Middle East Technical University Northern Cyprus Campus
Computer Engineering Program

CNG491 Computer Engineering Design I

Iris Analyzer

2269421 Aref Khademi Najafabadi
2246452 Sameh Farid S. Algharabli
2269454 Evans Muthugumi Kimathi
2175263 Elbaraa Elsaadany

Supervised by
Asst. Prof. Dr. Meryem Erbilek

Report No.2

Contents

1 Introduction **3**
1.1 Motivation . . . 3
1.2 Aims and Objectives . . . 3
1.3 Research Methodology . . . 4

2 Related Work **5**

3 Requirements **8**
3.1 Stakeholders . . . 8
3.2 Functional System Requirements . . . 8
3.3 Non-functional System Requirements . . . 9
3.4 Domain Requirements . . . 10
3.5 Assumptions and Justifications . . . 10

4 System Modelling **11**
4.1 Structured Use Case Diagram . . . 11
4.2 Sequence Diagram of Major User Cases . . . 12
4.3 Context Model . . . 14
4.4 Architectural Model . . . 15
4.5 Process Model . . . 16

5 Graphical User Interface **18**
5.1 Main Page GUI . . . 18
5.2 User GUI . . . 18
5.3 System Administrator GUI . . . 19

6 Agile Development with Scrum **21**
6.1 Sprint Backlog . . . 21
6.2 Sprint Burndown Chart . . . 22
6.3 Sprint Review . . . 24
6.4 Sprint Retrospective . . . 24

7 Estimation **26**
7.1 Input . . . 26
7.2 Complexity Adjustment Table . . . 27
7.3 Line of Code . . . 28
7.4 Estimate the Efforts . . . 28
7.4.1 Constructive Cost Model COCOM0 Basic . . . 28
7.4.2 jones's first -order effort estimation methods . . . 28

7.4.3 Schedule rule of thumb 29

8 Project Management **30**
8.1 Milestones and Tasks 30
8.2 Gantt Chart 32

9 References **33**

Chapter 1

Introduction

1.1 Motivation

Iris Recognition is a biometric method of identifying people based on unique patterns within the ring-shaped region surrounding the pupil of the eye. Every iris is unique to an individual, making it an ideal form of biometric verification. One of the reasons Iris Recognition is such a sought after method of identifying individuals, especially in sectors such as law enforcement and border control, is that the iris is a very strong biometric, highly resistant to false matches as well as high search speed against large databases. Iris Recognition is an extremely reliable and strong method to accurately identify individuals.[10]

Iris recognition can be used for verifying owner identity (biometric data) and demographic data of the identified person (soft biometric data). Verification(1-to-1 comparison) is the process where the biometrics information of an individual is compared with biometrics on the record[4] while identificaton(1-to-many comparison) is used to discover the origin of certain biometrics to prove or disprove the association of that information with a certain person/individual. [4] Research and studies have been done on gender prediction[1], age prediction[2] and biometric technology[3] using iris analysis using image processing and machine learning techniques.

In biometrics, iris data cannot be the same in two different individuals and cannot wear out as seen in some people's fingerprints, hence, iris recognition can be a secure way to identify and authenticate individuals. Our main motivation is to make iris recognition(in biometrics) a more widely used option, by improving the output accuracy using deep learning techniques like CNN's(Convolution Neural Networks) to extract features instead of the old-school image processing feature extraction ways. This makes the prediction process more accurate as the image is passed through a network of filters to extract different features over and over again.

The aim this semester is to work on the deep learning techniques used in the research papers, get the results and try to improve them by adding new techniques or by deleting ones used.

1.2 Aims and Objectives

The main aim of this project is to acquire identification and their demographic information by analyzing their eyes (iris). The objectives are:

1. Get eye data from a database.
2. Pre-processing where the eye image will be segmented using image processing techniques. This step will partition the eye image into image objects (iris and pupil) and this will be used to locate the objects and their boundaries(diameter and radius of pupil and iris). This will help us work on the image in the next stages.
3. Use deep learning to retrieve the identity and demographic data.

1.3 Research Methodology

The project will be done in two parts: image processing part and deep learning part. Image processing techniques shall be used to segment the eye images. This will define the iris and pupil objects(pixel values) and their boundaries(radius and diameters) which will later be needed in deep learning process to work on specific objects i.e. the iris alone. The image processing part will be achieved using the following methods:

1. Images shall be acquired from the BioSecure Moltimodal Database(BMDB). The BMDB is a database that contains eye images collected from a standard office environment using the LG Iris EOU3000 system. During acquisition, spectacles weren't allowed but contact lenses were allowed.
2. Pre-processing: The chosen eye image will be segmented using image processing segmentation algorithms. This process will localise the iris region and detect the sclera and pupil boundaries.

The deep learning part will use the method explained below:

1. The segmented eye images in the database shall be divided into training and testing sets.

   The training set: The actual dataset that we use to train the model (weights and biases in the case of a Neural Network). During each epoch, our model will be trained over and over again on this same data in our training set, and it will continue to learn about the features of this data.

   Testing set: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. Once the model is built, the test dataset is used to get the accuracy of the hypothesis.
2. The system shall use deep learning algorithms to train the input images using the training set, and to predict the output based on the model using the testing set.

Chapter 2

Related Work

The first successful and commercially available iris recognition system was in 1993, and was suggested by Daugman[5]. In the system, the inner and outer boundaries are detected using an integro-differential operator. Afterwards, the iris region template is transferred into normalised form form using Daugman's rubber sheet mehod. This is followed by using a 2D Gabor filter to extract the iris feaures and the hamming distance for decision making.[6] The limitation to this method is that it requires a high resolution camera to capture the iris image and its accuracy significantly decreases under non-ideal imaging conditions due to sensitivity of the iris localization stage.

After decades of comparative reserch, many iris recognition methods have been introduced to enhance the reusability and reliability. In recent years, with the vigorous development of the annual ImageNet Large Scale Visual Recognition Challenge(ILSVRC)[8], deep learning networks, especially CNNs, have shown obvious improvement in the perfomance of computer vision tasks such as image classification, single object localization, and object detection.[7]

In Tianming Zhao et al. [7],"A Deep Learning Iris Recognition Method Based on Capsule Network Architecture", they start by doing pre-processing on the iris image where they locate and segment the iris texture part from the original iris image through quality evaluation. The Region of Interest(ROI) is extracted, normalized and enhanced.

They move on to convolution step where low level features are extracted from pixel intensities of the input images and form the primary features used for the primary capsule layer. Two strategies are used to construct the convolution part:

1. Applying transfer learning method -¿ The entire network is divided into subnetworks with the vector part and a series of different networks is instantiated. This gets the outputs of different levels in the same network as the result of the convolution part to find the most suitable option of convolution structure for the entire network through experiments.

2. Building several several shallow convolution networks as the convolution part without pretraining.

The next step is the use of routing algorithm in the vector layer and is done in 2 stages:

1. Dynamic Routing in Capsule Network: provides non-linear mapping for two adjacent capsule layers. The dynamic routing algorithm uses the cosine similarity of two vectors to measure their agreement. However, this manner is not quite good at judging quite good agreement and very good agreement, which usually makes it difficult for the network to converge after training in the experiments.

2. Adjusted Dynamic Routing: Used to improve applicability of the capsule structure and make it easy for the network to converge when dealing with input images of large size.

The last step us the network architecture which consists of the convolution part, vector part, classification and reconstruction.

Finally,they simulate an environment with different illumination intensities, and train and test iris images with different pupil sizes (represent different light intensities) to show the recognition ability of the method when the environment is varied. Experiments show that a deep network with capsule architecture is feasible in iris recognition. In the test of the JluV3.1 iris dataset, InceptionV3-5blocksCDRDL achieves the highest accuracy of 99.37 percent. In the test of the JluV4 dataset, Iris-DenseCDRDL achieves the highest accuracy of 99.42 percent. In the test of the CASIA-V4 Lamp dataset, VGG16-4blocksCDRDL achieves the highest accuracy of 93.87 percent.

Kien Nguyen et al.,[9] in "Iris Recognition with off-the-shelf CNN features" start by segmenting the eye image. Here, they localize the iris by extracting two circular contours pertaining to the inner and outer boundaries of the iris region. This is followed by normalization step which maps the iris to a region of fixed dimension. This helps to remove the effects caused by dilation and contraction of the pupil causing iris stretching.

The normalized iris image is fed into the CNN feature extraction module to extract features. They use 5 state-of-the-art and off-the-shelf CNNs(AlexNet, VGG, Google Inception, ResNet and DenseNet).

