PREDICTION OF HUMAN EMOTIONS THROUGH ROBOT FACIAL RECOGNITION

by

Melissa Ho Hui Qi
HOM 14581956

A dissertation submitted in partial fulfilment of the requirements for the degree of

BSc (Hons) Computer Science

School of Computer Science, University of Lincoln

UNIVERSITY OF LINCOLN

APRIL 2018

University of Lincoln

School of Computer Science

Project Title: **PREDICTION OF HUMAN EMOTIONS THROUGH ROBOT FACIAL RECOGNITION**

Student ID: **HOM 14581956**

Name: **Melissa Ho Hui Qi**

Project Supervisor: **Dr Heriberto Cuayahuitl**

Module: **CMP 3060M - Project**

Assignment 2: **Project Report**

Due Date: **26th April 2018**

ACKNOWLEDGEMENT

Although this is an individual project, this project would not have made such a remarkably progress without all those people whom have helped, and supported me throughout the highs and lows of the project development.

First of all, I would like to express my sincere gratitude to my supervisor, **Dr Heriberto Cuayahuitl** for his continuous support and guidance towards my proposed project. His consistent pressure regarding to the project development has reminded me and allowed me to complete the project on time.

Secondly, I would like to thank my personal tutor, **Dr Michael Mangan** for providing me continuous emotional support throughout the development of this project.

Thirdly, I would like to thank **all other academic staff of the Computer Science** department for their teaching of valuable knowledge which allows me to implement them within this project.

Additionally, I would like to take this opportunity to **thank the staff of English Language Center and my friends** whom have consistently providing guidelines and suggestions of ways in improving my project.

Other than that, I am thankful enough to have **my supportive family** whom have their full trust and encouragement for me in pursuing my own dream of being a part of the IT world.

Lastly, I would like to thank **all the image database providers**: Karolinska Directed Emotional Faces (KDEF) Database [(E.Lundqvist, D., Flykt, A., and Öhman, A., 1998)], Japanese Female Facial Expression (JAFFE) Database [(Lyons, M., Akemastu, S., Kamachi, M. and Gyoba, J., 1998)], and Yale Face Database [(Belhumeur, P., Hespanha, J. and Kriegman, D., 1997)] for their kindness and willingness to share their database.

ABSTRACT

Melissa Ho: Prediction of Human Emotions through Robot Facial Recognition

Emotional Recognition has been a popular research topic recently in order to create a more human-like robot which is able to learn and think reasonably like a human being. Hence, this dissertation is about programming the robot to be able to detect human faces through Facial Recognition. Using the detected faces, the system is able to analyse and predict the facial emotions whether the subject is either happy, neutral, or sad using the implementation of Convolutional Neural Network (CNN). With CNN, the system is able to achieve the highest accuracy of 85.9% for testing.

Keywords: Facial Recognition, Emotional Recognition, Convolutional Neural Network (CNN)

CONTENTS

Chapter 1: Introduction

Is it possible for future robots to recognise human emotions?

This report describes a graduation project which hoped to build a closer relationship between humans and robots by improving their understanding of social interaction with one another. It is proposed to change the opinions of most people in general whom categorise robots as "cold" and "close-minded" machines.

1.1 Intention of this Project

This project is intended to improve human-robot interaction, and resolve the limitation gap between a robot and a human. Communication and interaction have been an important part of the social life of humans. Hence, social robots like Aibo, Cosmo, Jibo, Nao, and Pepper (Figure 1) have been introduced into the world to communicate and interact with humans to make their lives more interesting everyday.

*Figure 1: Different Types of Social Robots (*Karthika, 2017)

For this project, the proposed system should be able to recognise, analyse, and predict the emotions of subjects through facial recognition and response to it appropriately. For instance, the robot should be able to comfort people who are sad instead of laughing at them inappropriately.

1.2 Aim and Objectives

Aim: To create a robot system able to predict human emotions through face detection and emotion recognition using data from a 2D camera

Objectives:

1. The proposed robot is able to capture and select the images from its real-time scenario.
2. The proposed robot is able to detect human faces among other objects.
3. The proposed robot is able to recognise the characteristics of human faces.
4. The proposed robot is able to detect, recognise and differentiate some basic human emotions.
5. The proposed robot is able to predict human emotions although only partial of the face is captured.

1.3 Research Scope

This project implements the field of Emotional Recognition. Emotional Recognition applies most of the techniques used within Image Processing besides constructing a Neural Network which enable the robots to learn and recognise the patterns of different categories in order to make a prediction for it. The following are the two main research scope for this project.

1.3.1 Static Image Recognition

This type of recognition is processed by manually inputting the image of a person into the system to be analysed. The input image used is clearly focused and less interruption at the background.

1.3.2 Real-Time Image Recognition

This type of recognition is processed by capturing the image at the moment through the webcam. This type of image tends to be a bit noisy compared to the input image used in Static Image Recognition.

1.4 Report Outline

The following outlines the contents within this report which basically describes the development process of the system. This report will be consisted of six chapters in total, including Chapter 1 as the Introduction.

Chapter 2: Background and In-Depth Literature Review

This chapter explains the background of the system as well as a few academic literature of the related projects. These descriptions of the related projects are then referred back to obtain useful ideas which could be contributed within this project.

Chapter 3: Methodology

This chapter highlights and compare several methodologies which could be implemented within this project before choosing the most suitable methodology which fits the requirement criteria of the project. Furthermore, this chapter also mentions tools and machine environments which are used to operate the artefact. Besides these, the research methods are then justified for this project.

Chapter 4: Design, Development and Evaluation

This chapter describes the development process of the artefact in detailed steps based on Software Development Life Cycle (SDLC) model.

Chapter 5: Conclusion

This chapter concludes and summarises the development process of the artefact by including the result findings.

Chapter 6: Reflective Analysis

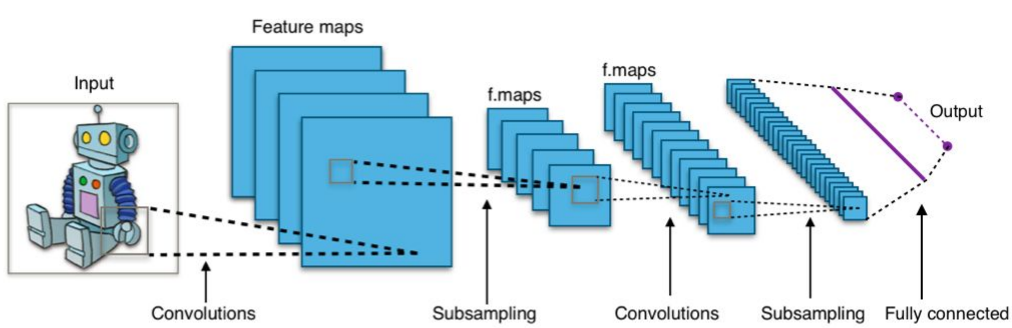
This chapter reflects on the development process where there is any potential changes needed to be made or improved towards the project.

Chapter 2: Background and In-depth Literature Review

This chapter will explain in more details regarding the background of the project as well as several related works.

2.1 Background

This project implements the concept of Emotional Recognition which has been a popular field in the recent years. The proposed system uses Convolutional Neural Network (CNN) to create an artificial "brain" for the robot[1] to analyse and predict human emotions that are detected and extracted. The purpose of choosing CNN as the selected neural network model as it has higher accuracy in performing prediction which will later be explained and supported within the related works. Using Figure 2, this Neural Network requires a collection of grayscale images which made up of four different human emotions (angry, happy, neutral, and sad) acting as its training datasets. These datasets are then trained and tested separately to obtain the accuracy result of the system. Furthermore, different pixels sizes (48x48, 70x70, and 100x100) of the same datasets are later created to test the difference in accuracy results.



[1] The robot used for this project is Pepper: https://www.ald.softbankrobotics.com/en/robots/pepper

Figure 2: Outline Structure of Convolutional Neural Network (CNN)

Although the pixels size for each datasets are different, they are of the same dataset which is made up of 1,007 images in total of four different emotions. The specific dataset is a combination of datasets obtained from 3 different sources which are Karolinska Directed Emotional Faces (KDEF), Japanese Female Facial Expression (JAFFE) Database, and Yale Face Database.

2.2 In-Depth Literature Review

This section will list out and describe in details with the related works on how they achieve their desirable results. Hence, the useful information regarding to the development of the related works will be highlighted in this section to support and to be implemented within the development of the artefact for this project.

2.2.1 Emotion Recognition using Convolutional Neural Network (CNN)

As mentioned in Section 2.1, Convolutional Neural Network (CNN) is one of the neural networks which is selected to recognise human facial emotions. This is because CNN can recognise and result in a higher prediction accuracy of the correct facial emotions. This statement could be supported by Sighthound[2] of having a high accuracy of predicting the correct facial emotions using Convolutional Neural Network as shown in Table 1. Dehghan, A., Ortiz, E., Shu, G. and Masood, S. (2017) analyses around 2,156 images with the size of 48x48 to predict the accuracy of the seven emotions (“angry”, “disgusted”, “fearful”, “happy”, “neutral”, “sad”, “surprised”) after over 4 million data is

[2] Sighthound for Emotional Recognition is available at https://www.sighthound.com/products/cloud

trained within the system. Figure 3 shows the process on how the system starts to learn to recognise the displayed human face features and then tested to predict the emotion displayed on the tester by comparing with the results of the learned data at the same time.

Methods	Accuracy
Sighthound	76.10 %
Microsoft	61.30 %

Table 1: Averaged Accuracy of the Seven Emotions Prediction

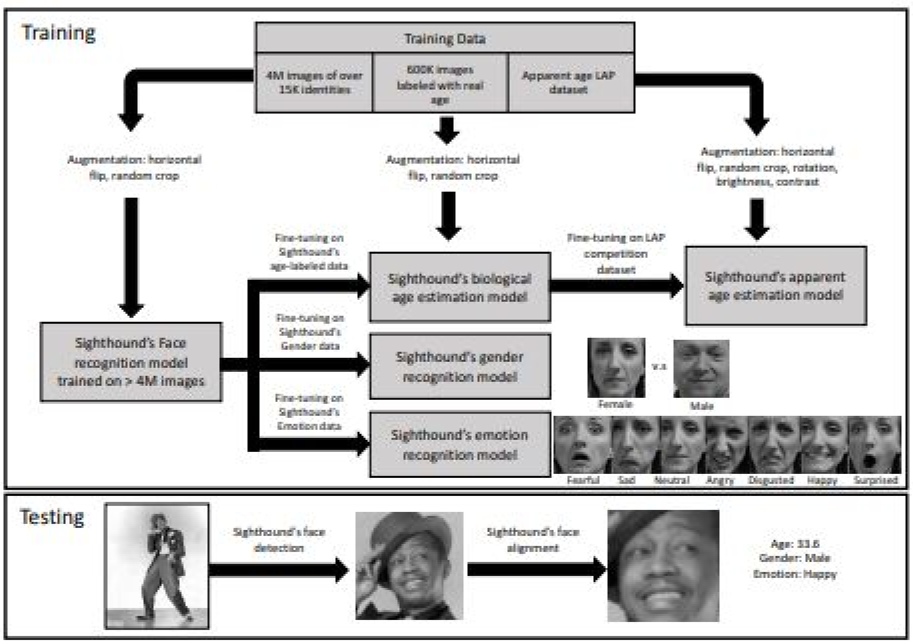


Figure 3: Analysing Process using Sighthound

Besides that, Convolutional Neural Network(CNN) or commonly known as a model of Deep Learning implements the same topology as Local Binary Patterns (LBP) (Xi, M., Chen, L., Polajnar, D. and Tong, W., 2016).

