

***INSTITUTION:*** GALWAY MAYO INSTITUTE OF TECHNOLOGY

***COURSE:*** BENG(H) IN SOFTWARE AND ELECTRONIC ENGINEERING - YEAR 4

***MODULE:*** PROJECT ENGINEERING

***SUPERVISOR:*** NIALL O’KEEFFE

***YEAR:*** 2020/2021

**Ross Monaghan – G00376556**

EXPOSITION DISPLAY GAZE TRACKING SYSTEM

Project Graphic (Optional)

**Declaration**

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software & Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

Signed: Ross Monaghan

**Acknowledgements**

I would like to express my gratitude and appreciation to the academic staff at GMIT who have provided continuous support and guidance to me throughout the duration of the academic year.

Specifically, my thanks go out to my project supervisor Niall O’Keeffe, who has been extremely helpful in providing technical feedback and direction to areas of my project which required support.

I would also like to thank my Project Engineering lecturers, Brian O’Shea, Michelle Lynch and Paul Lennon, who have been present and available to provide help and guidance.

Michael Murray, Des O’Reilly, Dan Maher

**Table of Contents**

[1 Summary 7](#_Toc71748800)

[2 Poster 8](#_Toc71748801)

[3 Introduction 9](#_Toc71748802)

[4 Convolutional Neural Networks 10](#_Toc71748803)

[4.1 Convolution Operation 10](#_Toc71748804)

[4.2 Convolutions for Pattern Recognition 11](#_Toc71748805)

[4.3 Neural Networks 12](#_Toc71748806)

[4.3.1 Neural Networks 12](#_Toc71748807)

[4.3.2 Neural Networks 12](#_Toc71748808)

[4.3.3 Neural Networks 12](#_Toc71748809)

[5 Deep Learning 13](#_Toc71748810)

[6 Computer Vision 15](#_Toc71748811)

[7 Project Architecture 17](#_Toc71748812)

[8 Project Plan 18](#_Toc71748813)

[9 Head Pose Estimation 19](#_Toc71748814)

[9.1 Subheading 19](#_Toc71748815)

[10 Saliency Mapping 21](#_Toc71748816)

[11 Training 23](#_Toc71748817)

[12 Results 25](#_Toc71748818)

[13 Conclusion 27](#_Toc71748819)

[14 References [5] 28](#_Toc71748820)

[15 Appendix 29](#_Toc71748821)

# Summary

The goal of the Exposition Display Gaze Tracking System was to develop a computer vision system that estimates the gaze of attendees at an exposition, so that data may be gathered of what component of a display most frequently caught a person’s eye. The result of this would be a more detailed understanding of how a display performs, so that its design may be altered to grab attention more effectively.

This projects scope focused on the computer vision aspect of the system, with the main deliverable being the development of a gaze following program, which can detect the most likely area in a frame that an individual is looking.

