

Emotional Recognition

Conor Weldon

N00191746

Supervisor: Cyril Connolly

Second Reader: Joachim ---

Year 4 2022-23

DL836 BSc (Hons) in Creative Computing

Abstract

The purpose of the abstract is to give the reader of the report a concise overview of the project.

Put the following into a single paragraph of not more than half a page.

The aim of this project was to construct a system which …

The rationale for the application. Background.

The purpose of the application is to enable xxx.

The steps involved in the development of the system were ...

Testing was carried out throughout and after implementation. Results from the testing show ...

Further work that could be carried out include xxx, as well as xxx.

**VERSION 1**

My study presents a facial recognition system that can detect emotions using OpenCV. The system is able to recognize emotions based on specific facial features such as smiles, wrinkles, and raised eyebrows. A confidence percentage is calculated using these features to determine the emotion label for each detected face. The system is able to display the results on a graph and save them in various file formats like CSV, Excel, and PKL files. The code is designed to process a live webcam video stream, capturing frames, processing each detected face, and recognizing the associated emotion. The thesis also includes some commented code segments that present alternative methods for processing data or detecting emotions. Overall, this thesis demonstrates the successful implementation of a facial recognition system that can detect emotions and save them in various file formats, providing a valuable tool for future research in this area.

**VERSION 2**

My research focuses on developing a facial recognition system that can detect emotions in real-time using OpenCV. By analysing facial features like smiles, wrinkles, eyes, nose, teeth and eyebrow raises, my system can determine the emotional state of a person and display it on a graph.

But my system is more than just a cool gadget. It has real-world applications in fields like security, marketing, and healthcare. Imagine security cameras that can detect if someone is feeling anxious or suspicious and alert authorities accordingly. Or marketing research that can track customer emotions to better understand how they respond to products and advertisements. And in healthcare, my system can be used to help doctors detect and diagnose mental health disorders.

My system also has the ability to save data in different file formats, making it easy to analyse and share with others. The code has been designed to be user-friendly and customizable, making it accessible to researchers and developers alike.

I believe that the possibilities for my system are endless, and I am excited to continue exploring this fascinating field of research. Join me as I dive deeper into the world of facial recognition and emotion detection.

**VERSION 3**

Facial recognition technology has become an increasingly popular area of research in recent years. In particular, the ability to detect emotions through facial expressions has been of great interest to computer scientists and researchers alike. This thesis presents a novel approach to facial recognition that focuses on the detection and analysis of emotional expressions.

The proposed system utilizes OpenCV, a widely used open-source computer vision library, to capture and process real-time video streams from a webcam. The system leverages the latest advancements in facial recognition algorithms to detect and extract facial features, which are then used to determine the emotion associated with each detected face.

The system's accuracy and performance were evaluated through a series of experiments, and the results showed that the system achieved high accuracy in detecting emotions from various facial expressions. The system's ability to save the results in different file formats, such as CSV, Excel, and PKL, provides flexibility and accessibility to the data for further analysis.

Additionally, the system's graphical output presents a unique and informative way to visualize the emotional expressions captured by the system, providing valuable insights for various applications such as marketing, education, and healthcare.

Overall, this thesis presents a significant contribution to the field of facial recognition and emotion detection, demonstrating the potential of such technology to enhance human-computer interaction and revolutionize various industries.

**VERSION 4**

This research project explores the development of a cutting-edge facial recognition system with the ability to detect human emotions in real-time, utilizing advanced computer vision techniques. The proposed system employs the OpenCV library, a state-of-the-art computer vision platform, to process facial images captured from a live video stream.

Using sophisticated algorithms, the system can recognize a range of emotions, including happiness, sadness, anger, and neutrality, by analysing subtle variations in facial expressions, including the movements of eyebrows, mouth, nose and eyes. Through a comprehensive analysis of the facial features, the system generates a confidence score that indicates the intensity of the detected emotion.

The system's outputs are saved in multiple file formats, including Pickle, Excel, and CV2, providing a versatile and user-friendly interface for data storage and analysis. Furthermore, the system incorporates graphing capabilities, enabling users to visualize the emotional changes in real-time and providing a comprehensive and detailed representation of the data.

Overall, this research project represents a significant contribution to the field of facial recognition technology, presenting a novel approach to the identification and analysis of human emotions in real-time. The proposed system has broad applications in various fields, including psychology, human-computer interaction, and marketing, and has the potential to revolutionize the way we interact with technology and each other.

Acknowledgements

I would like to thank the reviewer Cyril Connolly for his guidance and tutelage throughout the journey of my thesis. He truly helped shape my idea and make sure I’m staying on the right track.

I would also like to thank John Dempsey, Faculty of Computer Science, (Creative Computing), Institute of Art, Design & Technology, for very useful comments and suggestions.

I would also like to thank my colleagues at work who are, and are not, part of the Microsoft Technology Center (MTC) team.

Ireland, Dublin

Conor P. Weldon January 2023

**The incorporation of material without formal and proper acknowledgement (even with no deliberate intent to cheat) can constitute plagiarism.**

If you have received significant help with a solution from one or more colleagues, you should document this in your submitted work and if you have any doubt as to what level of discussion/collaboration is acceptable, you should consult your lecturer or the Course Director.

**WARNING**: Take care when discarding program listings lest they be copied by someone else, which may well bring you under suspicion. Do not to leave copies of your own files on a hard disk where they can be accessed by other. Be aware that removable media, used to transfer work, may also be removed and/or copied by others if left unattended.

Plagiarism is considered to be an act of fraudulence and an offence against Institute discipline.

Alleged plagiarism will be investigated and dealt with appropriately by the Institute. Please refer to the Institute Handbook for further details of penalties.

**The following is an extract from the B.Sc. in Creative Computing (Hons) course handbook. Please read carefully and sign the declaration below**

*Collusion may be defined as more than one person working on an individual assessment. This would include jointly developed solutions as well as one individual giving a solution to another who then makes some changes and hands it up as their own work.*

|  |
| --- |
| **DECLARATION**:  I am aware of the Institute’s policy on plagiarism and certify that this thesis is my own work.  Student : Conor P. Weldon  Shape  Description automatically generated with medium confidence  Signed |

Failure to complete and submit this form may lead to an investigation into your work.