Shan, C., Gong, S. and McOwan, P. (2008) has conducted the experiments of combining different machine learning methods with Local Binary Patterns (LBP) in predicting the changes of facial expression. This is where the most discriminant LBP features are collected within the process. Table 2 below concludes that a combination of Support Vector Machine (SVM) and LBP can generate a higher accuracy than other machine learning methods.

Machine Learning Method + LBP	7 Classification Accuracy (%)
Template Matching + LBP	79.1 ± 4.6
Support Vector Machine (SVM) + LBP	88.1 ± 3.8
Linear Discriminant Analysis (LDA) + LBP	73.4 ± 5.6
Linear Programming + LBP	82.3 ± 3.5

Table 2: Prediction Accuracy using a combination of LBP and different Machine Learning Methods

2.2.2 Emotional Recognition using Other Neural Network

In comparison with CNN, emotional recognition can also be done by implementing other neural networks like Multi-Layer Perceptron (MLP), Recurrent Neural Network, and others more. For instance, Barman, A. and Dutta, P. (2017) proposed in performing emotional recognition using Multi-Layer Perceptron (MLP) as its selective type of neural network for the self-learning process of the system. Both Figure 4 and Table 3 have mentioned about implementing Normalised Distance and Shape Signature before training the data using MLP neural network. In this context, Distance Signature is explained as the distance is measured from its central geometric location like the nose region and forms a grid whereas the grid formation for Shape Signature is formed using Salient Landmarks like the eyes, eyebrows, nose, mouth and others.

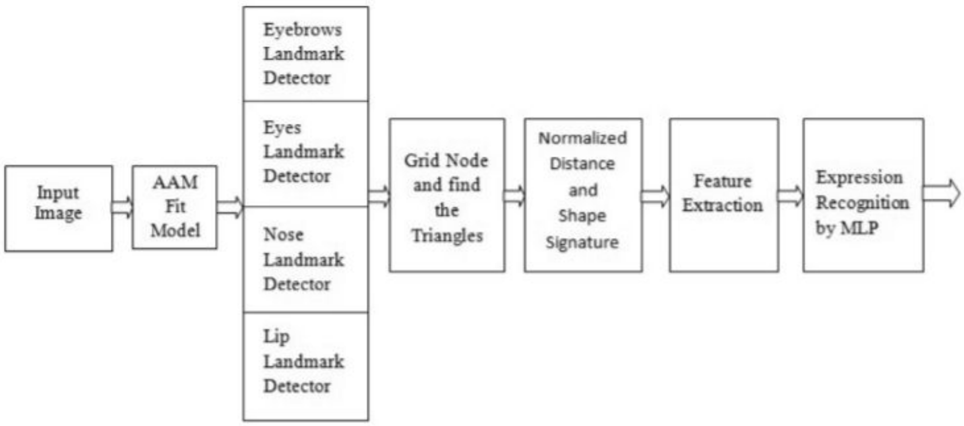


Figure 4: Flow Chart on the Proposed Method Used to Recognise Emotions

Type of Database Used	Distance Signature	Shape Signature	Distance - Shape Signature (D-S)
JAFFE	94.0	94.9	96.4
MMI	81.5	81.8	81.9

Table 3: Prediction Accuracy after Implementing MLP to train data

Similarly, Khanchandani, K. and Hussain, M. (2009) has conducted emotional recognition through analysis and recognition of input speech before categorised them into seven different emotions (i.e. Anger, Boredom, Disgust, Fear, Happiness, Neutral, and Sadness). This process are done using the Multi-Layer Perceptron Neural Network (MLPNN) and Generalised Feed Forward Neural Network (GFFNN) separately. Figure 5 shows the architecture of GFFNN which is a simplified version of MLPNN. This architecture implements several layers feed forward networks to calculate back propagation of errors in order to identify static pattern of emotion.

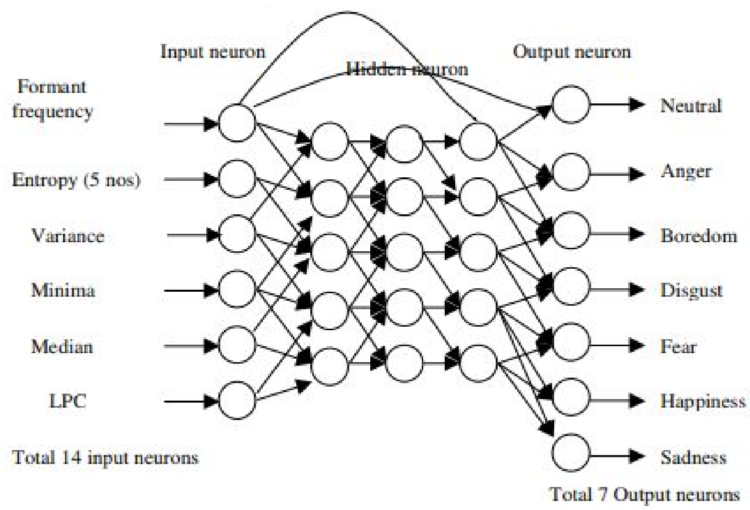


Figure 5: Architecture of a simplified version of MLP Neural Network

Type of Neural Network	Average Accuracy
GFFNN	98.08 %
MLPNN	89.62 %

Table 4: Average Accuracy for GFFNN and MLPNN

	Angry	Boredom	Disgust	Fear	Happiness	Neutral	Sadness
GFFNN	99.04	100.00	92.68	96.49	100.00	98.41	100.00
MLPNN	98.02	98.36	78.79	84.48	84.48	86.21	100.00

Table 5: Prediction Accuracy for each Emotion

Other than Multi-Layer Perceptron(MLP) and CNN neural network, Zhang, T., Zheng, W., Cui, Z., Zong, Y. and Li, Y. (2017) has proposed in implementing Hierarchical Recurrent Neural Network (RNN) which is a combination of two different layers of RNN for emotional recognition, such as, quad-directional spatial RNN (SRNN), and bi-directional temporal RNN layer (TRNN). With further explanation, Zhang, T., Zheng, W., Cui, Z., Zong, Y. and Li, Y. (2017) explained that SRNN is used to compare each temporal spatial of different angles in order to create discriminative dependency sequence while TRNN continues after SRNN to obtain long-term temporal dependencies by comparing the temporal sequences back and forth. Figure 6 below shows an overall process of the proposed Hierarchical RNN or known as Spatial-Temporal Recurrent Neural Network (STRNN) for analysing and predict the emotion of the input data. Its accuracy results of its prediction is summarised by comparing with other choice of methods as shown in Table 6.

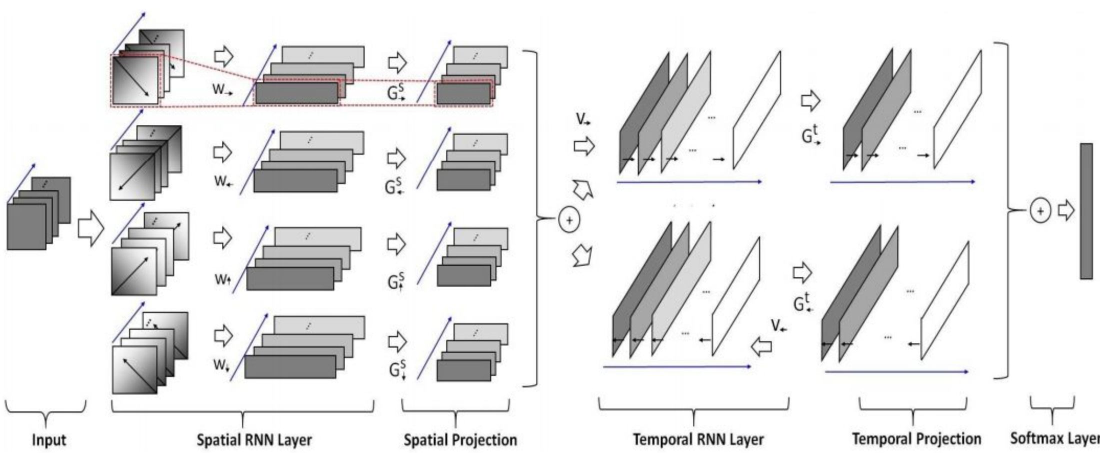


Figure 6: Overview Process of the Proposed STRNN method

Methods	Prediction Accuracy (%)
Support Vector Machine (SVM)	86.65
Linear Discriminant Analysis (LDA)	80.32
Canonical Correlation Analysis (CCA)	76.16

Deep Belief Network (DBN)	86.08
Spatial-Temporal Recurrent Neural Network (STRNN)	**89.50**

Table 6: Prediction Accuracy using Different Methods

Similarly, Graves, A., Schmidhuber, J., Mayer, C., Wimmer, M. and Radig, B. (2008) has introduced two different types of Recurrent Neural Network (RNN): Bidirectional RNN (BRNN) and Unidirectional RNN (URNN) in order to predict the correct emotion for the input data. Figure 7 illustrates the overall architecture of RNN model which one or more layer of it could be linked back to one another while Figure 8 shows and compares the difference of both architecture of the normal URNN and the BRNN. In addition, the concept of BRNN is also explained and supported by Lee, C. (2017) and Schuster, M. and Paliwal, K. (1997) that BRNN is an extension of URNN in which BRNN is a combination of two independent RNN to be able to be trained without the limitation of the input data to generate a better prediction results. This can be supported with the accuracy results listed under Table 7.

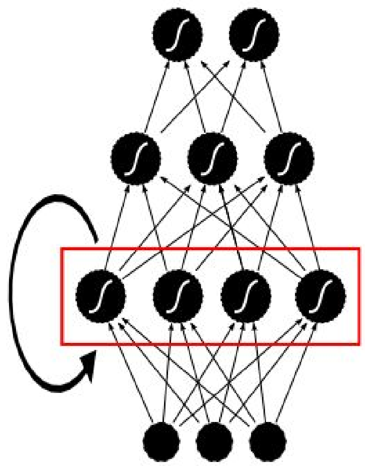


Figure 7: Overall Architecture for RNN

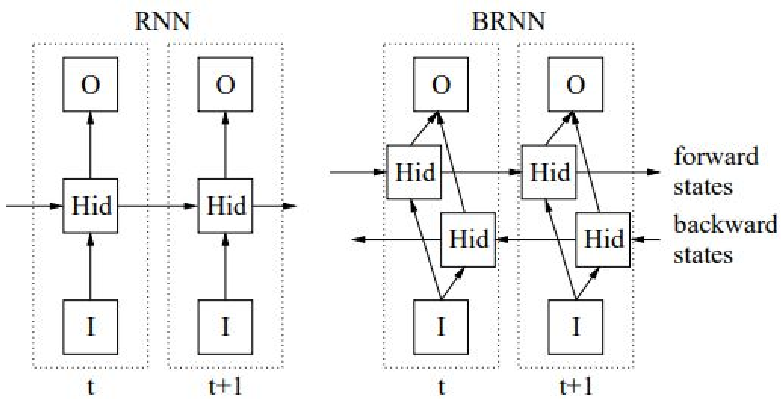


Figure 8: Partial Architecture of Normal RNN and Bidirectional RNN

Type of RNN	Average Accuracy (%)
Bidirectional RNN	85.4 ± 0.9
Unidirectional RNN	81.8 ± 0.6

Table 7: Average Accuracy of Prediction Results for both BRNN and URNN

2.2.3 Emotional Recognition using Speech Recognition

Comparatively, Zhang, S., Zhang, S., Huang, T. and Gao, W. (2017) has proposed in predicting human emotions through speech with the same approach of implementing Deep Convolutional Neural Network (DCNN) but along with Discriminant Temporal Pyramid Matching (DTPM). According to Figure 9, the whole proposed system implements Figure 10 (DCNN) and Figure 11 (DTPM). Figure 10 implements the proposed DCNN architecture by Krizhevsky, A., Sutskever, I. and Hinton, G. (2012) which implements several layers of max poolings in order to downsizing the previously obtained feature maps whereas Figure 11 continuously attach the temporal clues and obtain the optimal results.