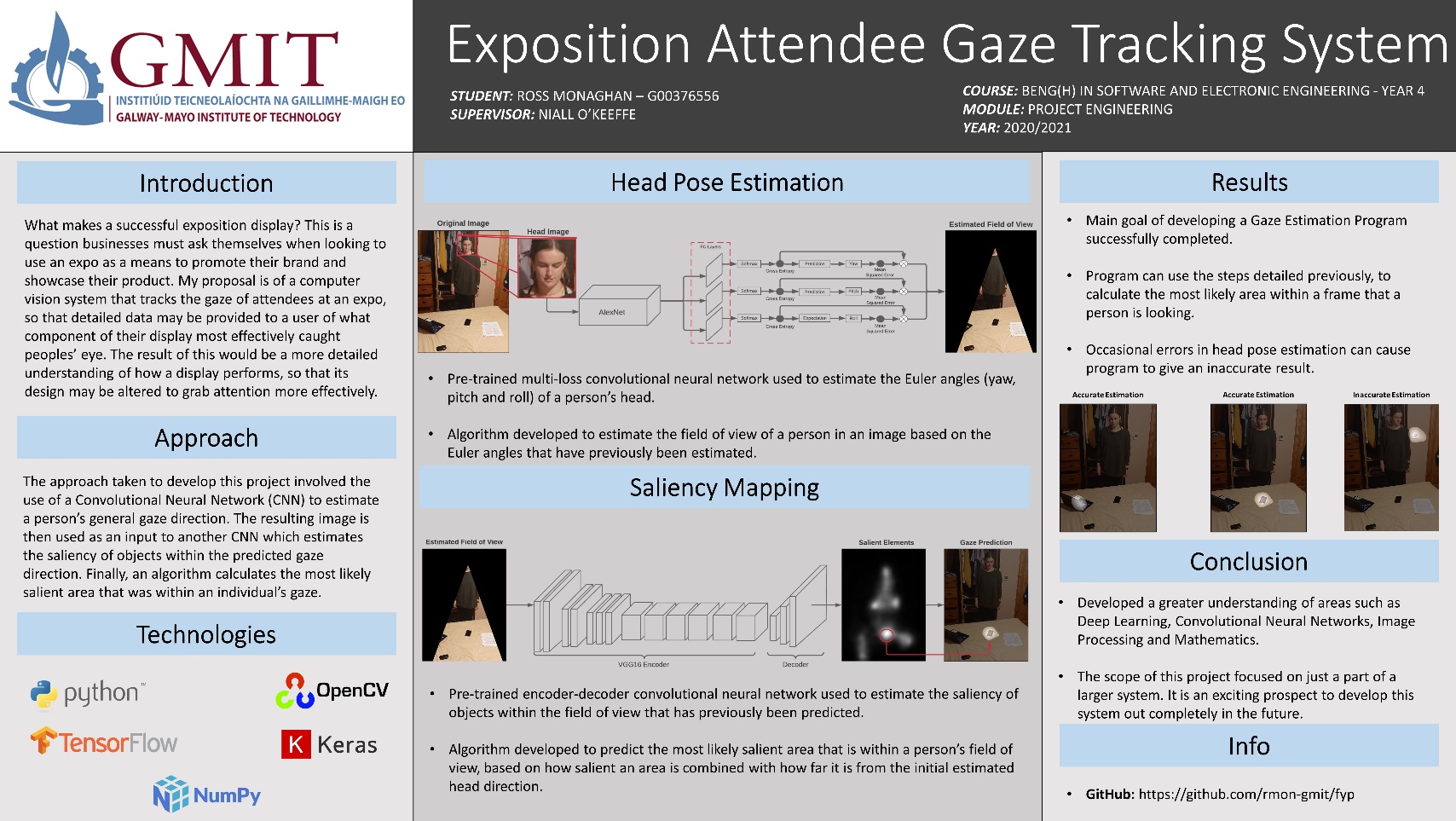
The approach taken to implement this functionality involved the use of a Convolutional Neural Network (CNN) to estimate a person’s general gaze direction. The resulting image then being used as an input to another CNN which estimates the saliency of objects within said gaze direction. An algorithm would then calculate the most likely salient area that was within an individual’s gaze.

The Project has been developed using Python. It has involved the use of TensorFlow/Keras libraries for its Deep Learning and CNN areas and has used the NumPy library for the mathematical aspects of the code. OpenCV has been used for the various image processing tasks that the project requires.

Much was accomplished during the duration of The Project, with the main goal of developing a Gaze Estimation Program successfully completed. This program can use the steps detailed previously, to calculate the most likely area within a frame that a person is looking.

Conclusions made during the development of this project have led to a greater understanding of areas such as Deep Learning, Convolutional Neural Networks, Image Processing and Mathematics. The resulting program has much opportunity for further development, and this will undoubtably be pursued in the future. As was stated previously, the scope of this project focused on just a part of a larger system, and it is an exciting prospect to develop this system out completely in the future.

# Poster



# Introduction

The motivation for pursuing this project stems largely from the increasing use of Deep Learning algorithms to improve processes or tasks which are benefited by the access to more sources of information. Computer Vision is a field which has on many occasions implemented Deep Learning in this way, providing a user or system with useful and accurate information which has been gathered from an image or images.

Gaze following is an established computer vision topic which is used to estimate the most likely area that an individual in a frame is looking. Different from eye tracking, which relies on a clear view of a person’s eye to calculate the point which they are looking. Gaze following instead relies on deducing the direction a person is looking based on an image of their head, followed by a saliency prediction on the objects in that direction [1]. What results will not be an exact point in an image, but instead a general area.

