.

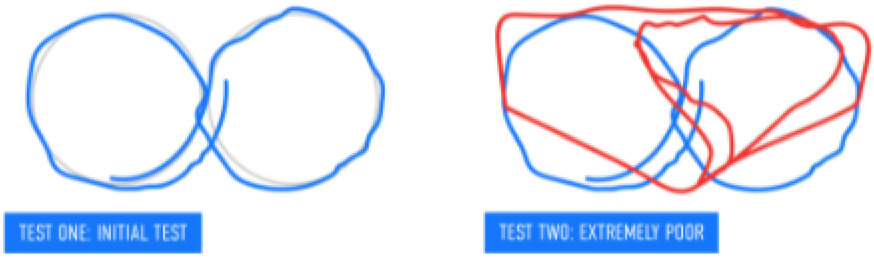

Figure 3.2: Tests 1 and 2, EyeGuide Technology

3.3.2 NeuroFlex

The company Saccade Analytics, founded in 2016, developed NeuroFlex [14] which is a VR based solution. Their technology is used as an assessment and recovery aid. Using a Fove Virtual Reality Headset they perform six types of ocular motor function tests, see below list. Using the data gathered they develop a graphical representation of an individuals recovery rate over time. Saccade Analytics also offer VR based recovery packages, which they maintain will shorten and enhance the healing process. Their set up can be seen in fig 3.3.

Figure 3.3: Fove in use, Saccade Analytics Twitter

NeuroFlex Tests:

- Smooth Pursuit
- Vestibulo-Ocular Reflex
- Saccades
- Antisaccades
- Optokinetic Nystagmus
- Spontaneous & Gaze-Evoked Nystagmus

In fig 3.4 we see an sample plot from the Smooth Pursuit test. NeuroFlex technology randomly generates the plot to avoid it been predictable and therefor cheat prone. The graph shows a blue line as the output to be followed and orange represents the result of the tracking.

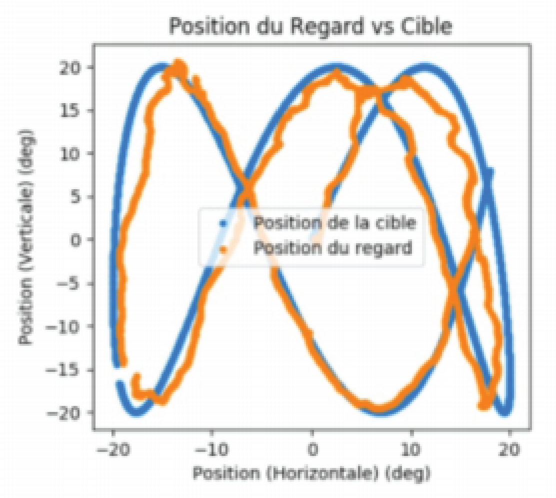

Figure 3.4: Example output from Smooth Pursuit Test, NeuroFlex

Chapter 4

Methodology

The first task was to understand the domain area. Researching concussion detection techniques (section 3.2) and current systems (section 3.3) led to an understanding of possible solutions. By establishing the available resources, the most viable solution to implement would be an application that would track ocular motor function and simulate a concussion detection technique. To create this application five main objectives were identified, see figure 2.1.

The Software Development Life Cycle (SDLC) that was implemented in the development of this application was the Rapid Application Development (RAD) Model. This would allow for the iterative development style that was needed in order to achieve an acceptable result.

Python was quickly established as the programming language of choice. This decision was based on several factors, outlined in section 5.1. The Ubuntu OS was chosen due to Linux's open-source nature. This allowed for the installation of older versions of python libraries needed in the detection process. MongoDB was incorporated because of its scalable nature, in the event of rapid growth of the application this advantage could reduce the risk of data loss.

4.1 Capture Data

This task involves using a devices webcam to track the eyes movement while following a pattern on screen. Solving this began with developing an application that would detect eyes. Using images classifiers, two versions of eye tracking were implemented.

Implementation one used two OpenCV models, the first to detect and extract the face from the input creating a frame, the second to detect and extract the eyes from the face frame. Implementation two, uses the dlibs shape predictor model 'shape_68'. This is the version used in the final product as it worked faster and smoother than iteration one.

Once satisfied with the tracking method the next step was to generate data. By modifying the tracking script, it was possible to output the X,Y coordinates of each eye. These values would be vital for the following objectives to be met.

Based on research findings, subsection 3.2.2, VOMS Smooth Pursuit detection method was incorporated. This was accomplished by revising the tracking script once again. A moving target was added to the display, when followed the eyes position would change therefor changing the X,Y coordinates recorded each frame.

4.2 Process Data

By leveraging the extracted X,Y coordinates and writing them to file it was then possible to create graphical representations of the eyes movement. These graphs would provided the data necessary to perform analyses on.

The use of a visualisation library was important in the creation of these graphs, as the project was developed in python, Mathplotlib was chosen for this task. This decision was also made with the report objective in mind, see section 4.5.

Using Mathplotlib, a number of representations were created. It was found that the scatter plot and line graph models were the most intuitive to understand.

With the ability to process the captured data and compile graphs it was time to consider how to analyse it.

4.3 Analyse Data

The research showed that a concussed individual's ocular motor function would be effected. This would then be mirrored in the graphs created in the processing phase. Therefor, by comparing graphs from pre and post incident a deviation would be apparent. There were two solution developed for this process, the first method employed was a simple standard deviation calculation, mid-way through development it was decided to change this method to a trained AI model that would make the prediction.

While the standard deviation method showed some potential as an analyse technique it had its flaws. During testing both the baseline and post incident data would show massive deviation if the starting positions were not the same. This led to the only major change implemented in the development cycle.

Moving to an AI model had a number of advantages. Training the model of simulated data, both good and bad examples, means that no matter what starting position the test was taken, the model would be enable to interpret and decide if the test fell into the good or bad data-set. This model can then be better trained with the addition of further test data. While this change increased the scope of the project it benefits outweighed negatives.

4.4 Store Data

The ability to store data remotely was an important factor as the client would need to be a portable device. An additional requirement was being able to access all of the collective test data for further research.

In order to analyse data we would need two tests. The first test would be taken under ideal circumstances, that is the individual would be in a rested, non-injured state. This test may be performed days, weeks or months in advance of a concussive injury occurring. Therefor storage was crucial.

Another consideration was, the data collected through the use of this application could be used to create data-sets for future research and development in this domain.

The solution incorporated was MongoDB. This framework was chosen because of its free tier storage and its ability to scale in parallel with the applications growth.

In the early stage of development the database schema was minimal, only holding an ID value and the .csv file containing the X,Y coordinates. The schema was scaled up a number of times during development, to include tuples for other relevant data points such as incident description and incident activity.

Because MongoDB is user friendly, the addition of further data points is a simple process.

4.5 Report Data

Data visualisation would be important to give an understanding of the process involved and also to provide results from testing to the operator. The Python plugin PyQt was used to create a GUI to output not only the aforementioned, but to allow an operator to input details important to specific tests while interacting with the application. This tied the database development to the GUI development, if one was modified the changes would need to be reflected in the other.

4.6 Validation

To validate this application, real world concussion data is needed. The intent at the beginning of development was to use our application to gather data from local rugby and GAA clubs, including GMIT teams. This was to be facilitated by Ed Daly. Unfortunately, this data was impossible to collect due to public health restrictions that were ongoing throughout both cycles of development.

Simulated data was then used in place of real data. Both good and bad examples were recorded by the development team, creating a simulated dataset.

4.7 Development Cycle 1

October 2020 - December 2020.

This cycle began with been presented with the domain problem. There were consistent weekly development and supervisor meetings during this cycle. These meetings were generally held in a consecutive order on the same day each week. During this cycle there were a number of milestones achieved, listed below.

- Project approval.
- Researched domain areas.
- Began tracking application.
- Began database implementation.
- Implemented deviation metrics.
- Began dissertation.
- Mid-development interview board.

4.8 Development Cycle 2

January 2021 - May 2021.

By the beginning of this stage the application was well under development. The tracking application was successfully integrated with the database. During this phase it was decided to change the applications analyse method, moving from a deviation metric to an AI model which was trained on sample data. Supervisor meetings continued on a weekly basis but development time was increased steadily each week leading up to the submission date. Below is a list of milestones reached.

- AI trained model introduced.
- Sample data collected.
- AI model trained.
- Database enhanced.
- Complete architecture integration.
- Submission.

Chapter 5

Technology Review

5.1 Python

Python is an open-source programming language designed by Guido van Rossum in the late 1980's and released in 1991 [15]. It is an interpreted language that supports object-oriented programming and dynamic-semantics. It can leverage high-level data structures to easily build applications without the developer needing to declare data types being returned by a method or function. This makes it a very attractive programming language for Rapid Application and Prototype Development.

The language used by Python has a very simple syntax that is easy to understand with a strong emphasis on readability. Most of the redundant boiler plate code associated with other high-level programming languages is removed, along with other common syntax features such as curly braces to indicate the body of a method or semi-colons to finish declaring a statement.

The result is a very minimalist programming language that almost looks like pseudo code at first glance. This enables developers to be able to quickly read a code base and reduce the overall time required to maintain the program. Python is extremely popular with developers due to the increased productivity it offers and the vast collection of extensible packages available [16]. The code samples below give an example of a simple python program.

```
        # define class
        class Person:
        # constructor
        def __init__(self, name, age):
            self.name = name
            self.age = age
```

```python

    # member functions
    def hello(self):
        print("Hello my name is " + self.name)

        # convert the num into string
        my_age = str(self.age)
        print("My age is " + my_age)

# initialise object
p1 = Person("Peter", 25)

# call function
p1.hello()
```

Advantages:

- **Open Source** - It is open source, meaning its is completely free. This low barrier of entry is ideal for projects, small businesses and research groups.
- **Ease of use** - Python focuses on code readability. The language is versatile, neat, easy to use and learn, readable, and well-structured.
- **Vast Libraries** - There are libraries and packages to support the development of a wide range applications types such as: web development, computer vision, game design and machine learning.

Disadvantages:

- **Performance** - Python is an interpreted language, so it does have slightly slower performance compared to compiled languages such as C or C++.
- **Not Native to Mobile** - Python is not officially supported by Android or iOS. However, Python can still be used on mobile but it does require cross platform libraries such as Kivy[17] to handle GUI and business logic and Buildozer[18] to generate the APK along with the applications dependencies.

5.2 Numpy

Numpy is a Python library that converts multidimensional arrays into an object and offers a comprehensive selection of algorithms for high performance array operations such as: sorting, shape manipulation, transforms and statistics [19].

The fundamental data structure in Numpy is an array object. This abstracts and encapsulates multidimensional arrays of similar data types. Many of the underlying operations within Numpy are performed in compiled code for increased performance.

This helps to reduce the amount of overhead required using a python interpreter and manipulating the native object contained in a collection.

Numpy uses a form of array programming where operations are applied to whole arrays instead of individual elements. This gives the appearance that it functions without loops or index references, when in-fact these operations take place in the background on pre-compiled C code.

