

**Auto Raise Hand in Microsoft Teams (API/Extension)**

**BY**

**TEH BOON HIN**

**A REPORT**

**SUBMITTED TO**

**Universiti Tunku Abdul Rahman**

**in partial fulfillment of the requirements**

**for the degree of**

**BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION**

**SYSTEMS**

**Faculty of Information and Communication Technology**

**(Kampar Campus)**

**JAN 2023**

## REPORT STATUS DECLARATION FORM

**Title:** \_\_\_ Auto Raise Hand in Microsoft Teams (API/Extension) \_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Academic Session:** \_\_\_ Jan 2023 \_\_\_

I \_\_\_ TEH BOON HIN \_\_\_

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

\_\_\_ TBH \_\_\_  
(Author's signature)

\_\_\_  \_\_\_  
(Supervisor's signature)

**Address:**

\_\_\_ 18, Lebuhrapat Baru 13 B, \_\_\_

\_\_\_ Taman Lapangan Indah \_\_\_

\_\_\_ 31350, Ipoh Perak \_\_\_

\_\_\_ Dr. Aun Yichiet \_\_\_

Supervisor's name

**Date:** \_\_\_ 12/4/2023 \_\_\_

**Date:** \_\_\_ 28/4/2023 \_\_\_

<b>Universiti Tunku Abdul Rahman</b>			
Form Title : <b>Sample of Submission Sheet for FYP/Dissertation/Thesis</b>			
Form Number: <b>FM-IAD-004</b>	Rev No.: <b>0</b>	Effective Date: <b>21 JUNE 2011</b>	Page No.: <b>1 of 1</b>

**FACULTY OF** \_\_\_\_\_ **Information and Communication Technology** \_\_\_\_\_

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: \_\_\_\_\_12/4/2023\_\_\_\_\_

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that \_\_\_\_\_ ***Teh Boon Hin*** \_\_\_\_\_ (ID No: **19ACB03289** ) has completed this final year project entitled “\_\_Auto Raise Hand in Microsoft Teams (API/Extension)\_\_\_” under the supervision of Dr Aun Yichiet (Supervisor) from the Department of \_ Computer and Communication Technology\_\_\_\_, Faculty/Institute\* of Information and Communication Technology\_ .

I understand that University will upload softcopy of my final year project / dissertation/ thesis\* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

**TBH**  
\_\_\_\_\_  
(*Teh Boon Hin*)

\*Delete whichever not applicable

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**Auto Raise Hand in Microsoft Teams (API/Extension)**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_ **TBH** \_\_\_\_\_

Name : \_\_\_\_\_ Teh Boon Hin \_\_\_\_\_

Date : \_\_\_\_\_ 12/4/2023 \_\_\_\_\_

## **ACKNOWLEDGEMENTS**

I would like to express thanks and appreciation to my supervisor, Dr. Aun Yichiet and my moderator, Cik Zanariah Binti Zainudin who have given me a golden opportunity to involve in the Artificial Intelligence field study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

Other than that, I would like to thank my friend, Lim Jia Hui who has provided a lot of assistance in completing this project. She shared knowledges that I do not know and information on solving problems.

## **ABSTRACT**

This project is working on a growing trend – Artificial Intelligence. To be more specific, it is regarding facial expression recognition based on deep learning. Artificial Intelligence focuses on developing intelligences of machines, by developing algorithms, machines are able to learn from data and patterns, even perform tasks that require human intelligence, such as visual perception, speech recognition, and decision-making. Facial expression recognition is one of the popular topic in computer vision as its extensive applicability in various fields. In spite of that, people are becoming concerned about the issues brought about by deep learning as the trend expands at a staggering pace. For example, efficiency, accuracy, data confidentiality, user-friendliness and so on. This is a research based project for academic purposes which aims to investigate methods on training a CNN model with better performance that is applicable to project – auto raise hand API/extension in Microsoft Teams. Three systems related to the project are selected for reviewing their technology used, strengths, weaknesses, with some suggestions provided. The reviewed products are MorphCast, FaceReader, and Live 3D. After reviewing, the project scope is defined and there are three objectives to be archived in this project. The proposed solution is to develop a high performance CNN model. Also, the hardware and software requirements to conduct this project are listed and briefly introduced.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>REPORT STATUS DECLARATION FORM</b>	<b>ii</b>
<b>FYP THESIS SUBMISSION FORM</b>	<b>iii</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>TABLE OF CONTENTS</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement and Motivation	2
1.2 Objectives	2
1.3 Project Scope and Direction	3
1.4 Contributions	4
1.5 Report Organization	4
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
2.1 Previous Works on Facial Expression Recognition	
2.1.1 MorphCast	6
2.1.2 FaceReader	8
2.1.3 Live 3D	10
2.2 Summarization of Previous Works	12
2.3 Data Collection	13
2.4 Critical Remarks of Selected Dataset	15
<b>CHAPTER 3 SYSTEM MODEL</b>	<b>16</b>
3.1 Model Design Diagram	16
3.2 General Working Procedure	17

<b>CHAPTER 4 SYSTEM DESIGN</b>	<b>19</b>
4.1 Deep Learning	19
4.2 Transfer Learning	22
4.3 Locating Corrupted Image	23
4.4 Data Pre-processing – Image Classification	24
4.5 Data Pre-processing – Image Cropping	24
4.6 Under Sampling	26
<b>CHAPTER 5 EXPERIMENT/SIMULATION</b>	<b>27</b>
5.1 Hardware Setup	27
5.2 Software Setup	27
5.3 Setting and Configuration	27
5.4 Implementation Issues and Challenges	28
5.5 Concluding Remark	29
<b>CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION</b>	<b>30</b>
6.1 System Testing and Performance Metrics	30
6.2 Testing Setup and Result	32
6.3 Project Challenges	36
6.4 Objectives Evaluation	36
6.5 Concluding Remark	37
<b>CHAPTER 7 CONCLUSION AND RECOMMENDATION</b>	<b>38</b>
7.1 Conclusion	38
7.2 Recommendation	38
<b>REFERENCES</b>	<b>40</b>
<b>APPENDIX</b>	<b>42</b>
Data Pre-processing – Image Sorting	42
Identification of Corrupted Image – Train Set	43
Data Pre-processing – Image Cropping	44



Under Sampling	46
<b>WEEKLY LOG</b>	<b>47</b>
<b>POSTER</b>	<b>49</b>
<b>PLAGIARISM CHECK RESULT</b>	<b>50</b>
<b>FYP2 CHECKLIST</b>	<b>52</b>

# LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 1.4.1	Flow Chart Illustration of Board Model Training Process	2
Figure 2.1.1.1	Flow Chart Illustration of Adaptive-Media	6
Figure 2.1.1.2	Illustration of Adaptive-Media Use Case	7
Figure 2.1.2.1	Work Flow Chart of FaceReader	8
Figure 2.1.2.2	UI of FaceReader	9
Figure 2.1.3.1	Work Flow Chart of Live 3D	10
Figure 2.1.3.2	Example of Model Clipping	11
Figure 3.1.1	Code Snippet of CNN Layers	16
Figure 3.1.2	Visualization of CNN Layers	16
Figure 3.2.1	General Working Procedure	17
Figure 4.1.1	Artificial Neuron In Deep Learning	19
Figure 4.1.2	Equation of Summation Function in Neuron	19
Figure 4.1.3	Threshold Function Equation	20
Figure 4.1.4	Sigmoid Function Equation	20
Figure 4.1.5	Rectifier Linear Function Equation	21
Figure 4.1.6	Visualization of Neuron Network	21
Figure 4.2.1	Traditional Machine Learning VS Transfer Learning	22
Figure 4.3.1	Flow of Identifying Corrupted Image	23
Figure 4.4.1	Flow of Image Classification	24
Figure 4.4.2	Sorted Dataset	24
Figure 4.5.1	Flow of Image Cropping	25
Figure 4.5.2	Before Cropping (Left) VS After Cropping (Right)	25
Figure 4.6.1	Flow of Under Sampling	26
Figure 5.4.1	Raw AffectNet Dataset	28
Figure 5.4.2	Error of Reporting Corrupted Image	28
Figure 5.4.3	Information Available in the CSV Provided	29
Figure 6.1.1	Confusion Matrix	31

Figure 6.1.2	Equations of Precision Score, Recall Score (Left) and F1 Score (Right)	31
Figure 6.2.1	Results of Each Model & Its Configurations	32
Figure 6.2.2	Results of binary(2) CNN	32
Figure 6.2.3	Results of binary CNN	32
Figure 6.2.4	Results of 5 class CNN	33
Figure 6.2.5	Results of vgg16 CNN	33
Figure 6.2.6	Results of vgg19 CNN	33
Figure 6.2.7	Results of vgg16(2) CNN	34
Figure 6.2.8	Results of mobilenet CNN	34
Figure 6.2.9	Results of 3 indicators CNN	35
Figure 6.2.10	Results of resnet50 CNN	35
Figure 6.2.11	Results of fyp2 CNN	35
Figure 6.4.1	Objectives Evaluation Result of binary(2) CNN	36
Figure 6.4.2	Objectives Evaluation Result of binary CNN	36
Figure 7.2.1	Example of Face Landmarks	39

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 2.2.1	Comparison of Previous Works Related to Proposed Project	12
Table 2.3.1	Comparison of Available Datasets in Data Collection	14
Table 3.1.3.1	Specifications of Laptop	27

## LIST OF ABBREVIATIONS

<i>OLT</i>	Online teaching and learning
<i>CNN</i>	Convolutional Neural Networks
<i>CV</i>	Computer Vision
<i>SDK</i>	Software Development Kit
<i>API</i>	Application Programming Interface
<i>FER</i>	Face Expression Recognition
<i>GUI</i>	Graphical User Interface
<i>GB</i>	Gigabytes
<i>CSV</i>	Comma Separated Values (file)
<i>GPU</i>	Graphical Processing Unit
<i>cuDNN</i>	CUDA Deep Neuron Network
<i>ReLU</i>	Rectifier Linear Unit

# Chapter 1: Introduction

Online teaching and learning (OTL), also called e-learning is the education that takes place over the internet. Teachers and lecturers conduct the classes through the online platforms and students attend their courses and learn relatively.