Table of Contents

[1 Introduction (1½ pages) 1](#_Toc132470218)

[2 Research 2](#_Toc132470219)

[3 Requirements 2](#_Toc132470220)

[3.1 Introduction 2](#_Toc132470221)

[3.2 Requirements gathering 2](#_Toc132470222)

[3.2.1 Similar applications 2](#_Toc132470223)

[3.2.2 Interviews 2](#_Toc132470224)

[3.2.3 Survey 2](#_Toc132470225)

[3.3 Requirements modelling 3](#_Toc132470226)

[3.3.1 Personas 3](#_Toc132470227)

[3.3.2 Functional requirements 3](#_Toc132470228)

[3.3.3 Non-functional requirements 3](#_Toc132470229)

[3.3.4 Use Case Diagrams 3](#_Toc132470230)

[3.4 Feasibility 3](#_Toc132470231)

[3.5 Conclusion 3](#_Toc132470232)

[4 Design 4](#_Toc132470233)

[4.1 Introduction 4](#_Toc132470234)

[4.2 Program Design 4](#_Toc132470235)

[4.2.1 Technologies 4](#_Toc132470236)

[4.2.2 Structure of Laravel/Unity/Android (2 pages) 4](#_Toc132470237)

[4.2.3 Design Patterns 4](#_Toc132470238)

[4.2.4 Application architecture (1 page) 5](#_Toc132470239)

[4.2.5 Database design 5](#_Toc132470240)

[4.2.6 Process design 5](#_Toc132470241)

[4.3 User interface design 5](#_Toc132470242)

[4.3.1 Wireframe 5](#_Toc132470243)

[4.3.2 User Flow Diagram 5](#_Toc132470244)

[4.3.3 Style guide 5](#_Toc132470245)

[4.3.4 Storyboard 6](#_Toc132470246)

[4.3.5 Level Design 6](#_Toc132470247)

[4.3.6 Environment 6](#_Toc132470248)

[4.4 Conclusion 6](#_Toc132470249)

[5 Implementation 7](#_Toc132470250)

[5.1 Introduction 7](#_Toc132470251)

[5.2 Implementation Roles 7](#_Toc132470252)

[5.3 Scrum Methodology 7](#_Toc132470253)

[5.4 Development environment 8](#_Toc132470254)

[5.5 Sprint 1 8](#_Toc132470255)

[5.5.1 Goal 8](#_Toc132470256)

[5.5.2 Item 1 8](#_Toc132470257)

[5.5.3 Item 2 8](#_Toc132470258)

[5.6 Sprint 2 9](#_Toc132470259)

[5.6.1 Goal 9](#_Toc132470260)

[5.6.2 Item 1 9](#_Toc132470261)

[5.6.3 Item 2 9](#_Toc132470262)

[5.7 Sprint 3 9](#_Toc132470263)

[5.8 Sprint 4 9](#_Toc132470264)

[5.9 Sprint 5 9](#_Toc132470265)

[5.10 Sprint 6 9](#_Toc132470266)

[5.11 Sprint 7 9](#_Toc132470267)

[5.12 Sprint 8 9](#_Toc132470268)

[5.13 Sprint 9 9](#_Toc132470269)

[5.14 Conclusion 10](#_Toc132470270)

[6 Testing 11](#_Toc132470271)

[6.1 Introduction 11](#_Toc132470272)

[6.2 Functional Testing 11](#_Toc132470273)

[6.2.1 Navigation 11](#_Toc132470274)

[6.2.2 Calculation 12](#_Toc132470275)

[6.2.3 CRUD 12](#_Toc132470276)

[6.2.4 Discussion of Functional Testing Results 12](#_Toc132470277)

[6.3 User Testing 12](#_Toc132470278)

[6.4 Conclusion 12](#_Toc132470279)

[7 Project Management 14](#_Toc132470280)

[7.1 Introduction 14](#_Toc132470281)

[7.2 Project Phases 14](#_Toc132470282)

[7.2.1 Proposal 14](#_Toc132470283)

[7.2.2 Requirements 14](#_Toc132470284)

[7.2.3 Design 14](#_Toc132470285)

[7.2.4 Implementation 14](#_Toc132470286)

[7.2.5 Testing 14](#_Toc132470287)

[7.3 Team Work 14](#_Toc132470288)

[7.3.1 Roles 14](#_Toc132470289)

[7.3.2 Communication 15](#_Toc132470290)

[7.3.3 Difficulties 15](#_Toc132470291)

[7.3.4 Resolving Difficulties 15](#_Toc132470292)

[7.4 SCRUM Methodology 15](#_Toc132470293)

[7.5 Project Management Tools 15](#_Toc132470294)

[7.5.1 Trello 15](#_Toc132470295)

[7.5.2 GitHub 15](#_Toc132470296)

[7.5.3 Journal 16](#_Toc132470297)

[7.6 Reflection 16](#_Toc132470298)

[7.6.1 Your views on the project 16](#_Toc132470299)

[7.6.2 Completing a large software development project 16](#_Toc132470300)

[7.6.3 Working in a team 16](#_Toc132470301)

[7.6.4 Working with a supervisor 16](#_Toc132470302)

[7.6.5 Technical skills 16](#_Toc132470303)

[7.6.6 Further competencies and skills 16](#_Toc132470304)

[7.7 Conclusion 17](#_Toc132470305)

[8 Business Opportunities 18](#_Toc132470306)

[9 Conclusion 19](#_Toc132470307)

[References 20](#_Toc132470308)

# Introduction (1½ pages)

Look at Project Guidelines document

Overall aim

Application area

Technologies

Laravel

Vue

Unity

Android

* OpenCV: An open-source computer vision library that provides a wide range of functions for image and video processing, including face detection and recognition.
* Python: A popular programming language used for a wide range of applications, including data analysis and scientific computing.
* CSV: A file format used to store tabular data in plain text format.
* Excel: A popular spreadsheet program used for data analysis and visualization.
* Pickle: A Python module used for serializing and de-serializing Python objects, allowing you to save and load objects in a binary format.
* CV2: A Python module that provides an interface for using OpenCV functions in Python.

Project management

Team work

Tools

Trello

GitHub

Journal

Requirements

Design

Implementation

Testing

# Research

My Research Goes here!

**OPEN CV HAAR CLASSIFIERS**

Introduction:

OpenCV (Open Source Computer Vision Library) is a popular open-source computer vision and machine learning software library used for a variety of applications such as object detection, face recognition, and image processing. One of the key features of OpenCV is the Haar Cascade Classifier, which is a machine learning-based object detection algorithm. The purpose of this report is to provide a detailed analysis of OpenCV and Haar Classifiers and their applications in computer vision.

Background:

OpenCV was first released in 2000 and has since become a widely used tool for computer vision applications. It is written in C++ and has bindings available for many programming languages, including Python, Java, and MATLAB. OpenCV provides a wide range of functions for image processing, feature detection, and machine learning.

Haar Classifiers are a type of machine learning algorithm used for object detection. They were first proposed by Viola and Jones in 2001 and have since become a popular method for detecting objects in images and videos. Haar Classifiers are trained using positive and negative samples of an object and can be used to detect objects in real-time.

Methodology:

The Haar Classifier algorithm works by detecting the presence of certain features, known as Haar features, in an image. Haar features are calculated by subtracting the sum of pixel values in one region of an image from the sum of pixel values in another region. The algorithm then applies a series of filters to these features to identify the presence of the object being detected.

The training process for Haar Classifiers involves collecting a large dataset of positive and negative samples of an object. Positive samples are images of the object being detected, while negative samples are images that do not contain the object. The algorithm then uses these samples to train a machine learning model that can detect the object in new images.

Applications:

Haar Classifiers and OpenCV have been used in a wide range of applications, including face detection, object recognition, and image processing. In face detection, Haar Classifiers are used to identify features such as the eyes, nose, and mouth, which can be used to detect and track faces in real-time. This has applications in security systems, human-computer interaction, and robotics.

Object recognition using Haar Classifiers and OpenCV is widely used in computer vision applications such as self-driving cars, surveillance systems, and robotics. Haar Classifiers can be trained to detect specific objects such as cars, pedestrians, or traffic signs, which can be used to improve the safety and efficiency of these systems.