Note that there are multiple layers in each CNN. Every layer models different levels of visual content in the image, with later layers encoding finer and more abstract information, and earlier layers retaining coarser information. One of the key reasons why CNNs work very well on computer vision tasks is that these deep networks with tens or hundreds of layers and millions of parameters are extremely good at capturing and encoding complex features of the images, leading to superior performance. To investigate the representation capability of each layer for the iris recognition task, we employ the output of each layer as a feature descriptor and report the corresponding recognition accuracy.

The extracted CNN feature is the fed to the Support Vector Machine (SVM) classification module for classification. The multi-class SVM for N classes is implemented as a one-against-all strategy, which is equivalent to combining N binary SVM classifiers, with every classifier discriminating a single class against all other classes. The test sample is assigned to the class with the largest margin.

CNNs are effective in encoding discriminative features for iris recognition. However according to Kien et al., deep learning offers the following problems and open questions:

1. Computational complexity: very high during training phase due to millions of parameters used in the network.

2. Domain adaptation and fine tuning: entails freezing early layers and only re-training a few selected later layers to adapt the representation capability of the CNNs to iris images. Fine tuning is expected to learn and encode iris-specific, as opposed to generic image features.

3. Few-shot learning: limited number of training images can be partially solved with few-shot learning, which allows the network to perform well after seeing very few samples from each class.

4. Architecture evolution: evolve off-the-shelf CNNs in order to generate powerful networks that are more suited for iris recognition.

5. Other architectures: Architectures like Deep Belief Network(DBN), Stacked Auto-Encoder(SAE) and Recurrent Neural Network(RNN) have their own advantages and can be used to extract features for iris images.

Chapter 3

Requirements

3.1 Stakeholders

1. System administrator : is responsible for updating and managing the iris analyser system.
2. Application users : every person that will use the iris analyser system to get an identity and demographic output i.e. students, teachers, workers.

3.2 Functional System Requirements

1. The system shall use an authentication for the system administrator.
    1. The system shall display an authentication message to the administrator.
    2. The system shall give the administrator access to the database if the pin is correct.
    3. The system display an error message if the pin entered is correct.
    4. The system shall display a add confirmation message to the system administrator.
2. The system administrator shall be able to add eye images in the database.
    1. The system shall authenticate the system administrator.
    2. The system shall send the added images to the database.
    3. The system shall display an added image successfully message to the system administrator.
3. The system administrator shall be able to delete eye images in the database.
    1. The system shall authenticate the system administrator.
    2. The system shall discard the selected image from the database.
    3. The system shall display a deleted image successful message to the system administrator.

4. The system administrator shall be able to create a model to be used in deep learning section.

1. The system shall authenticate the system administrator.
2. The system administrator shall divide the eye images in the database into 2 parts: training and testing set.
3. The system shall save the 2 sets in the database.
4. The system shall use the training set for modelling and the testing set for prediction using deep leaning algorithms.

5. The user shall be able to choose eye images as inputs.

1. The system shall display eye images as input to the screen.
2. The system shall mark all the eyes images selected by the user as wanted input.
3. The system shall move copies of the chosen eye inputs to the next stage (pre-processing stage).

6. The user shall be able to choose the expected output (identification and demographic output).

1. The system shall read checkbox input of either or both identification and demographic data.
2. The system shall display a confirmation message for the selected output.
3. The system shall use the chosen input to process the data after the output confirmation from the user.

3.3 Non-functional System Requirements

1. The system shall be available to users 24hrs a day so that users can access it at anytime in the day.

1. The system shall check every 2hrs (system troubleshoot) that all parts are functioning properly.
2. The system shall send a notification to the system administrator about any errors that occurred.

2. The system shall work on Windows, Linux and Mac operating system to work on all devices and be accessible to all users.

1. The system shall run on all devices with Windows 7 and above, all Linux and Mac devices.
2. The system shall display an error message if the device is incompatible.

3. The system interface input response time should not be more than 1 second to users for fast interaction process.

1. The system shall shall detect keyed and clicked inputs at a response time of 50ms.

3.4 Domain Requirements

1. The eye images in the database should be negatives (black and white) so as to used in the pre-processing and prediction part.

2. The system administrator should seperate the training and testing datasets. This is to prevent the model seeing the test dataset and giving the exact results. (The model gives impartial results because it hasn't seen the test dataset)

3.5 Assumptions and Justifications

1. The eye images used are not captured in real time but are taken from a database. These is due to the following reasons:

1. It needs special cameras and scanners.

2. The eye is sensitive to flashes and they might be damaged.

3. Ethic and privacy concerns, because users may not be sure of their safety of their images.

Chapter 4

System Modelling

4.1 Structured Use Case Diagram

The use case diagram shows how the relationship between the user and different use cases. The user can insert an image for the prediction but he has to set an output option first. Prediction model is then called to process the user input. The system administrator is entitled to manage the system. The system administrator can modify the processing model of the system later on. However, the user will be notified by the new modification(named "Display notification") through a message on the screen.

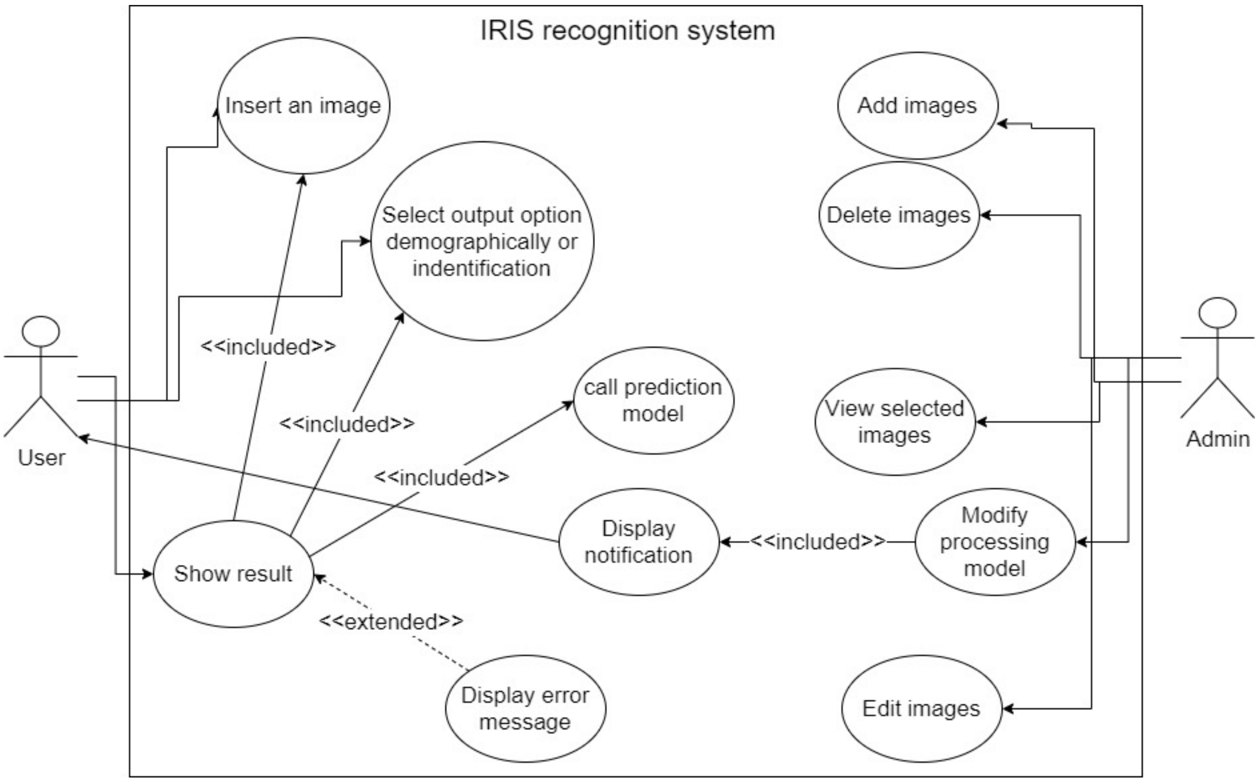

Figure 4.1: Structured Use Case Diagram for Iris Analyzer System

4.2 Sequence Diagram of Major User Cases

The system shall require authentication from the system administrator and give access to the system and Database. The System administrator shall choose training and testing sets from the images in the database. The sets will be used as a model to train the user inputs using deep learning algorithms and test them to get the output data.