The goal of this project is to develop a gaze following program which will be used to estimate the general area in an image that a person is looking. In the context of a larger Exposition Display Gaze Tracking System, this program would be fundamental to that systems functionality.

The work required to develop the gaze following program will involve extensive research into both deep learning and convolutional neural networks. The knowledge gained from this research will be used to develop a head pose estimation deep learning algorithm which will estimate gaze direction based on a person’s head. This estimation will be used to generate a field of view for that person, which can then be subject to a saliency prediction using another deep learning algorithm. The result of this saliency prediction will then be overlayed on the original image, creating a heat map of the area that a person is looking.

This report will detail the underlying technologies and concepts involved in building a gaze following program, followed by the steps taking in completing the project. What will follow is a description of the results, and a comparison with existing systems.

# Convolutional Neural Networks

When dealing with computer vision applications of deep learning, Convolutional Neural Networks (CNN’s) are almost universal in their usage. The robustness in which CNN’s learn and recognise patterns, combined with their ability to break larger data down into smaller chunks, make them excellent for deep learning tasks that involve images [2].

## Convolution Operation

The core operation of a Convolutional Neural Network is that of the convolution. In essence, a convolution involves taking two matrices and performing an element-by-element multiplication on them, the resulting values being summed together [3].

When applied to images, convolutions involve the creation of a ‘kernel’, which is to say a matrix of fixed value and of a size that is usually much smaller than that of the original image. This kernel is then applied to a section of the original image, creating an output for that section. When this process is repeated throughout the image, an output image can be created which will vary greatly depending on the values which have been assigned to the kernel. A diagram of a convolution is contained below:

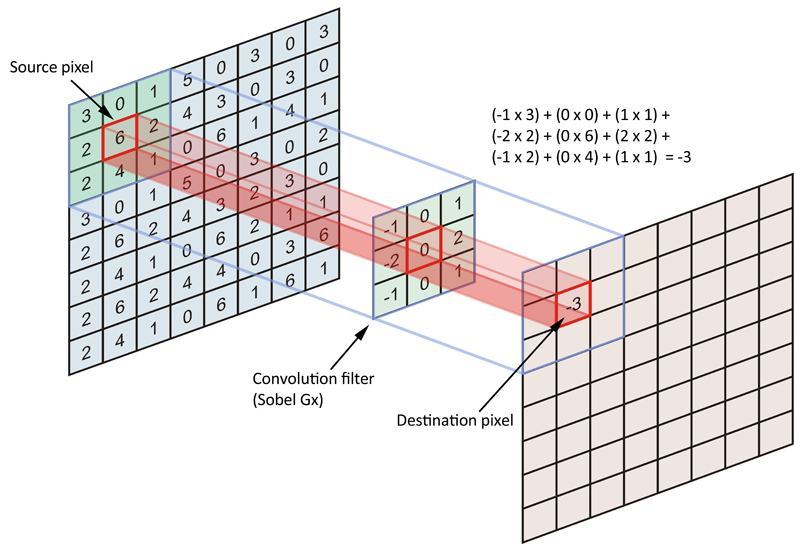
**** 

Figure 1: Convolution Diagram [4]

## Convolutions for Pattern Recognition

The fact that different kernel values can have drastically different effects on the resulting image is extremely powerful and results in the ability to use kernels of specifically selected values to carry out operations such as filtering or edge/contour detection.

Take for example the kernel present in *Figure 1,* which is used as part of the common Sobel operator used in edge detection and will have the effect of emphasising vertical lines in an image. The entire Sobel operator will output a result of both vertical and horizontal kernels applied to an image and can be compared with just the vertical kernel in the figures below:



Figure 2: Emphasized Vertical Lines Figure 3: Full Sobel Operator

In the context of CNN’s, the ability to extract very basic details (such as edges and contours) from an image is essential to the effectiveness of the network. This is because the underlying functionality of a CNN is that these very basic features may be combined to create more complex patterns which the network can recognise and use to form its prediction [5].