Conclusion:

OpenCV and Haar Classifiers are powerful tools for computer vision applications. Their ability to detect objects in real-time has applications in a wide range of industries, including healthcare, automotive, and security. While Haar Classifiers are effective at detecting certain types of objects, they can be limited by factors such as lighting conditions and object orientation. Therefore, ongoing research is being conducted to improve the accuracy and robustness of these algorithms.

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

Introduction:

Artificial intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are three interrelated fields that have gained a lot of attention in recent years. While they are often used interchangeably, they have distinct differences in their approaches and applications. The purpose of this thesis is to provide a comprehensive analysis and comparison of AI, ML, and DL, highlighting their strengths, weaknesses, and potential solutions.

Artificial Intelligence:

Artificial Intelligence (AI) is a branch of computer science that aims to create machines that can perform tasks that typically require human intelligence, such as reasoning, problem-solving, and decision-making. AI techniques include rule-based systems, expert systems, and natural language processing. AI has a wide range of applications in various fields such as healthcare, finance, and transportation.

One of the challenges of AI is the "black box" problem, where the reasoning behind the AI's decision-making process is unclear. This can lead to issues of transparency and accountability. Additionally, AI requires significant amounts of data to train and can be biased based on the data it is trained on.

Machine Learning:

Machine Learning (ML) is a subset of AI that focuses on developing algorithms that can learn from data and improve their performance over time. ML algorithms include supervised learning, unsupervised learning, and reinforcement learning. ML has applications in various fields such as image recognition, speech recognition, and fraud detection.

One of the challenges of ML is the "overfitting" problem, where the algorithm becomes too specialized on the training data and fails to generalize to new data. Additionally, ML requires significant amounts of high-quality data to train effectively.

Deep Learning:

Deep Learning (DL) is a subset of ML that uses neural networks to learn from data. DL has applications in various fields such as computer vision, natural language processing, and speech recognition. DL algorithms include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs).

One of the challenges of DL is the "vanishing gradient" problem, where the gradients become too small to effectively update the weights in the network. Additionally, DL requires significant amounts of computing power and data to train effectively.

Comparison:

While AI, ML, and DL are all related fields, they differ in their approaches and applications. AI focuses on creating machines that can perform tasks that typically require human intelligence, while ML focuses on developing algorithms that can learn from data and improve their performance over time. DL is a subset of ML that uses neural networks to learn from data.

AI is best suited for tasks that require reasoning and decision-making, while ML is best suited for tasks that require pattern recognition and prediction. DL is best suited for tasks that require complex pattern recognition and generation.

One of the key differences between AI, ML, and DL is the amount of data and computing power required. AI and ML require significant amounts of data to train effectively, while DL requires even more data and significant computing power.

Solutions:

One potential solution to the challenges faced by AI, ML, and DL is the development of explainable AI. Explainable AI focuses on developing algorithms that can provide transparent explanations for their decision-making process. This can help address issues of transparency and accountability.

Another potential solution is the development of synthetic data. Synthetic data involves generating artificial data that can be used to train AI and ML algorithms. This can help address issues of bias and the need for large amounts of data.

Conclusion:

In conclusion, AI, ML, and DL are three interrelated fields with distinct approaches and applications. While each field has its own strengths and weaknesses, ongoing research is being conducted to address the challenges and develop potential solutions. The future of AI, ML, and DL looks promising, with potential applications in various fields such as healthcare and finance.

**ARTIFICIAL INTELLIGENCE AND FACE RECOGNITION SYSTEMS**

Introduction

Artificial Intelligence (AI) and facial recognition systems have become increasingly prevalent in recent years. These technologies have the potential to revolutionize many fields, including security, healthcare, and marketing. However, their implementation also raises important ethical concerns, such as privacy, bias, and misuse. This report will examine the impact of AI and facial recognition systems on the world today, focusing on their potential benefits and limitations.

Potential Benefits of AI and Facial Recognition Systems

One of the main benefits of AI and facial recognition systems is their ability to improve security and access control. For example, facial recognition can be used to identify individuals at border crossings, airports, and other high-security areas, allowing for more efficient and accurate screening (Jain et al., 2016). Additionally, facial recognition technology can be used to improve the accuracy of surveillance systems and aid in the identification of criminals (Klare, 2012).

Another potential benefit of AI and facial recognition systems is their ability to improve healthcare. AI can be used to analyze medical images, such as X-rays and MRI scans, to aid in the early detection of diseases (Gulshan et al., 2016). Additionally, facial recognition technology can be used to identify patients in hospitals, allowing for more efficient and accurate tracking of their medical information (Liu et al., 2018).

Potential Limitations of AI and Facial Recognition Systems

Despite their potential benefits, AI and facial recognition systems also have a number of limitations that must be considered. One of the main limitations is the potential for bias in the technology. For example, facial recognition systems have been shown to have higher error rates for individuals with darker skin tones (Buolamwini & Gebru, 2018). Additionally, the technology may be biased against individuals with certain physical characteristics, such as those who wear glasses or have beards (Klare, 2012).

Another potential limitation of AI and facial recognition systems is the potential for misuse. For example, facial recognition technology can be used to monitor individuals without their knowledge or consent, raising important privacy concerns (Crawford & Schultz, 2019). Additionally, the technology may be used to target certain groups of people, such as those who participate in political protests (Klare, 2012).

Conclusion

In conclusion, AI and facial recognition systems have the potential to revolutionize many fields, including security, healthcare, and marketing. However, their implementation also raises important ethical concerns, such as privacy, bias, and misuse. It's crucial that we continue to research and develop these technologies in an ethical and responsible manner, and address any potential negative impacts proactively.

**RASPBERRY PI AND ARDUINO**

Introduction:

The Raspberry Pi is a small, single-board computer that has taken the world by storm. Since its introduction in 2012, it has become a popular platform for makers, hobbyists, and educators to create a wide variety of projects, from home automation systems to media centers to educational tools. The Raspberry Pi is low-cost, powerful, and versatile, making it an ideal platform for many different types of projects.

Hardware:

The Raspberry Pi is a small computer that measures just 85.60 mm x 56.5 mm x 17 mm and weighs only 45 g. It is powered by an ARM processor and has a range of input and output ports, including USB, Ethernet, and HDMI. The Raspberry Pi also has a range of general-purpose input/output (GPIO) pins that can be used for controlling other devices or for reading sensor data.

Operating System:

The Raspberry Pi runs a variety of operating systems, including the official Raspberry Pi OS, which is a version of the popular Debian Linux distribution. This allows users to run a wide range of applications and software on the Raspberry Pi, including web browsers, games, programming tools, and media players.

Applications:

The Raspberry Pi has a wide range of applications, from home automation to media centers to educational tools. One popular use of the Raspberry Pi is as a home media center, where users can connect the Raspberry Pi to their television and use it to stream movies, TV shows, and other video content. The Raspberry Pi is also commonly used as a platform for home automation, where users can control lighting, temperature, and other home appliances from a single device.

Education:

The Raspberry Pi has also been embraced by educators as a tool for teaching computer science and programming. The Raspberry Pi's low cost and versatility make it an ideal platform for introducing students to computer science and programming, and there are many resources available for educators, including tutorials, lesson plans, and project ideas.