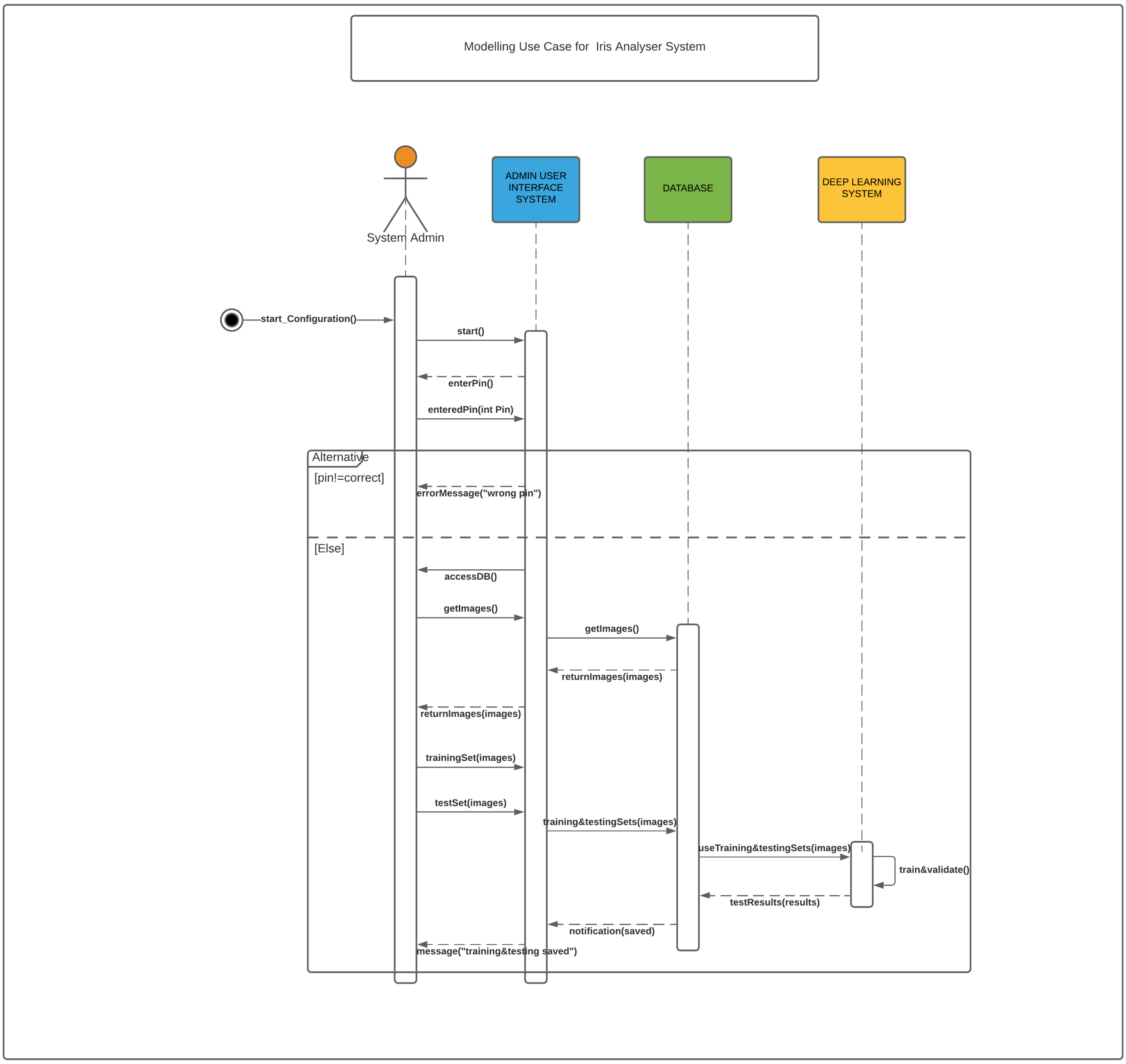

Figure 4.2: Modelling Image Use Case Sequence Diagram

The user shall choose an image and output option. The image shall be segmented first then passed through the Convolution Neural Network system for feature extraction. The features extracted shall be trained, validated and tested. The system through the test set shall give the output.

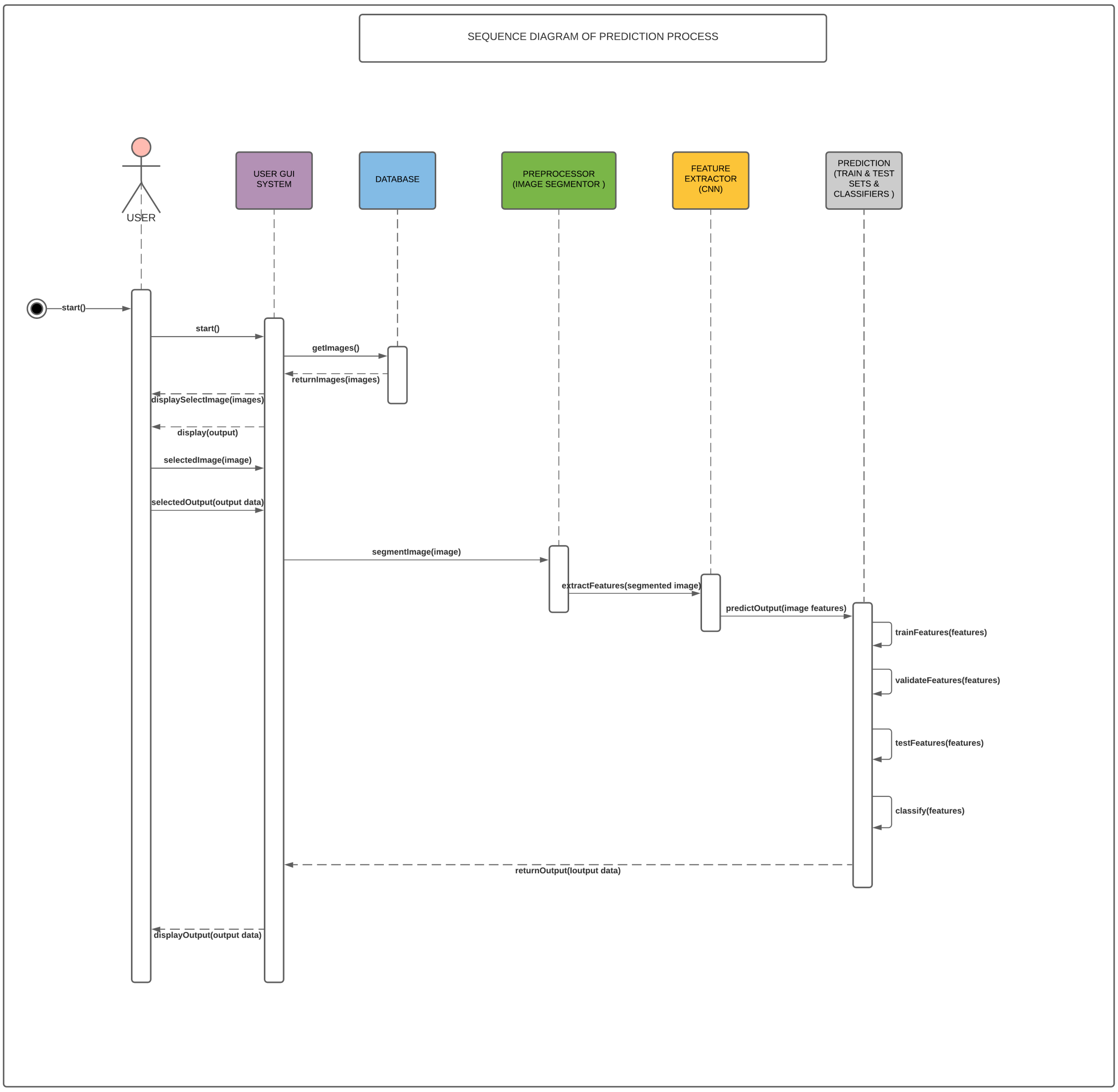

Figure 4.3: Prediction Image Use Case Sequence Diagram

4.3 Context Model

The context model shows the external and actors on the applications environment which are the Users, system administrator and Database.