This allows for the development of python applications that are both easier to read and maintain, while offering the performance advantages that come with using compiled C code.

Numpy example: Adding two arrays of numbers together

```
import numpy as np
x = [1, 2, 3, 4, 5]
y = [10, 20, 30, 40, 50]
nums1 = np.array(x)
nums2 = np.array(y)
result = nums1 + nums2
print(result)
>[11 22 33 44 55]
```

Numpy is highly optimized for the manipulation of array data in the field of scientific computing. Popular Machine Learning packages such as TensorFlow and Scikit-learn use numpy arrays as inputs to represent complex data structures.

Advantages:

- **Performance** - Numpy is significantly faster than base Python due to the use of C extensions.

- **Flexability** - Packages such as Keras, scikit-learn and openCV make extensive use of numpy. So it provides a universal data structure between various libraries.
- **Features** - It come with a wide range of algorithms for sorting, reshaping and filtering data. While also enabling mathematical functions such as standard deviation.

Disadvantages:

- **Memory** - Data is stored in contiguous memory locations, so insert and delete functions are computationally expensive due to shifting.

5.3 Amazon Web Services (AWS)

AWS are market leaders in the field of cloud computing. They offer a highly reliable, scalable and low-cost platform in the cloud [20].

They provide rapid access to low-cost IT resources. This allows the development effort to save on investing and maintaining extensive hardware. Rather, they can simply decide on the amount of computing resources needed to meet the demands of the application.

Cloud computing provides a simple and convenient way for applications to access storage, databases, servers, and a broad set of application services over the Internet fig 5.1.

Figure 5.1: Amazon Web Services

Cloud computing has four main characteristics: [21]

- **Elasticity** - Services are provided on demand when the customer needs them and they will be decremented if the customer does not need them.
- **Self-service** - Customers can request additional resources directly.
- **Application Programming Interfaces** - API's allow for the efficient deployment of online resources and communication to the application via HTTP/HTTPS.
- **Billing based service usage** - The operating cost will only be for the resources that are consumed. It operates with a per-as-you-go style of pricing.

Ultimately it was decided to choose AWS over the services provided by Google Cloud Platform and Microsoft Azure. This was due to AWS's high reliability, scalability and cost-effectiveness.

Advantages:

- **Cost Effective** - Only pay when computing resources are consumed, instead of investing heavily in data centers and servers. The free tier is very generous and is well suited for small scale application development.
- **Scalability** - AWS can achieve high economies of scale. This is due to a vast client base aggregated in the cloud. The net effect is lowing the pay-as-you-go prices and reducing IT labor costs.
- **Accessibility** - Grants access to a highly distributed, fully-featured platform at a fraction of the cost compared to traditional infrastructure.
- **Modular Upgrades** - It is easy to add additional CPU, Memory or Storage upgrades to the virtual machine as required.
- **Documentation** - The AWS documentation is excellent and well constructed. They also provide videos, courses and tutorials to assist with getting started.

Disadvantages:

- **Amazon's EC2 Limits** - Amazon Web Service resources are limited by region. The customers location will determine how many resources that they will have access to.

5.4 MongoDB

MongoDB is an open source, NoSQL, documented-oriented database that is very powerful, flexible, and scalable. It is a non-relational database but it implements many features of relational databases, such as secondary indexing, sorting and range queries. Instead of storing the data in tables with columns and rows, MongoDB stores the data in documents. The documents are serialized naturally as Javascript Object Notation (JSON) objects, and are stored internally using a binary encoding of JSON called BSON [22]. These documents are gathered together in collections. The databases can store one or more collections of documents.

The use of JSON-like objects makes the data very easy to access and manipulate. We can access the data from MongoDb using CRUD operations. Which stands for create, read, update and delete.

```
{
    "person": {
        "name": "michael"
    }
}
```

5.4.1 Create Operation

The Create or insert operations allows us to add a new document/documents to a collection. If the collection does not exist, this operation will create the collection. We used the following to preform a create operation to insert a single document into the collection see 5.2.

```
db.collection.insert()
db.collection.insertMany()
```

1. It is used to insert a single document in the collection.
2. It is used to insert multiple documents in the collection.

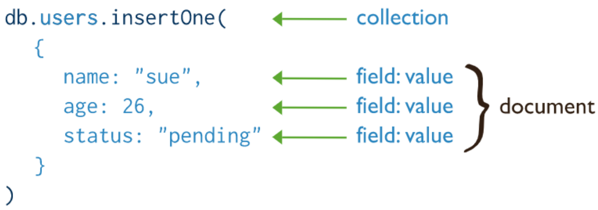

Figure 5.2: Create Operation Example

5.4.2 Read Operation

The Read operation returns the document from the collection. We used this operation to query a collection for a document. We used the following to preform a read operation to retrieve a document from the collection see 5.3.

```
db.collection.find_one()
```

1. It is used to retrieve documents from the collection.

Figure 5.3: Read Operation Example

5.4.3 Update Operation

The Update operation is used to modify the documents that already exist in the collection. We used the following to preform update operations on the documents in the collection see 5.4.

```
db.collection.update_one()
db.collection.updateMany()
db.collection.replaceOne()
```

1. It is used to update a single document in the collection that satisfy the given criteria.

2. It is used to update multiple documents in the collection that satisfy the given criteria.

3. It is used to replace single document in the collection that satisfy the given criteria.

Figure 5.4: Update Operation Example

5.4.4 Delete Operation

The Delete operation removes selected documents from the collection. We used the following to preform delete documents from the collection see 5.5.

```
db.collection.delete_one()
db.collection.deleteMany()
```

1. It is used to delete a single document from the collection that satisfy the given criteria.

2. It is used to delete multiple documents from the collection that satisfy the given criteria.

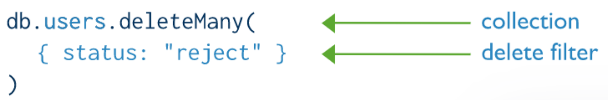

Figure 5.5: Delete Operation Example

Advantages:

- **Flexible** - MongoDB is a schema-less database. That means we can have any type of data in a separate document. This thing gives us flexibility and a freedom to store data of different types.

- **Sharding** - The ability to scale the cluster linearly by adding more ma-chines which allows us to store a larger data.
- **High Availability** - If supports replication facility called, replica set. The replica set is a group of servers that maintains the dataset. This provides automatic redundancy, increased data availability and fail over.
- **Rich query language** - It provides us with the features of RDBMS. Some of the features include dynamic queries, sorting, secondary indexes, rich updates and easy aggregation.
- **Ease of use** - It is very easy to install, use, maintain and configure.
- **High performance** - It provides high performance data persistence. It is easy to access documents by indexing and this supports faster queries.
- **Support for multiple storage engines** - It supports multiple storage engines and supports pluggable storage engine API that allows third party to develop storage engine for MongoDB.

Disadvantages:

- **Joins not supported** - There are no joins in MongoDB like there is in the relational model. If someone needs to use the join functionality they must add it to the code level manually. This could result in reduced flexibility and slow execution.
- **No transactions** - Operations are not treated as transactions in MongoDB. To ensure transaction we have to check manually for completion of transaction and from that choose to commit or rollback.
- **High Memory Usage** - Due to no functionality of joins, there is data redundancy. This results in increasing unnecessary usage of memory.
- **Limited Data Size** - The document size can be no more than 16MB.

5.5 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library [23]. OpenCV was developed as a gateway for commercial products to utilize computer vision based applications through the use of a common infrastructure. As a BSD-licensed product, OpenCV makes it convenient for businesses, developers and data scientists to leverage and modify the code base at will [24].

The OpenCV library contains over 2500 algorithms for computer vision and machine learning. These algorithms can be used to recognise peoples faces, classify 3D objects, follow eye movements, identify hand gestures and body movement and establish markers to overlay with augmented reality see 5.6. OpenCV has a user community of more than 47 thousand people and an estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies [25].

Figure 5.6: OpenCV detecting people in a crowd

OpenCV is natively written in C++, but also includes Python and Java interfaces. It is supported on most conventional operating systems such as Windows, Linux, Android and Mac OS. OpenCV specialises in real-time computer vision applications.

The OpenCV library is utilised by several major company's such as Google, Microsoft, Intel, IBM, Sony and Toyota. It has been successfully deployed in a number of real world applications such as helping robots navigate and pick up objects [26], video-based drowning detection for swimming pools [27] and automatic license plate recognition [28].

OpenCV example: Conversion from color to greyscale

```
img = cv.imread('image.jpg')
_grey = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
```

OpenCV example: Visualizing images

```
img = cv.imread('image.jpg')
cv.namedWindow('image', cv.WINDOW_AUTOSIZE)
cv.imshow('image', img)
cv.waitKey()
```

Advantages:

- **Open Source** - It is open source, meaning its is completely free. This low barrier of entry is ideal for projects, small businesses and research groups.
- **High Speed** - The OpenCV library is written in C++. As a compiled language it performs very well in terms of computational speed.
- **Low Memory** - Its algorithms are optimised to be highly efficient with memory usage. Allowing it to be used on low-end hardware.
- **Extensive Community** - With a vast network ranging from: students, enthusiasts, researchers and professional developers. It is often easy to fall on the wisdom of others to trouble shoot problems.
- **Efficient Solution** - It provides efficient algorithms that are optimised to process real-time programs. Moreover, it has been engineered to take advantage of hardware acceleration and multi-core systems.

Disadvantages:

- **Complexity** - OpenCV is not as user friendly when compared to other computer vision based solutions such as MATLAB. There is a steep learning curve that will require a fair amount of self learning to overcome.
- **Installation** - It can be tricky to setup and install. Careful attention needs to be given to the various packages and dependencies it requires. Incorrect package versions can lead to conflicts.

5.6 Matplotlib

Matplotlib is a Python library for making 2D plots of arrays. It is written in Python and makes heavy use of NumPy and other code extensions for the delivery efficient performance on even large arrays. Its was designed to allow users to create simple plots with just a few commands [29]. It provides users with an object-oriented approach for embedding plots and graphs into applications see 5.7. It can take advantage of several GUI tool-kits suck as: Thinker, wxPython and Qt.