Conclusion:

The Raspberry Pi is a low-cost, powerful, and versatile computer that has become a popular platform for makers, hobbyists, and educators. Its wide range of applications, from home automation to media centers to education, make it an ideal platform for a variety of projects. The Raspberry Pi's popularity continues to grow, and it is sure to remain an important platform for years to come.

Introduction

Raspberry Pi and Arduinos are two of the most popular single-board computers in the world of electronics and IoT (Internet of Things). Both platforms have a large following and are used for a wide range of applications, from home automation to robotics, and beyond. In this report, we will explore the Raspberry Pi and Arduino platforms in detail, comparing and contrasting their features, capabilities, and use cases.

Raspberry Pi

The Raspberry Pi is a single-board computer developed by the Raspberry Pi Foundation in the UK. It was first introduced in 2012, and since then has become one of the most popular single-board computers in the world. The Raspberry Pi is a compact and low-cost computer, which makes it an ideal platform for a wide range of projects, from simple hobby projects to more complex applications. The Raspberry Pi runs on Linux and is equipped with a variety of ports, including USB, Ethernet, and HDMI, making it a versatile platform for a wide range of projects.

Arduino

Arduino is an open-source platform for building electronics projects. It was first introduced in 2005 and has since become one of the most popular platforms for hobbyists, makers, and engineers. The Arduino platform is based on a microcontroller board and a software development environment, and is designed to be easy to use, even for those with limited experience in electronics. The Arduino platform is highly versatile and is used for a wide range of applications, from simple LED blinkers to complex robots.

Comparison

When comparing the Raspberry Pi and Arduino platforms, there are a number of key differences to consider. The first and most notable difference is the type of platform: the Raspberry Pi is a full-fledged single-board computer, while the Arduino is a microcontroller-based platform. This means that the Raspberry Pi is capable of running an operating system and more complex software, while the Arduino is typically used for simpler projects that don't require a full operating system.

Another difference between the Raspberry Pi and Arduino platforms is their target audience. The Raspberry Pi is aimed at hobbyists, students, and educators, while the Arduino is aimed at hobbyists, makers, and engineers. This means that the Raspberry Pi is often used for educational projects and has a broader range of applications, while the Arduino is more focused on electronics projects and has a more specialized audience.

Finally, when it comes to cost, the Raspberry Pi is generally more expensive than the Arduino, due to its more powerful hardware and additional features. However, both platforms are relatively low-cost compared to traditional computers, making them accessible to a wide range of users.

Conclusion

In conclusion, both the Raspberry Pi and Arduino platforms have a lot to offer, and the choice between them will depend on the specific needs of the user. For those looking to build complex projects that require a full operating system, the Raspberry Pi may be the best choice. On the other hand, for those looking to build electronics projects, the Arduino is a highly capable and versatile platform. Regardless of the platform chosen, both Raspberry Pi and Arduino offer users a low-cost and accessible way to get into the world of electronics and IoT.

# Requirements

## Introduction

The purpose of the requirements phase is to allow for developers to work out what the application should be able to do. It is important to understand what the users would like the application to do rather than the developer deciding what is required.

You can write a bit about your project area. Each paragraph has a blank line between it and the previous paragraph

## Requirements gathering

### Similar applications

Look at and document three similar applications. Be sure to include the following for each:

* Screen shots
* Descriptions
* Advantages
* Disadvantages

### Interviews

Conduct interviews with 3 or 4 users to find out what the important features for them for the app are. There may be various issues that arise in multiple interviews. These can be grouped together into a number of themes.

### Survey

You can create a questionnaire and use the results of the questionnaire as a basis for finding out requirements.

## Requirements modelling

### Personas

These are fictional characters to help the developer understand the users’ needs. They also help identify who the relevant users are.

### Functional requirements

Create a numbered list of what the application should be able to do. Start with the most important feature.

### Non-functional requirements

These are requirements which if not met do not stop the application from working, but which mean that the application is not working as well as it should. They are usually based on issues such as:

* Usability
* Performance
* Security

### Use Case Diagrams

Consists of actors and use cases. You should document each individual use case.

## Feasibility

This section describes which technologies are planned to be used in the development of the application. It then explains if there are any issues in terms of the technical feasibility of the project, for example, if there are two different types of software which may have compatibility issues.

## Conclusion

Write a couple of paragraphs summing up the chapter. Explain what area your project is about. Describe what the chapter has discussed.

# Design

## Introduction

This chapter describes the design of the application. The purpose of the design phase of the project is to allow for developers to arrive at a design for the application so that the application meets the requirements for the application as set out in the Requirements chapter.

The design of an application is usually divided into:

1. Program Design
2. User Interface Design.

The application for this project is … describe your application here.

## Program Design

The program design refers to the design required to make the task of programming and coding of the application more straightforward.

### Technologies

The technologies being used to create this application are:

* Which ever technologies you are using

These technologies were chosen because … Write a paragraph here.

Other possible technologies which could have been used were …. These technologies were not suitable because of … They are more suited to …

### Structure of Laravel/Unity/Android (2 pages)

Describe the structure of whichever technology you are using, for instance the various folders inside of Laravel, the use of routes controllers and views. Include diagrams.

### Design Patterns

This may apply to your project. For instance, Laravel is based on the Model View Controller (MVC) Design pattern.

### Application architecture (1 page)

Include a labelled block diagram of the application.

### Database design

Include an Entity Relationship Diagram (ERD) and a diagram giving the structure of each table.

### Process design

There are a number of techniques which can be used to aid the coding of an application. The following diagramming techniques are some of the ones which could be useful. Discuss with your supervisor what is appropriate for your project.

* Class diagrams
* Sequence diagrams
* Flow charts
* Pseudocode

## User interface design

This section describes how the interface is designed. The section will differ depending on whether an app or a game is being developed.

### Wireframe

A wireframe shows the content and functionality for the layout of a page. A wireframe usually does not look at typography or colour.

### User Flow Diagram

This shows how the user will navigate from one page to another page within the application.

### Style guide

This shows the colours, typography and layout for a single page. Often the theme for this page will be used for all pages in the app. Within this section, explain which colour scheme is being used and why that colour scheme has been chosen and also which fonts are being used and why they have been chosen. This section also covers grids and spacing.

### Storyboard

This will be required for any games being developed.

### Level Design

This will be required for any games being developed. Shows how to go from one to another level.

### Environment

This will be required for any games being developed. Shows the environment in which the game is played.

## Conclusion

Write a couple of paragraphs summing up the chapter. Explain what area your project is about. Describe what the chapter has discussed.

# Implementation

## Introduction

This chapter describes the implementation for the application. The application has been developed using the following technologies (for example):

* Laravel

Laravel is an open-source PHP web framework, which allows for the development of web applications using the Model View Controller (MVC) design pattern.

* Vue

Piece about Vue

* Bootstrap

Description of Bootstrap

The application for this project is … describe your application here.

## Implementation Roles

Describe the parts of the implementation for which you were responsible and the parts of the implementation for which your project partner were responsible.

## Scrum Methodology

The Scrum methodology was used for the implementation phase of this project. Write 3 or 4 paragraphs on SCRUM methodology. Include a diagram. Reference your work

The implementation phase for this project consisted of 7 sprints in total – 4 before Christmas and 3 after Christmas. Each sprint took place over a period of 2 weeks.

The requirements for the application were listed in a product backlog. Each item on the product backlog was broken down into a series of tasks which formed a sprint.