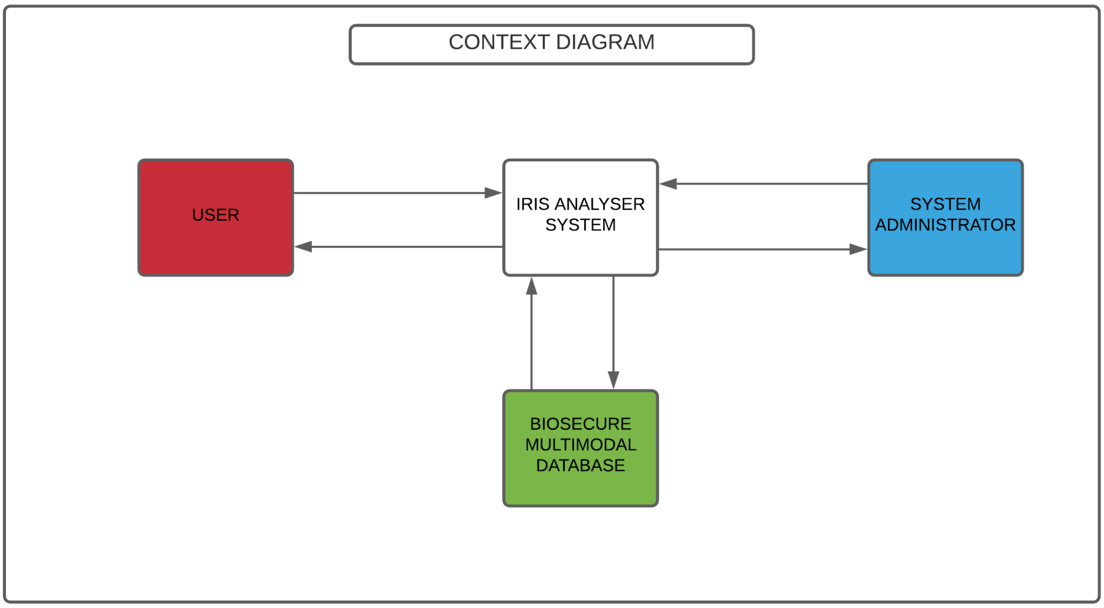

Figure 4.4: Context Model for the Iris Analyzer System

4.4 Architectural Model

This is a pipe and filter architecture diagram since the flow of the diagram is very straight forward. We have a data-set in which some parts of it get segmented, to build the ımage processıng model. After that, the user can insert an image or set of images. These pictures will be first filtered and segmented by ımage processıng model, then passed to the prediction model which is developed by deep learning for the final result.

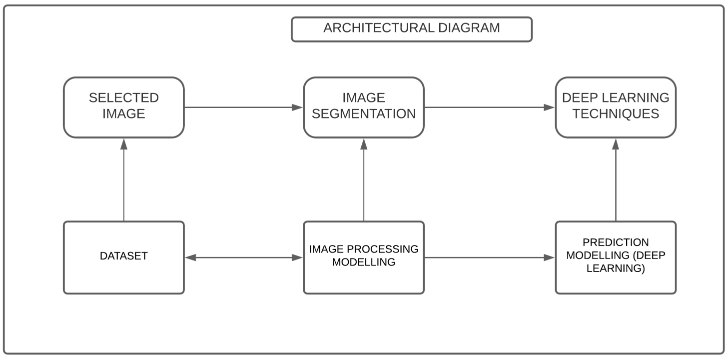

Figure 4.5: Architectural Model for the Iris Analyzer System

4.5 Process Model

The process model diagram shows the procedure to get the output from the inputted data. The user shall choose and image and the expected output wanted. The image shall be segmented and prediction done using deep learning methods to display the output.

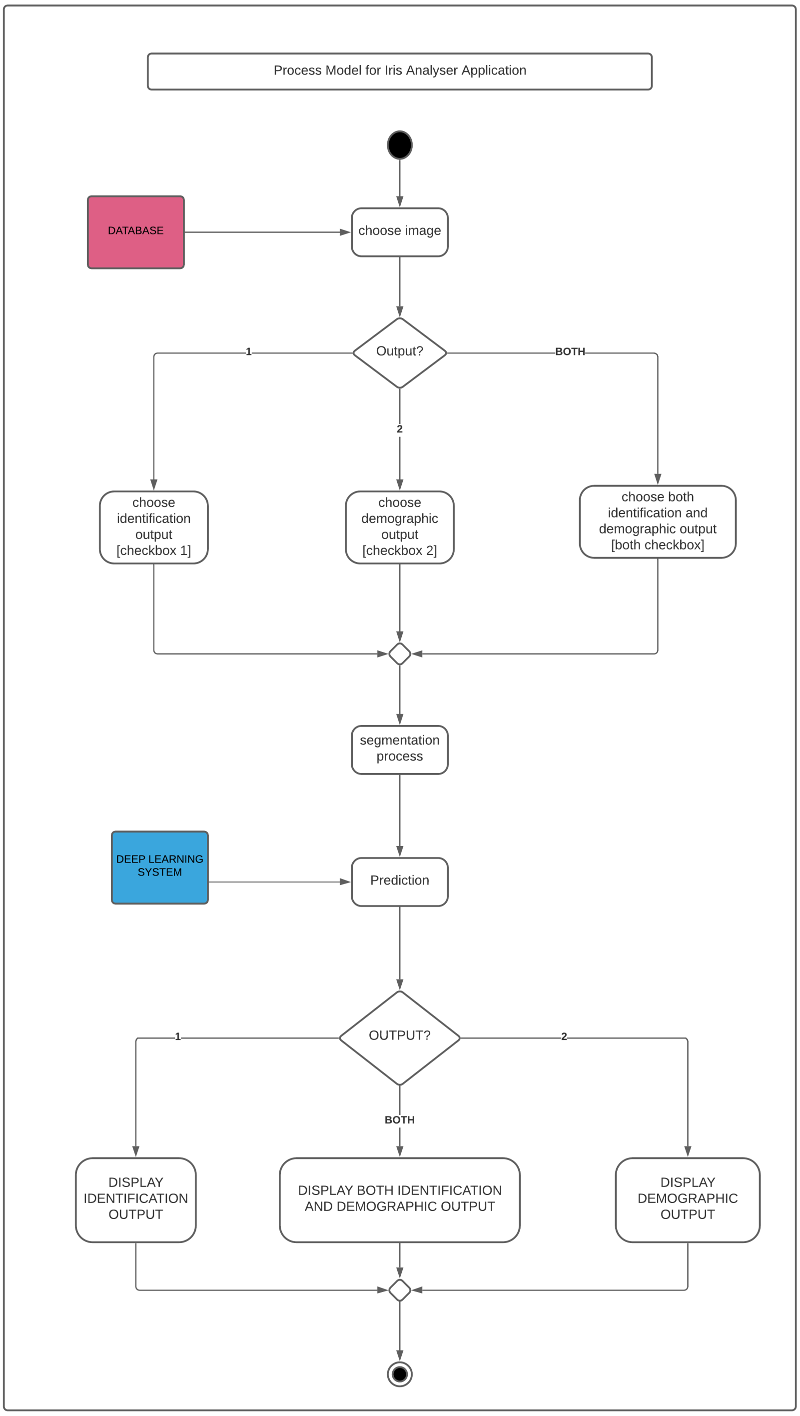

Figure 4.6: Process Model for the Iris Analyzer System

An image is taken through the first step which is segmentation process. The segmented image is then passed through the Convolution Neural Network system which extract the features. The features are then trained using the training set and validated. This validation process helps give information that may assist us with adjusting our hyperparameters. The trained data is then tested and if the accuracy threshold is above 85percent we accept the output otherwise it is trained again to improve the accuracy.