There are some differences between OTL and traditional class. As we all know, traditional class is conducted on the school campus and the interested parties have to go back physically to carry out the actions of a traditional class. Thus, we can conclude that the main difference between them is the place to conduct.

Due to the Covid-19 pandemic, the lockdowns implemented to control the pandemic have affected the global economy sorely. This incident has also agitated several industry sectors such as agriculture, manufacturing, retail trading, and so on. As one of the biggest victims of this tragedy, the education sector has made a lot of adjustments to past traditions to maintain its operation. One extensive change is that in-person teaching no longer has the permission to be conducted in various schools, colleges, and universities as most of the campuses are closed temporarily [2]. Thus, students are forced to continue their studies at home and some even choose to drop out of school as not able to adapt to a new study environment. According to the estimation of researchers, the chance of getting back to school for the physical class is low and uncertain. As a substitute for physical classes, people started to switch to online classes. The research observation from Lapitan et al. [4] indicates a distinctive shift to OTL as a contribution to further control the spread of the virus. Academic institutions have progressively implemented online education, and the number of new intake students enrolling in is skyrocketing while the existing students are changing their academic mode to OTL as well [3].

So which is better? OTL or physical class? Both of them have their advantages and it depends on different people and situations to justify which is better. For example, if there are no lockdowns and moment control orders, most of the students are tend to go back to campus. This is because they are more accustomed to the physical academic mode for several reasons. But to put in this case, students have no choice but to conduct OTL since it is more convenient and suitable in this current situation.

## 1.1 Problem Statement and Motivation

Considering that it might be the first time to conduct OTL, most of the people including students and teachers do not have the experience of tackling problems that occur during OTL. For example, weak internet connection and lagging affect the student's class experience, and microphone or screen sharing problem that slows down the course progress. Besides these technical issues, another significant problem is the adaptation of students to OTL. A common manifestation is that students do not dare to ask even if they have questions. This is mainly due to they are still not familiar with and adapting to this new study environment. Based on the journal of Tani [13], this phenomenon is especially common in Asia as Asian students are shy and contribute a low level of in-class participation.

The aim of the project is to introduce a solution to the problem statement stated above. There are always some inactive students who are not responding although the lecturers attempt to engage with them. In this proposal, an extension of Microsoft Teams is proposed as a solution to the student side. Since they are not willing to ask question by themselves, this extension is made to assist them when facing problems.

## 1.2 Objectives

The core value of this system is the accuracy of facial expression detection. Thus, the project aims to train a CNN model that hold an accuracy of 0.8 and above and maintain the loss bellow 0.5. The board life cycle of model training is shown as below:

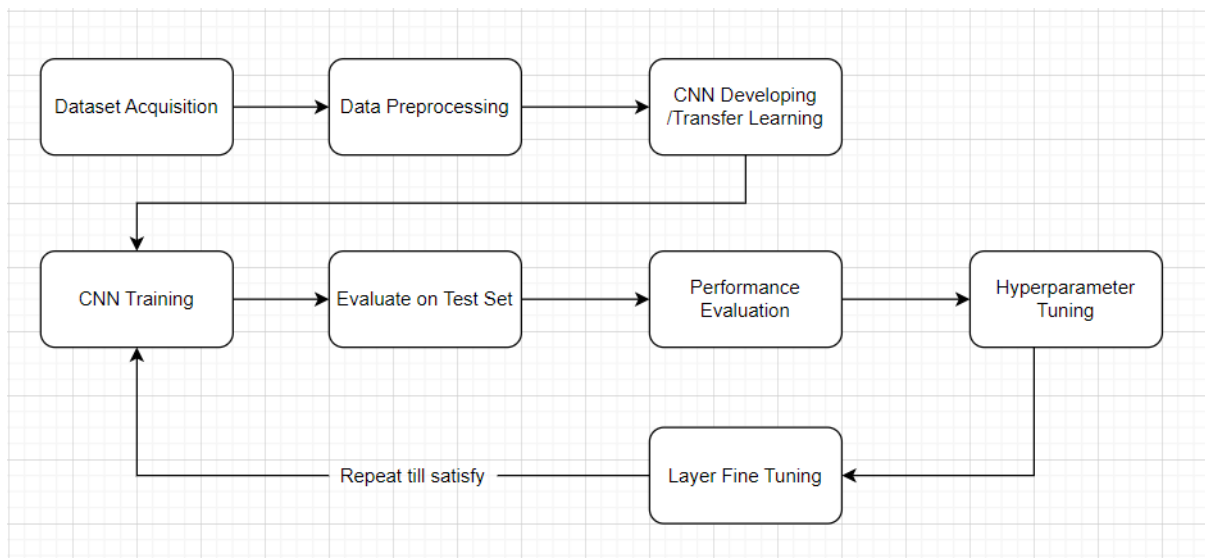


Figure 1.4.1: Flow Chart Illustration of Board Model Training Process

The dataset is initially downloaded from OneDrive provided by AffectNet [6]. After that, some pre-processing procedures are applied to categorize and normalize the data. The model is built with CNN and the model will be trained from shallow to deep. In this case, the first train is implemented on model with some simpler parameters and layer building and model training. After evaluating it, those set up will be improved and this is a recurring process till satisfied result is get from the model. Also, transfer learning technique is practiced in this project by training the CNN with various source domains.

Generally there are 3 milestones in this project:

**I. To perform data pre-processing on the selected dataset.**

Steps of data pre-processing are performed. Firstly, data cleaning is done to clean the data. For example, all images are cropped based on the height and weight provided to reduce the noises of data. Besides that, due to the size of dataset is too large to be handled, data reduction is performed and under sampling technique is applied to balance the dataset.

**II. To develop a facial expression recognition system using CNN framework.**

Several models are built based on CNN framework with different layers and parameters. Then deep learning is implemented on this model with dataset. Batch training is conducted to speed up the process of deep learning and save time and resources. Parameter tuning is performed to identify the best parameters that can maximize the performance of CNN model.

**III. To perform transfer learning using different source domains.**

Transfer learning can greatly shorten the time needed by reusing a pre-trained model as the initial point of training a new CNN model. Some common examples of pre-trained models are VGG, ResNet, and so on. It produces a notable better performance as it skipped the process of training from scratch with lesser data required.

### **1.3 Project Scope and Direction**

In this project, a CNN with high performance is developed. In order to better detect and classify different facial expression of students during classes, this model is trained based on the AffectNet dataset, the largest facial expression dataset to ensure the accuracy of this system. This CNN will be applied on a system that will detect the facial expression of students during class. When the student shows some facial expressions that indicate his incomprehension, this system will automatically raise hand in the Microsoft Teams class to notify the lecturer. Thus,



achieving high accuracy and low loss on this CNN is important to guarantee the performance of the system above.

#### **1.4 Contributions**

This project is aimed to develop a CNN model that can detect when the student “looks” confused in the Microsoft Teams online class. Firstly, some research about the relationship between sentiment and facial expression of a person is studied and investigated. I concluded that micro-expressions are unintentional brief facial expressions that can reflect the emotion of the person in current situation [5]. For example, if a student is nervous, he will frown and pout his lips to show his nervousness. This helps to analyze the input (user face expression) to evaluate the current mental state then decide whether to raise hand or not. Computer vision (CV) now is able to recognize the face of user and AI technology can decode the face gestures. TensorFlow and Keras implemented in Python is providing facial expression recognition libraries with a deep neural network. The next step is to build a model for facial expression recognition using CNN framework. In order to improve the accuracy of the system, a more integrated dataset will be used on training the model. Here I will be using AffectNet Dataset as its pictures are more in line with real-life scenarios and its dataset size larger compared to the others. After that, I can perform hyperparameter tuning to enhance the performance of CNN model.

#### **1.5 Report Organization**

This report is organized into 7 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Model, Chapter 4 System Design, Chapter 5 Experiment/Simulation, Chapter 6 System Evaluation and Discussion, Chapter 7 Conclusion and Recommendation. The first chapter is the introduction of this project which includes problem statement, project background and motivation, project scope, project objectives, project contribution, and report organization. The second chapter is the literature review carried out on several existing facial expression recognition products available in the public to evaluate the strengths and weaknesses of each product. The third chapter is discussing the overall system model of this project. The fourth chapter is regarding the details on how to implement the design of the system. Furthermore, the fifth chapter reports the hardware and software requirements, procedures used for data collection and data pre-processing, implementation

issues and challenges, and concluding remark. Chapter 6 presents the results of the research and evaluates the performance and limitations of each model based on the project objectives. It also suggests a preferred model which is performing the best. Lastly, chapter 7 summarizes the research findings, restates the research question and objectives, and offers recommendations for future research.

# Chapter 2: Literature Review

## 2.1 Previous Works on Facial Expression Recognition

### 2.1.1 MorphCast

MorphCast is the world-leading company in adaptive-media technology. Technologies such as interactive video, AI, machine learning, and facial expression recognition with emotion, gender, and age analysis can be found in this MorphCast. Adaptive-media is defined as the multimedia that adapts automatically to viewers in real-life by detecting the viewer's face and reactions. Let's further illustrate the concept of adaptive-media.

Try to imagine, Ali is searching on the Internet looking for some movies to watch. Firstly Ali will watch the movie trailers to decide whether that movie is his type. Just try to imagine if adaptive-media is implemented in this situation where Ali is watching a movie. If Ali frown, the system will rewind the trailer as Ali may not understand the plot; the system will skip playing next trailer if Ali feels bored. If Ali looks excited and interested in the movie, the system will suggest and push more similar movie trailers.