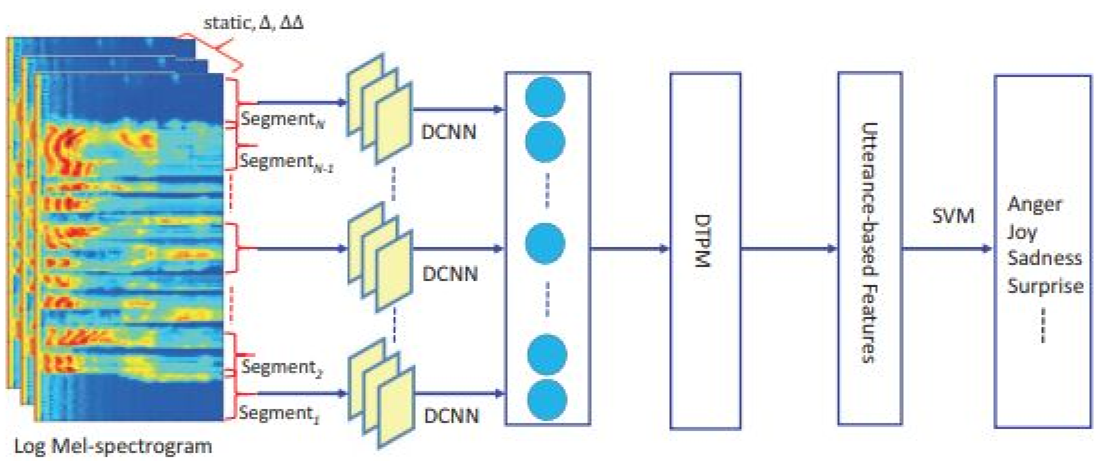


Figure 9: Overall Process of the Whole System (DCNN - DTPM)

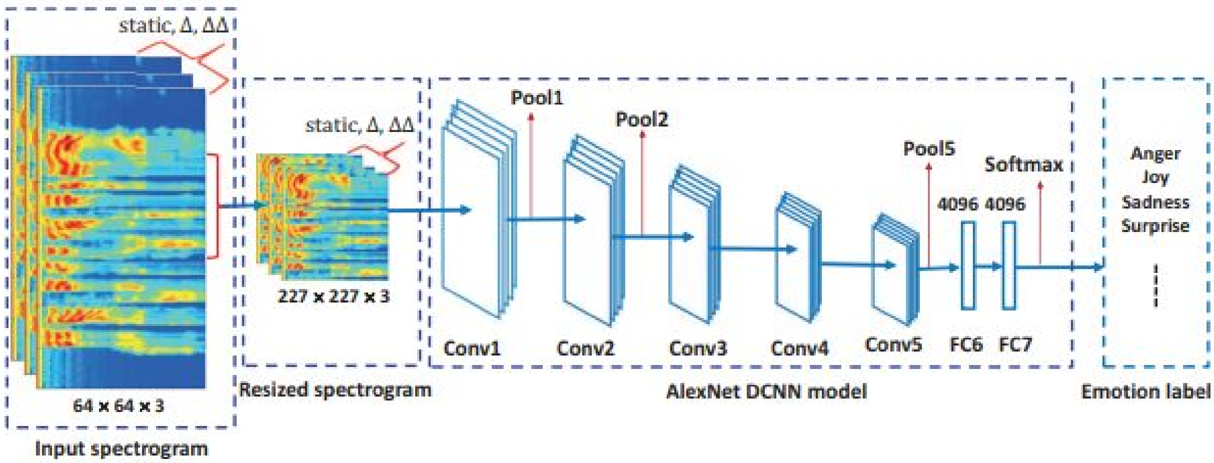


Figure 10: Overall Process of DCNN for Feature Extraction

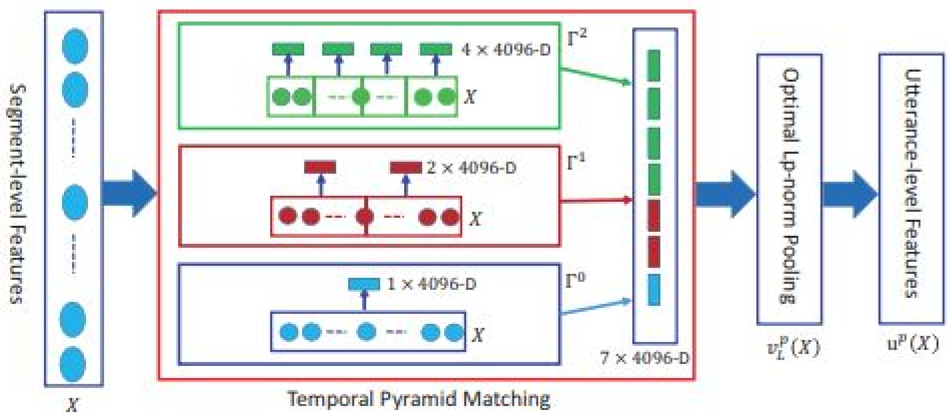


Figure 11: Overall Process of DTPM for Utterance-Level Feature Representation

2.3 SUMMARY

In sum, there are many kinds of Neural Network which can be implemented to learn and recognise the training data of human emotions before predicting the input data by comparing it with the output of the learning process within the process of Neural Networks. For instance, many different types of Neural Networks have been introduced by Veen, F. (2017) which can be used for emotional recognition like Deep Convolutional

Network (DCN), MultiLayer Perceptrons (MLP) or known as Deep Feed Forward (DFF), Recurrent Neural Network (RNN), and others more. So far, Deep Convolutional Neural Network is the most simplest concept in terms of theory and has the highest accuracy in predicting the results compared to the others Neural Networks.

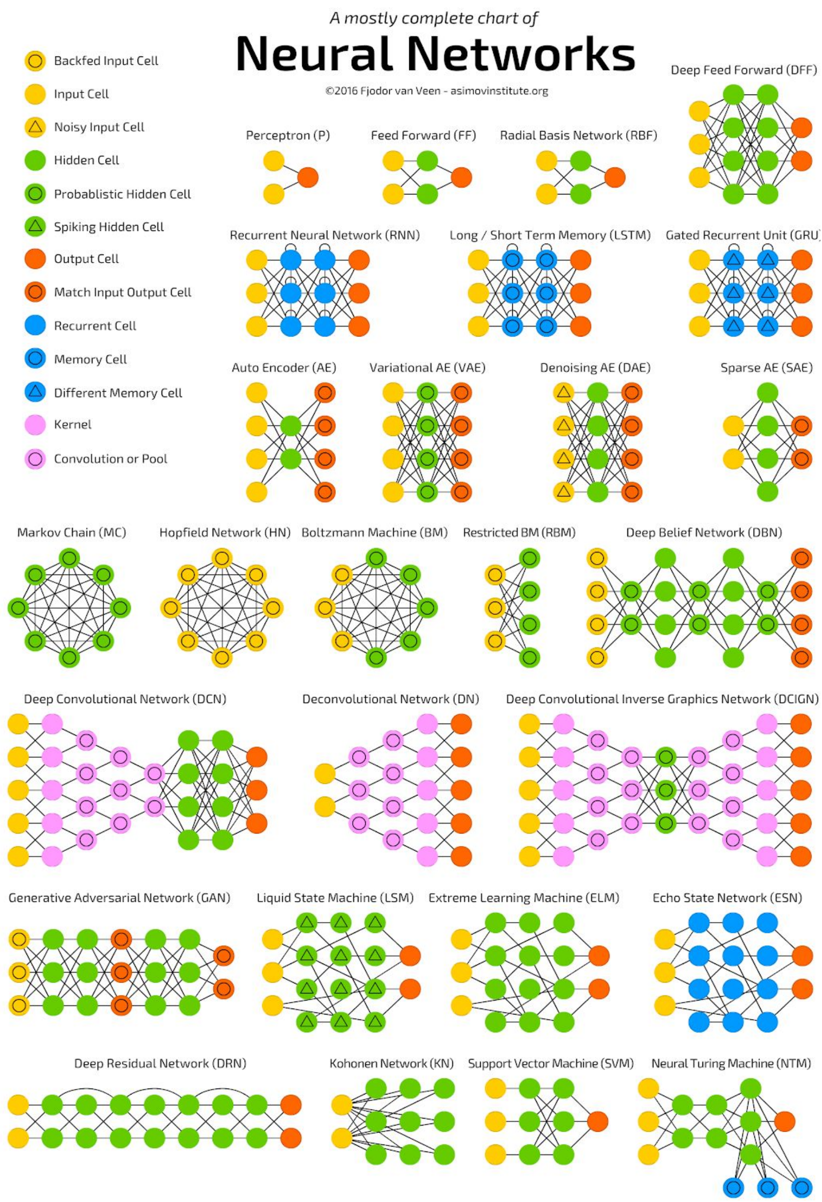


Figure 12: Types of Existing Neural Networks

Chapter 3: Methodology

This chapter will first introduce several different types of software methodologies which is popular to be used within the computing industry. Next, a best suitable software methodology is selected based on the development progress of the artefact as well as being supported with other projects of implementing the similar approach methodology throughout its development process.

3.1 Software Development Life Cycle (SDLC)

The development of the system can be related to the Software Development Life Cycle (SDLC) model. In relate to it, Singh, R. (2016) states the SDLC model represents the phases in producing a working software where each phases are related and interconnected with one another.

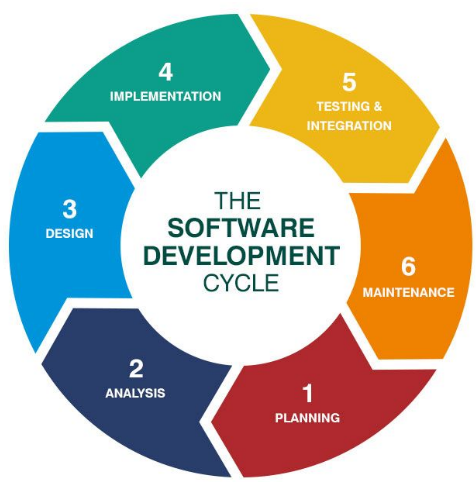


Figure 13: Software Development Life Cycle (SDLC) Model (Singh, 2016)

According to Figure 13, the following are the phases contained within the SDLC model.

1. **Planning:** decides on what actions should be taken as well as the solutions for the identified problems during this phase.
2. **Analysis:** find out the needs and the expectations from the end-users besides maintain a great communication among the client or the end-users and the programmer about the development progress of the artefact.
3. **Design:** describes the specification and define the overall architecture of the whole system.
4. **Implementation / Coding:** starts coding and developing the system by the programmers.
5. **Testing and Integration:** does several types of experiments to test on the system whether it has achieved the required requirements or not.
6. **Maintenance:** the process where system updates or upgrades happen after the system has been deployed to the users.

These general phases of Software Development Life Cycle (SDLC) are applied within the following examples of software methodologies which are popular to be used for developing a new software or system.

3.1.1 Waterfall Methodology

Waterfall Methodology is known as the traditional approach where each stages are arranged in a sequential order of one-by-one. Hence, it is also known as a linear-sequential life cycle model (SDLC Waterfall Model, undated).