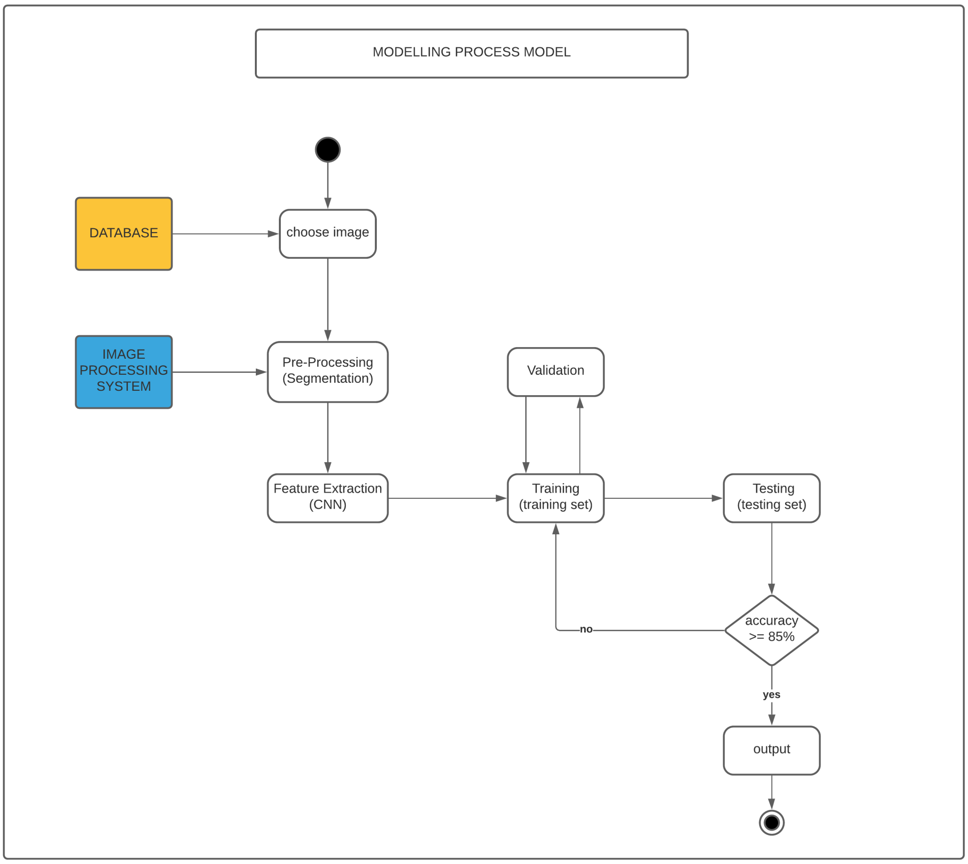

Figure 4.7: Modelling Process Model for the Iris Analyzer System

Chapter 5

Graphical User Interface

Below is a very early iteration of how the graphical user interface of the Iris Analyser application will look like. The interface is in early stages of design and is subject to many changes across the upcoming sprints.

5.1 Main Page GUI

The user will be given an option either to open the system administrator section or User section.

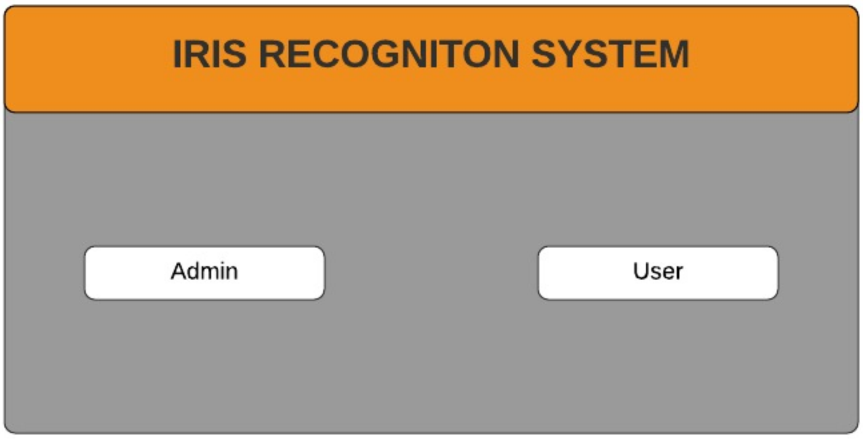

Figure 5.1: Graphical User Interface Main page view

5.2 User GUI

The User Interface works as follows; After the appearance of the interface below, the user must first click on "Choose eye image" in order to choose an eye image of their choice from the Iris Analyzer eye image database. If the user presses any other option before doing so an appropriate error message will be prompted. The user shall choose either the identification or demographic output or both. After having chosen an eye image, output and confirming, the user must press the "Start Configuration" button and will be immediately presented with the identification and demographic result in the "Identification output" and "Demographic output" boxes respectively.

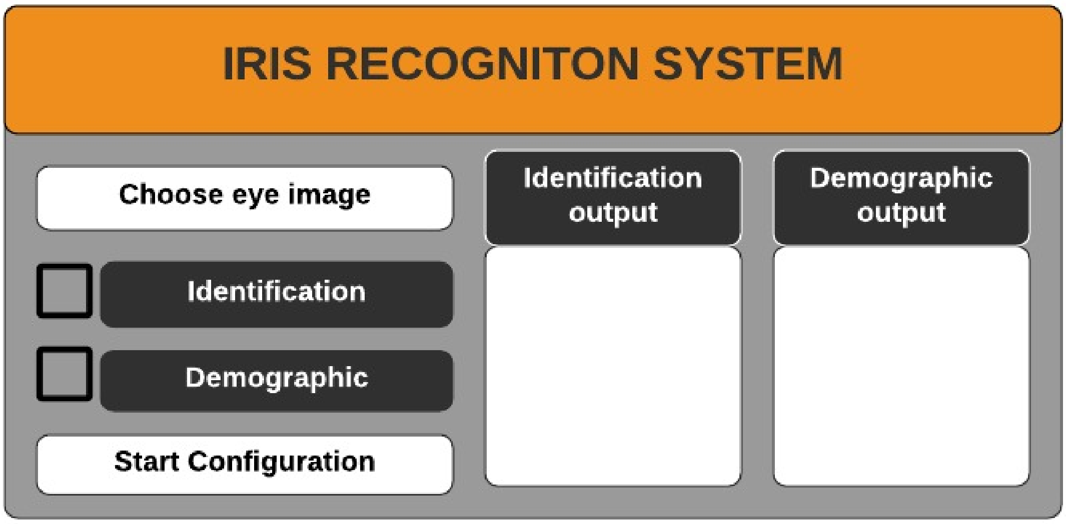

Figure 5.2: Graphical User Interface User view

5.3 System Administrator GUI

The system administrator interface works as follows; After the appearance of the interface below, the system admin shall be prompted to enter a 4-digit pin. The system shall move on to the next interface if pin is correct and if its wrong display an error message. In the next interface the system admin can add and delete images from the database.