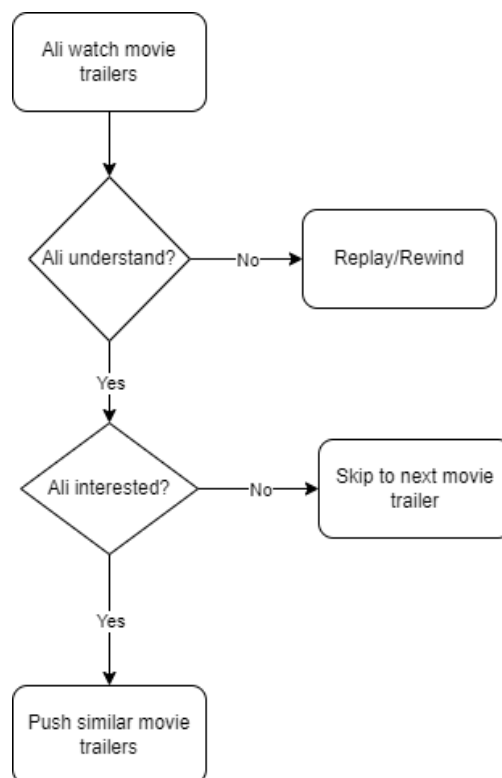


Figure 2.1.1.1: Flow Chart Illustration of Adaptive-Media

Another use case is Emotion AI Interactive Video. An example video can be found in this link: <https://info.verizonenterprise.com/inclusive-stories>. It is a firm that uses Emotion AI to create empathy and change the narrative according to the viewer's emotions. This helps to keep the viewer stick to the plot and increase the interest in it by improving the audience participation.

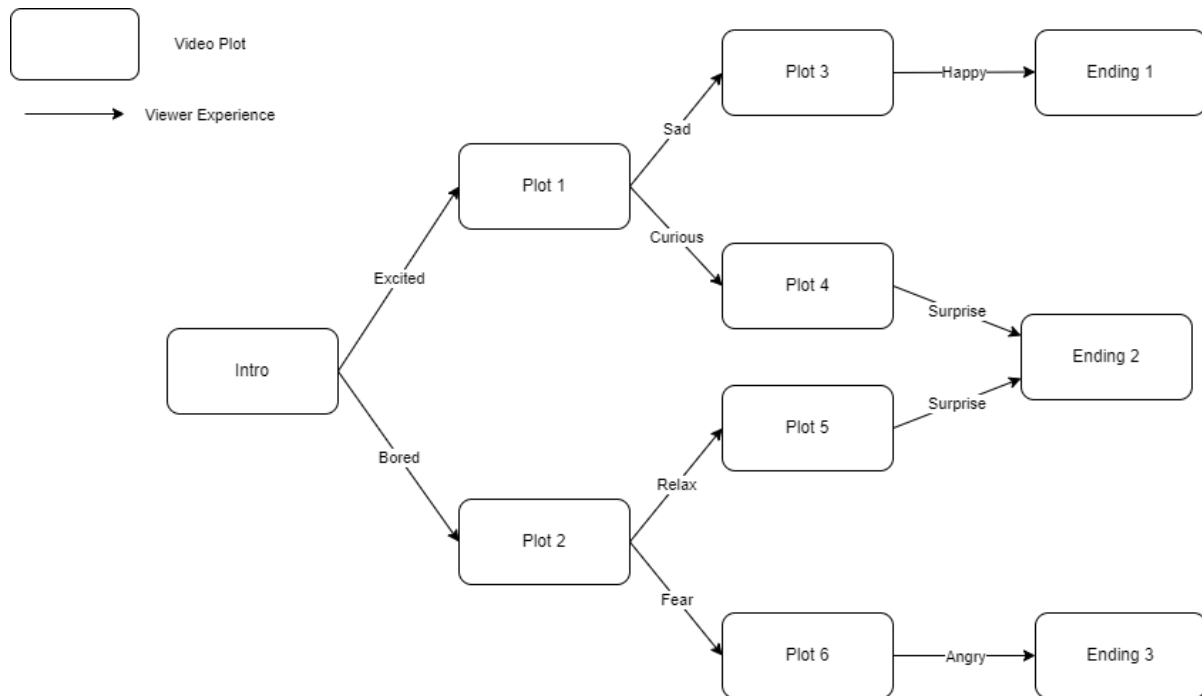


Figure 2.1.1.2: Illustration of Adaptive-Media Use Case

As shown in Figure 2.1.1.2, the plot of video is changing according to the viewer's experience. Not only on movie or films, adaptive-media can also be applied to e-branding. It helps companies efficiently captivate online customers' attention and enhance customer engagement by providing very remarkable interactive emotional experiences.

### **Strength**

This innovative technology helps to **enhance the user experience**. Users can find suitable interests and control the pace of receiving new information more effectively. As a result, the content creator can **maximize the efficiency of the video for marketing purposes**.

### **Weakness/Limitation**

Some further improvements are still needed in this technology especially on the **accuracy of facial expression recognition**. There are several factors that can affect the accurateness of such as the pixel of camera and capturing situation of the user.

## Suggestion/Improvement

MorphCast can **build a database to store user data** of using the system. So these data can be used for further training the model of facial expression recognition. The data should be deleted after use to ensure user privacy.

### 2.1.2 FaceReader

FaceReader claims itself to be the most powerful automated system that helps users to gain detailed and reliable data of facial expressions. It is an analysis system that is able to provide clear insight and the effects of different emotions. This system is well-integrated with eye-tracking data and physiology data which make its ability of precise analysis.

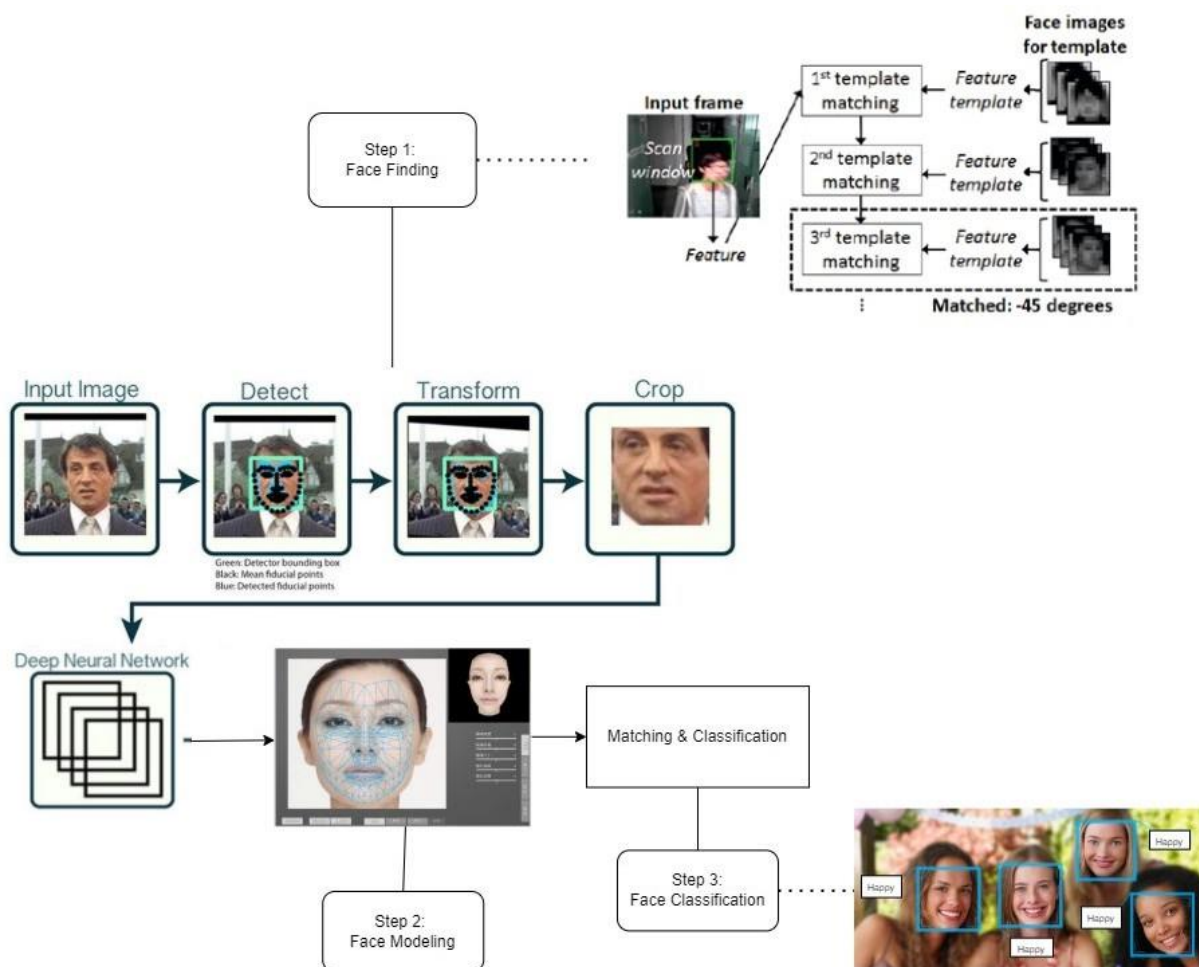


Figure 2.1.2.1: Work Flow Chart of FaceReader

The figure above illustrates the processes of how FaceReader function. There are mainly 3 steps, which are **face finding, face modelling, and face classification**. Firstly, an input (static image, live video, etc.) is inserted into the system. Then FaceReader system will utilize the algorithm set to locate the face of user. The top right image is explaining the face

finding algorithm. For example, if the input is a video, one frame of it will be captured and sent as an input frame. Then this input frame will be processed with AI to match with the templates of the face image to locate and identify the face. After that, unnecessary parts of the input frame are cropped and sent to process with deep neural network. In this step, the AI will generate an artificial face model which consists of up to 500 key points using 3D modelling. The process of face classification then will be carried out based on this generated face model to perform matching and classification. Lastly, the results such as happy or angry will be responded to the UI of user-side. Also, the user can make some analysis or report based on the result.