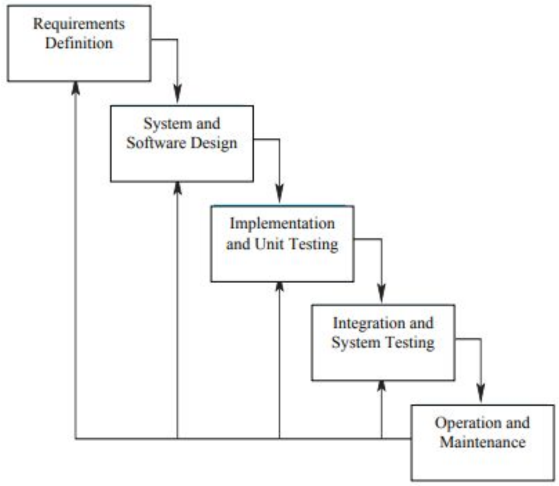


Figure 14: Flow Chart of Waterfall Methodology (Munassar and Govardhan , 2010)

However, there are pros and cons of implementing the concept of Waterfall methodology which are listed in Table 8 below (Powell-Morse, 2016).

Advantages	Disadvantages
Copes with the changes of team throughout the project	Allows limited or no changes to be made throughout the development process
Implements a fixed and structured schedule	No involvement or feedbacks from users or clients throughout the process
Allows Changes to be made early	Focus more at design (beginning) rather than testing (end)
Suited for Time-Limitation Project	

Table 8: Advantages and Disadvantages of Implementing Waterfall Methodology

Thus, Waterfall methodology is suitable for projects which do not require any changes to be made throughout the process like building an ATM system.

3.1.2 Agile Methodology

According to Figure 15, around 97% of organisations implement Agile methodology nowadays which contributes to 98 % success of Agile projects.

Figure 15: Statistics of Organisations Implementing Agile Methodology (VersionOne, 2018)

In other words, Agile Methodology is an improved version of Waterfall methodology which allows iteration of the whole process starting from designing the product till the finishing of the product. Unlike Waterfall Methodology, Agile Methodology strictly follows the concept within the Agile Manifesto which is explained by Lichtenberger, A. (2014).

The Agile Manifesto

Individuals and interactions	over	Processes and Tools
Working Product	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to change	over	Following a plan

Figure 16: List of Agile Manifesto (Lichtenberger, 2014)

From Figure 16, it can be concluded that Agile Methodology is basically focused on the development process of the artefact and the satisfactory of its customers instead of the success in fulfilling all the requirements of the artefact listed at the beginning of the project.

Furthermore, there are many types of Agile Methodologies like KanBan, SCRUM, Extreme Programming (XP), Crystal, and others more. For example, Agile SCRUM methodology is the most popular one which involves the engagement of the clients and users along the development process besides the process also tends to be flexible and adaptive to any changes that need to be made or added towards the final product. Figure 17 illustrates the overall workflow of implementing Agile SCRUM Methodology where the Product Backlog is the list of requirements discussed with the clients before classified them into Sprint Backlog (smaller classification of the most important requirements) before proceeding into Sprint process. Sprint process is where the iterations of designing, coding, and testing of the final product happens. Throughout the Sprint process, discussion of the development is often made within the daily Scrum.

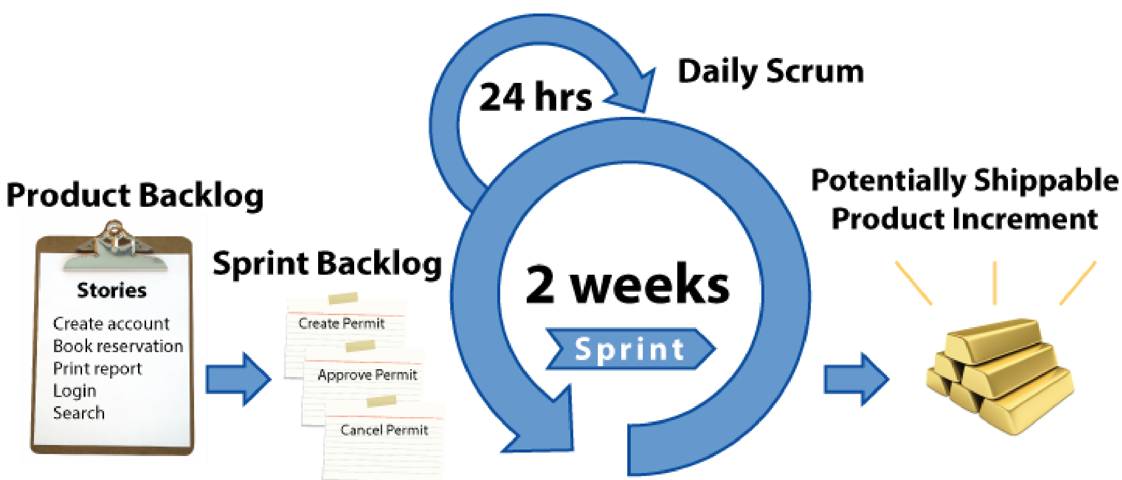


Figure 15: Overall Agile SCRUM Methodology (Ramusson, undated)

However, both Babar, M. (2009) and Haunts, S. (2014) mentioned about several pros and cons for implementing Agile Methodology throughout the development process as listed in Table 9 below.

Advantages	Disadvantages
Involve the developers at the early stage of product design	Difficult to estimate the overall effort needed at the start of the project
Prioritise the amount of work for each stage	Require large amount of time spent for meetings instead of developing the product
Able to deliver the main requirements on the agreed delivery date	Lack of designing and documentation leads to difficulty in understanding
Spends or Requires less amount of budget needed	No considerations for alternative, potentially better design choices can be missed
Sharings, discussions of progress, and changes are determined through frequent meetings	Focus more on satisfying users instead of producing a qualitative product

Table 9: Advantages and Disadvantages of Implementing

3.1.3 Spiral Model

Munassar, N. and Govardhan, A. (2010) mentioned that Spiral Model is similar to the Incrementation Methodology but with mostly focus on its risk analysis. On the other hand, Powell-Morse, A. (2016) states that Spiral Model is used to help in determining the optimal process model for a given project.

According to Figure 16, the four main stages are Planning, Risk Analysis, Coding and Customer Evaluation. In further explanation, Planning is where all the requirements, objectives, and alternatives are considered at this stage while Risk Analysis is where all the alternatives are considered for issues that will be potentially happened. Meanwhile, Coding is where development and testing of the product happens while Customer Evaluation is where the final product of the stage is reviewed as well as its cost.

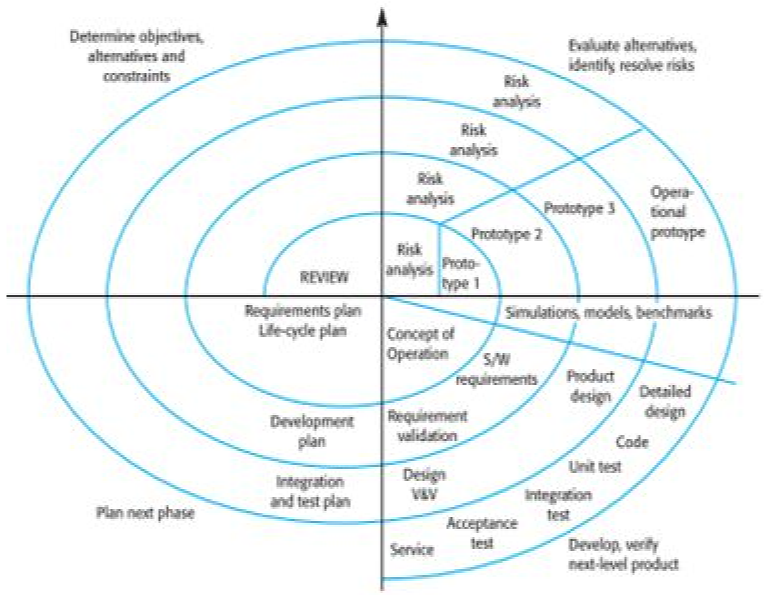


Figure 16: Stages within Spiral Methodology (Munassar and Govardhan, 2010)

However, Kienitz, P. (2017) listed in Table 10 about several pros and cons based on implementing Spiral Model within the development of the product.

Advantages	Disadvantages
Prevents most of the risks from happening and disrupting the development progress	Not suitable for smaller projects which do not contain much risks - might costly for them
Software is produced at the early stage of the project lifecycle	Require the presence of risk specialist expertise
Well-suited for complex yet large scale projects and mission-critical developments	The success in project highly depends on its risk analysis
Highly emphasis on documentation control and the approval of client	Rules and protocols must be adhered to strictly throughout the development
Monitoring is easy and effective	Documentation can be heavy, due to the number of intermediate stages

Extremely flexible model which allows additional functionality to be added at the later stage	End of project may be difficult to define beforehand
Requirements might change throughout the project lifespan	Considered mostly as a complex process

Table 10: Advantages and Disadvantages of Implementing Spiral Model

3.1.4 Comparison among the Software Methodologies

Overall, Rastogi, V. (2015) has made a comparison and summarised the above methodologies.

	Waterfall	**Agile**	**Spiral**
Objectives	Emphasis on quality control	Emphasis Customer Satisfaction	Emphasis on risks and costs
Suitable for	Project with Clear List of Requirements	Project requires up-to-date functionalities	Complex Project where risks and costs are important

3.2 Project Management

This section explains how Agile Methodology is the most suitable methodology to be implemented within the context of my artefact development progress.

Among the choices listed in Section 3.1, Spiral Model is not suitable to be implemented throughout the artefact development as there are ***not much risks to be considered throughout the artefact development*** according to the Risk Matrix in Appendix 2. Besides Spiral Model, Waterfall methodology is also not suitable to be implemented as the progress of the artefact development do not work or follow the timescale as planned

Gantt Chart in Appendix 1 presents an overview of the timescale for the whole development of the project. However, this timescale is only used as a guideline throughout the process because the requirements of the artefact are unclear at the beginning of the project.

Therefore, in this context, a combination of Agile Methodology has been applied throughout the whole process. The following are several supportive reasons in the following which supports the above statement.

First of all, the aim of having this project is for us to ***learn from the process of developing the artefact*** instead of focusing on developing of the successful working artefact. Thus, this is similar to the main concept of implementing Agile methodology.

Secondly, during the process of coding the artefact, ***my friend and I have both worked together and helped out with one another for our own projects***. In the context of my project, my friends have helped me a lot in looking out for any errors made within my code whenever I am stuck in trying to solve the errors. This situation is similar to Pair Programming of Extreme Programming (XP) where my friend takes the role as the observer while I am the driver. Along this process, we share our knowledge and makes coding more efficiently and interesting than coding alone.

Thirdly, we are required to ***update our progress of the project as frequent as possible with our supervisor***. Hence, this can be related to the Daily SPRINT of Agile SCRUM where our supervisor acts as our client while we are the development team who do the jobs of designing, coding, and testing.

Lastly, ***a progression board is made to keep track of the stages left till the completion of the whole project***. For this project, I have used Trello where I listed all the requirements of the proposed artefact under the To-Do list at the start of the project. Later some of the requirements are prioritized and selected to be worked on at each stages. This process will be repeated once the ongoing stage is completed and a new stage is started. In relate to Agile methodology, this overall process is similar to KanBan process which allows additional requirements to be added into the To-Do list easily and completed in a cyclic manner.