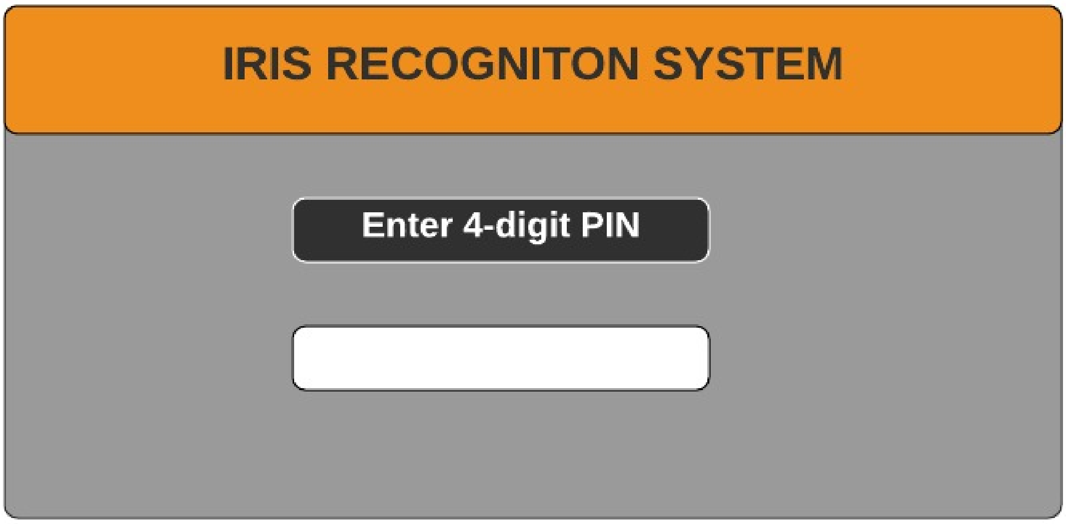

Figure 5.3: Graphical User Interface Authentication view

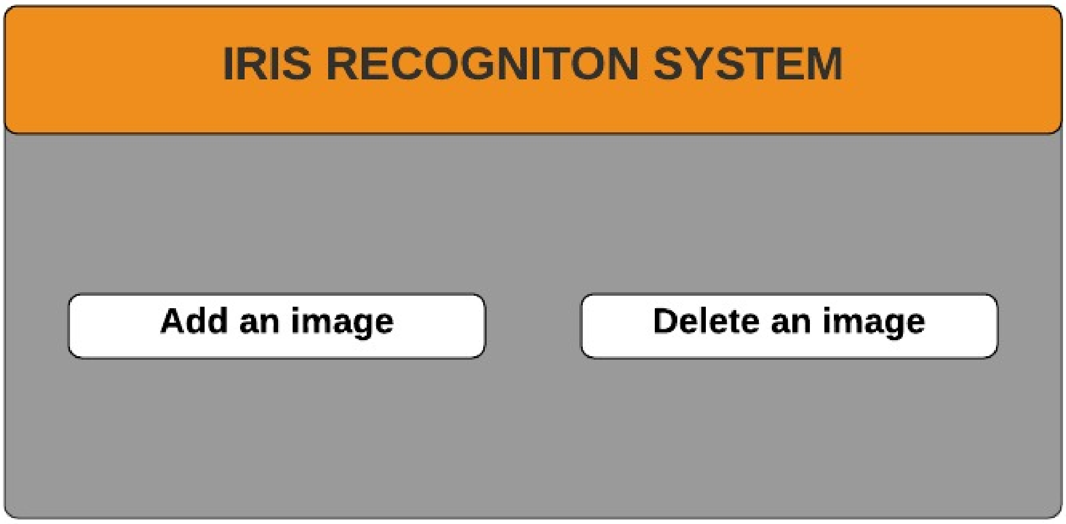

Figure 5.4: Graphical User Interface User database add/delete image view

Chapter 6

Agile Development with Scrum

6.1 Sprint Backlog

Table 6.1 shows the list of tasks completed so far and the time spent on each one. It also includes the priority level for each task. During the first sprint since we did not have many tasks to complete, the group was not divided in any particular way, and we shortly after decided on dividing the task of writing up a literature review corresponding to the two parts, we were completing research on at the time. After a consensus decision, Aref and Evans completed the first literature review while Sameh and Baraa took charge of the deep learning literature review that followed. However, for the research part itself, all 4 members worked on the Research for both Image processing techniques as well as Deep learning basics. During the second sprint, the tasks became more defined and we were able to distribute them more efficiently. As shown in the sprint backlog, each individual was responsible for a certain task from that point forwards, with the exception of the project report and deep learning research which was done as a group effort. We met for the research part once a week, but for the higher priority tasks we met twice a week to ensure all members were kept informed and up to date. At the end of this sprint, we now have our images segmented and ready to feed to the deep learning model.

Table 6.1: Sprint Backlog table

	Sprint Backlog Tasks	**Priority (High, Med or Low)**	**Responsible By**	**Time Spent (Hours)**	**Sprint**
1	Research (Image Processing techniques)	L	All	6	1
2	Write a Literature Review (on Image Processing)	L	Aref, Evans	3	1
3	Research (Deep Learning basics)	L	All	7	1
4	Write a Literature Review (on Deep learning basics)	L	Baraa, Sameh	3	1
5	Write a Project Reprort (1)	M	All	10	1
Sprint 1 (12.10.20 - 29.11.20) Gather enough information regarding deep learning in Iris Recognition studies					
6	Research (Deep Learning Algorithms)	L	All	5	2
7	Research (Image Segmentation Techniques)	M	Aref, Sameh	4	2
8	Research (Convolutional Neural Networks)	M	Baraa, Evans	5	2
9	Segmentation of Iris in Eye Images	H	Aref, Sameh	12	2
10	Preparation of Data Test and Training Sets	H	Baraa , Evans	2 (ongoing)	2
11	Write a Project Report (2)	M	All	10	2
Sprint 2(30.11.20 - 27.12.20) Segment Iris from eye images + work on data set preparation + continued research					

6.2 Sprint Burndown Chart

Table 6.2 shows the remaining time of the tasks of the first sprint over the first 7 weeks period. The sprint is shown against a general sprint guideline in red. It is important to note that although the group was not formed officially on week 1, we had already met at the beginning of the semester as a group and we had a general idea regarding the type of project we wanted to handle,therefore , there was basic research ongoing by each group member from the first week.

Table 6.2: Sprint 1 Burndown table

Task #	Sprint Backlog Tasks	Week 1 (12/10-18/10)	Week 2 (19/10-25/10)	Week 3 (26/10-1/11)	Week 4 (2/11-8/11)	Week 5 (9/11-15/11)	Week 6 (16/11-22/11)	Week 7 (23/11-26/11)	*Sprint 1 Ends*
	Guideline	**29**	**25**	**20**	**15**	**10**	**5**	**0**	
	Remaining Values	**29**	**23**	**15**	**13**	**10**	**7**	**0**	
1	Research (Image Processing techniques)	6	0	0	0	0	0	0	
2	Write a Literature Review (on Image Processing)	3	3	0	0	0	0	0	
3	Research (Deep Learning Basics)	7	7	2	0	0	0	0	
4	Write a Literature Review (on Deep Learning Basics)	3	3	3	3	0	0	0	
5	Write a Project Report (1)	10	10	10	10	10	7	0	

Figure 6.1 shows the graphical format of the above Sprint 1 Burndown chart with the red line indicating the guideline to follow, and the green line representing the group progress on the hours spent and remaining on the tasks.

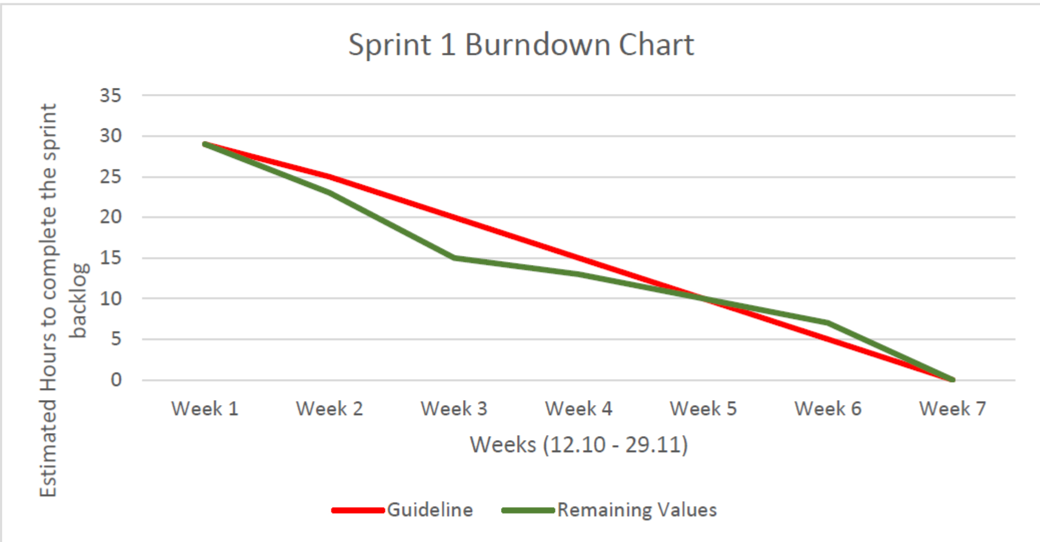

Figure 6.1: Sprint Backlog graph

Table 6.3 shows the remaining time of the tasks of the second sprint over the following 4 weeks, it is shown against a general sprint guideline

Table 6.3: Sprint 2 Burndown table

Task #	Sprint Backlog Tasks	Week 8 (30/11-6/12)	Week 9 (7/12-13/12)	Week 10 (14/12-20/12)	Week 11 (21/12-27/12)	Sprint 2 Ends
	Guideline	**46**	**30**	**15**	**0**	
	Remaining Values	**46**	**32**	**15**	**7**	
1	Research (Deep Learning Algorithms)	5	0	0	0	
2	Research (Image Segmentation Techniques)	4	0	0	0	
3	Research (Convolutional Neural Networks)	5	0	0	0	
4	Segmentaion of Iris in Eye Images	12	12	0	0	
5	Preparation of Data Test and Training Sets	10	10	10	7 (ongoing)	
6	Write a Project Report (2)	10	10	5	0	

Figure 6.2 shows the graphical format of the above Sprint 2 Burndown chart with the red line indicating the guideline to follow, and the green line representing the group progress on the hours spent and remaining on the tasks.