## Neural Networks

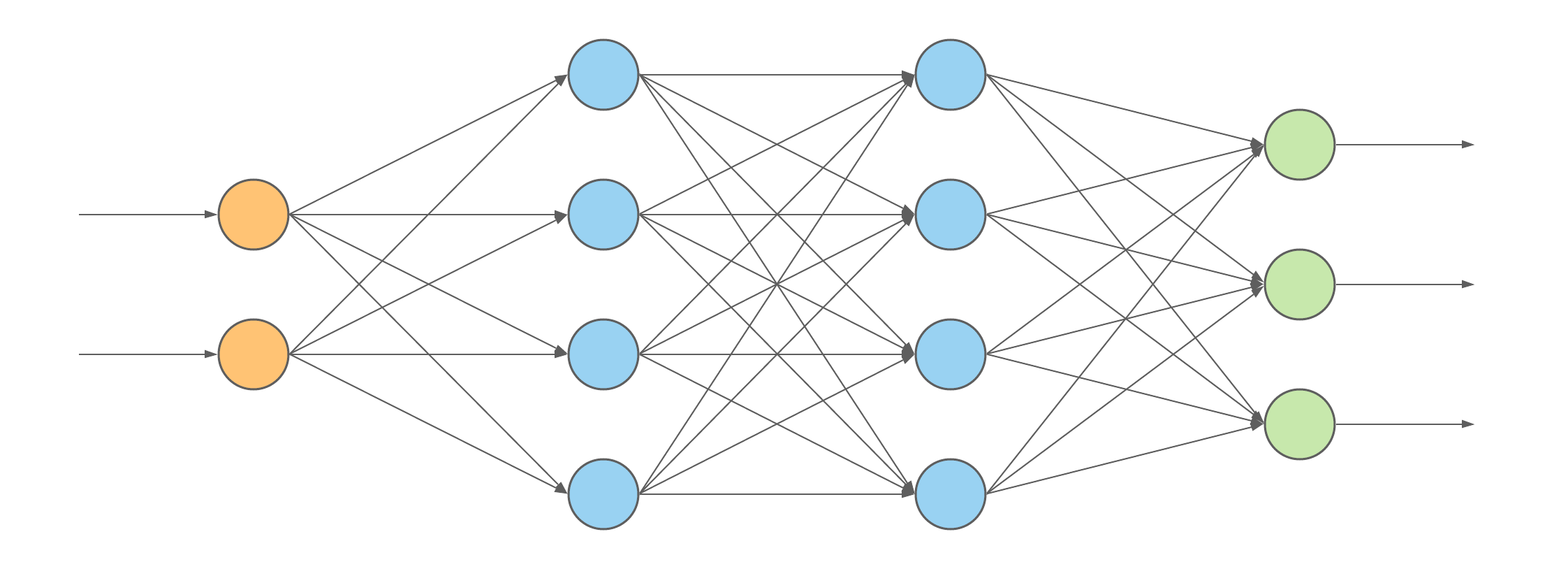
Drawing inspiration from neuroscience and biology, neural networks are a collection of artificial neurons that ‘fire’ or activate depending on what is input to the network. Similarly to how different stimuli will fire different collections of neurons in an organic brain, the data that is input to a neural network will activate different collections of artificial neurons. This property of neural networks can be used to make predictions on the data that has been input. Though the uses of neural networks vary widely, when applied to computer vision tasks neural networks are used because of their inherent ability to recognise patterns [2].

Figure 4: Simple Neural Network

### Artificial Neuron

The most basic component of a neural network is that of the artificial neuron. Artificial neurons are mathematical functions whose inputs are obtained from the outputs of previous neurons or the initially input data. The determining factor to whether a neuron is activated or not is decided by that neuron’s activation function, that is a mathematical function that normalises the input data to be a value between 0 and some positive number [5]. Should the neuron become active (i.e. the output of its activation function is greater than 0), its output value is fed into neurons further in the network to be subject to their own activation functions.

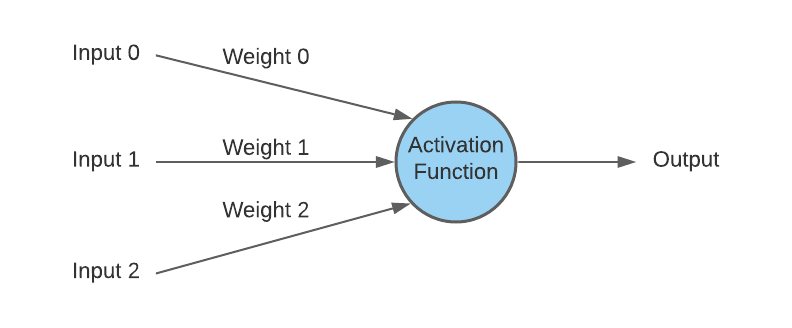


Figure 5: Artificial Neuron Model

### Activation Functions

How a neuron treats its input data is determined by the activation function that is used by said neuron.