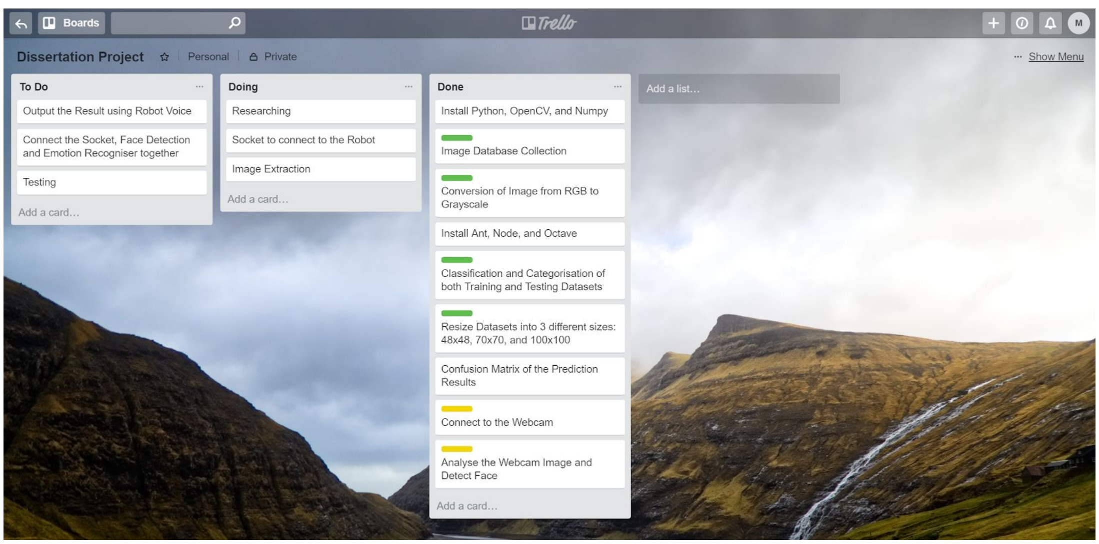


Figure 17: KanBan Board of Progression

3.3 Software Development

This section evaluate on which Software Methodology will be suitable to my project in terms of the requirements of my artefact which is about recognising human emotions by predicting from the detected facial expression.

This software development fits along with the concept of Agile methodology as the artefact is a ***non-critical system*** which will not propose life-threatening threats for the users. In other words, it is a system which ***focus more on user satisfactory*** and requires in implementing up-to-date trends like which method is the best to be implemented within the robot self-learning system. In this context, this artefact implements Convolutional Neural Network (CNN) which is the trending neural network in achieving the highest results accuracy compared to the other methods like Multi-Layer Perceptron (MLP) and Recurrent Neural Network (RNN).

Unlike Waterfall Methodology hardly allows any changes for the artefact requirements, the requirements for this artefact changes along the development process in order to create a higher satisfactory results for the client and user. For instance, the artefact was first planned to be tested on one fixed size for the image dataset used (48x48 pixels), but it is added later to include another two different sizes of image datasets, such as, 70x70 pixels and 100x100 pixels. In addition, one of the proposed objectives regarding on recognising facial expression with only partial face detected is removed and changed during the development process.

3.4 Toolsets and Machine Environment

This section will be mentioning and listing all the tools used for the development and documentation of this artefact.

3.4.1 Software Development Tools

All the tools listed below are installed and used on Windows 10 x64-bit.

1. **Python 2.7**

Website : https://www.python.org/download/releases/2.7/

Description :

Python is one of the object-oriented programming and high-level language (Python Overview, undated). Python also supports a large range of libraries besides containing a large range of built-in functions. Owen-Hill, A. (2016) emphasis that applying Python in Robotics field has gained more popularity than applying C++ due to its simplicity in understanding the code as Python programming language is more similar to the English we speak.

Machine Environment: Windows, Linux, and Mac OS

Usage in this Context: It is used as a selective programming language to work along OpenCV.

2. **PyCharm - Python IDE**

Website : https://www.jetbrains.com/pycharm/

Description :

PyCharm is free Integrated Development Environment (IDE) for coding in Python which has a similar interface and features of another JetBrain's IDE. Most important of all, PyCharm is an IDE that enable NumPy and Matplotlib to work with arrays and interactive plots. Furthermore, PyCharm also support JavaScript and Node.js which will later be used for Emotional Recognition (Vasconcellos, P. H., 2017).

Machine Environment: Windows, Mac OS and Linux

Usage in this Context: PyCharm is used as an interface for coding mainly in Python while provide support with other programming language as well.

3. **OpenCV 3.4.1**

Website: https://sourceforge.net/projects/opencvlibrary/files/opencv-win/

Description :

OpenCV is a short form for Open Source Computer Vision Library which is a free and available computer vision and machine learning software library building up more than 2500 built-in functions.

"This software library is able to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, , find similar images from an image database, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, and others more." (About - OpenCV library, undated)

Machine Environment: Windows, Linux, Android, and Mac OS

Usage in this Context: OpenCV is implemented along with Python to manipulate the images, for instance, resize image, convert image from colour to grayscale, and others more.

4. **NumPy 1.11.2**

Website : https://sourceforge.net/projects/numpy/files/NumPy/

Description :

NumPy is a short form of Numerical Python which is the prime software tools for scientific computing along with Python. This is because NumPy support reading a large array besides being able to perform simple mathematical and scientific calculations. Other than its scientific usage, NumPy is also able to integrate a large variety of databases quickly without any difficulties. (Willems, 2017)

Machine Environment: Windows, Linux, and Max OS X

Usage in this Context:

NumPy is implemented along with OpenCV and Python to manipulate the large array of the image.

5. **Java JDK 9.0.4**

Website: http://www.oracle.com/technetwork/java/javase/downloads/index.html

Description :

Java Development Kit or known as JDK is where Java applications and applets are developed using Java Runtime Environment (JRE), a programming language translator (.java), a compiler (.javac), and other more tools. Both tools are loaded from command prompt where Java source files are basic text files saved with an extension of .java. After writing and saving Java source code, the javac compiler is invoked to create .class files to run the java program. (What is Java Development Kit (JDK)? - Definition from Techopedia, undated)

Machine Environment: Windows, Linux, Solaris and Mac OS (additional adaptions tool needed)

Usage in this Context:

Java is installed to run Ant and Node system respectively which are written in Java programming language.

6. **Apache Ant 1.9.10**

Website : https://ant.apache.org/srcdownload.cgi

Description : Ant is used to build and deploy Java-written programmes using .xml file for configuration (Vogel, 2016).

Machine Environment: Any as long with Java 8.0 at least

Usage in this Context: Ant is used to build program where it will import the images into the link of the Node for Convolutional Neural Network (CNN).

7. **Node 0.10.25**

Website : https://nodejs.org/en/blog/release/v0.10.25/

Description : Node.js is a free open-source development platform to execute JavaScript code (What is Node.js? - Definition from WhatIs.com, undated).

Machine Environment: Any

Usage in this Context: Node is used to start Training or Testing for the Convolutional Neural Network (CNN).

8. **GNU Octave 4.2.1**

Website : https://www.gnu.org/software/octave/download.html

Description : Octave is a free open-source software that is used for numerical computations.

Machine Environment: Any

Usage in this Context: Octave is used to plot the results into graph.

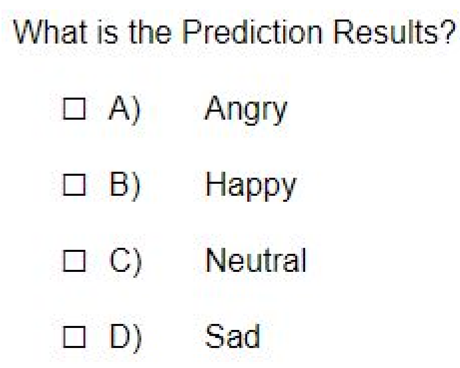
3.4.2 Project Development Tools

1. Google Docs
2. Trello
3. Microsoft Excel

3.5 Research Methods

For my research testing, Focus Group has been chosen as my selected research method while its results are collected quantitatively.

According to the collection results of data, it is a mixture of both an objective and observable data as well as subjective and self-reported data. This is because subjective and self-reported data is tested with the images manually imported into the database whereas the objective and observable data is tested with people at the real-time scenario. Overall, the data collected is considered as nominal as in Figure 18.



What is the Prediction Results?

- ☐ A) Angry
- ☐ B) Happy
- ☐ C) Neutral
- ☐ D) Sad

Figure 18: Nominal Data Collection Method

3.6 SUMMARY

In short, Agile methodology is considered as the most suitable methodology towards the development of this artefact. This is due to its unpredictable and user focused nature either from the view of project management or software development. Hence, focus group is carefully selected as the chosen research method due to the nature of the artefact is targeted towards the market of the younger generation whom is more of a fan of technology and have the higher possibility of purchasing this product.

Chapter 4: Design, Development, and Evaluation

This chapter will be documenting on the development process of the artefact in details. For this, the artefact is consisted of two main systems (face recognition and emotion recognition) which are done separately.

4.1 Software Development Projects

Using Agile methodology within the software development as mentioned in Chapter 3, the requirements of the overall proposed system are divided into several stages to be completed. For instance, the overall proposed system is divided into two main stages which are known as Facial Recognition and Emotional Recognition as listed below.

4.1.1 Facial Recognition

This is the process of enabling the camera which connects to the system is able to detect and recognise human faces in real-time scenario.

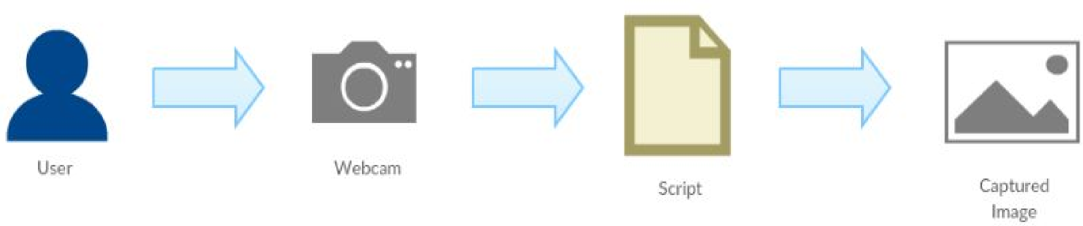


Figure 19: Overall Process of Facial Recognition

4.1.1.1 Requirements Extraction, Collection, and Analysis

This is the phase where it is determined to collect the detected face photos only while the other objects in the background are ignored. Besides that, it is also the phase to determine which programming language shall be used as well as the tools needed. For instance, Python is selected as the programming language to be worked on along with OpenCV.

4.1.1.2 Design

This is the phase regarding the process on facial recognition is determined and planned as in Figure 20.

Figure 20: Flow Chart of Planned Facial Recognition Process

4.1.1.3 Building or Coding

The camera-device is connected to the system through vc = cv2.VideoCapture(0) and started recording vc.read().

Later region of face within the frame of the captured video can be detected with the help of Haar-cascade files provided by OpenCV (OpenCV: Face Detection using Haar Cascades, undated). Haar-cascade files can be opened through cv2.CascadeClassifier(file_location) as the following.

```
eye_cascade = cv2.CascadeClassifier('C:\Python27\Lib\opencv\sources\data\haarcascades\haarcascade_eye.xml')
faceCascade = cv2.CascadeClassifier("C:\Python27\Lib\opencv\sources\data\haarcascades\haarcascade_frontalface_default.xml")
```

The face region within the captured video using faceCascade.detectMultiScale(gray_image, minSize, maxSize) and marked them with a coloured rectangle as below.

```
for (x,y,w,h) in faces:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0,0, 255), 2)
```

These steps are similarly applied to detect the eyes region except only changing the input image of the detectMultiScale() and rectangle() with a cropped face area only.

4.1.1.4 Testing

Figure 20 shows the result of face and eyes regions being detected within the captured video frames.