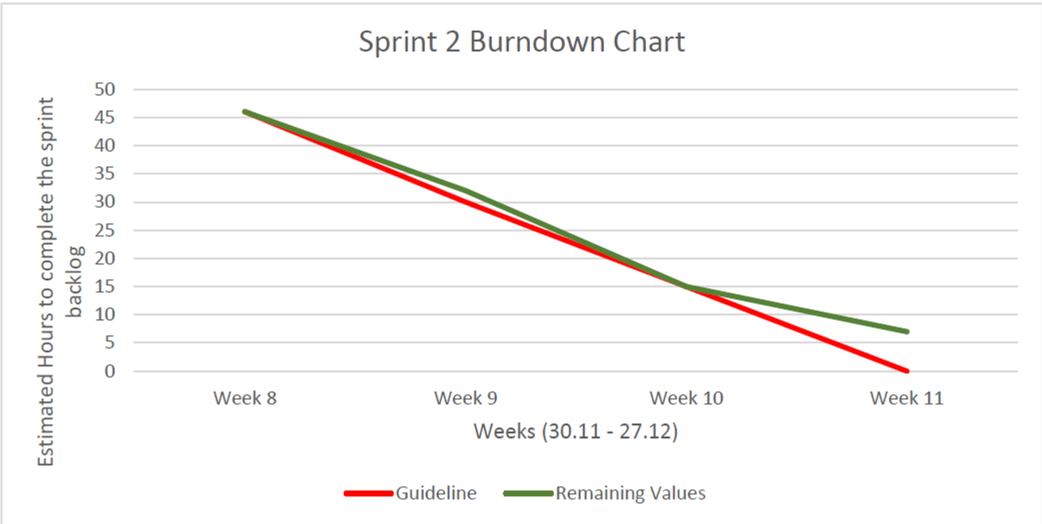

Figure 6.2: Sprint 2 Backlog graph

6.3 Sprint Review

Once we had completed the first sprint, which lasted for 6-7 weeks, we gathered strong knowledge regarding the very foundation and fundamentals of the image processing and deep learning basics we were planning to use in the second sprint further on, which is reflected in the first phase of the product backlog figure. This research-intensive work we completed revolved mostly around finding specific research papers which included many image segmentation processes as well as information on neural networks in general which were found through online means as well as being provided by our supervisor. The first sprint helped us transition easily to the next sprint which although had its fair share of research as well, this sprint was more focused on the application of the knowledge we have gathered up to this point. This helped us in completing the segmentation of the iris from the eye images and preparing the images in the correct format needed for the deep learning model we want to use. The research portion of the sprint could have been completed and managed better as it halted us from completing the division of the data test and training sets. We will keep in mind to complete research more efficiently for the upcoming future in order to match our sprint guideline.

6.4 Sprint Retrospective

At the beginning of the project, it was difficult to look at the shape of the sprint, although it looked like we were on track, there were very few and limited tasks to complete as well as the fact that most of the tasks revolved around individual research by each group member rather than strongly defined tasks which made the sprint chart obsolete. This caused us to encounter our first problem which was communication; although we met with our supervisor, the weekly meetings left many topics and questions regarding the project itself unanswered and unclear as we did not have a strong basis to work upon. This also resulted in miscommunication regarding a major aspect of the project as a whole: whether we were doing an application or research-based project. However, during the second sprint which occurred after the 7th week of the project development process, we learned to become more collaborative and work better together as a team. The first step was to make ourselves more experienced with the Trello Board and learn to use it in a correct manner. This immediately had a positive impact on the shape of the group, as we started to become more expressive in our opinions. On the other hand, we did not have to change everything, for example, our method of communication from start to present is done through Google Meets as we found it to be the easiest, fastest and most clear way of staying in touch within the group and with the supervisor as well. We also kept in touch with our supervisor more often through Messenger which was initially only used to set up and approve meeting dates and timings, however, on week 7 and on-wards we started to share more ideas and ask more questions through the Messenger medium as well where we were able to receive almost immediate answers. In summary;

What went well so far:

- Thorough research was made during the research-oriented weeks.
- Gathered many related works due to strong research.

- Were able to clearly grasp the end goal by the end of the project deadline.

What went wrong:

- Misunderstanding about the nature of the project (Research-based).
- Work done at the start and towards the end of the sprint more often, instead of being spread throughout the duration (caused fatigue).
- We were not able to complete the preparation of the data sets (training and test sets)
- Some tasks took longer than expected, mostly due to the 2nd sprint being much shorter (sprint 1 was 7 weeks, sprint 2 was only 4 weeks)

What was improved:

- Improved expressiveness during meetings and chats.
- Better use of communication software such as Trello, Messenger.
- Incorporate supervisor more strongly in development process.

What can still be improved:

- Manage time more efficiently in order to complete more demanding tasks on time whilst still doing research on the side.
- Look even further ahead in order to plan better and more accordingly.

Chapter 7

Estimation

7.1 Input

- **Inputs** : User image insertion-L, log-in form-M, Add images form(admin)-L, Delete images(admin)-L, View selection history(admin)-M, Edit processing model(admin)-H, edit images(admin)-H.
- **Outputs**: Showing the resulted image-H, predicted results-H
- **Inquiries**: User image insertion-L, admin image insertion-L, admin image deletion-L, Admin history display-M, admin modeling process modification-H, admin image modification-H, Password authentication-M.
- **Logical internal files**: IRIS Images dataset-M, Coordinates file-M

Figure 7.1 shows the

	Simple	Average	Complex
Number of User Inputs	3	2	2
Number of User Outputs			2
Number of User Inquiries	3	2	2
Number of Files		2	
Number of External Interfaces			

Figure 7.1: Estimation Input

7.2 Complexity Adjustment Table

		0	1	2	3	4	5
1.	Does the system require reliable backup and recovery?	○	○	○	○	●	○
2.	Are data communications required?	●	○	○	○	○	○
3.	Are there distributed processing functions?	●	○	○	○	○	○
4.	Is performance critical?	○	○	○	○	○	●
5.	Will the system run in an existing, heavily utilized operational environment?	○	○	○	●	○	○
6.	Does the system require on-line data entry?	●	○	○	○	○	○
7.	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	●	○	○	○	○	○

		0	1	2	3	4	5
8.	Are the master files updated on-line?	●	○	○	○	○	○
9.	Are the inputs, outputs, files, or inquiries complex?	○	○	○	○	○	●
10.	Is the internal processing complex?	○	○	○	○	○	●
11.	Is the code designed to be reusable?	●	○	○	○	○	○
12.	Are conversion and installation included in the design?	○	○	●	○	○	○
13.	Is the system designed for multiple installations in different organizations?	○	○	○	○	●	○
14.	Is the application designed to facilitate change and ease of use by the user?	○	○	○	○	●	○

Figure 7.2: Estimation Complexity Adjustment Table

Result	
Project Function Points	89.6999999

7.3 Line of Code

To calculate the number of code lines we chose Python as the language to use and had a unit size of 29 which we multiplied by ATFP. This is the average unit size for both.

Python 100%

Loc = 29 * 89.7 = 2601.3

7.4 Estimate the Efforts

7.4.1 Constructive Cost Model COCOM0 Basic

Development mode: organic.

a=2.4, b=1.05, c=0.38

Number of thousand delivered source instructions = (KDSI) = ATFP* Language unit size/1000 = 2601.3 /1000=2.6013

Effort in staff months =effort in man-moths=MM=a*KDSIb = 2.4*($2.6013^{1.05}$) = 6.54

The development time=TDEV=2.5*MM^c = 2.5($6.54^{0.38}$) $= 5.2 = 6 months$

7.4.2 jones's first -order effort estimation methods

We are building a shrink-wrap kind of software in an average organization.

Rough schedule estimation = ATFP^{exp} = $90^{0.042}$ $= 6.61 = 7 months.$

7.4.3 Schedule rule of thumb

Efficient Schedule (Shrink-Wrap)
    Schedule(Months)= 5.9   6 months
    Effort(Man-month) = 8

Table 8-9. Efficient Schedules

	Systems Products		Business Products		Shrink-Wrap Products	
System Size (lines of code)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)
10,000	8	24	4.9	5	5.9	8
15,000	10	38	5.8	8	7	12
20,000	11	54	7	11	8	18
25,000	12	70	7	14	9	23
30,000	13	97	8	20	9	32
35,000	14	120	8	24	10	39
40,000	15	140	9	30	10	49
45,000	16	170	9	34	11	57
50,000	16	190	10	40	11	67
60,000	18	240	10	49	12	83
70,000	19	290	11	61	13	100
80,000	20	345	12	71	14	120
90,000	21	400	12	82	15	140
100,000	22	450	13	93	15	160
120,000	23	560	14	115	16	195
140,000	25	670	15	140	17	235
160,000	26	709	15	160	18	280
180,000	28	910	16	190	19	320

Figure 7.3: Estimation Efficient Schedules

Chapter 8

Project Management

8.1 Milestones and Tasks

Table 8.1 shows all tasks to be completed by the end of the academic semester (January 2020) as well as the duration of each task, in addition to an estimate for the tasks in the upcoming weeks. The dependencies show which tasks depend on other tasks to be completed before they can be started. The group members responsible for the corresponding tasks are also mentioned. The colors indicate different sprint duration's (Blue = Sprint 1, Purple = Sprint 2, Green = Sprint 3).