Figure 6: Activation Function Comparison

### Convolutional Layer

### Pooling Layer

### Fully-Connected Layer

### Output Layer

# Supervised Deep Learning

## Backpropagation

## Loss

### Classification

### Regression

## Labelling

# Project Architecture

Your project architecture diagram should go here. This is an important section, and one most readers of your report will view.

Your diagram should be self-documenting. Use subsequent sections in your report to elaborate on technologies / software / hardware in your diagram.

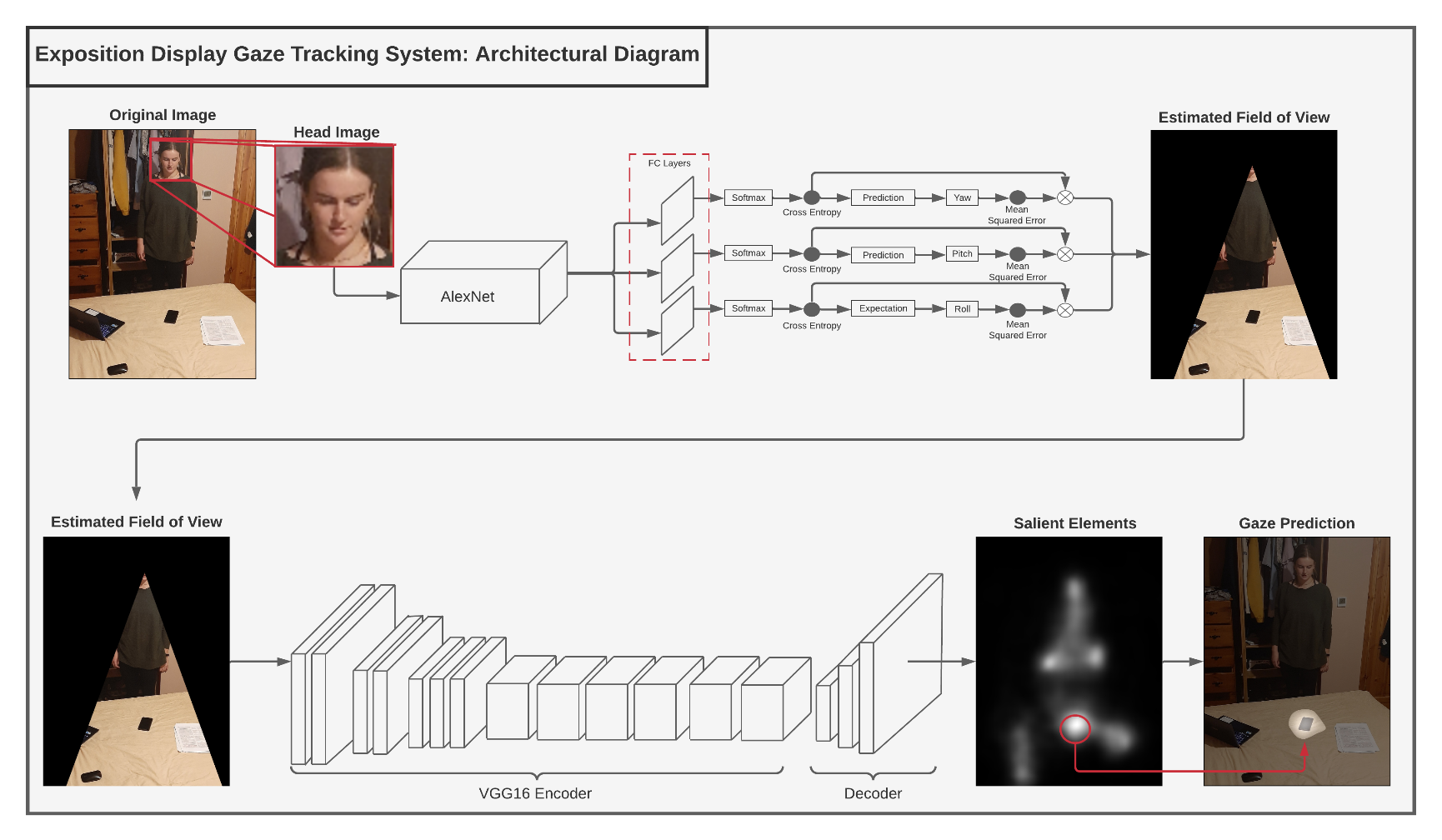


Figure 6‑1 Architecture Diagram

# Project Plan

# Head Pose Estimation

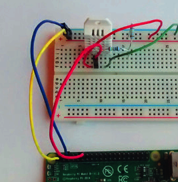
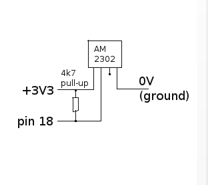
This is an example heading for a section in a project. You choose your sections to suit your project.

## Subheading

This is a subheading, use subheadings to break up a large topic into smaller sections.

Notes on content:

Photographs are not technical diagrams and are not a good substitute for professional technical diagrams. Use photographs to enhance a report, but not as a replacement for diagrams.



V

Figure 8‑1 A photograph is not a replacement for a circuit diagram

In describing software, you need diagrams and/or summaries of software design & layout. It is not sufficient to just paste some code. You should describe what your code is designed to do, in English. If you decided to put your code in functions or libraries or objects, describe this architecture. One good layout is to include a snippet(s) of code alongside an explanation. You do not have to explain every part of your code, pick the important parts.

Write out any mathematical equations or calculations that are important in your project and explain them.

Include details of any major problems or challenges you encountered in an area, and how you solved them.

# Saliency Mapping

# Training

# Results

# Conclusion

Write a short conclusion. What is the outcome of the project? Perhaps you have a product prototype, or some results, or a demonstratable system.