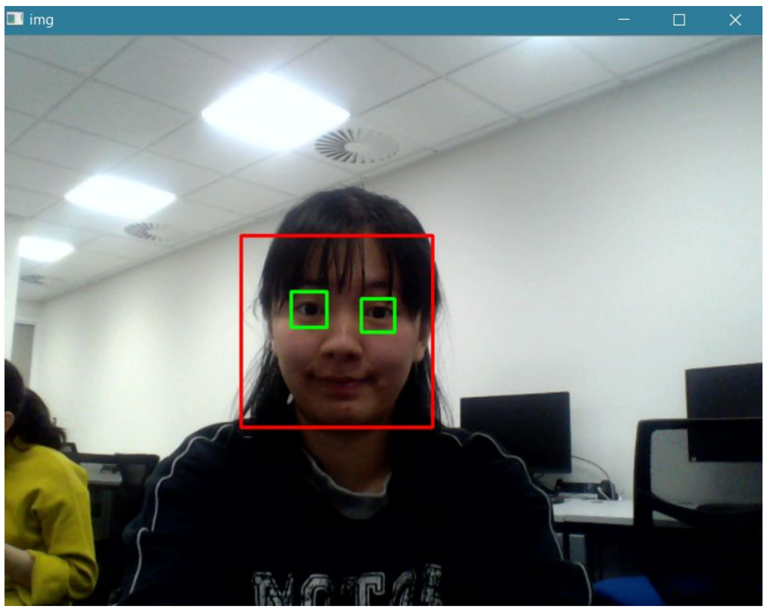


Figure 20: Testing Results of Facial Recognition

4.1.1.5 Operations and Maintenance

4.1.2 Emotional Recognition

Emotional Recognition is the process where the images are input into the Convolutional Neural Network to be trained and tested for its accuracy results in predicting the facial emotions as illustrated in Figure 21.

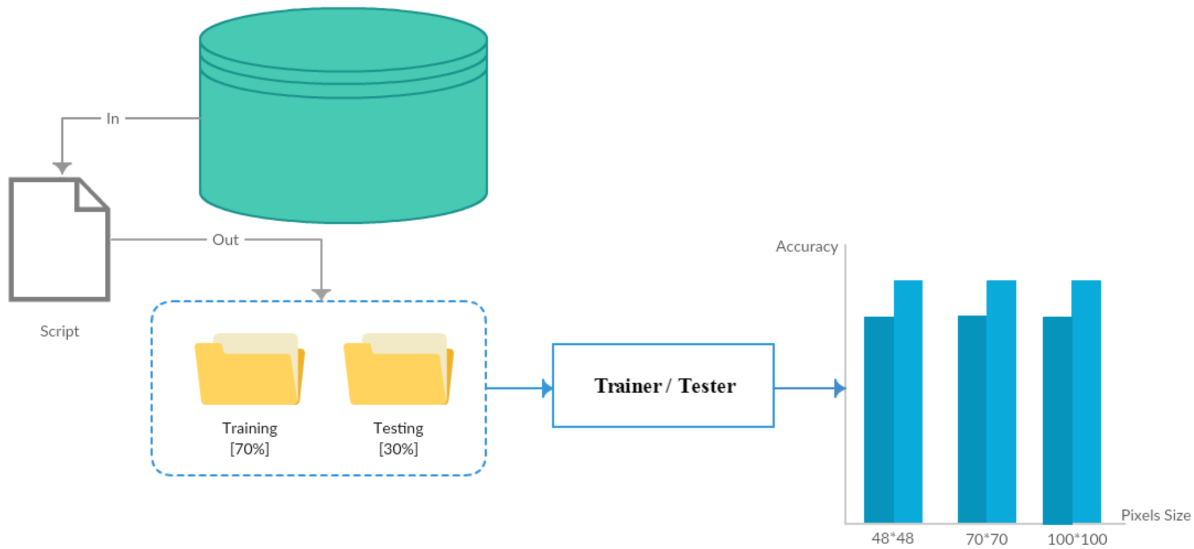


Figure 21: Overall Process of Emotional Recognition

4.2.1.1 Requirements Extraction, Collection, and Analysis

This phase is where all the Requirements for the Emotional Recognition are considered and listed, such as, the input database and the software needed to run the code like Java JDK, Ant, Node, and Octave which are mentioned in Section 3.4.

The overall image database is collected and a combination from three different sources which are Karolinska Directed Emotional Faces (KDEF) Database [(E.Lundqvist, D., Flykt, A., and Öhman, A., 1998)], Japanese Female Facial Expression (JAFFE) Database [(Lyons, M., Akemastu, S., Kamachi, M. and Gyoba, J., 1998)], and Yale Face Database [(Belhumeur, P., Hespanha, J. and Kriegman, D., 1997)]. These images are converted into grayscale before performing the emotional recognition process.

4.2.1.2 Design

Figure 22 illustrates the overall process for emotional recognition till achieving the learning results.

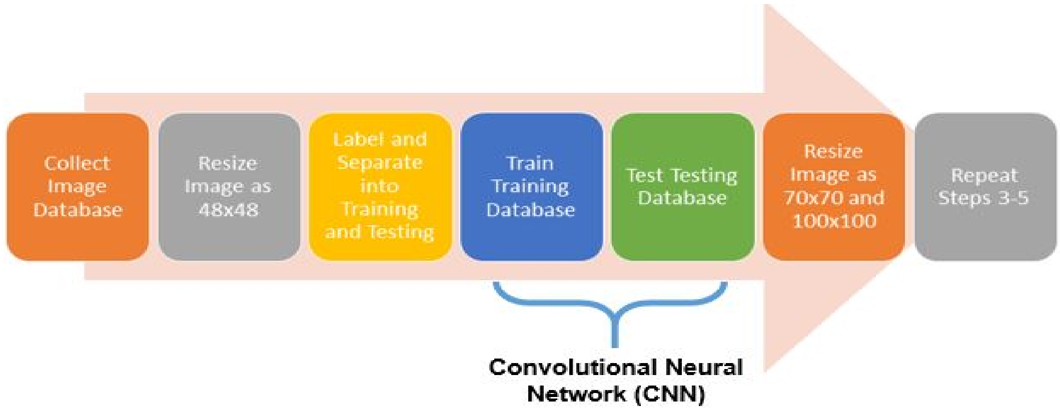


Figure 22: Flow Chart of Panned Emotional Recognition Process

4.2.1.3 Building or Coding

With the help of Convolution Neural Network (CNN) created by Cuayahuitl, H. (2017), the collected image database are then imported into the learning CNN through Ant (SimpleDS_ImageRecognitionTrainer) as shown in Figure 23.

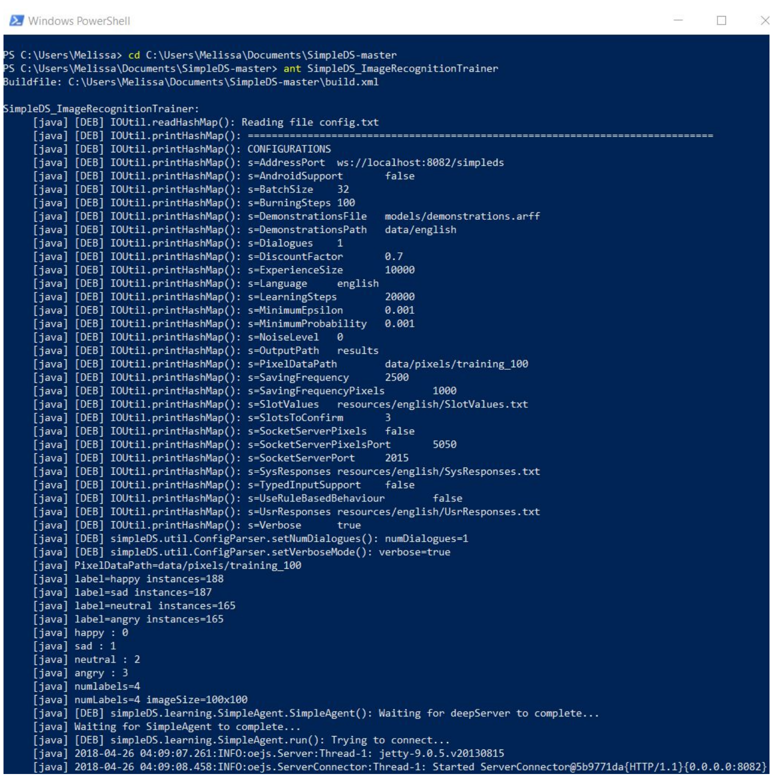


Figure 23: Screenshot of Importing Images using the Ant System

At the same time, the Ant Trainer system is connected to another Window PowerShell Terminal which runs the Node system of the CNN in Training Mode.

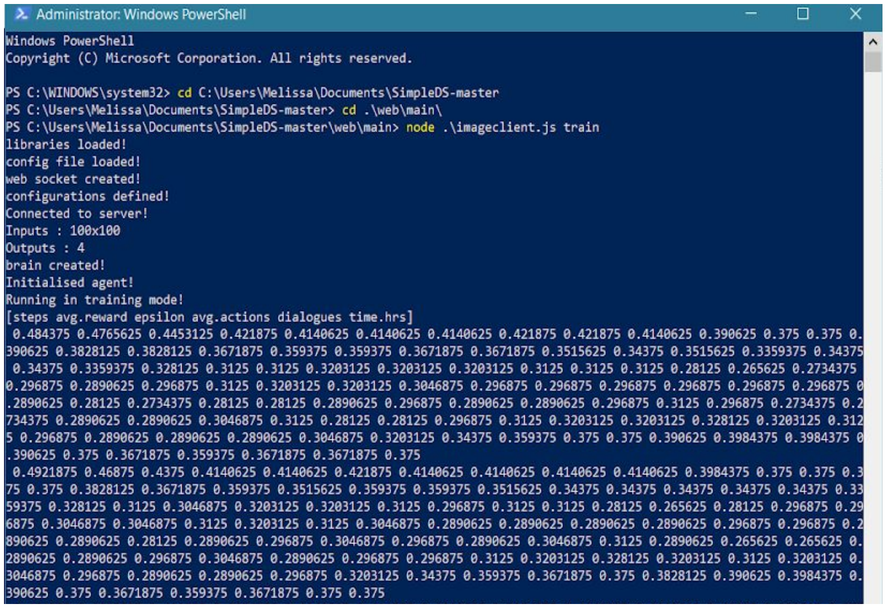


Figure 24: Screenshot of Training Images using Node System

Later GNU Octave is run at another new terminal to plot the learning results outcome of CNN which run in the Node system previously. The plotted results is then shown in graph like the following.