Table 8.1: Tasks for the semester

Task #	Task	Duration (weeks)	Dependencies	Resonsible By
T1	Research (Image Proc Techniques) + Literature Review	2		All
T2	Research (Deep learning Basics) + Literature Review	2		All
T3	Write a Project Report (1)	2	T1,T2 (M1)	All
T4	Research (Deep Learning Algorithms)	2		All
T5	Research (Image Segmentation Techniques)	2		Aref, Sameh
T6	Research (Convolutional Neural Networks)	2		Baraa, Evans
T7	Segmentation of Iris region from Eye Images	1	T5 (M2)	Aref, Sameh
T8	Crop and Prepare Segmented Images	1	T7	Baraa, Evans
T9	Divide Images into Training and Test set	1	T8	Baraa, Evans
T10	Write a Project Report (2)	2	T4,T5,T6 (M2)	All
T11	Convolutional Neural Network (Code)	1	T6	Baraa, Evans
T12	Train Deep Learning Model	1	T7,T8,T11(M3)	All
T13	Gather Outcome (Accuracy) Information	1	T12	Baraa, Evans
T14	Analyze Deep Learning Model Results	1	T13	Aref , Sameh
T15	Compare Deep Learning Outcome vs Traditional Methods	1	T13 (M5)	Aref, Sameh
T16	Write a Project Report (3/Final)	2		All

Table 8.2 shows the milestones occurring throughout the academic semester. Milestones represent any significant goal or activity in the project cycle.

Table 8.2: Milestones for the semester

Milestone #	Milestone
M1	Project Fundamentals Understood
M2	Sufficient Fundamental Research for Deep Learning Model
M3	Images are Segmented and Ready (Finalized Format)
M4	Deep Learning Model Complete
M5	Results (Output) Analysis Ready

8.2 Gantt Chart

Figure 8.1 shows the visual representation of tasks that are scheduled throughout a time period (one semester's length). The tasks mentioned in table 8.1 are shown on the left of the chart and the corresponding scheduled dates are followed. The progress of each task and completion status of each milestone is shown with a percentage.In the case of this project cycle, the tasks are between 1 to 2 weeks and are represented with the bars shown on the timeline. Each milestone from table 8.2 is indicated with an orange diamond and each color corresponds to a different sprint duration (Blue = Sprint 1,Purple = Sprint 2, Green = Sprint 3).

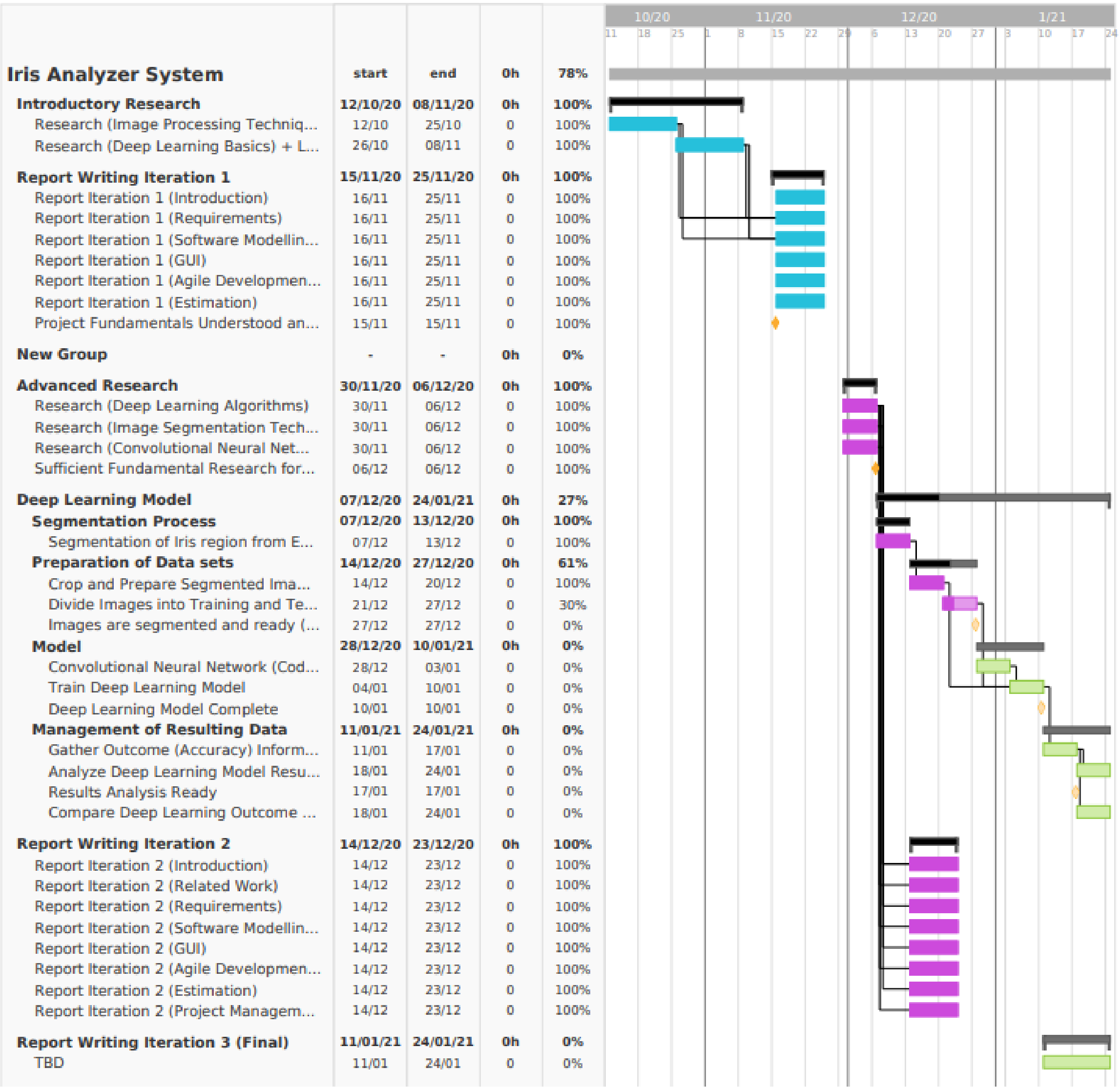

Figure 8.1: Gantt Chart

Chapter 9

References

1. M. Fairhurst, M. Erbilek and M. C. D. C. Abreu, "Exploring Gender Prediction from Iris Biometrics", International Conference of the Biometrics Special Interest Group (BIOSIG), pp. 1-11, 9-11 September, 2015, Darmstadt, Germany.

2. M. Da Costa-Abreu, M. Fairhurst, M. Erbilek, "Age predictive biometrics: predicting age from iris characteristics", Iris and Periocular Biometric Recognition, IET, Chap. 10, pp. 213- 234, 2017.

3. R. P. Wildes, "Iris recognition: an emerging biometric technology," in Proceedings of the IEEE, vol. 85, no. 9, pp. 1348-1363, Sept. 1997, doi: 10.1109/5.628669.

4. Bhattacharyya D., Das P., Bandyopadhyay S.K., Kim T. (2009) Feature Extraction for IRIS Recognition. In: Kim H., Kim T., Kiumi A. (eds) Advances in Security Technology. SecTech 2008. Communications in Computer and Information Science, vol 29. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-10240-0-3.

5. Daugman JG(1993) High confidence visual recognition of persons by a test of statistical independence. IEEE Trans Pattern Anal Mach Intel 15(11):1148-1161

6. Alaa S. Al-Waisy, Rami Qahwaji, Stanley Ipson, Shumoos Al-Fahdawi, Tarek A.M., Nagem "A muulti-biometric iris recognition system based on a deep learning approach". Pattern Anal Applic(2018) 21:783-802. https://doi.org/10.1007/s10044-017-0656-1

7. Tianming Zhao, Yuanning Liu, Guang Huo and Xiandong Zhu. "A Deep Learning Iris Recognition Method Based on Capsule Network Architecture." doi: 10.1109/ACCESS.2019.2911056

8. O. Russakovsky et al., "Imagenet large scale visual recognition challenge," Int. J. Comput. Vis., vol. 115, no.3, pp. 211-252, 2015

9. Kien Nguyen, Clinton Fookes, Arun Ross, Sridha Sridharan, "Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective". doi: 10.1109/ACCESS.2017.2784352

10. NEC. 8th October,2019. "What is Iris Recognition and how does it work?". https://www.nec.co.nz/market-leadership/publications-media/what-is-iris-recognition-and-how-does-it-work/