Matplotlib Example: Plotting Data

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)

fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='time (s)', ylabel='voltage (mV)',
       title='About as simple as it gets, folks')
ax.grid()

fig.savefig("test.png")
plt.show()
```

Matplotlib has many real world applications. It can be used to generate post-script files to a printer or publisher. It can be deployed on a web application to dynamically embed PNG outputs from a set of sample inputs. It is commonly used as a visual aid by data scientists and financial institutions to help reveal “gold nuggets” buried in raw data.

Advantages:

- **Multi-Platform** - It Works seamlessly with many operating systems including: Windows, Linux and Mac OS and supports several graphics back-ends.
- **Performance** - Matplotlib is built on the NumPy and PySide framework. Giving it high speed and efficient memory usage.
- **Flexibility** - The library possesses high-quality graphics and plots to print and view for a range of graphs such as histograms, bar charts, pie charts, scatter plots and heat maps.

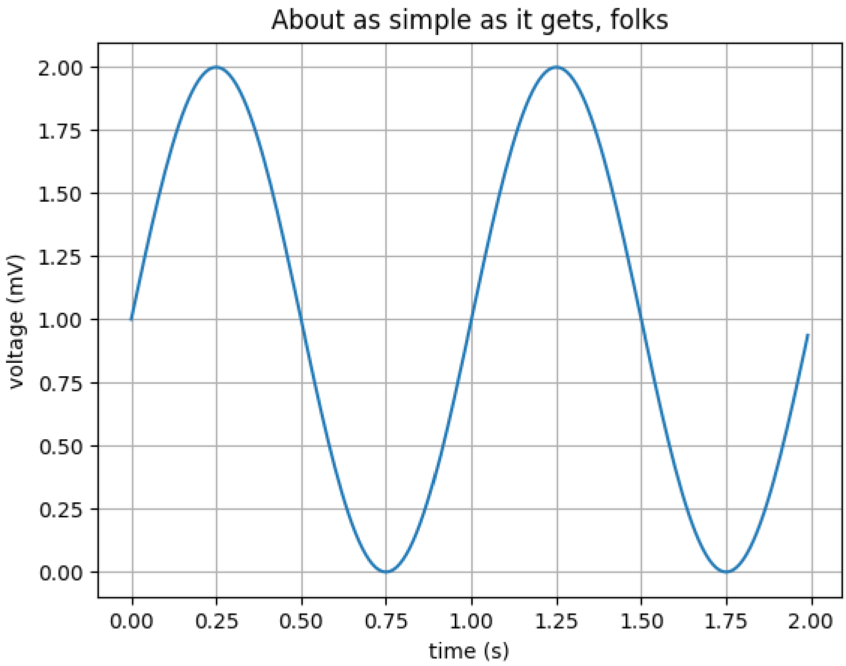

Figure 5.7: Matplotlib Visualisation of Data

- **Customisation** - Users have full control over graph or plot styles such as line properties, thoughts, and access properties.
- **Community** - It has large community support and cross-platform support as it is an open source tool.

Disadvantages:

- **Package Dependencies** - It is heavily reliant on other packages, in particular to NumPy.
- **Language Dependencies** - It only works for Python, so it is extremely difficult to implement into other languages.

5.7 dlib

Dlib is a cross-platform software library written in the C++ programming language [30]. It is currently operating as an open-source project released under a Boost Software License. Dlib is used extensively in both academic and industrial environments in a vast array of fields including: robotics, machine learning, image classification and data mining.

The particular use case for the dlibs library in our project was to leverage the pre-trained facial landmark detector 5.8. This is used to help estimate the location of 68 X,Y coordinates that map to facial structures on the face such as: left eye, right eye, mouth, nose etc.

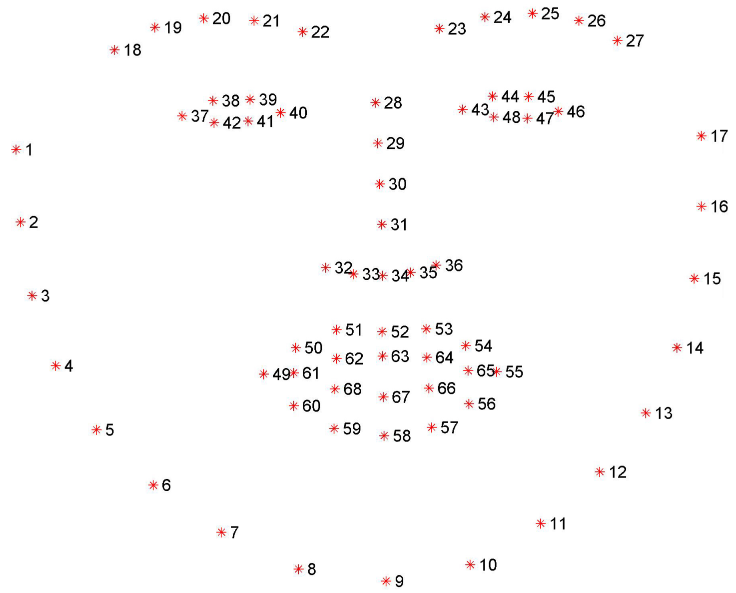

Figure 5.8: The indexes of the 68 coordinates on the facial model

dlib Example: Using dlib pre-trained facial landmark detector

```
import cv2
import dlib
import numpy as np

# facial detector and predictor
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor('shape_68.dat')

# video input
cap = cv2.VideoCapture(0)

while(True):
    # get image from video
    ret, img = cap.read()

    # greyscale the image
```

```
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        # detect objects
        rects = detector(gray, 1)
        for rect in rects:

            # predict facial structures
            shape = predictor(gray, rect)

            # rest of image processing...
```

Advantages:

- **Robustness** - Dlib when used in combination with OpenCV, can handle bad and inconsistent lighting and various facial positions such as tilted or rotated faces.
- **Performance** - Dlib is superior to the Haar cascade classifier in terms of speed, and accuracy.
- **Training** - The training is done using a sliding sub-window on the image so no sub-sampling or parameter manipulation is required like it is in the Haar classifier. Allowing for quicker training times with less data.

Disadvantages:

- **Minimum Size** - Does not detect small faces as it is trained for minimum face size of 80×80. Thus, you need to make sure that the face size should be more than that in your application.

5.8 Scikit-learn

Scikit-learn is an open source machine learning library developed for the Python programming language [31]. It provides a powerful toolkit within the programming environment, yet is still easy to use for non-experts in machine learning. This allows it to be used for a variety of projects ranging from scientific research to application development.

It features a wide array of tools to empower users including: regression, classification, gradient boosting, random forest and k-means clustering see

5.9. Its popularity can be quantified by the myriad of citations within scientific publications and impressive results in various machine learning challenges such as; Diabetic Retinopathy Detection [32] and Malignant Tumor Detection [33].

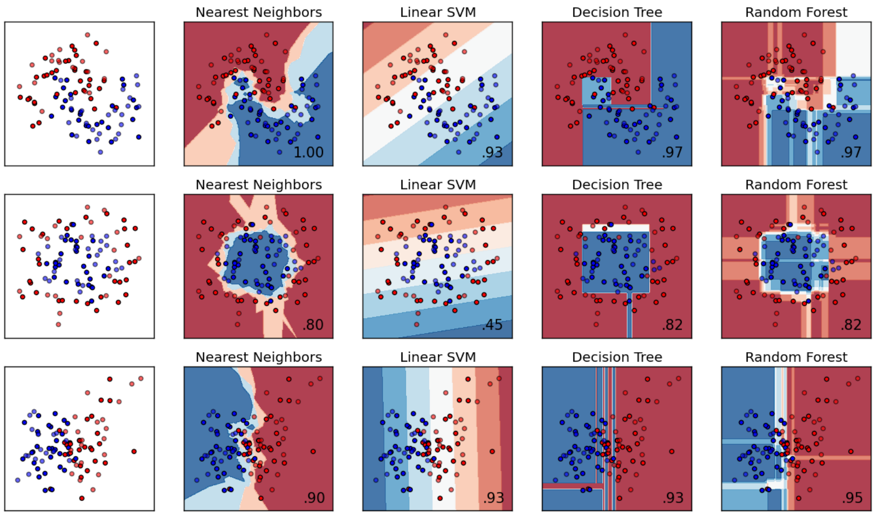

Figure 5.9: Scikit-learn classifier examples

Typically, in machine learning tasks the data is modeled as a set of variables. The aim being to map the input variables to a set of output variables. A common approach to this is to use a pair of matrices. One to represent the inputs and another to represent the outputs. Each row of the matrices equates to one sample of the data-set and each column corresponds to one variable of the problem.

In scikit-learn the representation of data is as close as possible to the original matrix representation. The data-sets are encoded as either NumPy multidimensional arrays for dense data or SciPy sparse matrices for sparse data. This not only improves the overall performance through the use of the high Performance algorithms used by NumPy and SciPy, but also frees the developer from the burden of having to format the inputs and outputs themselves.

Advantages:

- **Open Source** - It is open source, meaning its is completely free. This low barrier of entry is ideal for projects, small businesses and research groups.

- **User Friendly** - It is very easy to setup and and configure, so you can spend less time learning a complex framework and more time developing the application.

- **Versatility** - The scikit-learn library is wide enough to cover almost any machine learning challenge. It can do anything from detecting tumors, identifying market trends, feature extraction and data normalization
- **Community** - Scikit-learn is backed and updated by numerous authors, contributors and a vast international online community.
- **Well Documented** - The scikit-learn website provides elaborate API documentation that is both very well explained and easy to understand with several working examples. It allows users to easily integrate the algorithms within their applications.

Disadvantages:

- **Learning Outcomes** - Scikit-learn uses a convenient abstraction to segregate the user from much of the complexity of the underlying algorithms. This could result in developers not fully understanding the core fundamentals of how the application works under the hood.

5.9 Keras

Keras is an AI deep learning API written in the Python programming language [34]. It is built on-top of the machine learning platforms TensorFlow [35] as its foundation. It was developed to enable users to rapidly develop complex deep learning models to help facilitate research and application development.

Keras is the high-level API of TensorFlow: a powerful and user friendly interface for solving complex machine learning problems, with a strong emphasis on deep learning. It provides elegant abstractions and flexiable building blocks for the development and deployment of applications utilising cutting edge machine learning techniques.

Keras Example: Simple Nerural Network

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# initialise sequential model
model = Sequential()
```

```python

    # add 2 dense layers
    model.add(Dense(units=64, activation='relu'))
    model.add(Dense(units=10, activation='softmax'))

    # configure learning process
    model.compile(loss='categorical_crossentropy',
                  optimizer='sgd',
                  metrics=['accuracy'])

    # configure the optimizer
    model.compile(loss=keras.losses.categorical_crossentropy,
        optimizer=keras.optimizers.SGD(learning_rate=0.01,
        momentum=0.9, nesterov=True))

    #  iterate on training data
    model.fit(x_train, y_train, epochs=5, batch_size=32)

    # evaluate the model
    loss_and_metrics = model.evaluate(x_test, y_test,
        batch_size=128)

    # generate predictions
    results = model.predict(x_test, batch_size=128)

    # show predictions
    print(results)
```

It enables developers, engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow. Keras can run on a wide range of devices from single CPU laptops, up to industry scale clusters of high-end GPU's. Trained Keras models can also be deployed to operate in web browsers and mobile devices.

Advantages:

- **User Friendly** - Keras is simple to use and get started with. It has a user-friendly API that allows users to quickly create neural network models. It is excellent for implementing deep learning algorithms into applications.
- **Well Documented** - Keras is a prime example of how to do user documentation correctly. It gradually introduces you to each function

in an organized and sequential manner. It also provides quality working examples to help explain the concepts being discussed.

- **Community** - Keras also has great community support. There are lots of community code repositories on various open-source platforms. Many developers and Data Science enthusiasts prefer Keras for competing in Data Science challenges. Many of the researchers publish their code and tutorials to the general public.