## **Strength**



*Figure 2.1.2.2: UI of FaceReader*

FaceReader can provide **detailed analysis** on the input frame. The system is able to indicate different expression intensity of the user by checking each micro-expression part of him such as upper lip raiser, cheek raiser, lid tightener and so on. By providing these objective assessments, FaceReader is able to **benefit the user's work by saving valuable efforts and resources**.

## **Weakness/Limitation**

The **extensibility of FaceReader is restricted** as it can only be used on analysis and reporting. The outside developers are not allowed to integrate FaceReader into their development projects.

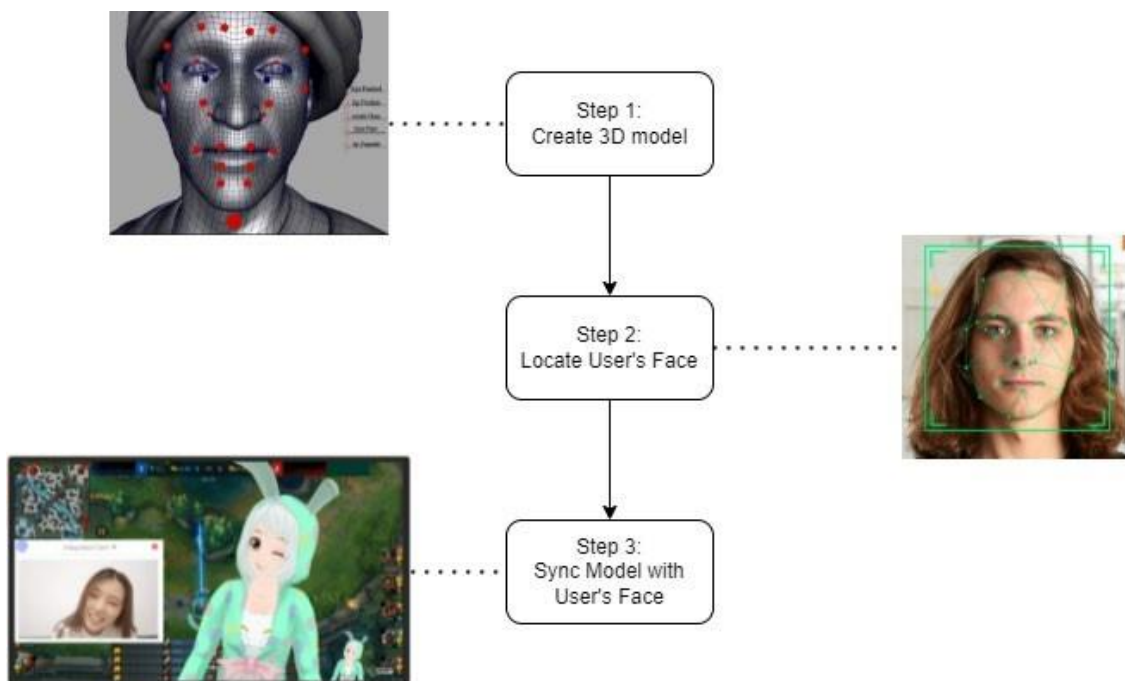
## **Suggestion/Improvement**

FaceReader can **publish free SDK or open API** to grant the user access to develop extensions based on the system. So the FaceReader is more flexible in different areas.

### 2.1.3 Live 3D

Live3D is a popular Vtuber Software Suite that provides various aids to serve its users. With serving almost 1 million Virtual YouTubers in the worldwide, it is dedicated to building amazing platform for Vtubers. Live3D accommodated abundant vtubing resources such as avatars, 3D models, and background scenes which user is allowed to utilize them to create amazing shows.

The main function of Live3D is virtual live broadcasting. Facial capture technology is the fundamental of this function while it connects the 3D model (virtual) and the user's face (real) by relaying the user's expression to the 3D model.



*Figure 2.1.3.1: Work Flow Chart of Live 3D*

In this example, a 3D model with only a face is created and rigged. Then the user turns on the camera to allow Live 3D to detect his face and locate his five sense organs by plotting them out. After that, the system will dynamically sync the model with the user's face during broadcasting. For example, when the user winks his left eye, the system can detect the changes of the markers plotted in step 2 then update them to the virtual face model. The system will then adjust the bones rigged of the face model according to the previous step.

#### **Strength**

Live 3D provides **real-time capture and flexible control** functions. User is able to easily record and stream 3D videos in live show. By detecting the face, user can enable pendant mode (greenscreen mode) to change his real-life background to movable virtual wallpaper.

### Weakness/Limitation



*Figure 2.1.3.2: Example of Model Clipping*

Some **clipping issues** might happen sometimes due to the incompleteness of modelling and rigging. Model clipping refers to the whole or part of the model passes through another model unrealistically. For example in *Figure 2.1.3.2*, the hand of black shirt basketball player “passed through” another player’s head, which is impossible to happen in real life.

### Suggestion/Improvement

This problem can be solved by **refining the process of modelling and rigging**. For example, this process can be classified into character modelling and fabric modelling. By conducting them separately, the clipping issues such as character model “passes through” the garment can be prevented effectively.



## 2.2 Summarization of Reviewing Previous Works

Previous Works	Technology/ Method	Strength	Weakness	Suggestion
MorphCast	Adaptive-Media	<ul style="list-style-type: none"> <li>- Enhance user experience</li> <li>- Maximize the efficiency in marketing purpose</li> </ul>	<ul style="list-style-type: none"> <li>- Unstable accuracy of facial expression recognition</li> </ul>	<ul style="list-style-type: none"> <li>- Build a database to store one-time user data for further training machine</li> </ul>
FaceReader	Face Finding	<ul style="list-style-type: none"> <li>- Detailed analysis</li> <li>- Report generating</li> </ul>	<ul style="list-style-type: none"> <li>- Restricted extensibility of system</li> </ul>	<ul style="list-style-type: none"> <li>- Publish free SDK</li> <li>- Open API</li> </ul>
Live 3D	Virtual live broadcasting	<ul style="list-style-type: none"> <li>- Real-time capture</li> <li>- Flexible control</li> </ul>	<ul style="list-style-type: none"> <li>- Model clipping</li> </ul>	<ul style="list-style-type: none"> <li>- Refine modelling and rigging process</li> </ul>

*Table 2.2.1: Comparison of Previous Works Related to Proposed Project*

### 2.3 Data Collection

Based on the criteria that have been listed out above, a comparison among 8 available dataset is conducted and the most suitable dataset, which has the most availability of useful characteristics will be selected. The 8 dataset are: **AffectNet, Ascertain, Dreamer, Extended Cohn-Kanade Dataset (CK+), EMOTIC, FER-2013, Google Facial Expression Comparison Dataset, K-EmoCon.**

		<b>Continuous Dimensional Model</b>	<b>Accessibility of Data</b>	<b>Features / Advantages</b>
1.	<b>AffectNet</b>	yes	<b>Images/Video:</b> More than 1.5 M <b>Emotion related keywords:</b> 1250	About half of the data are annotated manually based on <b>seven discrete facial expressions</b> (categorical model) and the <b>intensity of valence and arousal</b> (dimensional model)
2.	<b>Ascertain</b>	Yes	<b>Images/Video: ?</b> <b>Emotion related keywords: ?</b>	Relationship of <b>affective ratings and personality scales</b> of participants are examined to ensure the accuracy of data annotation. Data are <b>evaluated from the modalities physiological and facial activity data</b> of participants which are synchronously recorded during data collection.
3.	<b>Dreamer</b>	Yes	<b>Images/Video: ?</b> <b>Emotion related keywords: ?</b>	The participants have <b>conducted self-assessment</b> on their emotion state.
4.	<b>CK+</b>	No	<b>Images/Video:</b> 5,976 <b>Emotion related keywords:</b> 123	Data are <b>categorized</b> into different classes according to the <b>7 universal expressions</b>
5.	<b>EMOTIC</b>	Yes	<b>Images/Video:</b> 23,571 <b>Emotion related keywords:</b> 26	Data are evaluated through <b>the level of 3 continuous dimensions</b> , which are <b>Valence</b> (negative / positive), <b>Arousal</b> (calm / active) and <b>Dominance</b> (dominated / in control)
6.	<b>FER-2013</b>	Yes	<b>Images/Video:</b> 35,000 <b>Emotion related keywords:</b> 7	Dataset consists of 3 separate sets: <b>training set, development set and test set.</b>
7.	<b>Google Facial Expression Comparison Dataset</b>	Yes	<b>Images/Video:</b> 500,000 triplets (156,000 images) <b>Emotion related keywords: ?</b>	<b>Triplet loss:</b> each data (triplet) is composed of <b>Anchor, Positive and Negative</b> . This helps to identify the most similar facial expression of 2 images by comparing the anchor to the positive and negative ( <b>reduce the deviation between anchor and positive &amp; increase the deviation between anchor and negative</b> ).

8.	K-EmoCon	Yes	<b>Images/Video: ? Emotion related keywords: 7</b>	The data are annotated in 3 aspects: <b>self, debate partner</b> (other participants) and <b>external observers</b> .
----	----------	-----	--	---

*Table 2.3.1: Comparison of Available Datasets in Data Collection*

**Explanation**

**Continuous Dimensional Model:** Determine whether the data in the dataset is continuous, which means by the existence of valence and arousal or intensity dimensions in the dataset.

