

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

Project management

Team work

Tools

Trello

GitHub

Journal

Requirements

Design

Implementation

Testing

# Research

My Research Goes here!

# 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

### 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.

**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.

### Item 2

The same as Item 1.

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

## 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.