- **Multiple GPU Support** - Keras allows for model training on a single GPU or to leverage the power of multiple GPUs. It provides built-in support for data parallelism. This makes it an excellent choice for deep learning in fields such as big-data or image classification.

Disadvantages:

- **Debugging** - Keras can sometimes generate low-level backend errors that may be difficult to troubleshoot. These errors may appear as a result of trying to perform some operations that Keras was not designed for. As Keras is using TensorFlow to handle the back-end processing, it can be challenging to figure out what exactly is causing the problem in some cases.

5.10 PyQt5

PyQt5 is a set of Python bindings that allows us the ability to use The Qt Company's Qt application framework. The bindings are a set of Python modules and contain over one thousand classes.

The development toolkit which is a set of cross-platform C++ libraries, contains everything you need in order to build applications on Windows, Linux, MacOS, Android and embedded systems. Using Python with Qt, we get the advantages of the C++ toolkit for making applications, such as User Interfaces (UI) and Graphical User Interfaces (GUI) widgets [36].

Graphical User Interfaces (GUI) programming with Qt is created around the concept of signals and slots for establishing communication with objects. This gives us flexibility when dealing with GUI events and results in a smoother code base.

PyQt5 Example: - Basic operation

```
button.clicked.connect(self.slot_method)
```

The button click (signal) is connected to the action (slot). In this example, the slot-method will be called if the signal emits. This principle of connecting slots methods or function to a widget, applies to all widgets,

```
widget.signal.connect(slot_method))
```

or we can explicitly define the signal:

```
QtCore.QObject.connect(widget, QtCore.SIGNAL("
    signalname"), slot_function)
```

Advantages:

- **Flexibility:** - Not just a simple GUI framework, Qt it makes use of a wide array of native platform APIs for the purpose of networking, database creation, and many more. This means that it can be used across several platforms and they should function universally.
- **Stability:** - Qt is a very stable framework and has very few issues when deployed on large-scale applications.
- **Variety:** - Huge selection of widgets, such as menus or buttons and many more. These are all designed with a basic appearance and can be used across all supported platforms.
- **Graphic Design:** - Qt also has an in App-Designer. It allows for the generation Python code from the Designer.

Disadvantages

- **Documentation** - There is a lack of Python-specific documentation for classes in PyQt5.
- **Complexity** - There is a quite steep learning curve once you get beyond basic functionality.

5.11 Image Classifiers

In order to track the eyes movement image classification is a critical step. These classifiers are machine learning based data-sets. There were two methods tested during development. Iteration one made use of two OpenCV, see 5.5, Haar Cascade classifiers [37][38], iteration two uses a dlibs, see 5.7, classifier [39].

5.11.1 Haar Cascade

Haar Cascade classifiers are an effective object detection approach proposed by Paul Viola and Michael Jones [40]. In this iteration two of these classifiers were used. The first (haarcascade_frontalface_default.xml) was used to isolate the face from an image frame, creating a face frame. The second (haarcascade_eye.xml) isolated the eyes from the face frame creating a eye frame. It is this eye frame that is used to detect the pupil.

5.11.2 Shape Predictor

Iteration two employs the dlibs model shape_predictor_68_face_landmarks.dat. see section 5.7 and figure 5.8. More detail on this predictor can be found at the previous references. As this method was employed in the production version of the application it is covered in depth.

Chapter 6

System Design

6.1 Overview

The application was designed using a client-server architecture [41]. This decision was made to facilitate multiple users that can share the same database and contribute to the growth of the data set. The data set can then be used for further research into the subject domain such as: case studies, statistics, recovery rates and machine learning. Below is a high-level overview of the system design see 6.1.

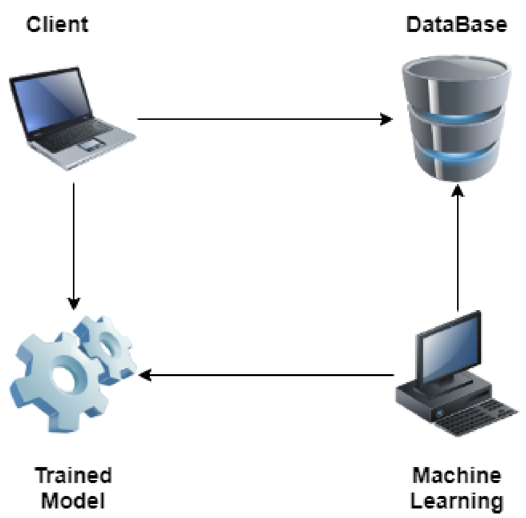

Figure 6.1: High-Level System Design

6.2 Client Architecture

The client was designed using a model/view architecture. This was enabled through the use of the PyQt5 GUI framework. Qt contains a set of item view classes that use a model/view architecture to manage the relationship between data and the way it is presented to the user. A Model-View-Controller (MVC) is a design pattern originating from Smalltalk that is often used when building user interfaces [42].

If the view and the controller objects are combined, the result is the model/view architecture. This still separates the way that data is stored from the way that it is presented to the user, but provides a simpler framework based on the same principles see 6.2. This separation makes it possible to display the same data in several different views, and to implement new types of views, without changing the underlying data structures [43].

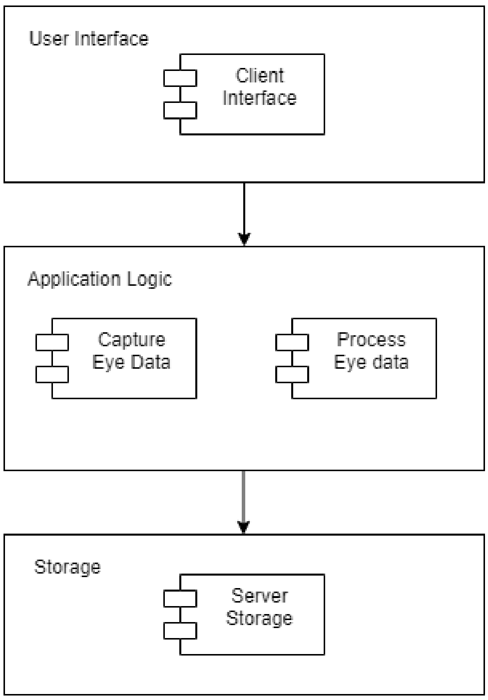

Figure 6.2: Client Architecture

- **Client Interface** - Acts as a gateway into the application. The user is presented with GUI using PyQt5 that allows them to navigate through the various application features.

- **Capture Eye Data** - A red dot will move around the screen in a cross pattern (up, down, left, right). Using OpenCV the application will then track the users eye movement and record this data into a .CSV format.

- **Process Eye Data** - Eye data is requested from the database. The data is then presented to the user in a series of graphs using Matplotlib and Scikit-learn. Finally, the graph data is then classified as pass/fail via machine learning using a Keras pre-trained model.

- **Server Storage** - Read and writes operations are performed using MongoDB via a cloud based data storage system hosted on AWS.

6.3 Client Subsystems

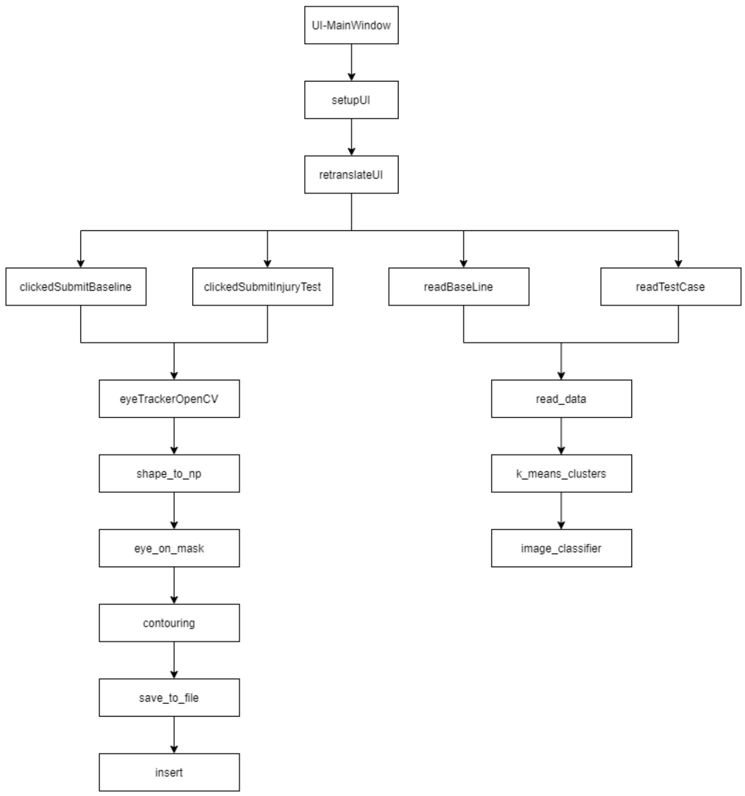

Figure 6.3: User Interface Subsystems UML

- **UI MainWindow** - Implementation of the PyQt5 model/view architecture. Provides access to client navigation via a graphical user interface and directs the application logic.
- **Setup UI** - Creates GUI elements such as frames, windows, widgets, labels and buttons.
- **Retranslate UI** - Navigation controller for the GUI. Re-translates the user from one winow to another when they click the various UI buttons.
- **Submit Baseline** - Selects option 1 to write to the baseline database.
- **Submit injuryTest** - Selects option 2 to write to the injuryTest database. Gets the currents time and prompts user for injury details.
- **EyeTracker OpenCV** - Requests access to the device webcam. Displays user prompts on screen to start recording data once user is ready. Moves the red dot around the screen.
- **Shape to Np** - Implements the 68_shape.dat model to return the (x,y) coordinates for each facial landmark.
- **Eye on Mask** - Draws a black mask on the face. Takes in image points and colour arguments. Returns an image with the area between those points filled with the specified colour.
- **Contouring** - Locates both eyes by finding the largest contours from the mask. Dividing the mask by the mid point will then give the center of the eyes. Finally, draws red circles around each eye.
- **Save to File** - Saves the (x, y) coordinates for both the left and right eye into a .csv file.
- **DB.insert** - Inserts the user details along with the contents of the eye tracking data into a collection contained on the server storage.
- **Read Data** - Takes input from the user and requests the given test_id from the database. Upon success the data set will be generated into a graph for image classification.
- **kmeans Clusters** - Converts the input data into a column stack and performs a k-means clustering function using scikit-learn. This groups data records into clusters of similar objects. These clusters are then shown to the user in a series of graphs.

- **Image Classifier** - Loads the pre-trained AI model to classify the input data. The results of the pass/fail metrics are printed to the console.

6.4 Database Architecture

AWS provides a convenient way to create, maintain and deploy a collection of cloud based resources. MongoDB utilises replication to maintain the same data set across a cluster of databases. Replication enables high availability and redundancy in the event of a failure to the primary system [44].