**Accessibility of Data:** The richness and completeness of data will directly affect the efficiency of machine learning and accuracy of facial emotions detection.

**Features / Advantages:** Some features of the dataset that will benefit the process of machine learning in this project.

## 2.4 Critical Remarks of Selected Dataset

Based on the criteria that have been listed out above, a comparison among 8 available dataset is conducted and the most suitable dataset, which has the most availability of useful characteristics will be selected. The 8 dataset are: **AffectNet, Ascertain, Dreamer, Extended Cohn-Kanade Dataset (CK+), EMOTIC, FER-2013, Google Facial Expression Comparison Dataset, K-EmoCon.**

**AffectNet** is chosen as the dataset for conducting the machine learning due to several reasons.

1. Some **useful discrete categories** of face are annotated in AffectNet. For example:
  - a. **Anger**: distressed from incomprehensiveness
  - b. **Disgust**: dislike this subject
  - c. **Fear**: feel of the anxiety of failing the subject
  - d. **Sadness**: feeling of self-denial
2. **Data availability** of AffectNet is large enough and well-suitable for this project; dataset and annotations of 2 type of dataset, manually annotated dataset and automatically annotated dataset.
3. AffectNet provides:
  - Face images
  - Coordination (x and y) of the faces in the images
  - Location of the 68 facial landmarks
  - Eleven emotion and non-emotion categorical labels (0: Neutral, 1: Happiness, 2: Sadness, 3: Surprise, 4: Fear, 5: Disgust, 6: Anger, 7: Contempt, 8: None, 9: Uncertain, 10: No-Face)
  - Valence and arousal values of the facial expressions in continuous domain

[7]