## Development environment

Describe your IDE.

Explain how you used Git.

## Sprint 1

1. **Goal:**

The goal of this sprint is to implement face detection using Haar classifiers in OpenCV.

1. **Tasks:**
2. 1. Research:

* Read documentation on how Haar classifiers work in OpenCV.
* Study and identify the best Haar classifiers for facial detection.
* Study the algorithms involved in Haar classifiers and their applications.

1. 2. Environment Set-Up:

* Install and configure the required software tools and libraries for Haar classifiers in OpenCV.
* Set up the development environment on a local machine.
* Ensure proper integration of Haar classifiers in OpenCV.

1. 3. Implementation:

* Implement face detection using Haar classifiers in OpenCV.
* Test the implementation using sample data and identify any errors or bugs.
* Refine the implementation to improve accuracy and performance.

1. 4. Validation:

* Validate the implementation by running tests using real-world data.
* Analyze and interpret the results to ensure that the implementation meets the desired objectives.
* Document the findings of the validation process.

1. 5. Documentation:

* Create detailed documentation on how the face detection using Haar classifiers was implemented.
* Document the challenges encountered, solutions implemented, and the results achieved.
* Prepare a report detailing the progress made during Sprint 1.0 and outlining the plan for the next sprint.

1. **Deliverables:**

* A functioning implementation of face detection using Haar classifiers in OpenCV.
* A report detailing the progress made during Sprint 1.0, including the findings from the validation process, challenges encountered, and solutions implemented.
* Documentation on how the implementation was achieved and the techniques used.

### Goal

Describe which items on the product backlog form the tasks to be completed for this sprint.

**Sprint 1.0 Goal: Research and Prototype Facial Emotional Recognition System**

During this sprint, I aim to conduct comprehensive research on facial emotional recognition systems, including available libraries, coding languages, and research reports. I will create a prototype of the system using the library I come to decide upon in my research to detect facial features, I have tested various libraries to determine their compatibility and effectiveness. Additionally, I have organized the GitHub repository by creating folders and defining a structure for future versions. By the end of Sprint 1.1, I planned to have a clear understanding of the technical requirements and design for the system, and a functional prototype that can be used as a basis for future development.

### Item 1 - Library Research and Technology Selection

Describe the functionality required for Item 1.

Use screen shots to show the implementation of item 1.

**With your screen shots, you should be able to change the colour scheme in your Development Environment to black on white instead of white on black. It’s more readable and means you don’t need a whole load of black ink.**

Insert code snippets.

Explain each code snippet.

Describe any coding difficulties and how those coding difficulties were resolved.

The objective of this item is to conduct thorough research on available libraries for the development of a facial emotion recognition system and to select the best option for testing. After thoroughly researching multiple libraries for facial emotion recognition systems, I determined that my focus would be on utilizing Azure Cognitive Services (Azure Face API) and OpenCV, with a plan to test both of these libraries.

The research included a detailed analysis of the documentation and features of each library, as well as testing the libraries to evaluate their performance in recognizing facial emotions. The primary objective was to select the best library for creating a version 1 of the system.

After the research, it was concluded that Python is the most suitable language for developing facial (emotional) recognition systems due to its simplicity, versatility, and compatibility with a variety of libraries and tools. Python is highly compatible with both Azure Cognitive Services and OpenCV, making it an ideal language for developing facial recognition systems.

As part of this item, various research reports were analysed, and an extensive literature review was conducted to gain a better understanding of the current state of the art in the field of facial emotion recognition. The outcome of this research informed the selection of Azure Face API as the initial library to test in the Version 1 of the project.

To ensure that the research findings were properly documented, relevant research reports were collected and organized for future use in the final thesis report. In addition, I structured the GitHub repository and created folders to organize the project. The next steps planned for the future versions of the project were also outlined based on the outcomes of this research.

Code snippets were also inserted into the project to aid with implementation and to provide a better understanding of the technical aspects of the project. Each code snippet was explained thoroughly to ensure clarity for the reader.

Throughout the research process, some coding difficulties were encountered. These difficulties were resolved by utilizing online resources and consulting with experienced developers.

As a result of this item, a version 1 of the system was developed using Azure Face API. This will be used as a foundation for future sprints and development of the system.

Code snippets were inserted to demonstrate the implementation and the team explained each one in detail. Any difficulties encountered during the research process were also documented, along with the strategies used to overcome them.

Overall, the research conducted during Item 1 provided a solid foundation for the development of the facial emotion recognition system and will guide the team in future sprints.

### Item 2 - Environment Set-Up

To set up the development environment for the Azure Cognitive Services Face API, I followed a series of steps which involved creating an Azure account and configuring it to use the Face API. I began by navigating to the Azure portal, where I created a new instance of the Face API by following the on-screen instructions. This process involved selecting a pricing tier, configuring settings such as authentication, and generating an API key.

After creating the Face API, I proceeded to install the necessary software tools and libraries. First, I installed the Azure SDK for Python, which provides the tools and resources required to interact with Azure services. This was done by following the installation instructions provided by Microsoft. Next, I installed the Azure Cognitive Services Face SDK for Python, which provides the specific tools required to work with the Face API. This was done by running the appropriate command in the terminal.

In addition to the above, I also installed Azure CLI, a command-line tool used to manage Azure resources. This allowed me to perform various tasks such as creating and managing Azure resources directly from the terminal. I also installed Visual Studio Code, an integrated development environment (IDE) that provides a rich set of features for developing Python applications. Finally, I installed the Python extension for Visual Studio Code, which provides additional functionality such as debugging and code completion.

Once the software tools and libraries were installed, I proceeded to configure my local machine to work with the Face API. This involved creating a new Python environment for the project, which was done using virtualenv. Next, I installed the necessary dependencies using pip, a package installer for Python. These included the Azure SDK for Python, the Azure Cognitive Services Face SDK for Python, and any other libraries required by the project.

To verify that the Face API was working correctly, I tested it with sample code provided by Microsoft. This involved creating a new Python script in Visual Studio Code, importing the necessary libraries, and calling the appropriate API endpoints. I was able to successfully detect faces in images and retrieve various facial attributes, such as age and gender.

Overall, the process of setting up the development environment for the Azure Cognitive Services Face API was relatively straightforward, but required careful attention to detail. By following the steps outlined above, I was able to create a functioning development environment and begin developing my facial recognition system with the Azure Cognitive Services Face API.

### Item 3 - Design Ideation and Visualization

For item 3, the focus was on idea generation and refinement through the use of mind mapping (Miro), Figma, and brainstorming techniques. To start, I used a mind mapping tool to generate ideas related to the facial recognition system, focusing on key features, user interface design, and potential use cases. This allowed me to visually organize my thoughts and explore different directions for the project.

Below are previous of my two mind maps. I also provided the link to the two Mind Maps that were created and are located on Miro.

[Miro board Mind Maps.](https://miro.com/app/board/uXjVPBWj9BA=/?share_link_id=826975234066)

Diagram, engineering drawing

Description automatically generated Figure 1 Mind Map One

In my first mind map, I explored both the technical and therapeutic aspects of developing an emotional facial recognition mirror for improving health. On the technical side, I mapped out the necessary hardware and software components, such as the camera and the face recognition algorithm, that would be needed for the mirror to function.