All applications typically interact with the primary node for read and write operations see 6.4. It is however possible to elect a secondary node as a preference for read operations, but write operations always go to the primary node and get replicated asynchronously in the secondary nodes.

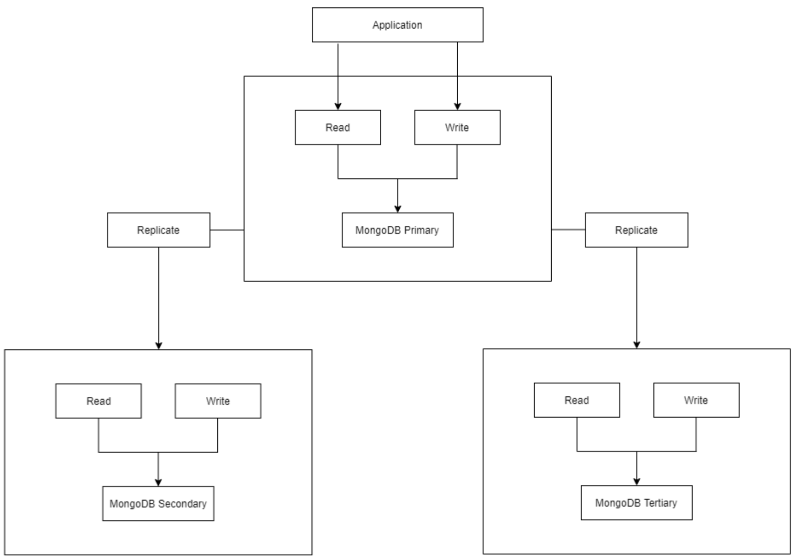

Figure 6.4: Database Architecture

6.5 DBM - Data Structure

The main objective of a data structure is to organize data to suit a specific purpose so that data can be accessed and worked both efficiently and effectively. In the case of the application a simple data structure was implemented that could easily be expanded upon if required. This is one of the key advantages of using a flexible NoSQL [45] database as opposed to a rigidly defined relational database [46].

```
    {
        "_id": "objectId",
        "test_id": "Example",
        "Date": "02/05/2021 12:07:33",
        "name": "John Doe",
        "contents": "268, 239, 386, 235
                     268, 239, 386, 235
                     269, 239, 387, 234
                     269, 239, 386..."
    }
```

- **_id** - ObjectId's are the default value of the _id field for each document. MongoDB generates these values during document creation and are used to identify individual documents.

- **test_id** - This value is entered by the user while creating a new baseline or injury test and is used as a search parameter when reading from the database.

- **Date** - This value is automatically generated by the application at run-time and is appended to the document during creation.

- **Name** - This value is entered by the user while creating a new baseline or injury test. Since a user might have multiple tests performed over a period of time, this value does not need to be unique and serves to help group collections of tests to specific users.

- **contents** - The contents is the eye data recorded during the test case. Each row represents a single frame of the captured video data. The columns are (x1, y1, x2, y2), being the 2D-coordinates for the left and right eye respectively.

6.6 Machine Learning Framework

Keras is the high-level API for TensorFlow, which enables the construction of data flow graphs by taking inputs as a multi-dimensional array called a Tensor. This functions as a flow chart of operations that can be performed on these inputs [47]. The model training process can be broken into a number of key steps see 6.5.

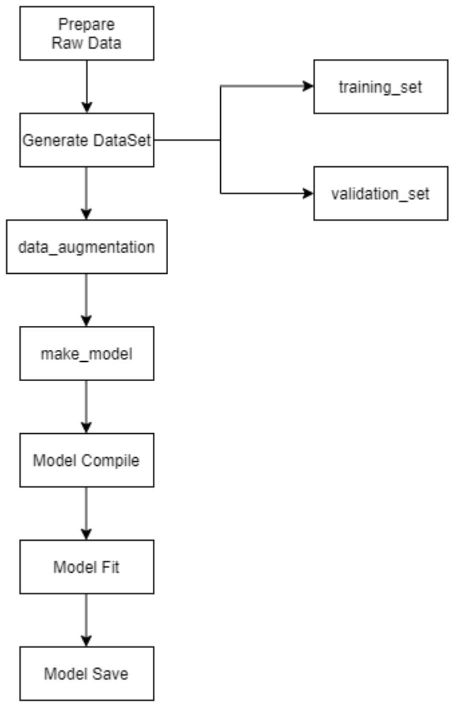

Figure 6.5: Keras Training Pipeline

- **Prepare Raw Data** - Loops through the provided data set and filters out any corrupted images.
- **Generate Data Set** - Defines the image and batch size. Separates the raw data into the training and validation sets at a 80:20 ratio respectively.
- **Data Augmentation** - Artificially expands the size and diversity of the training data set by generating modified versions of the images contained within the data set. The type of modification used on the training set was rotation. However, other enhancements such as image

flipping, colour inversion and magnification can also be implemented if more variation is required [48].

- **Make Model** - Groups the layers into an object with training and inference features. The Entry block contains the input layer and data_augmentation preprocessor. A dropout layer is included before the final classification layer.
- **Model Compile** - Configures the model training with an optimizer that implements the Adam algorithm [49], the in build loss functions and metrics used to evaluate the model during training and validation.
- **Model Save** - Stores the trained model to the local device as a HDF5 file. The contents of this file include the models topology, weights and optimizer state.

6.7 Training Data

The construction of the data set used to train the model was achieved by performing a series of eye tests using the client application. Each eye test was then manually labelled and placed into its respective collection, to define each class to be identified by the model. In total 1351 test cases were performed and divided between the pass and fail classes, with 1081 used for the training set and 270 for the validation set, fig 6.6. The full data set used for training the AI model can be found in the main GitHub repository **Here**.

```
Found 1351 files belonging to 2 classes.
Using 1081 files for training.
Found 1351 files belonging to 2 classes.
Using 270 files for validation.
```

Figure 6.6: Breakdown of Training and Validation data sets

The pass data represents test cases that accurately followed the red dot around the screen in a consistent manner. This was done to keep parity in the training data, as any major outliers could adversely effect the model training see 6.7.

The fail data represents test cases that used simulated concussion data, by looking around the screen in a random pattern see 6.8. Ed Daly was involved with the local rugby teams and had proposed using the application to gather live data from injured players.

Due to Covid-19 sporting activities were suspended, so unfortunately the opportunity to collect live data did not present itself. In light of this setback, simulated concussion data was used as a proof of concept to train the model.

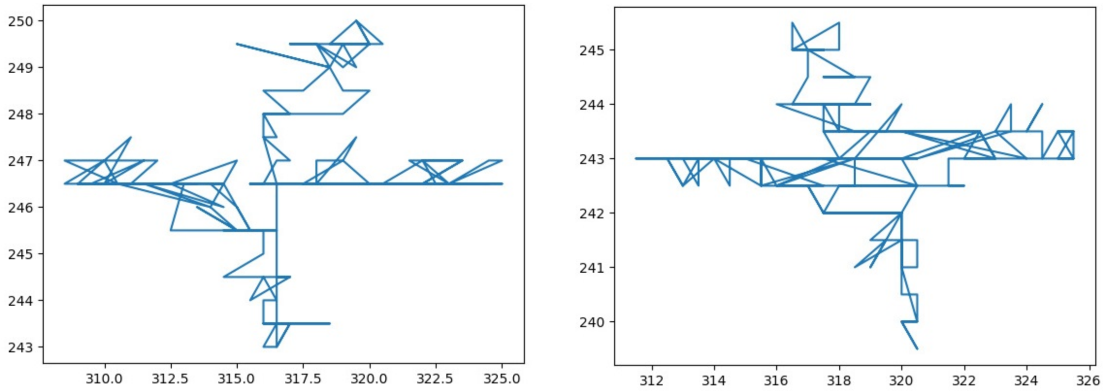

Figure 6.7: Examples of Successful Eye Tests (Pass)

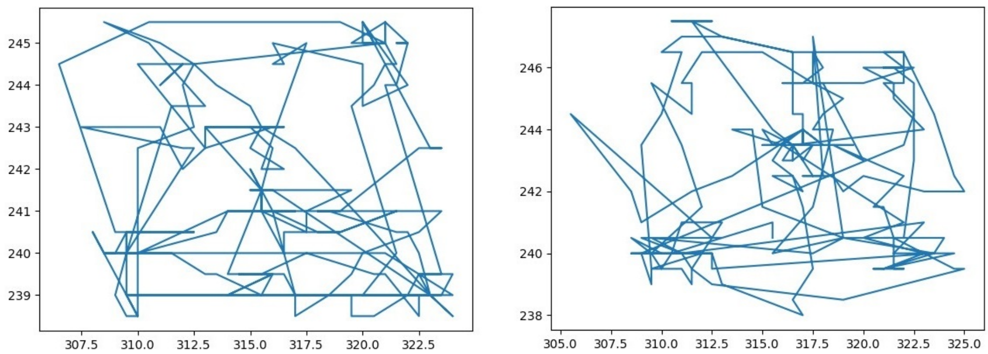

Figure 6.8: Examples of Simulated Concussion Data (Fail)

6.8 AI Model

A model is a collection of layers with training and inference features. Each layer contains a set of variables that are modified in response to input training data. The input layer is the entry point of the model, leading into a chain of additional layers. The output layer takes inputs from the penultimate layer and calculates the final result [50].

Below in fig 6.9 is a sample of the model used by the application. Due to the size of the model and page size limitations it could not be shown fully in this paper. The original copy of the model can be found on the project GitHub repository at the following link: **(Model Architecture)**.

- **Input** - The Input layer is used to instantiate a tensor.

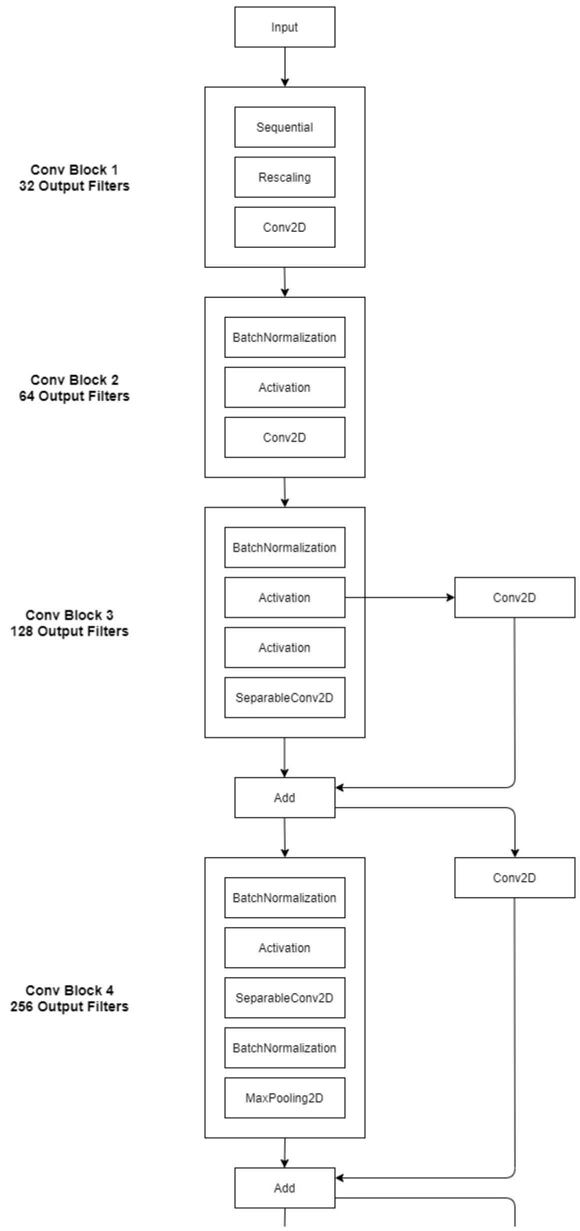

Figure 6.9: Sample of Model Architecture

- **Sequential** - The sequential layer creates a plain stack of layers that has a single input and output tensor.
- **Rescaling** - Multiplies the given inputs by a scaling factor. The Rescaling multiplier is applied both during training and inference.