# Chapter 3: System Model

## 3.1 Model Design Diagram

```
# Initialising the CNN
cnn = tf.keras.models.Sequential()

# Step 1 - Convolution
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=[64, 64, 3]))

# Step 2 - Pooling
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

# Adding a second convolutional layer
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

# Step 3 - Flattening
cnn.add(tf.keras.layers.Flatten())

# Step 4 - Full Connection
cnn.add(tf.keras.layers.Dense(units=64, activation='relu'))

# Step 5 - Output Layer
cnn.add(tf.keras.layers.Dense(units=11, activation='softmax'))

# Part 3 - Training the CNN

# Compiling the CNN
cnn.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

Figure 3.1.1: Code Snippet of CNN Layers

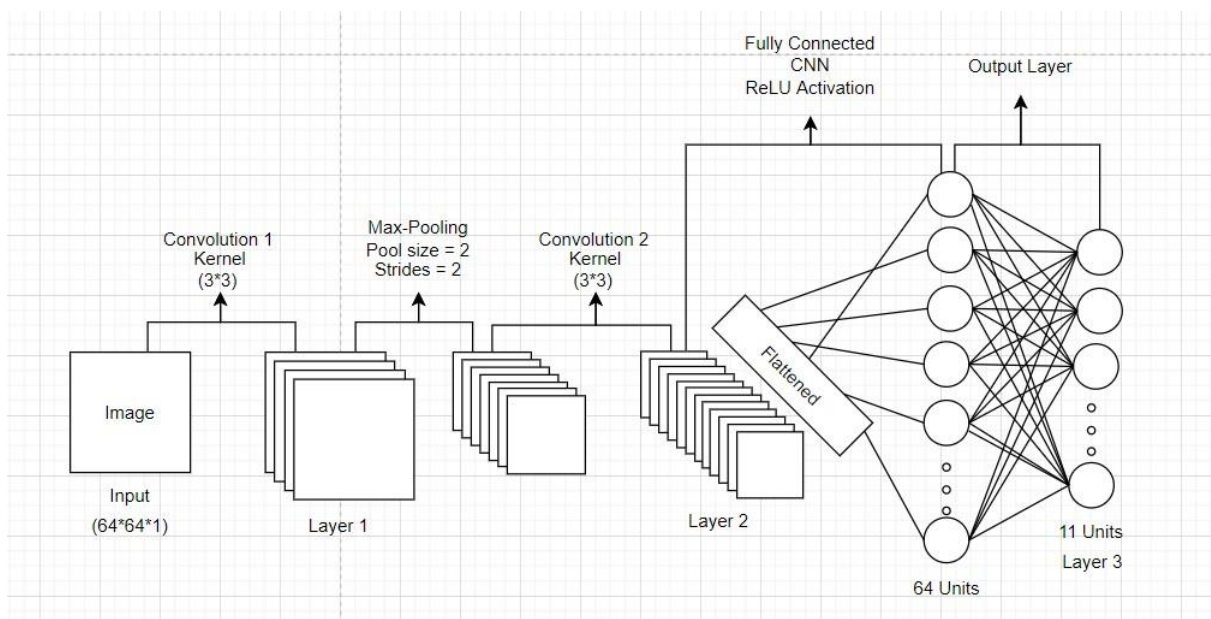


Figure 3.1.2: Visualization of CNN Layers

The two picture above shows the code snippet and visualization of a simple CNN model. This helps the user to better understand the architecture of a CNN. Three layers were built in this CNN model, which are two convolution layers and one output layer. It reads a total of 420,297 images that were classified into 11 classes as input and their size are set to 64\*64 pixel. Also, some other image pre-processing are applied on both train set and test set, such as shear\_range and zoom\_range to better formalize the dataset.

On the first convolution layer, a kernel of 3x3 is set to filter out the key features of the image. Its purpose is to condense the images into a format that is simpler to process without sacrificing elements that are essential for obtaining an accurate prediction. After that, those input is passed to a pooling layer, which has a similar role with convolution layer. By processing inputs through max pooling layer, the maximum value (the most outstanding feature) within every 2x2 of them is returned to create a new input with noises reduced. By doing this, the input has lesser noises which will affect the accuracy of prediction. Then the input is passed to another second convolution layer, repeating same process as the first convolution layer. Next, all input are flattened to convert them from multidimensions to 1 dimension before reading them into the next layer. Lastly, after connecting all of the layers previously created, all of them are linked to an output layer that consist of 11 classes, which is going to display the predicted output.

### 3.2 General Working Procedure

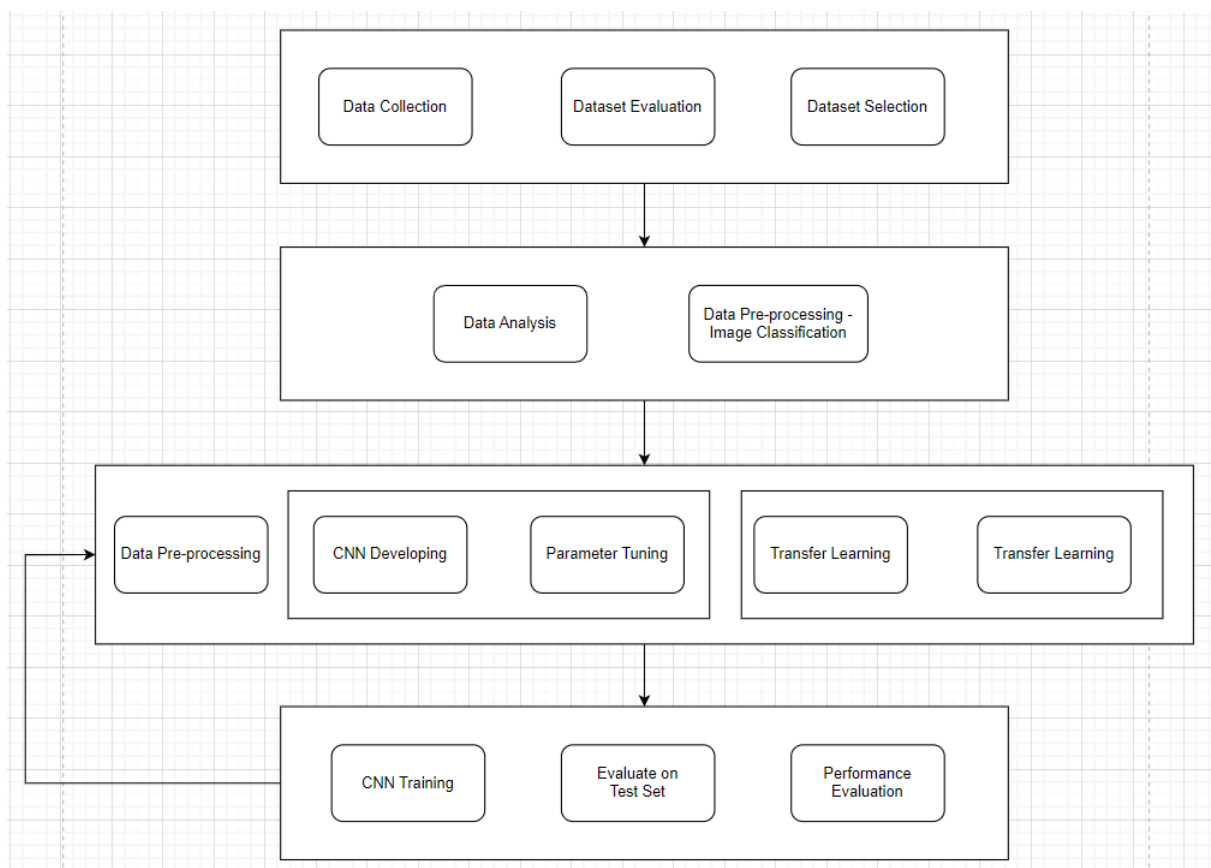


Figure 3.2.1: General Working Procedure

First of all, data collection is performed to gather suitable data for model training. Here secondary data are selected for convenience purpose and several datasets are shortlisted for evaluation. AffectNet dataset, which is the most suitable dataset is selected after evaluation.

After that, the data is studied and analyzed to have an overview and comprehend to the dataset. The dataset is then classified based on the annotation provided. After that, a CNN with several layers is built for training. Each layer has its purposes and works on different processes. This model is then trained for a certain number of epochs. Meanwhile, another CNNs with different setting are created while waiting for the first CNN to complete training. By evaluating on the test set, the performance of the model are able to be viewed by evaluating its accuracy and loss on both training set and test set. This training process is re-run till the desired performance is achieved.

# Chapter 4: System Design

## 4.1 Deep Learning

Deep learning is an artificial neural network-based branch of Machine Learning Science. It has various variations, including Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), which may be utilized in a variety of fields such as Computer Vision, Natural Language Processing, and so on. Some concepts and fundamental knowledge of deep learning is discussed in this topic.

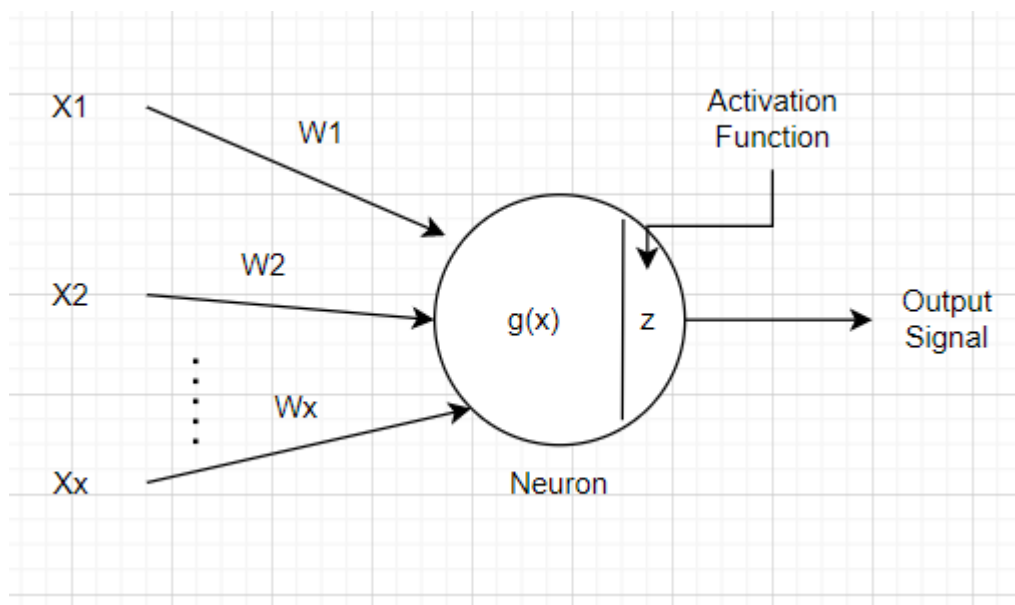


Figure 4.1.1: Artificial Neuron In Deep Learning

An artificial neuron (which also known as node) receives one or multiple input signals (X), either from its precedent neuron or the original feature. W refer to weights of each input, it indicates the importance of its input in the process of making prediction of final output.

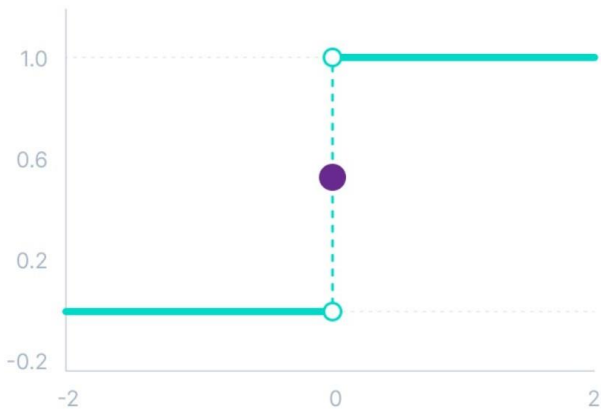
Artificial neuron is a mathematical function that sum up all the input values with their corresponding weights together, and add bias (b) to it. Later, the neuron produces an output signal to its consequence neuron.

$$\sum_{i=1}^{i=x} X_i W_i + b$$
$$= (X_1 W_1) + (X_2 W_2) + \dots + (X_i W_i) + b$$

Figure 4.1.2: Equation of Summation Function in Neuron



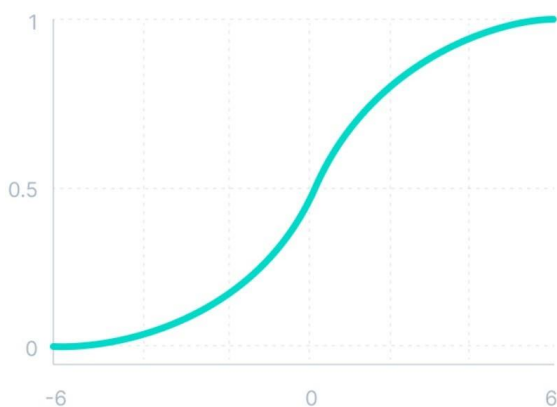
These signals are eventually passed to a non-linear function called activation function ( $z$ ), which act as a decision maker to decide the importance of each neuron's input in making prediction. The goal of adding an activation function is to increase the non-linearity of the neural network. There are many type of activation functions, 3 common activation functions are explained in simple here.



$$\phi(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Figure 4.1.3: Threshold Function Equation

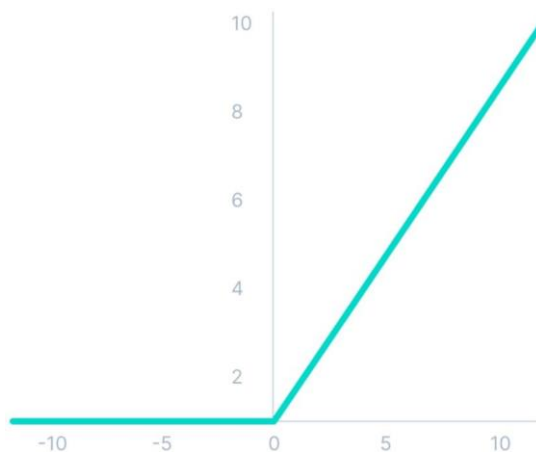
Threshold function, also known as Binary Step Function is a activation function only available for **binary classification**. It determine either yes or no. Thus, this function is not applicable on multi-class classification situations.



$$f(x) = \frac{1}{1 + e^{-x}}$$

Figure 4.1.4: Sigmoid Function Equation

This function is commonly used in logistic regression. It is smoother compared to threshold function, therefore is frequently implemented in models where the **probability is required to be predicted as the output**. This is because this function is more in line with the reality situation, where the probability of anything occurs only between 0 and 1.



$$f(x) = \max(0, x)$$

Figure 4.1.5: Rectifier Linear Function Equation

Rectifier Linear or ReLU is the most popular activation function for artificial neuron networks. When  $x$  is less than zero,  $f(x)$  equals zero, and when  $x$  is more than or equal to zero,  $f(x)$  equals  $x$ . ReLU is typically applied as activation function when one have no idea which to use.

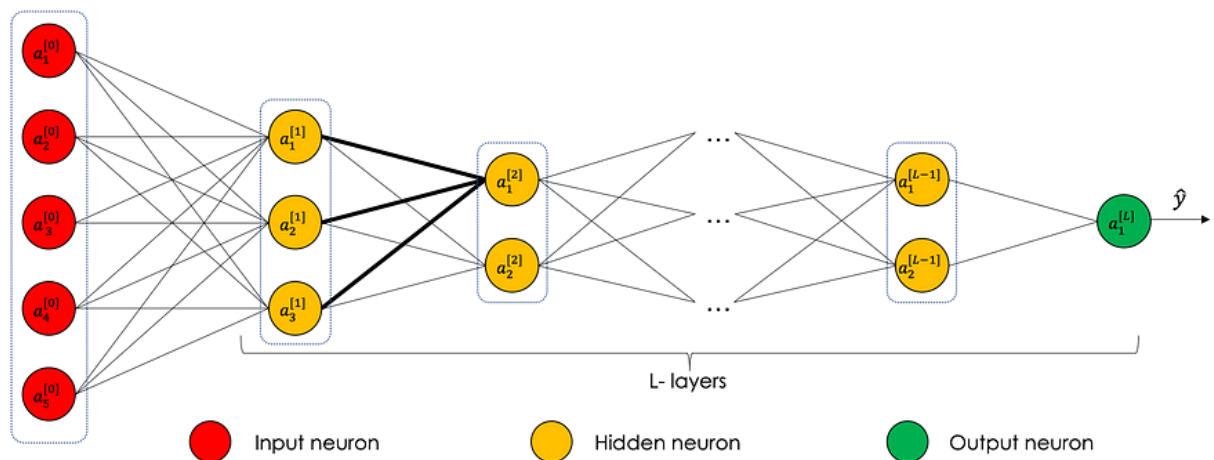


Figure 4.1.6: Visualization of Neuron Network

Picture above is the visualization of a neuron network. All of the neurons are connected to their corresponding one from layer to layer. The input layer represent the layer where all neurons receive input signal from original features. The subsequence layers are hidden layers which consists of different number of neurons, working for different purpose. Eventually, all of the value a summed and parsed to the output layer where the prediction is made.

## 4.2 Transfer Learning

Transfer learning is a machine learning and deep learning approach in which a pre-trained model is utilized as a starting point for a new task. As an optimization that enables quick progress when modelling the second work, a model that has been trained on the first job is reused on the second, related task. A pre-trained model is often utilized as an initial foundation in transfer learning, and it is then fine-tuned to make the new model more compatible with its required task. When fine-tuning, some layers of the pre-trained model are retrained on the new task while the general layers are frozen. The general layers are frozen so that they will not be retrained, because as mentioned earlier, pre-trained model for similar task is chosen, so its general layers are applicable to the new different but similar task. Let's see a scenario of transfer learning in term of human being. You have learnt driving a car before. When you are learning to drive a boat, some common concepts of driving car can be applied in this situation as they are actually two similar tasks. For example, step on the gas pedal to move forward, step on the brake pedal to lower the speed, and hold the steering wheel to control the direction. This is so called transfer learning on boat driving with car driving concepts.

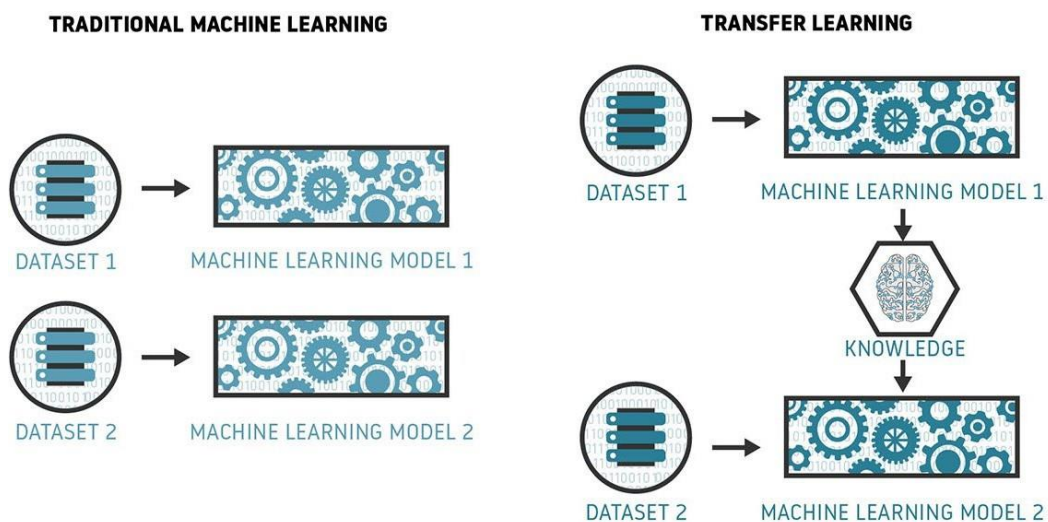


Figure 4.2.1: Traditional Machine Learning VS Transfer Learning

Traditional machine learning and deep learning require user to train the model from scratch, which is very time and resource consuming. Each model is trained only for its task as the knowledge is not retained or reused and learning process is conducted isolatedly of each model.

On the other hand, most of the researchers prefer to perform transfer learning. ResNet, VGG, and MobileNet are some typical pretrained model that can be used as the source domain for transfer learning. It is commonly used where the task at hand is similar to one that has

already been studied. Transfer learning may assist in cutting down on the time and computing resources needed to train a model for a new task while enhancing its performance in the same time by utilizing the information obtained from a pre-trained model.