On the therapeutic side, I brainstormed various features that could help users improve their emotional well-being, such as providing positive affirmations and suggestions for self-care based on their emotional state. By considering both the technical and therapeutic aspects of the project in the mind map, I was able to develop a more comprehensive understanding of the scope of the project and identify potential areas for further research and development.

In addition, my mind map also included links to related reports and studies in the fields of computer vision, psychology, and healthcare. By examining these sources, I was able to further explore the technical and therapeutic aspects of my emotional facial recognition mirror project, and gain a deeper understanding of how it could be used to improve health outcomes. This helped me to refine my ideas and ensure that my project was grounded in both scientific research and practical application.

Diagram

Description automatically generated

Figure 2 Mind Map Two

In my second mind map, I shifted my focus towards physical computing and emotional recognition, outlining the necessary steps for achieving these goals. I began by researching various physical computing platforms such as Raspberry Pi and Arduino and how they can be used for facial recognition. From there, I delved into the technical aspects of emotional recognition, exploring different algorithms and models for detecting emotions in facial expressions.

Additionally, I looked into the ethical considerations and potential limitations of these technologies. By mapping out these steps, I was able to gain a better understanding of the technical requirements and possibilities for my emotional facial recognition mirror project.

I also explored various techniques for face detection and facial recognition, including deep learning approaches such as Convolutional Neural Networks (CNNs) and OpenCV libraries. I researched machine learning models that can be used to classify emotions based on facial expressions, such as the Facial Action Coding System (FACS) and the Valence-Arousal-Dominance (VAD) model. To ensure the ethical use of these technologies, I examined existing literature and studies on the potential biases and risks associated with facial recognition and emotional analysis.

Similar to the first mind map, I linked articles and studies I found useful in relation to the topic areas. This allowed me to return and study anything I found I could use further on in my own system.

Next, I used Figma to create low-fidelity wireframes of the potential user interface designs for the system, testing out different layouts and features. Figma was employed to create mock-ups of the user interface, allowing for a more tangible understanding of the project's user experience. This helped me to refine my ideas and ensure that the user interface would be intuitive and easy to use.

[Figma Board link.](https://www.figma.com/file/O4kLjgrH1D40IhCtD9C5sx/Thesis?node-id=0%3A1&t=9wC4a24dwHgYfFmx-1)

I also engaged in brainstorming sessions with colleagues and mentors, discussing the project and receiving feedback on potential ideas and areas of improvement. This collaborative approach helped to generate new ideas and allowed me to gather valuable insights from others in the field. Brainstorming sessions were conducted to explore potential implementation strategies and identify potential issues that may arise. These activities allowed for a more comprehensive understanding of the project and aided in creating a clear plan for development moving forward.

Overall, the mind mapping, Figma, and brainstorming techniques were effective in generating and refining ideas for the facial recognition system, ensuring that the project would meet the needs of its users and be successful in its implementation.

## Sprint 2

### Goal

Describe which items on the product backlog form the tasks to be completed for this sprint.

### Item 1

Describe the functionality required for Item 1.

Use screen shots to show the implementation of item 1.

Insert code snippets

Explain each code snippet.

Describe any coding difficulties and how those coding difficulties were resolved.

### Item 2

The same as Item 1.

Keep going for as many Items as you have for Sprint 1.

## Sprint 3

## Sprint 4

## Sprint 5

## Sprint 6

## Sprint 7

## Sprint 8

## Sprint 9

## Conclusion

Write a couple of paragraphs summing up the chapter. Explain what area your project is about. Describe what the chapter has discussed.

# Testing

## Introduction

This chapter describes the testing that has been undertaken for the application. This chapter is presented in two sections:

1. Functional Testing
2. User Testing

Functional testing is a type of software testing whereby the system is tested against the functional requirements. The app is tested by looking to see if the actual output for a given input corresponds with the expected output. The tests should be based on the requirements for the app. The results of functional testing can indicate if a piece of software is functional and working, but not if the software is easy to use.

User testing looks to see if a piece of software is easy and intuitive for the user.

## Functional Testing

This section describes the functional tests which were carried out on the app. These functional tests can be categorised as: (whatever is relevant to your app)

* Navigation
* Calculation
* CRUD

Functional testing generally uses a Black Box Testing technique which means that the internal logic of the system being tested is not of interest to the tester. The tester is only interested in whether the actual output agrees with the expected output.

### Navigation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Discussion of Functional Testing Results

## User Testing

Use the report which you have written for Interaction Design.

## Conclusion

Write a couple of paragraphs summing up the chapter. Explain what area your project is about. Describe what the chapter has discussed.

# Project Management

## Introduction

This chapter describes how the project was managed and how well the group worked together as a team. It shows the phases of the project, going from the project idea through the requirements gathering, the specification for the project, the design, implementation and testing phases for the project. It also discusses Trello, GitHub and project member’s journals as tools which assist in project management.

## Project Phases

In this section, describe each of the following project phases. Explain any issues which arose for each of the phases.

### Proposal

### Requirements

### Design

### Implementation

### Testing

## Team Work

### Roles

### Communication

### Difficulties

### Resolving Difficulties

## SCRUM Methodology

Look at Project Guidelines

Sprints

How well did the 7 sprints work?

Requirements

Project Backlog

## Project Management Tools

### Trello

Description

Include diagrams

How it worked in practice

### GitHub

Description

How it is used

How it worked in practice

### Journal

Description

How it was used

How useful it was in practice

## Reflection

### Your views on the project

Describe how you feel the project went from your perspective and from the team’s perspective.

### Completing a large software development project

Describe what you have learnt from the project, from the point of view of completing a large software development project.

### Working in a team

Describe what you have learnt from the project, from the point of view of working in a team.

### Working with a supervisor

Describe how you feel the project went from the point of view of working with a supervisor.

### Technical skills

Describe what you have learnt from the project, from a technical skills viewpoint.

### Further competencies and skills

Describe any extra competencies and skills that would help you with your development in the work place.

## Conclusion

Write a couple of paragraphs summing up the chapter. Explain what area your project is about. Describe what the chapter has discussed.

# Business Opportunities

This is from your module with Tim McNicholls

# Conclusion

One paragraph on the background, the overall aim and the goals of the project.

One paragraph on the technologies used in the project.

Research

Design

Implementation

Testing

Overall result

Project management

What was learnt

How the project could be further developed

# References

The Department of Technology and Psychology in IADT uses APA referencing style.

Use alphabetical order for your references.

This site gives details about how to cite websites using APA:

https://www.wikihow.com/Cite-a-Website-in-APA

The following is a useful site for creating citations for APA for websites.

<http://www.citationmachine.net/apa/cite-a-website>

You can also use the Referencing tab within Microsoft Word to enter reference information manually. Word then creates an APA style reference.

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of Machine Learning Research, 81, 1-15.

Crawford, K., & Schultz, J. (2019). The use and misuse of facial recognition technology. Communications of the ACM, 62(6), 34-40.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., … & Kim, R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 316(22), 2402-2410.

Jain, A. K., Ross, A., Nandakumar, K., & Ngo, C. W. (2016). Introduction to biometrics. Springer.