- **Conv2D** - Initialises a convolution kernel that is convolved with the previous layer to produce a tensor of outputs.
- **BatchNormalization** - Normalizes a set of inputs. Applies a transform that maintains the mean output close to 0 and the standard deviation of the output close to 1.
- **Activation** - Applies an activation function to an output. This calculates the weighted sum of its input, applies a bias and then decides if it needs to be fired or not.
- **SeparableConv2D** - Performs a depthwise spatial convolution followed by a point wise convolution. This mixes the resulting output channel. This is a method of factorising a convolution kernel into two smaller kernels.
- **MaxPooling2D** - The input representation is downsampled by taking the maximum value of the window. This is defined by the pool_size for each dimension along the features axis.
- **Add** - Adds a list of inputs. The input is a list of tensors of the same shape and return a single tensor of the same shape
- **GlobalAveragePooling2D** - Global average pooling operation for spatial data. Outputs a 2D tensor of a given shape.
- **Dropout** - Randomly sets the the inputs to 0 with a frequency of rate at the end of each training epoch. This helps to prevent model over fitting.

The model achieved a validation accuracy of 96% on 270 samples after training for 20 epochs on the test data provided. Below is a summary of the model evaluation results fig 6.10.

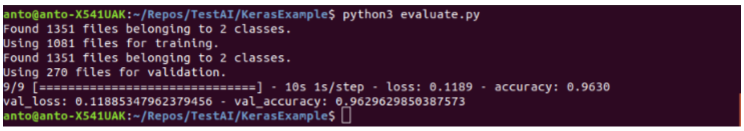

Figure 6.10: AI Model Evaluation Results

6.9 Development Environment

The application was developed on the Linux operating system [51] using the Ubuntu 18.04.5 LTS distribution [52]. This decision was made as Ubuntu 18.04 comes pre-installed with a Python 3.6 interpreter.

It is worth noting that OpenCV and dlibs will not work correctly on newer versions of Python, therefore having the development environment setup correctly was crucial for this project.

To streamline the development process, the development team used the same operating system and python versions. This reduced the amount of time spent trouble shooting problems and allowed for more efficient collaboration.

A copy the Ubuntu 18.04.5 desktop image file can be downloaded - **here**

Another advantage of using the linux operating system is that, it is completely open source and enables the use of python commands through the terminal without needing to install additional software such as Anaconda.

This allows for the convenient installation and management of python packages through the use of pip commands [53] via the linux terminal.

Using the default pip command will install packages for Python 2.x.x. However, Since Python 3.6 is used for development, **pip3** must be used to install the required packages.

Some packages such as Keras and TensorFlow will require newer versions of pip3 installed on the device as a prerequisite. To upgrade pip3 enter the following command.

Example: Using sudo commands to upgrade pip3.

```
sudo pip3 install --upgrade pip
```

Below is an example of using pip3 to install a package. The same process can be used for each package required by the application referenced in 6.9.1.

Example: Using pip3 commands to install OpenCV version 3.4.5.20.

```
pip3 install opencv-python==3.4.5.20
```

6.9.1 Required Packages

Provided is a summary of the external packages needed to run the application as per the Technology Review. Links have been embedded into the **version number** for each Package, to allow for convenient reference to the installation documentation.

- **dlib** - version 19.21.0
- **dnspython** - version 1.16.0
- **keras** - version 2.4.3
- **matplotlib** - version 3.3.4
- **numpy** - version 1.19.5
- **opencv-python** - version 3.4.5.20
- **pip3** - version 21.0.1
- **pymongo** - version 3.11.3
- **pyqt5** - version 5.15.4
- **scikit-learn** - version 0.24.1
- **tensorflow** - version 2.4.1

6.10 User Guide

To start the application run the following command:

```
python3 client.py
```

The client application is driven by the PyQt5 GUI framework. The user will have a number of options for both creating and processing eye tests fig 6.11. The baseline and injury tests are stored in their respective collections on the server database.

Prior to starting to capture the eye data, the user should correctly position them selves in-front of the device and get into a comfortable seating position. A distance of approximately 40cm (16 inches) away from the webcam is optimal fig 6.12.

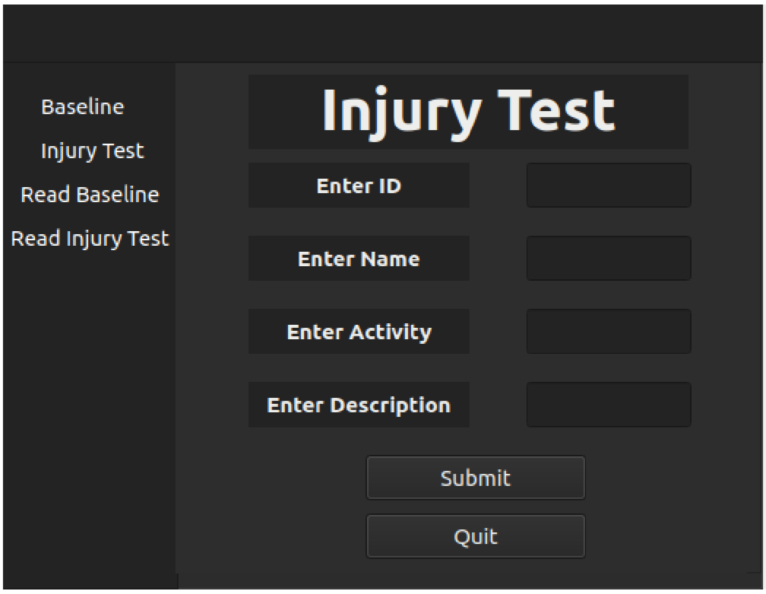

Figure 6.11: Client Graphical User Interface

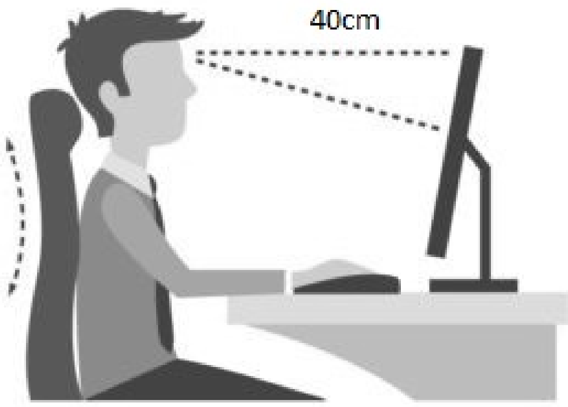

Figure 6.12: Optimal Seating Position (Side View)

The viewing angle in-front of the device must also be consistent while capturing eye data. The user should be looking directly forward with their face perpendicular to the camera. The width of the viewing window for OpenCV is approximately 15 cm wide. Given the user is sitting approx 40 cm away this gives a total viewing angle of 21.2° fig 6.13.

The threshold slider can then be adjusted to suit the lighting conditions and detect the eyes. The head should be placed in the center of the screen, with the red dot in the cross section between the two eyes fig 6.14. When the user is ready they can press ”R” to begin the eye test. After the eye test

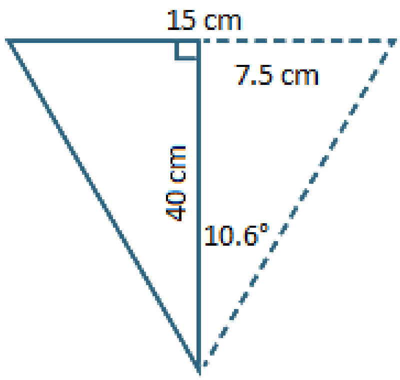

Figure 6.13: Optimal Viewing Angle (Top View)

has concluded the data will be stored and the user will be returned to the main menu.

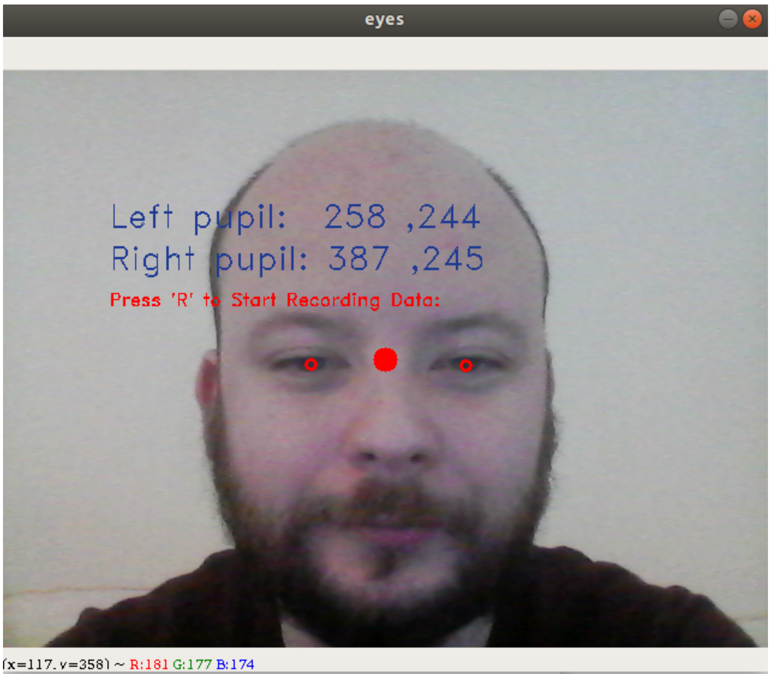

Figure 6.14: Eye Tracking Interface

Eye data that is read from the server is presented to the user in a series of graphs using matplotlib. The first graph is the default plotted eye data fig 6.15, while the second is a combination of kmeans clustering with scatter and plot fig 6.16. These images can also be stored locally to facilitate creating test data for AI training.