Do not use your conclusion to tell the reader what you might have done if you had more time, but keep it focussed on what you actually have done. You can mention future opportunities for further development of the work, but keep this part short.

# References

|  |  |
| --- | --- |
| [1] | D. Lian, Z. Yu and S. Gao, "Believe It or Not, We Know What You Are Looking at!," ACCV, Shanghai, 2018. |
| [2] | F. Chollet, Deep Learning With Python, New York: Manning Publications, 2017. |
| [3] | M. Lynch, "Image Processing," *Digital Signal Processing,* vol. Galway Mayo Institute of Technology, 04 2021. |
| [4] | T. Bluche, "Deep Neural Networks Applications in Handwriting Recognition," DOCBOX, 2017 03 09. [Online]. Available: http://technodocbox.com/3D\_Graphics/70716176-Deep-neural-networks-applications-in-handwriting-recognition.html. [Accessed 10 05 2021]. |
| [5] | J. Portilla, "Python for Computer Vision with OpenCV and Deep Learning," Udemy, 10 05 2020. [Online]. Available: https://www.udemy.com/course/python-for-computer-vision-with-opencv-and-deep-learning/. [Accessed 11 2020]. |
| [6] | N. Ruiz, E. Chong and J. Rehg, “Fine-Grained Head Pose Estimation Without Keypoints,” Georgia Institute of Technology, Atlanta, USA, 2018. |
| [7] | N. Ruiz, "Deep Learning Head Pose Estimation using PyTorch," 1 10 2019. [Online]. Available: https://github.com/natanielruiz/deep-head-pose. [Accessed 12 2020]. |
| [8] | A. Recasens, A. Kholsa, C. Vondrick and A. Torralba, "Where are they looking?," Advances in Neural Information Processing Systems (NIPS), Boston, 2015. |
| [9] | A. Kroner, M. Senden, K. Driessens and R. Goebel, "Contextual encoder–decoder network for visual saliency prediction," Elsevier Ltd., Maastricht, 2020. |

|  |  |
| --- | --- |
| [3] | M. Lynch, *Digital Signal Processing,* Topic: “Image Processing”*,* Galway Mayo Institute of Technology, Galway, 04 2021. |

# Appendix