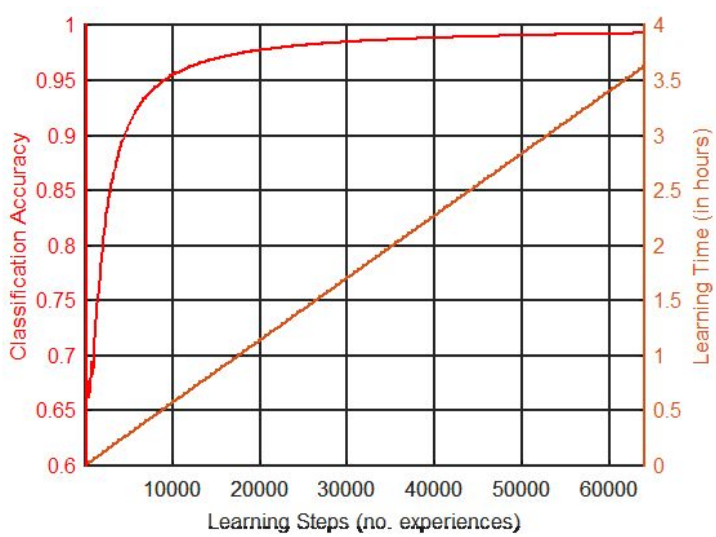


Figure 25: Line Graph of the Learning Curve

4.2.1.4 Training and Testing Results

Pixels Size	Purpose	Accuracy
48 Pixels	Training	0.970
	Testing	0.810
70 Pixels	Training	0.990
	Testing	0.859
100 Pixels	Training	0.992
	Testing	0.855

Table 11: Tabulation of Results for Training and Testing

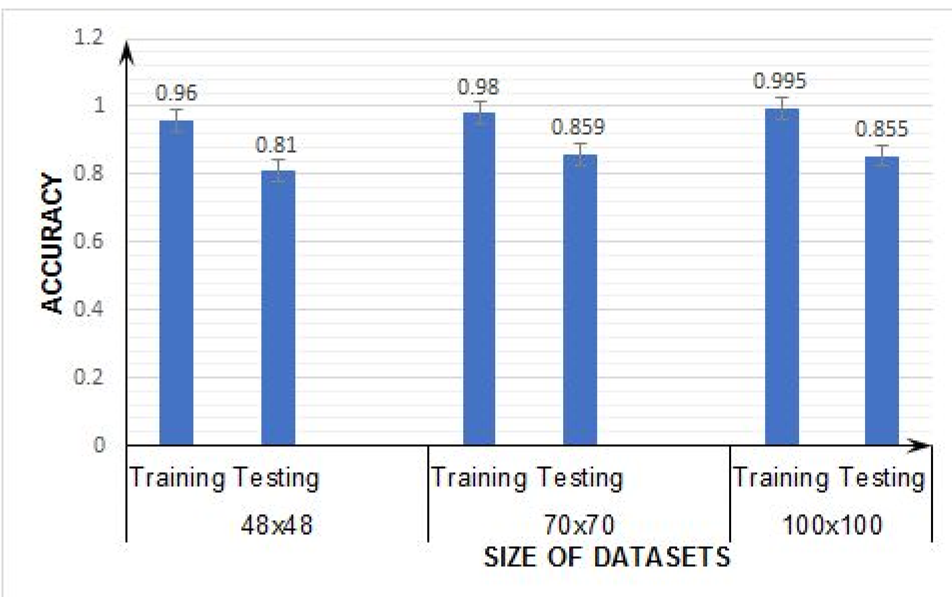


Figure 21: Bar Graph of Accuracy Results

4.2.1.5 Operations and Maintenance

4.2 Research Projects

This artefact is tested on a focused group of people with the same background of being a student of University of Lincoln. Before the experiment is conducted, a Confidential Agreement (See Appendix 3) is made to comply with the ethical procedures. After giving consent towards the experiment, the participants are required to face the webcam provided directly and instructed to express the expression shown at the PowerPoint slides once at a time.

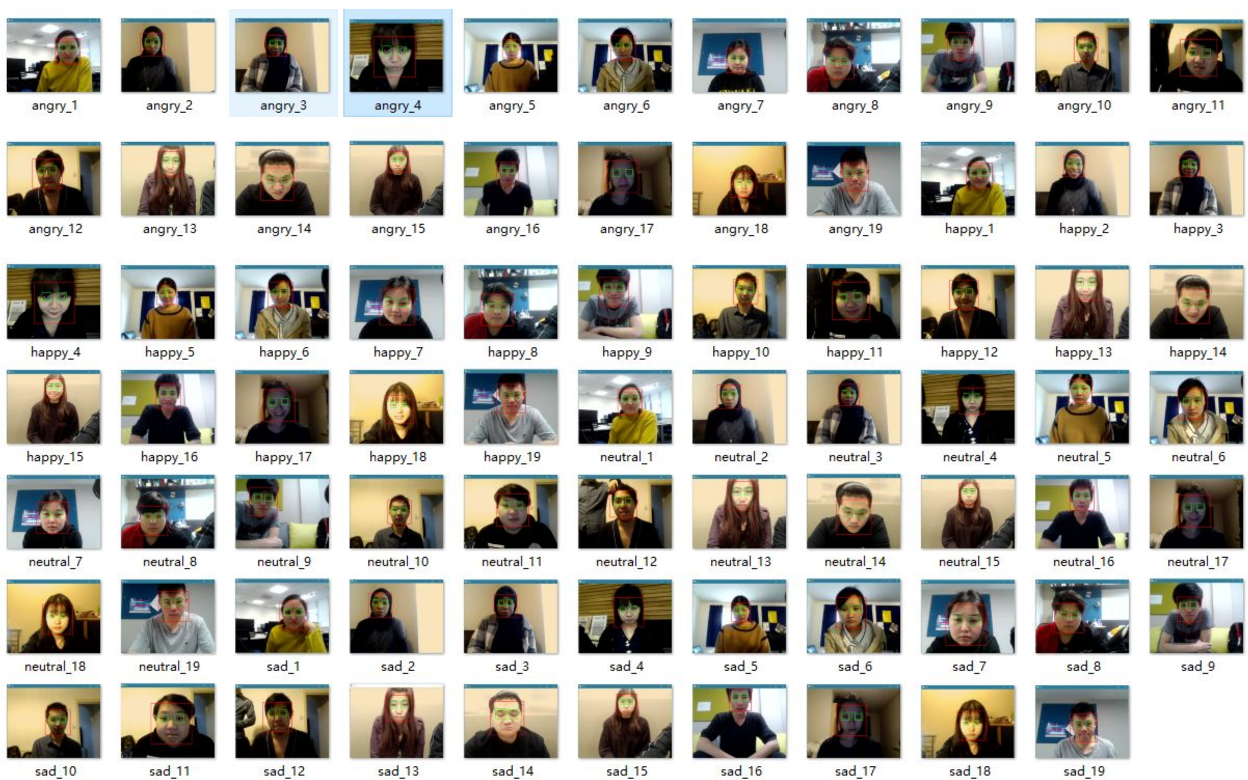


Figure 22: Overall of Participants Involved in the Experiment

Figure 22 are the images produced by Face Recognition system. It is then converted into grayscale and resized into 48x48, 70x70, and 100x100 respectively before importing them into Emotional Recognition to be analysed.

48 x 48	**Recognised Emotions**			
	Angry	**Happy**	**Neutral**	**Sad**
Angry	**81.30%** *300*	3.79% *14*	6.78% *25*	8.13% *30*
Happy	0.00% *0*	**90.55%** *345*	6.04% *23*	3.41% *13*
Neutral	5.14% *20*	10.28% *40*	**77.12%** *300*	7.46% *29*
Sad	11.64% *42*	8.03% *29*	5.54% *20*	**74.79%** *270*

Table 12: Confusion Matrix of 48x48 Image Pixels Size

70 x 70	**Recognised Emotions**			
	Angry	**Happy**	**Neutral**	**Sad**
Angry	**86.81%** *316*	1.37% *5*	1.11% *4*	10.71% *39*
Happy	0.00% *0*	**92.88%** *326*	4.27% *15*	2.85% *10*
Neutral	1.02% *4*	6.09% *24*	**78.17%** *308*	14.72% *58*
Sad	1.02% *4*	8.12% *32*	4.82% *19*	**86.04%** *339*

Table 13: Confusion Matrix of 70x70 Image Pixels Size

100 x 100	**Recognised Emotions**			
	Angry	**Happy**	**Neutral**	**Sad**
Angry	**96.37%** *345*	0.88% *3*	2.51% *9*	0.28% *1*
Happy	0.00% *0*	**91.03%** *345*	5.28% *20*	3.69% *14*
Neutral	9.78% *35*	7.54% *27*	**75.42%** *270*	7.26% *26*
Sad	9.38% *38*	5.43% *22*	7.41% *30*	**77.78%** *315*

Table 14: Confusion Matrix of 70x70 Image Pixels Size

Table 12, Table 13, and Table 14 are the Confusion Matrix Results of the Prediction Accuracy of Emotional Recognition system.

4.3 SUMMARY

In short, image database of 70x70 is the best size for the image in achieving the highest accuracy rate using Convolutional Neural Network. Using this scenario, the image database of 100x100 is overfitting as there is a slight decrease in accuracy rate as shown in Figure 21 and Table 11.

Chapter 5: Conclusion

In conclusion, the image database of 70x70 pixels size is the best size to be conducted for Emotional Recognition. The Convolutional Neural Network (CNN) is a neural network which is able to produce a testing accuracy rate as high as 85.9%.

Chapter 6: Reflective Analysis

Due to the time constraint, I am unable to complete the artefact as proposed completely and perfectly as predicted. This is mainly because of the unexpected events like I have spend too much time at the beginning in learning several new languages like Python, OpenCV and others to try in figuring out the code. Luckily, things get better and clear along the way with some guidelines and help provided by my supervisor. Although it is surely an extremely challenging research which I have proposed, I have learned and liked the experience and knowledge that I have gained from it. Agile methodology is the most suitable methodology for my conditions as I prefer to do things step by step in an orderly manner. However, there are some improvements need to be made while using this methodology like time limitation fixed for each stages so that the final product can be finished fulfilling all the aim and objectives as proposed.

Bibliography

Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. (2002) Agile software development methods: Review and Analysis. Espoo [Finland]: VTT.

About - OpenCV library. Available from https://opencv.org/about.html [accessed 26 April 2018].
Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. (2002) Agile software development methods: Review and Analysis. Espoo [Finland]: VTT.

Alwan, M. (2015) What is System Development Life Cycle?. Available from https://airbrake.io/blog/sdlc/what-is-system-development-life-cycle [accessed 23 April 2018].

Babar, M. (2009) An Exploratory Study of Architectural Practices and Challenges in Using Agile Software Development Approaches. In: 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, Cambridge, UK. IEEE, 81 - 90. Available from https://ieeexplore.ieee.org/document/5290794/?denied [accessed 22 April 2018].

Barman, A. and Dutta, P. (2017) Facial expression recognition using distance and shape signature features. Pattern Recognition Letters 1-8. Available from https://www.sciencedirect.com/science/article/pii/S0167865517302246 [accessed 17 April 2018].

Cuayahuitl, H. (2017) Deep Reinforcement Learning for Conversational Robots Playing Games. In: *IEEE RAS International Conference on Humanoid Robots*, REP Theatre, Birmingham, UK. IEEE.

Dehghan, A., Ortiz, E., Shu, G. and Masood, S. (2017) DAGER: Deep Age, Gender and Emotion Recognition Using Convolutional Neural Network. *arXiv preprint arXiv:1702.04280.* Available from https://arxiv.org/pdf/1702.04280.pdf [accessed 29 March 2017].

Graves, A., Schmidhuber, J., Mayer, C., Wimmer, M. and Radig, B. (2008) Facial Expression Recognition with Recurrent Neural Networks. In: *Proceedings of the International Workshop on Cognition for Technical Systems,* Munich, Germany.

Available from https://mediatum.ub.tum.de/doc/1289310/246627.pdf [accessed 19 April 2018].

Haunts, S. (2014) Advantages and Disadvantages of Agile Software Development. Available from https://stephenhaunts.com/2014/12/19/advantages-and-disadvantages-of-agile-software-development/ [accessed 22 April 2018].

Karthika (2017) RiotZ.org - Top 10 Amazing Social Robots That Really Exist. Available from http://www.rtoz.org/2017/11/03/top-10-amazing-social-robots-that-really-exist/ [accessed 22 February 2018].

Khanchandani, K. and Hussain, M. (2009) Emotion recognition using multilayer perceptron and generalized feed forward neural network. *Journal of Scientific and Industrial Research*, 68(5) 367-371. Available from http://nopr.niscair.res.in/handle/123456789/3787 [accessed 18 April 2018].

Kienitz, P. (2017) The pros and cons of Spiral Software Development | DCSL Software Ltd. Available from https://www.dcslsoftware.com/pros-cons-spiral-software-development/ [accessed 22 April 2018].

Krizhevsky, A., Sutskever, I. and Hinton, G. (2012) ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 1097-1105. Available from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf [accessed 20 April 2018].