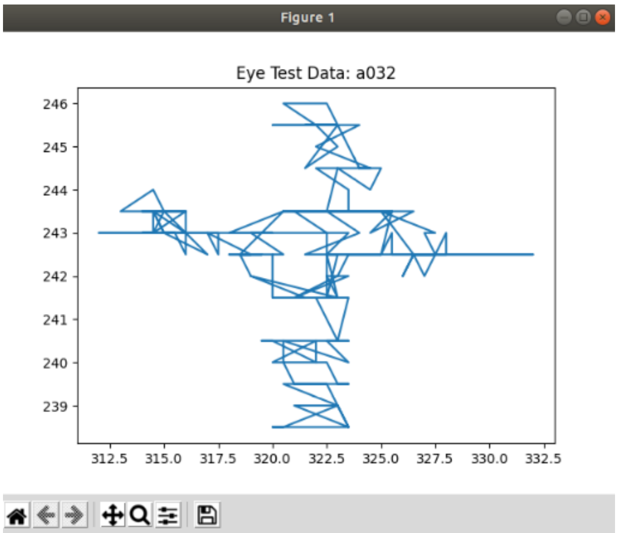

Figure 6.15: Example of plotted Eye Data

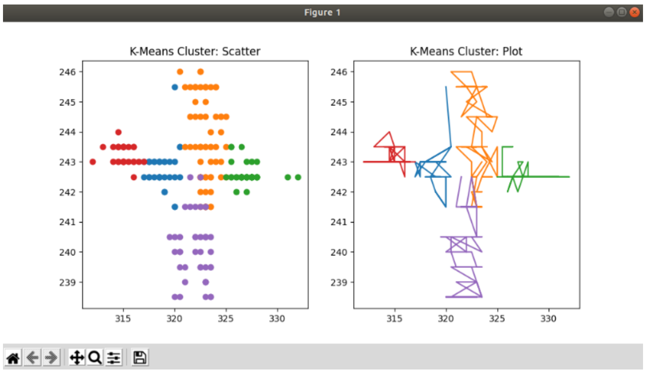

Figure 6.16: Example of plotted data with K-means clustering

The default image generated from the eye data is then processed by keras using the pre-trained model. The results of the image classification are then displayed in the terminal fig 6.17.

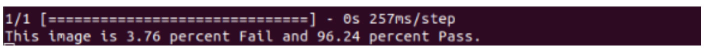

Figure 6.17: Image Classification Results for Pass/Fail

Chapter 7

System Evaluation

A key step in improving, is the ability to critically analyse ones work. In this chapter an evaluation of the system is recorded. This is achieved by breaking the project into its five objective structure, and performing an evaluation on the individual components. Before dissecting the project though, it must be noted that all objectives were met. While there is room for enhancement, it can be said that the application is a proof of concept. Having access to more data could help train the model to a higher degree and access to more sensitive hardware such as IR cameras could enhance the applications performance.

7.1 Capture

OpenCV performs sufficiently at detecting eyes, especially as a low budget option using a standard webcam. Using basic devices for computer vision based applications is an attractive option as it significantly lowers the barrier of entry through readily accessible devices such as laptops, external webcams and smart devices.

Webcam eye tracking performs well at detecting large changes in the users eye movement fig 6.7. Moving in wide sweeping patterns such as: up, down left and right going from one side of the screen to the other gave the best results in terms of consistent data capture. And as such, implementing the VOMS smooth pursuit detection technique fig 1, was the best method employable.

However, webcam eye tracking does not perform well at detecting subtle movements or complex patterns such as trying to follow a bezier curve. It is also very sensitive to movement and requires the user to keep as still as possible for the duration of the recording. This may be an issue for athletes that have just suffered from a head injury as they may be in some discomfort.

There is also a large discrepancy in the standard of webcams that are installed on portable devices. Webcams have different frame rates, resolutions, lens exposure and placement. This makes it difficult to implement a standard operating procedure for setting up the camera correctly as all devices are different.

Future development should consider using Infrared Cameras to detect the eye movement. These offer increased accuracy by being able to detect the eye outside the visible spectrum of light. First a light source is used to illuminate the eye, causing a bright reflection from the cornea and pupil. The camera then captures this reflection and calculates the vector formed between the cornea and pupil reflections. From this vector the gaze direction can be determined [54].

Many commercial products and industry leaders in the field of eye tracking technologies are taking advantage of using infrared cameras such as: Tobii Pro Fusion, GP3 Eye Tracker and Smart Eye Aurora. Other related works doing research in relation to the application of mobile devices for eye tracking, have determined that while mobile devices can give decent results, they lack the precision of specialised eye tracking equipment for developing professional applications [55].

It is hard to deny the clear advantages that can be gained through the use of infrared cameras. However, this will raise the cost of the system and potentially reduce the scope of the target audience due to the additional specification requirements.

A low budget option that uses readily available mobile technologies does have some drawbacks in terms of accuracy. However, it does have obvious benefits in terms of scalability and accessibility.

7.2 Process

The coordinates obtained in the capture step were used to create graphical representations of the eyes movement as they follow a target onscreen. The graphs in this project are used to analyse the ocular motor function of the individual been tested. This method is echoed in both available solutions researched for this project, as recorded in section 3.3. Also required in these solutions is the need for a baseline test, performed under ideal circumstances. An individuals base line data is used to compare against data taken after an injury. Making use of Matpoltlib, which is excellent for plotting the 2D eye data, the graphs can also be saved as images and used to build data sets for training the AI model.

While the AI model in this project can evaluate the graphical data, it can be difficult to discern as a third party. By creating a more stable input method, such as with the use of chin and forehead straps or a form of wearable technology the graphical representations would be less noisy, see figure 6.15. It is worth noting that, while having clearer graphs would help with human understanding, the AI model has successfully demonstrated its ability to differentiate between good and bad data.

7.3 Analyze

The Image classifier performed well on the training and validation data set that we procured during the development and testing process fig. 6.10.

However, after about 20 epochs the validation accuracy starts to drop, indicating the model is starting to suffer from overfitting [56] due to the relatively small data set after a certain point.

Ideally having 1000 or more samples for each class should greatly enhance the AI models ability to generalise, as having a large data set is always preferable when it comes to training AI models [57].

This might take some time to collect if working on a small scale with local teams. However, if it received traction it could be done in a relatively short period of time through collaboration.

Without access to live data and in particular to eye data collected from individuals suffering from a suspected concussion, it will not be possible for the application to truly identify a genuine concussion.

However, based on the accurate pass data and simulated concussion data used during the model training process. The results obtained from using a trained AI for image classification on graphs of plotted eye data show some interesting potential.

Comparing baseline tests to injury tests was achieved by calculating the standard deviation of the two-dimensional eye data. This algorithm was then performed of two sets of data such as a baseline test and subsequent injury test. By calculating the percentage difference between them we could heuristically calculate how similar two sets of eye data were.

This however this does have some limitations depending on the consistency of the data being captured. If there is not parity between the baseline and injury test conditions it may result in a fail condition, despite the user following the red dot with their eyes correctly. A deviation of just a few millimeters, with the head slightly out of position between both tests will cause a huge variance between the two sets of data.

Using methods such as this will almost certainly require some form of chin strap or head rest in order eliminate errors caused due to inconsistencies with the setup process.

Another issue with this method, is if the user gazes at the centre of the screen for both the injury and baseline tests and does not attempt to follow the red dot around the screen. The standard deviation between both tests will be low as they will be very similar. So this can result in generating a false positive despite the user not completing the eye test as intended.

Heuristic methods are not good metrics when used in isolation for accessing eye movement, as it can be random by nature and difficult to quantify. However, they may be useful to provide additional insight when used in combination with other metrics to help determine pass/fail conditions.

7.4 Store

AWS and MongoDB are market leaders when it comes to cloud based NoSQL data storage. This is due to Amazons high reliability, scalability and cost-effectiveness. The free tier provided is well suited for small scale development.

However, if the application received widespread adoption it might be worth considering to pay for the premium features such as: configurable backup services, increased RAM and CPU processing capability.

Update and Delete functionality is not currently included in the front-end client. This is relatively easy to implement with MongoDB and was omitted due to it not being a major requirement during the development process. However, this would definitely need to be implemented for a more refined product.

The purpose of the database is to facilitate the client being used on a portable device and to assist with the collection of test data. This data will then be used fo d, generating star further research into the field, generating statistics, monitoring recover rates and training AI models.

Building a large data set to train AI models should be one of the long term goals for any future development and having an extensive collection of data to fall back on may prove invaluable to research groups. In this regard the data base storage system is robust and more than sufficient for the requirements.

7.5 Report

The plotting Graphs used give a good illustration of how the eye data was captured. As previously mentioned this seems to be standard for most eye tracking products that implement the VOMS smooth pursuit detection technique.

Kmeans clustering graphs in particular give good insight as to where the eyes were looking. Top, Bottom, Left, Right and centre. Clustering may also be useful for future development as coordinates that are densely clustered together could indicate points of interest that the user was staring at during the test.

The Image classifier can be used to give a suggested assessment about the condition of the injured party. However, simply deconstructing the entirety of the eye data down to a single pass/fail metric may not give an fair reflection of the users physical state.

Ideally the results should have extensive meta data. Listing a summary of all the tests that the specific user has taken previously and how the current test compares to these, as well as on its own merits.

The more information that can be generated from the data, the more robust the application will be at providing insight about the condition of the user. However the final decision about the health of an individual should always be made by a medical professional.

Chapter 8

Conclusion

Developing affordable, technology based solutions for real world problems such as the detection of concussion, has the potential to dramatically improve peoples quality of life and prevent long term health problems.

This would be most relevant in sport at amateur and youth level. Due to financial restrains often encountered at this level, the purchase of specialised eye tracking equipment is a massive investment that is usually out of reach.

This application, with more robust testing, could be distributed to these clubs at no charge. This would have multiple benefits, at club level the effect would be a quicker more accurate concussion test. At development level it would mean more data to train the AI model with, resulting with more and more accurate testing.

The outcome of this project hinged on the completion of each of the five main objectives identified early in cycle one. Each of these objectives were met and the result is the production version of the concussion detection application.

Capturing the data was the first objective, this was completed early in development as it was the foundation of the rest of the application. This process performs well at tracking large amounts of eye movement, using a standard webcam camera. The eye tracking does not perform well at detecting fine detailed movements. Using an infrared camera may significantly improve accuracy, but will also increase the cost and hardware requirements limiting its market penetration.

It would not be recommended to attempt to recreate this type of application for a mobile phone as the capture process would be inherently effected resulting in unusable graphical data. This would be due to the instability of the camera while tracking the eyes and the lower viewing angle available on smaller screens.

Using graphs of plotted eye data is common place amongst related technologies such as Eye Guide and NeuroFlex. Plotting the average position of both eyes on the screen is a good measure of where the user is looking. However, using standard laptop webcams is not as accurate as using more sophisticated and expensive hardware to calculate this data. The quality of the graphs are heavily dependent on the consistency of the data being captured and therefor making use of more advanced hardware would enhance the overall outcome achieved.

The image classifier used to analyse the processed data performed well, both on the training and validation data sets that were produced during the development and testing process. Heuristic methods are not good metrics when used in isolation due to the complexities of human eye movement. However, they may be useful to provide additional insight when used in combination with other metrics. The AI model trained and employed in this project has been consistently accurate in the determination of bad data.