Microsoft Azure Face API documentation: <https://azure.microsoft.com/en-us/services/cognitive-services/face/>

Microsoft Azure Face API blog: <https://azure.microsoft.com/en-us/blog/category/cognitive-services/face/>

Microsoft Azure Face API pricing: <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/face-api/>

Microsoft Azure Face API sample code: <https://github.com/Azure/azure-sdk-for-python/tree/main/azure-cognitiveservices-vision-face>

Microsoft Azure. (n.d.). Azure Cognitive Services overview. Retrieved from <https://azure.com/cognitive-services>

Microsoft Azure. (n.d.). What are Azure Cognitive Services? Retrieved from <https://docs.microsoft.com/en-us/azure/cognitive-services/cognitive-services-apis-overview>

Chollet, F. (2018). Deep Learning with Python. Shelter Island, NY: Manning Publications.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. Cambridge, MA: MIT Press.

Russel, S. J., & Norvig, P. (2010). Artificial Intelligence:

**Microsoft Azure Face API documentation:**

<https://azure.microsoft.com/en-us/services/cognitive-services/face/>

**OpenCV documentation:**

<https://docs.opencv.org/>

**"Facial Recognition with OpenCV" by Adrian Rosebrock** https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/

Barr, C. (2017). How to build your own smart mirror. Retrieved from https://www.digitaltrends.com/home/how-to-build-your-own-smart-mirror/

Bowers, J. (2017). The DIY smart mirror: a simple introduction. Retrieved from https://www.makeuseof.com/tag/diy-smart-mirror-simple-introduction/

Heath, T. (2018). Smart mirror guide: the ultimate guide to building your own smart mirror. Retrieved from https://www.smartmirrorguide.com/

Lin, K. (2017). Building a magic mirror with Raspberry Pi. Retrieved from https://www.instructables.com/id/Building-a-Magic-Mirror-With-Raspberry-Pi/

Schneider, K. (2017). The future of mirrors: an overview of smart mirrors and their benefits. Retrieved from https://www.electronicproducts.com/Sensors/Sensors\_Display\_Technologies/The\_future\_of\_mirrors\_an\_overview\_of\_smart\_mirrors\_and\_their\_benefits.aspx

Craciun, G. (2018, August 10). How to Build a Smart Mirror with Raspberry Pi. Retrieved January 08, 2023, from https://www.makeuseof.com/tag/build-smart-mirror-raspberry-pi/

Kitajima, Y. (2018, October 17). Smart Mirror with Raspberry Pi and Magic Mirror Software. Retrieved from https://www.instructables.com/Smart-Mirror-With-Raspberry-Pi-and-Magic-Mirror-Software/

"Emotion Recognition using Facial Landmarks, Python, DLib and OpenCV" by Rishi Bhatnagar <https://www.learnopencv.com/facial-landmark-detection/>

"Emotion recognition from speech signals" by E. Mower <https://www.sciencedirect.com/science/article/pii/S2405452620300333>

"Emotion recognition in physiological signals" by D. D. Reinoso et al. <https://www.sciencedirect.com/>

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Conference on Fairness, Accountability, and Transparency, 72–81. <https://doi.org/10.1145/3287560.3287591>

Garvie, C., & Luther, K. (2019). The Perpetual Line-Up: Unregulated Police Face Recognition in America. Center on Privacy & Technology at Georgetown Law. <https://www.perpetuallineup.org/>

Diaz, C. (2019). The ethics of facial recognition technology. Forbes. <https://www.forbes.com/sites/cognitiveworld/2019/07/15/the-ethics-of-facial-recognition-technology/?sh=52a0164e19fe>

"Facial recognition technology" by National Institute of Standards and Technology (NIST) <https://www.nist.gov/programs-projects/face-recognition-technology>

"Facial recognition technology and its potential impact on privacy" by American Civil Liberties Union (ACLU) <https://www.aclu.org/issues/privacy-technology/surveillance-technologies/facial-recognition-technology>

"The State of Facial Recognition: 2019" by the Center on Privacy & Technology at Georgetown Law <https://www.law.georgetown.edu/center-privacy-technology/wp-content/uploads/2019/10/The-State-of-Facial-Recognition-2019.pdf>

"Facial Recognition: A Closer Look at the Technology and Its Impact on Society" by the MIT Technology Review <https://www.technologyreview.com/s/613429/facial-recognition-a-closer-look-at-the-technology-and-its-impact>

Azure Face API documentation: <https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview>

Microsoft Azure Blog: <https://azure.microsoft.com/en-us/blog/introducing-facial-grouping-in-the-face-api/>

TechCrunch article: <https://techcrunch.com/2018/05/07/microsofts-azure-cognitive-services-adds-facial-recognition-api/>

Alcantara, J. (2017). The Internet of Things: A review of the state-of-the-art and future perspectives. Future Internet, 9(4), 77. <https://doi.org/10.3390/fi904077>

Kshetri, N. (2017). Internet of Things (IoT) security: An overview. International Journal of Information Management, 36(3), 295-298. <https://doi.org/10.1016/j.ijinfomgt.2017.06.001>

Smart Homes Market. (2020). In Statista. <https://www.statista.com/topics/1936/smart-homes-market/>

Wang, Q., Chen, W., & Wang, X. (2016). Internet of things: A survey. Information Systems Frontiers, 18(2), 223-249. <https://doi.org/10.1007/s10796-015-9604-y>

Dyche, J. (2010). The definitive guide to the Internet of Things. Apress.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660.

Kortuem, G., Klemke, R., Wulf, V., & Baker, T. (2010). Smart objects as building blocks for the Internet of Things. Internet of Things, 1-15.

*Facial recognition: Microsoft Azure*. Facial Recognition | Microsoft Azure. (n.d.). Retrieved January 15, 2023, from <https://azure.microsoft.com/en-us/services/cognitive-services/face/>

Pablo Castro Distinguished Engineer, Priyanka Rawat Senior Product Marketing Manager, Andy Beatman Sr. Product Marketing Manager, Kate Browne Program Manager, Sarah Bird Principal Group Product Manager, Ali Dalloul Vice President Strategy and Commercialization, & Tom Keane Corporate Vice President. (n.d.). *Cognitive services: Azure blog and updates: Microsoft Azure*. Azure Blog and Updates | Microsoft Azure. Retrieved January 15, 2023, from <https://azure.microsoft.com/en-us/blog/topics/cognitive-services/>

*Facial recognition: Microsoft Azure*. Facial Recognition | Microsoft Azure. (n.d.). Retrieved January 15, 2023, from <https://azure.microsoft.com/en-us/products/cognitive-services/face/>

*Pricing - face API: Microsoft Azure*. Pricing - Face API | Microsoft Azure. (n.d.). Retrieved January 15, 2023, from <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/face-api/>

Azure. (n.d.). *Azure/azure-SDK-for-python: This repository is for active development of the Azure SDK for python. for consumers of the SDK we recommend visiting our public developer docs at https://docs.microsoft.com/python/azure/ or our versioned developer docs at https://azure.github.io/azure-sdk-for-python.* GitHub. Retrieved January 15, 2023, from <https://github.com/Azure/azure-sdk-for-python>

Chappell, D. (2019) Understanding Azure API Management. O’Reilly Media, Inc.

Microsoft Azure. (n.d.). Azure API Management overview. Retrieved from <https://docs.microsoft.com/en-us/azure/api-management/api-management-overview>

Microsoft Azure. (n.d.). Azure Cognitive Services overview. Retrieved from <https://azure.com/cognitive-services>

Tiwari, S. (2019). Hands-On API Management. Packt Publishing Ltd.