Lee, C. (2017) *Understanding Bidirectional RNN in PyTorch – Towards Data Science*. Available from https://towardsdatascience.com/understanding-bidirectional-rnn-in-pytorch-5bd25a5dd66 [accessed 20 April 2018].

Lichtenberger, A. (2014) What IT Service Management can learn from the Agile Manifesto (and vice versa) - Disruptive agile Service Management. Available from https://blog.itil.org/2014/08/allgemein/what-it-service-management-can-learn-from-the-agile-manifesto-and-vice-versa/ [accessed 21 April 2018].

Munassar, N. and Govardhan, A. (2010) A Comparison Between Five Models Of Software Engineering. *IJCSI International Journal of Computer Science Issues*, 7(5) 94 - 101.

OpenCV: Face Detection using Haar Cascades. Available from https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html [accessed 25 April 2018].

Owen-Hill, A. (2016) What is the Best Programming Language for Robotics?. Available from https://blog.robotiq.com/what-is-the-best-programming-language-for-robotics [accessed 24 April 2018].

Powell-Morse, A. (2016) Waterfall Model: What Is It and When Should You Use It?. Available from https://airbrake.io/blog/sdlc/waterfall-model [accessed 21 April 2018].

Powell-Morse, A. (2016) Spiral Model: Software Development For Critical Projects. Available from https://airbrake.io/blog/sdlc/spiral-model [accessed 22 April 2018].

Python Overview. Available from https://www.tutorialspoint.com/python/python_overview.htm [accessed 24 April 2018].

Ramusson, J. Scrum. Available from http://www.agilenutshell.com/scrum [accessed 22 April 2018].

Rastogi, V. (2015) Software Development Life Cycle Models-Comparison, Consequences. International Journal of Computer Science and Information Security, 6(1) 168-172. Available from https://pdfs.semanticscholar.org/577c/fae86ee8bd01d64783c1c6d240523bea3b03.pdf [accessed 23 April 2018].

Schuster, M. and Paliwal, K. (1997) Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11) 2673-2681. Available from https://ai.intel.com/wp-content/uploads/sites/53/2017/06/BRNN.pdf [accessed 20 April 2018].

SDLC Waterfall Model. Available from https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm [accessed 21 April 2018].

Shan, C., Gong, S. and McOwan, P. (2008) Facial expression recognition based on Local Binary Patterns: *A comprehensive study. Image and Vision Computing*, 27(6) 803-816. Available from https://ac.els-cdn.com/S0262885608001844/1-s2.0-S0262885608001844-main.pdf?_tid=e826e03d-468b-4e3e-8ffb-a78893ae2771&acdnat=1523805845_60458d9c4ffe55f862d983fb70a5353e [accessed 15 April 2018].

Singh, R. (2016) Easy way to understand SDLC. Available from http://www.dignitasdigital.com/blog/easy-way-to-understand-sdlc/ [accessed 23 April 2018].

Vasconcellos, P. H. (2017) Top 5 Python IDEs For Data Science. Available from https://www.datacamp.com/community/tutorials/data-science-python-ide [accessed 26 April 2018]

Veen, F. (2017) NEURAL NETWORK ZOO PREQUEL: CELLS AND LAYERS. Available from https://www.asimovinstitute.org/author/fjodorvanveen/ [accessed 20 April 2018].

VersionOne (2018) *12th Annual State of Agile Report.* VersionOne. Available from https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report [accessed 22 April 2018].

What is Java Development Kit (JDK)? - Definition from Techopedia. Available from https://www.techopedia.com/definition/5594/java-development-kit-jdk [accessed 26 April 2018].

What is Node.js? - Definition from WhatIs.com. Available from https://whatis.techtarget.com/definition/Nodejs [accessed 26 April 2018].

Willems, K. (2017) Python Numpy Array Tutorial. Available from https://www.datacamp.com/community/tutorials/python-numpy-tutorial#a [accessed 26 April 2018].

Xi, M., Chen, L., Polajnar, D. and Tong, W. (2016) FACE RECOGNITION SYSTEM BY USING BLOCK BASED LOCAL BINARY PATTERN (LBP) AND DEEP NEURAL NETWORK (DNN). *International Journal of Advance Engineering and Research Development*, 4(10). Available from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7532955 [accessed 15 April 2018].

Zhang, T., Zheng, W., Cui, Z., Zong, Y. and Li, Y. (2017) Spatial-Temporal Recurrent Neural Network for Emotion Recognition. *IEEE Transactions on Cybernetics*, 13(9) 1-9. Available from https://arxiv.org/pdf/1705.04515.pdf [accessed 18 April 2018].

Zhang, S., Zhang, S., Huang, T. and Gao, W. (2017) Speech Emotion Recognition Using Deep Convolutional Neural Network and Discriminant Temporal Pyramid Matching. *IEEE Transactions on Multimedia 1-1*. Available from https://ieeexplore.ieee.org/abstract/document/8085174/ [accessed 20 April 2018].

Database

1. **Karolinska Directed Emotional Faces (KDEF) Database**

E.Lundqvist, D., Flykt, A., and Öhman, A. (1998). The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.

2. **Japanese Female Facial Expression (JAFFE) Database**

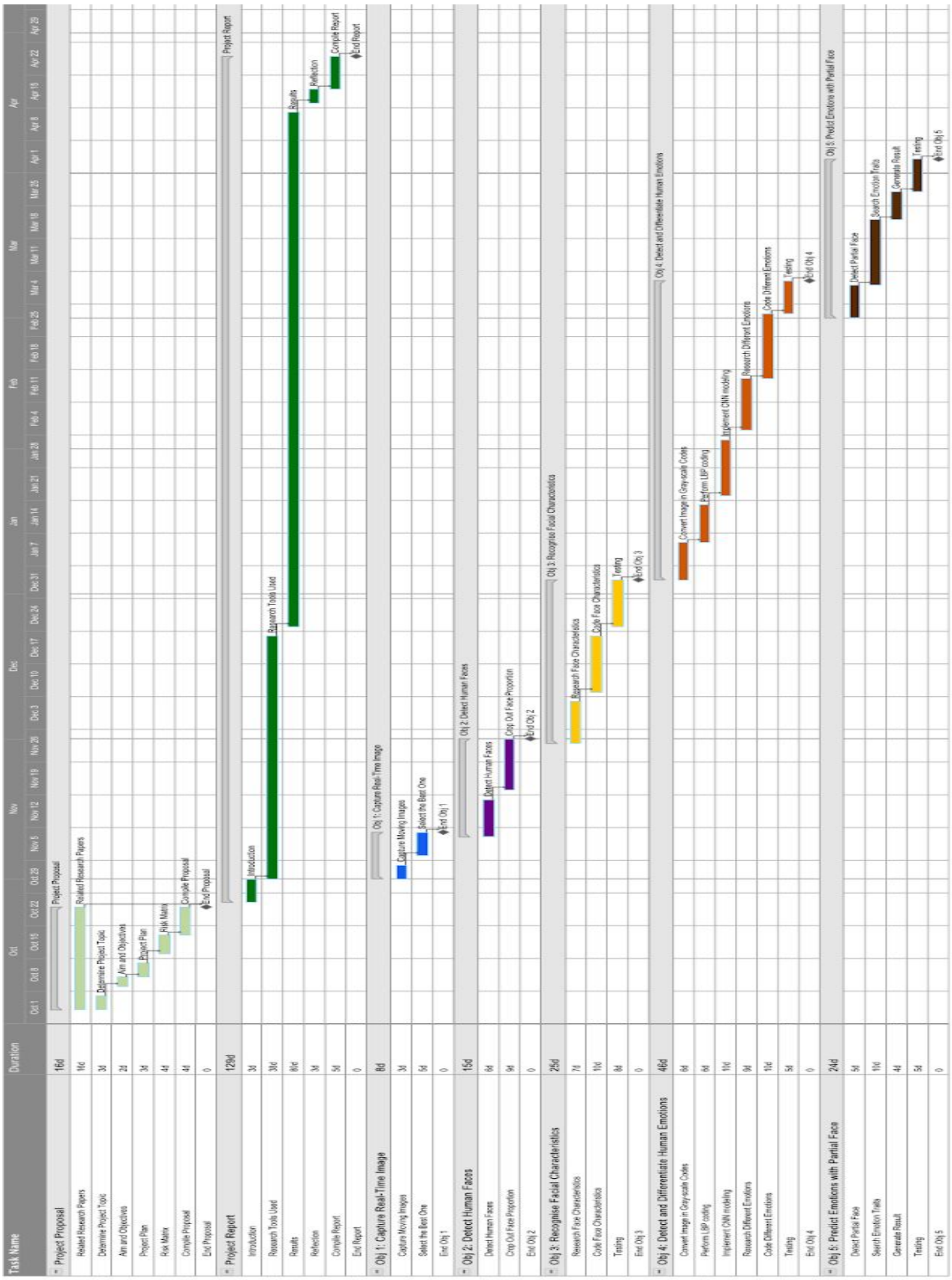
Lyons, M., Akemastu, S., Kamachi, M. and Gyoba, J. (1998) Coding Facial Expressions with Gabor Wavelets. In: *3rd IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan. Nara: IEEE, 200 - 205.

3. **Yale Face Database**

Belhumeur, P., Hespanha, J. and Kriegman, D. (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) 711-720. Available from https://ieeexplore.ieee.org/abstract/document/598228/ [accessed 26 April 2018].

Appendix

Appendix 1: Gantt Chart



Appendix 2: Risk Matrix

Risk ID	Risk	Likelihood of Occurrence	Impact on the project	Mitigation
1	Low Battery	**Possible**	**High Impact.** The result is unable to be generated.	Keep track of the battery usage level. Charge it when necessary.
2	Electricity Shock	**Low**	**Low Impact.** It might cause physical harms to the users who are in contact with it.	Keep away from damp areas.
3	Fallen of the Robot	**Low**	**Low Impact.** It might cause physical injuries to the people nearby.	Be careful while moving the robot.
4	Webcam of the Robot Not Functioning Properly	**Possible**	**High Impact.** No images could be input into the system to proceed.	Test and check the webcam if it is in good condition before testing the system. Use an alternative webcam to capture the images.
5	The lighting of the environment for image capturing (i.e. too dark or too bright)	**Very Likely**	**High Impact.** Neither too dark nor bright causes the system unable to detect human	Perform the Testing indoors with light bulbs to reduce the impact caused of the

			faces within the image captured.	sunlight.
6	The Face Movement is too fast to be captured and analysed at the same time	**Likely**	**High Impact.** The system might miss out the important key points that determine the person's emotions.	The system should try to capture as many frames per second as possible.
7	Error occurred within Convolutional Neural Network	**Rare**	**High Impact.** This may lead to system failed.	An alternative Neural Network modelling could be considered.
8	Robot being stolen or unavailable	**Unlikely**	**Low Impact**. This unable to prove that the system is compatible or functionable along with the robot.	The result can be obtained by using a laptop to carry out the process instead.

Appendix 3: Confidential Agreement

FOCUS GROUP: EMOTION RECOGNITION TESTING

This is a confidential agreement between the conductor (Melissa Ho) and the participants.

This experiment is conducted as the dissertation project for BSc Computer Science.

All participants of this experiment will be anonymous throughout the process, only the images of the participants will be kept as experimental results for academic purpose.

Throughout this experiment, all participants only required to express four different type of emotions (angry, happy, neutral, and sad) as instructed using the PowerPoint slides.

If you have any more questions, feel free to ask.

The experiment will only be conducted once the participant expresses his / her consent after reading this agreement letter.

MELISSA HO HUI QI – HOM 14581956

UNIVERSITY OF LINCOLN

20th APRIL 2018