### 4.3 Locating Corrupted Image

There was a corrupted image in the dataset. A python script is created to located the corrupted image. The code is appended in the appendix. The pseudocode of it is as follow:

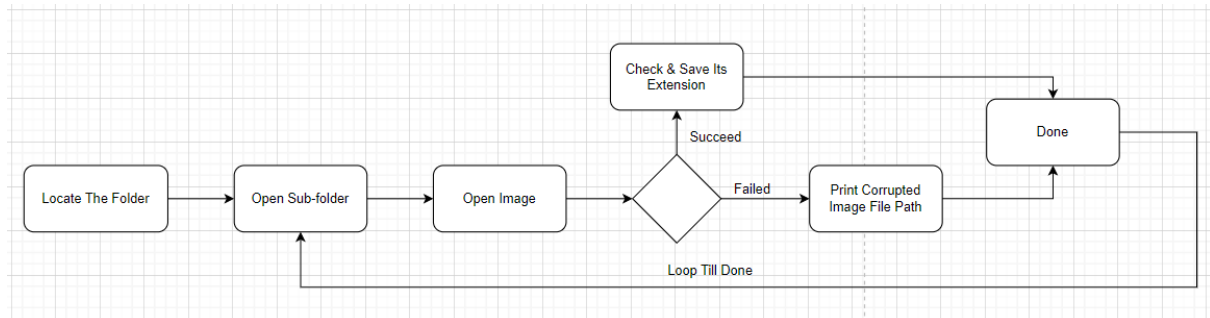


Figure 4.3.1: Flow of Identifying Corrupted Image

9. Define the `folder_path` variable to point to the folder containing the images.
10. Create two empty lists: `extensions` to store the file extensions of the valid image files, and `corrupted_files` to store the file paths of any corrupted image files.
11. Loop through each sub-folder in the specified `folder_path`.
12. For each sub-folder, loop through each file in the sub-folder.
13. For each file, create a file path using the `os.path.join()` function.
14. Use a try-except block to handle any exceptions that may occur when trying to open an image file:
  - a. If no exception occurs, open the image file using `Image.open()` and convert it to RGB format using `im.convert('RGB')`.
  - b. Check if the file extension is already in the `extensions` list, and if not, append it to the list.
  - c. If an exception occurs, print an error message indicating the file path that caused the error and append the file path to the `corrupted_files` list.
15. Print the list of corrupted files to the console.
16. Done.

One corrupted image is found in test set of manually annotated data, and it is removed from the dataset.

#### 4.4 Data Pre-processing – Image Classification

In order to classify all of the images into relative categories, 11 folders of classes are created in test folder and train folder. A python script is created for this, aiming to automate the process of classifying images. The code is appended in the appendix. The pseudocode of it is as follow:

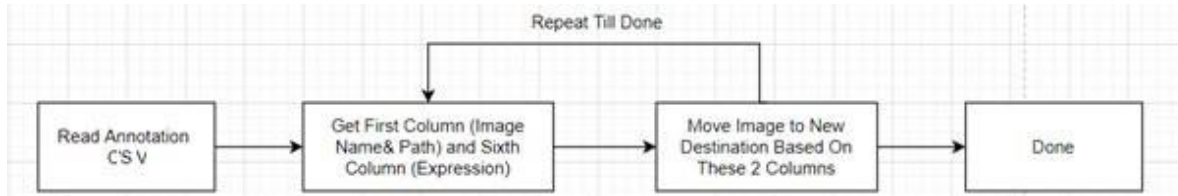


Figure 4.4.1: Flow of Image Classification

This python code will open up and read the CSV on the file path provided. Then it will look for the first column which consist of the image name and its corresponding file path and sixth column which consist of the image’s expression and get the values of these 2 columns. The image is moved to the new destination which is its class file based on the 2 values. This process is repeated till all images are moved to their matching expression class file.

After the script finish running, all of the images are sorted corrcctly according to their annotated expression as shown below.

Anger	26/8/2022 6:51 PM	File folder
Contempt	26/8/2022 6:51 PM	File folder
Disgust	26/8/2022 6:51 PM	File folder
Fear	26/8/2022 6:51 PM	File folder
Happiness	26/8/2022 6:51 PM	File folder
Neutral	26/8/2022 6:51 PM	File folder
No-Face	27/8/2022 8:07 PM	File folder
None	26/8/2022 6:51 PM	File folder
Sadness	26/8/2022 6:51 PM	File folder
Surprise	26/8/2022 6:51 PM	File folder
Uncertain	26/8/2022 6:51 PM	File folder

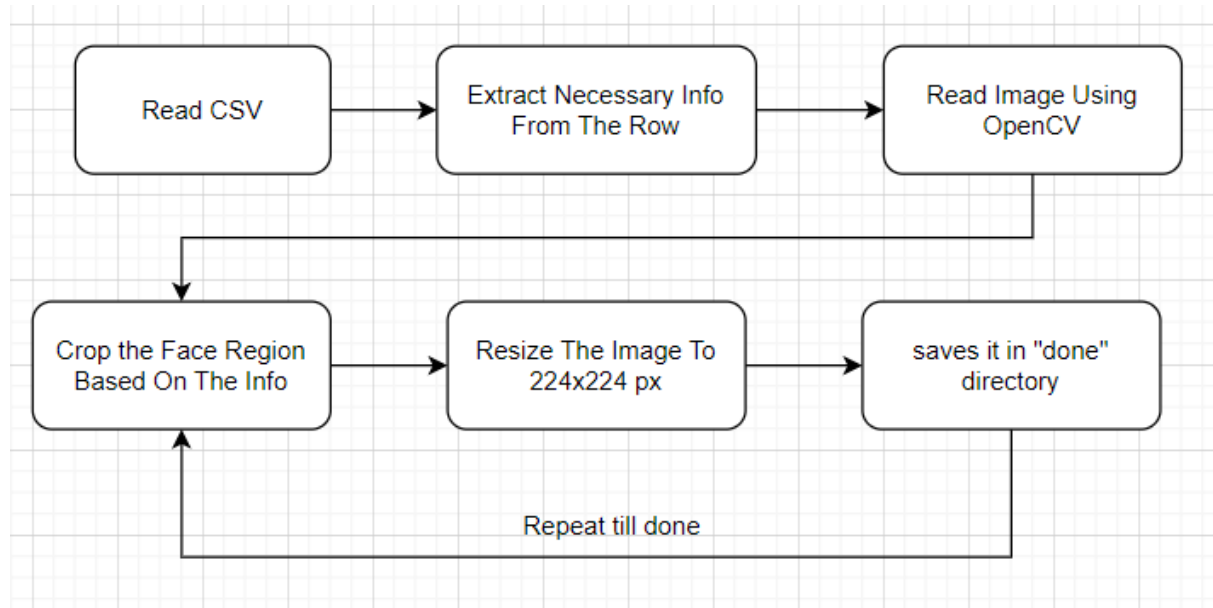
Figure 4.4.2: Sorted Dataset

#### 4.5 Data Pre-processing – Image Cropping

This procedure is performed to reduce the noises of each image. This helps to filter out the irrelevant part of the image so that the inputs are clearer for better feature extraction. A python script is created for this, aiming to automate the process of cropping the unnecessary parts while remaining only the face of the image, based on the **face\_x**, **face\_y**, **face\_width**, and **face\_height** provided in CSV. Below are the description for these 4 attributes:

- **face\_x**: the horizontal position of the top-left corner of the face bounding box, in pixels
- **face\_y**: the vertical position of the top-left corner of the face bounding box, in pixels
- **face\_width**: the width of the face bounding box, in pixels
- **face\_height**: the height of the face bounding box, in pixels

The code is appended in the appendix. The pseudocode of it is as follow:



*Figure 4.5.1: Flow of Image Cropping*

This Python script processes images based on the CSV file containing file paths, face\_x, face\_y, face\_width, face\_height, and expression. It crops the face region of the images based on the provided information, resizes them to 224x224, and saves them in a new directory. The script uses the OpenCV library for image processing and the multiprocessing module to speed up the processing by running multiple processes in parallel.



*Figure 4.5.2: Before Cropping (Left) VS After Cropping (Right)*

As shown in the pictures above, there a difference between before and after. Some unnecessary parts such as hair, neck, and blank background are removed while remaining the face in the center.

## 4.6 Under Sampling

An unbalanced dataset refers to dataset that are having different number of data in each of its classes. Under sampling is a method for balancing unbalanced datasets by retaining all data in the minority class while reducing the size of the majority class. In simple words, the class with lowest amount of data (minority class) is identified first, then the amount of data in all other classes (majority classes) are reduced to the same as the minority class.

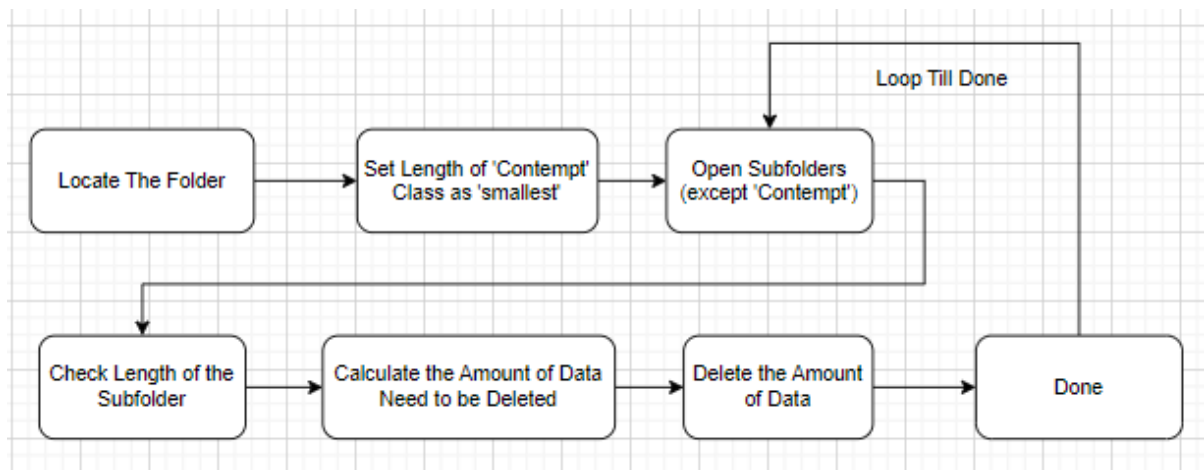


Figure 4.6.1: Flow of Under Sampling

1. Determines the number of images in the smallest subfolder ('Contempt') using `len(os.listdir(os.path.join(folder, 'Contempt')))`. It sets this value to `smallest`.
2. Creates a list of all subfolders (excluding 'Contempt') using a list comprehension. It sets this list to `subfolders`.
3. Creates a 'Pool' object for multiprocessing.
4. For each subfolder, the `process_subfolder` function is called with `(folder, subfolder, smallest)` as its arguments.
5. The `process_subfolder` function checks if the current subfolder is not 'Contempt' and if it exists as a directory.
6. The `process_subfolder` function determines the number of images in the subfolder using `len(os.listdir(subfolder_path))`.
7. If the number of images is greater than `smallest`, the `process_subfolder` function calculates the number of images to delete and removes them one by one using `os.remove(os.path.join(subfolder_path, os.listdir(subfolder_path)[0]))`.
8. After all subfolders have been processed, the code exits.

# Chapter 5: Experiment/Simulation

## 5.1 Hardware Setup

Description	Specifications
Model	HP Pavilion Gaming Laptop 15-dk1xxx
Processor	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 2592 Mhz, 6 Core(s), 12 Logical Processor(s)
Operating System	Windows 10
Graphic	NVIDIA GeForce GTX 1660Ti with Max-Q
Memory	16GB DDR4 RAM
Storage	1TB SSD

*Table 3.1.3.1: Specifications of Laptop*

Table above shows the hardware specifications of laptop being used to development of this project. GPU and processor is very useful on boosting the speed of model training, while this big enough storage is used to store the AffectNet dataset which is more than 100GB.

## 5.2 Software Setup

### Jupyter Notebook

Jupyter Notebook is an open-source application that allow user to run their code separately in cell. This is an extremely useful function for diagnosing and debugging the code as users do not have to rerun the entire code every time after modifying their code. And also, the users can share their code in a notebook with a user-friendly design as Jupyter Notebook supports rich text formatting so the users are allowed to create comprehensive markdown and clear documentation.

## 5.3 Setting and Configuration

### CUDA and cuDNN

CUDA and cuDNN is installed as the hardware GPU being used is NVIDIA. The main purpose of installing CUDA is for performing computation tasks on GPU. CUDA provides API for user to interact and communicate with the NVIDIA GPU. On the other hand, cuDNN deep neuron network library and indispensable for conducting deep learning duties. In this case, the power of GPU is being used in order to speed up model training process.



## Python environment

Upon pre-processing the database, some python scripts are created for different purposes. And also, with the help of python, some useful libraries are able to be utilized on each stage of development process.

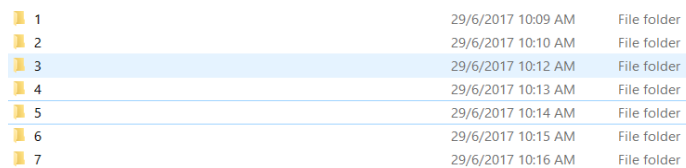
## TensorFlow and Keras

TensorFlow is a powerful library for machine learning and deep learning tasks. By calling Keras API, which is a high-level neuron network library based on TensorFlow, the process of CNN model building and training is much more easier as they provide user-friendly frameworks.

## 5.4 Implementation Issues and Challenges

### Classification of Dataset

Initially the dataset is provided in a raw form as shown in *Figure 5.4.1*.

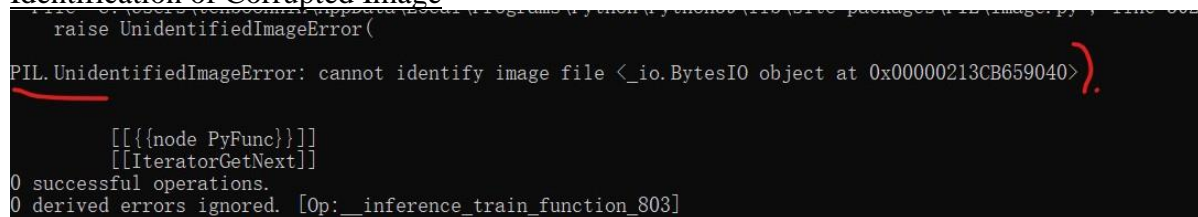


1	29/6/2017 10:09 AM	File folder
2	29/6/2017 10:10 AM	File folder
3	29/6/2017 10:12 AM	File folder
4	29/6/2017 10:13 AM	File folder
5	29/6/2017 10:14 AM	File folder
6	29/6/2017 10:15 AM	File folder
7	29/6/2017 10:16 AM	File folder

*Figure 5.4.1: Raw AffectNet Dataset*

All of the images are stored in folders without labelled directly on neither the folder or the image itself. The name of each image are also garbled unreadable which make it impossible to be used on training without the help of annotation CVS files.

### Identification of Corrupted Image