The storage solution incorporated performed as expected. The primary function of the database is to store data between tests. Baseline data is needed to compare with post injury data and these captures, under real circumstances, will not be taken at the same time. Meaning this step is critical.

The secondary objective was to collect data that can be used for further research into concussion and concussion detection systems. While been the secondary function of the database, building a data-set like this could be as important as the application itself.

The use of a graphical user interface such as the one employed in this project was not specifically needed. The result of the test could be displayed through the command line interface. Yet adding a GUI has enhanced the usability of the application. The GUI is used to both display results but also to interact with the information been added to the database. While the all relevant information is reported to the operator, the GUI has much room for improvement. This does not take away from the functionality of the application though.

Towards the end of cycle two it was announced that World Rugby will be trialing eye-tracking technology along side current assessments to detect concussion. The technology will be a collaboration between EyeGuide and NeuroFlex. This announcement further justifies the importance of this project.

This area of research and development is in its infancy and warrants further investigation. The application presented here has the potential of becoming an important tool used in these investigations.

There are plenty of opportunities available for an application such as this. Not only could it be of benefit in future research, if adopted into production it could be used to detect concussion under a range of circumstances.

Bibliography

[1] S. Media, "Six more former players join concussion lawsuit against rfu, welsh rugby union and world rugby," *Sky Sports*, Dec. 18, 2020. [Online]. Available: https://www.skysports.com/rugby-union/news/12321/12165318/six-more-former-players-join-concussion-lawsuit-against-rfu-welsh-rugby-union-and-world-rugby (visited on 12/17/2020).

[2] R. McLoughlin, M. Liston, T. Comyns, I. Kenny, R. Cahalan, M. Campbell, L. Glynn, A. Griffin, A. Harrison, T. Leahy, M. Lyons, H. Purtill, G. Warrington, and C. Yeomans, *The irish rugby injury surveillance (iris) project all-ireland league male female amateur community club rugby 2018-19 season report*, Sep. 2019.

[3] A. Nordström and P. Nordström, "Traumatic brain injury and the risk of dementia diagnosis: A nationwide cohort study," *PLoS medicine*, vol. 15, no. 1, e1002496, 2018.

[4] A. Association, *Traumatic brain injury (tbi)*, unknown. [Online]. Available: https://www.alz.org/alzheimers-dementia/what-is-dementia/related_conditions/traumatic-brain-injury.

[5] P. McCrory, W. Meeuwisse, J. Dvorak, M. Aubry, J. Bailes, S. Broglio, R. C. Cantu, D. Cassidy, R. J. Echemendia, R. J. Castellani, *et al.*, "Consensus statement on concussion in sport—the 5th international conference on concussion in sport held in berlin, october 2016," *British journal of sports medicine*, vol. 51, no. 11, pp. 838–847, 2017.

[6] P. McCrory, W. H. Meeuwisse, M. Aubry, R. C. Cantu, J. Dvorak, R. J. Echemendia, L. Engebretsen, K. M. Johnston, J. S. Kutcher, M. Raftery, *et al.*, "Consensus statement on concussion in sport—the 4th international conference on concussion in sport held in zurich, november 2012," *PM&R*, vol. 5, no. 4, pp. 255–279, 2013.

[7] C. for Disease Control and Prevention, “What is a consussion?” *Center for Disease Control and Prevention*, Feb. 12, 2019. [Online]. Available: `https://www.cdc.gov/headsup/basics/concussion_whatis.html#:~:text=A%20concussion%20is%20a%20type,move%20rapidly%20back%20and%20forth.` (visited on 10/11/2020).

[8] H. S. E. Ireland, “Concussion, symptoms of concussion,” *Health Service Executive*, Jul. 13, 2011. [Online]. Available: `https://www.hse.ie/eng/health/az/c/concussion` (visited on 10/11/2020).

[9] A. Baillargeon, M. Lassonde, S. Leclerc, and D. Ellemberg, “Neuropsychological and neurophysiological assessment of sport concussion in children, adolescents and adults,” *Brain injury*, vol. 26, no. 3, pp. 211–220, 2012.

[10] World Rugby, *Head injury assessment (hia)*, 2012. [Online]. Available: `https://www.sareferees.com/uploads/files/Eugene%5C%202017/180326%5C%20HIA%5C%20Protocol%5C%20and%5C%20Procedures.pdf`.

[11] ——, *Concussion guidelines*, 2012. [Online]. Available: `https://www.world.rugby//the-game/player-welfare/medical/concussion/concussion-guidelines`.

[12] A. Mucha, M. W. Collins, R. Elbin, J. M. Furman, C. Troutman-Enseki, R. M. DeWolf, G. Marchetti, and A. P. Kontos, “A brief vestibular/ocular motor screening (voms) assessment to evaluate concussions: Preliminary findings,” *The American journal of sports medicine*, vol. 42, no. 10, pp. 2479–2486, 2014.

[13] EyeGuide, *Eyeguide*, 2010. [Online]. Available: `https://www.eye.guide/`.

[14] Saccade Analytics, *Neuroflex*, 2016. [Online]. Available: `https://saccadeanalytics.com/`.

[15] G. Van Rossum *et al.*, “Python programming language.,” in *USENIX annual technical conference*, vol. 41, 2007, p. 36.

[16] python.org, *What is python?* 1991. [Online]. Available: `https://www.python.org/doc/essays/blurb/`.

[17] kivy.org, *Kivy: Cross-platform python framework for nui development*, 2012. [Online]. Available: `https://kivy.org/#aboutus`.

[18] buildozer.readthedocs.io, *Welcome to buildozer’s documentation!* 2018. [Online]. Available: `https://buildozer.readthedocs.io/en/latest/#`.

[19] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: `10.1038/s41586-020-2649-2`. [Online]. Available: `https://doi.org/10.1038/s41586-020-2649-2`.

[20] aws.amazon.com, *Aws general reference*, 2006. [Online]. Available: `https://docs.aws.amazon.com/general/latest/gr/aws-general.pdf`.

[21] S. Obrutsky, "Cloud storage: Advantages, disadvantages and enterprise solutions for business," in *Proceedings of the Eastern Institute of Technology Conference*, 2016, p. 10.

[22] E. Dede, M. Govindaraju, D. Gunter, R. S. Canon, and L. Ramakrishnan, "Performance evaluation of a mongodb and hadoop platform for scientific data analysis," in *Proceedings of the 4th ACM workshop on Scientific cloud computing*, 2013, pp. 13–20.

[23] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[24] GNU.org, *Various licenses and comments about them*, 2010. [Online]. Available: `https://www.gnu.org/licenses/license-list.html#ModifiedBSD`.

[25] R. Laganière, *OpenCV Computer Vision Application Programming Cookbook Second Edition.* Packt Publishing Ltd, 2014.

[26] S. S. Srinivasa, D. Ferguson, C. J. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. V. Weghe, "Herb: A home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.

[27] N. Salehi, M. Keyvanara, and S. A. Monadjemmi, "An automatic video-based drowning detection system for swimming pools using active contours," *International Journal of Image, Graphics and Signal Processing*, vol. 8, no. 8, p. 1, 2016.

[28] K. Sajjad, "Automatic license plate recognition using python and opencv," *Department of Computer Science and Engineering MES College of Engineering*, 2010.

[29] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: `10.1109/MCSE.2007.55`.

[30] dlib.net, *Introduction*, 2002. [Online]. Available: `http://dlib.net/intro.html`.

[31] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[32] B. Graham, "Kaggle diabetic retinopathy detection competition report," *University of Warwick*, 2015.

[33] A. Agarwal and A. Saxena, "Malignant tumor detection using machine learning through scikit-learn," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 15, pp. 2863–2874, 2018.

[34] Keras.io, *About keras*, 2018. [Online]. Available: `https://keras.io/about/`.

[35] TensorFlow.org, *Why tensorflow*, 2015. [Online]. Available: `https://www.tensorflow.org/about`.

[36] G. Create and J. Willman, "Modern pyqt,"

[37] OpenCV, *Haarcascade eye*, 2013. [Online]. Available: `https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_eye.xml`.

[38] ——, *Haarcascade frontal face*, 2013. [Online]. Available: `https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml`.

[39] dlib, *Shape predictor 68 face landmarks*, 2015. [Online]. Available: `https://github.com/davisking/dlib-models/blob/master/shape_predictor_68_face_landmarks.dat.bz2`.

[40] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, IEEE, vol. 1, 2001, pp. I–I.

[41] M. Bertocco, F. Ferraris, C. Offelli, and M. Parvis, "A client-server architecture for distributed measurement systems," *IEEE transactions on instrumentation and measurement*, vol. 47, no. 5, pp. 1143–1148, 1998.

[42] G. E. Krasner, S. T. Pope, *et al.*, "A description of the model-view-controller user interface paradigm in the smalltalk-80 system," *Journal of object oriented programming*, vol. 1, no. 3, pp. 26–49, 1988.

[43] Qt.io, *Model/view programming*, 1990. [Online]. Available: `https://doc.qt.io/qtforpython/overviews/model-view-programming.html`.

[44] MongoDB, *Replication*, 2009. [Online]. Available: `https://docs.mongodb.com/manual/replication/`.

[45] J. Han, E. Haihong, G. Le, and J. Du, "Survey on nosql database," in *2011 6th international conference on pervasive computing and applications*, IEEE, 2011, pp. 363–366.

[46] E. F. Codd, "Relational database: A practical foundation for productivity," in *Readings in Artificial Intelligence and Databases*, Elsevier, 1989, pp. 60–68.

[47] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[48] TensorFlow, *Data augmentation*, 2015. [Online]. Available: `https://www.tensorflow.org/tutorials/images/data_augmentation`.

[49] ——, *Adam optimizer*, 2015. [Online]. Available: `https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam`.

[50] Keras, *The model class*, 2015. [Online]. Available: `https://keras.io/api/models/model/`.

[51] S. Bokhari, "The linux operating system," *Computer*, vol. 28, no. 8, pp. 74–79, 1995.

[52] W. Von Hagen, *Ubuntu Linux Bible*. John Wiley & Sons, 2007, vol. 352.

[53] python.org, *Installing packages*, 2008. [Online]. Available: `https://packaging.python.org/tutorials/installing-packages/`.

[54] K. Różanowski and K. Murawski, "An infrared sensor for eye tracking in a harsh car environment," *Acta Physica Polonica A*, vol. 122, no. 5, pp. 874–879, 2012.

[55] P. Chynał, J. M. Szymański, and J. Sobecki, "Using eyetracking in a mobile applications usability testing," in *Asian Conference on Intelligent Information and Database Systems*, Springer, 2012, pp. 178–186.

[56] S. Mutasa, S. Sun, and R. Ha, "Understanding artificial intelligence based radiology studies: What is overfitting?" *Clinical imaging*, 2020.

[57] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine learning for user modeling," *User modeling and user-adapted interaction*, vol. 11, no. 1, pp. 19–29, 2001.