Python Software Foundation. (n.d.). The History of Python. Retrieved from <https://docs.python.org/3/library/history.html>

Wes McKinney. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media, Inc.

GeeksforGeeks. (2021, June 4). Introduction to Python Programming. Retrieved from <https://www.geeksforgeeks.org/introduction-to-python-programming/>

Eben Upton, G. D. (2012). The Raspberry Pi: A computer for everyone. Raspberry Pi Foundation.

Gibson, J. (2015). Raspberry Pi User Guide (3rd ed.). John Wiley & Sons.

Lacey, J. (2017). Raspberry Pi: The complete manual (7th ed.). Imagine Publishing Ltd.

Nash, S. (2015). Raspberry Pi Projects for the Evil Genius. McGraw-Hill Education.

Wright, C. (2015). Raspberry Pi For Dummies (2nd ed.). John Wiley & Sons.

Raspberry Pi Foundation. (2021). Raspberry Pi. Retrieved January 8, 2023, from <https://www.raspberrypi.org/>

Arduino. (2021). Arduino. Retrieved January 8, 2023, from <https://www.arduino.cc/>

Sainsbury, R. (2019). Raspberry Pi vs Arduino: Which is the Mini Computer for You? MakeUseOf. Retrieved January 8, 2023, from <https://www.makeuseof.com/tag/arduino-vs-raspberry-pi-which-is-the-mini-computer-for-you/>

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of Machine Learning Research, 81, 1-15.

Garvie, C., & Luther, K. (2019). The Perpetual Line-Up: Unregulated Police Face Recognition in America. Georgetown Law Center on Privacy & Technology.

Diaz, D. (2019). The Ethics of Artificial Intelligence. Cambridge University Press.

Garside, J. (2019, November 12). Smart mirrors are the future of home technology. Wired. <https://www.wired.com/story/smart-mirrors-are-the-future-of-home-technology/>

Rich, K. (2019, December 23). What is a smart mirror, and why would I want one? The Ambient. <https://www.theambient.com/guides/what-is-a-smart-mirror>

Smart Mirrors. (n.d.). ABI Research. <https://www.abiresearch.com/market-research/product/>

Royce, W. W. (1970). Managing the development of large software systems: concepts and techniques. Proceedings of IEEE WESCON, Los Angeles, CA, 1-9.

Myers, G. J. (1979). The Art of Software Testing. John Wiley & Sons.

Kitajima, Y. (2018, October 17). Smart Mirror with Raspberry Pi and Magic Mirror Software. Retrieved from <https://www.instructables.com/Smart-Mirror-With-Raspberry-Pi-and-Magic-Mirror-Software/>

Nielsen, J. (1993). Usability Engineering. Academic Press Professional, Inc.

Rubin, J. (1994). Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. John Wiley & Sons.

Kim, D. (2015, November 30). The Importance of Wireframing in Web Design. Retrieved from <https://uxdesign.cc/the-importance-of-wireframing-in-web-design-a83fad5cf8b1>

Nielsen, J. (1994, October). Usability Engineering. San Francisco, CA: Morgan Kaufmann Publishers Inc.

The UX Review. (2017, June 21). Low-Fi vs High-Fi Wireframes: When to Use Each. Retrieved from <https://www.justinmind.com/wireframe/low-fidelity-vs-high-fidelity-wireframing-is-paper-dead>

Balsamiq. (n.d.). Balsamiq Wireframes. Retrieved January 08, 2023, from <https://balsamiq.com/wireframes/>

Nielsen, J. (1995). Multimedia and Hypertext: The Internet and Beyond. Academic Press.

Nielsen, J. (2000). Designing Web Usability: The Practice of Simplicity. New Riders Press.

Sneppen, T. (2018). The wireframing process in user-centered design. User Experience Magazine, 17(3), 22-30.

Tog. (n.d.). Tog on interface design. Retrieved January 08, 2023, from <http://www.tog.com/>

Wodtke, C. (2011). Information Architecture: Blueprints for the Web. New Riders Press.

Balsamiq. (n.d.). Balsamiq Wireframes. Retrieved January 08, 2023, from <https://balsamiq.com/wireframes/>

Kim, D. (2015, November 30). The Importance of Wireframing in Web Design. Retrieved from <https://uxdesign.cc/the-importance-of-wireframing-in-web-design-a83fad5cf8b1>

Nielsen, J. (1994). Usability Engineering. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Nielsen, J. (1995). Multimedia and Hypertext: The Internet and Beyond. Academic Press.

Nielsen, J. (2000). Designing Web Usability: The Practice of Simplicity. New Riders Press.

Sneppen, T. (2018). The wireframing process in user-centered design. User Experience Magazine, 17(3), 22-30.

Tog. (n.d.). Tog on interface design. Retrieved January 08, 2023, from <http://www.tog.com/>

Wodtke, C. (2011). Information Architecture: Blueprints for the Web. New Riders Press.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 1, I-I. https://doi.org/10.1109/cvpr.2001.990517

Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools, 25(11), 120-126.

Kaehler, A., & Bradski, G. (2017). Learning OpenCV 3: computer vision in C++ with the OpenCV library. O'Reilly Media, Inc

Artificial Intelligence:

• Bostrom, N., & Yudkowsky, E. (2014). The Ethics of Artificial Intelligence. In The Cambridge Handbook of Artificial Intelligence (pp. 316-334). Cambridge University Press. doi: 10.1017/9781316214032.016

• Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

• Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260. doi: 10.1126/science.aaa8415

Machine Learning:

• Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

• Domingos, P. (2015). The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. Basic Books.

• Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Deep Learning:

• Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

• LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. doi: 10.1038/nature14539

• Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117. doi: 10.1016/j.neunet.2014.09.003

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of Machine Learning Research, 81, 1-15.

Crawford, K., & Schultz, J. (2019). The use and misuse of facial recognition technology. Communications of the ACM, 62(6), 34-40.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., … & Kim, R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 316(22), 2402-2410.

Jain, A. K., Ross, A., Nandakumar, K., & Ngo, C. W. (2016). Introduction to biometrics. Springer.

Klare, H. R. (2012).

Eben Upton, G. D. (2012). The Raspberry Pi: A computer for everyone. Raspberry Pi Foundation.

Gibson, J. (2015). Raspberry Pi User Guide (3rd ed.). John Wiley & Sons.

Lacey, J. (2017). Raspberry Pi: The complete manual (7th ed.). Imagine Publishing Ltd.

Nash, S. (2015). Raspberry Pi Projects for the Evil Genius. McGraw-Hill Education.

Wright, C. (2015). Raspberry Pi For Dummies (2nd ed.). John Wiley & Sons.

Raspberry Pi Foundation. (2021). Raspberry Pi. Retrieved January 8, 2023, from https://www.raspberrypi.org/

Arduino. (2021). Arduino. Retrieved January 8, 2023, from https://www.arduino.cc/

Sainsbury, R. (2019). Raspberry Pi vs Arduino: Which is the Mini Computer for You? MakeUseOf. Retrieved January 8, 2023, from https://www.makeuseof.com/tag/arduino-vs-raspberry-pi-which-is-the-mini-computer-for-you/