```
raise UnidentifiedImageError(  
PIL.UnidentifiedImageError: cannot identify image file <_io.BytesIO object at 0x00000213CB659040>)  
[[{{node PyFunc}}]]  
[[IteratorGetNext]]  
0 successful operations.  
0 derived errors ignored. [Op:__inference_train_function_803]
```

*Figure 5.4.2: Error of Reporting Corrupted Image*

This error occurs every time when during the process of model training. This error occurs when TensorFlow is not able to read a corrupt image file or a image file that is assigned with an invalid extension. Thus, the model training process is terminated every time TensorFlow try to read this image as an input.

### Utilization of Useful Information in the CSV Provided

subDirectory_filePath	face_x	face_y	face_width	face_height	facial_landmarks	expression	valence	arousal
324/27e086ee2eabdde2a009ce6	9	9	195	195	20.25;82.12;18.48;107.19;20.	1	0.685447	0.018762
213/b7a2d66132c29a1ce5096a5	48	48	325	325	106.82;245.68;104.14;273.39	0	0.015911	0.119697
292/1e07c59017e7560116db9f3	106	106	709	709	221.42;428.83;207.09;492.61	1	0.510148	0.08934
1453/e7bf4ba69a82b8de5d6f376	15	15	210	210	48.21;100.98;46.69;124.96;48	2	-0.23043	-0.10857
1078/3b3277db30720c3621427a	23	23	157	157	29.32;81.59;29.63;102.4;31.0	1	0.288721	0.008566
1136/68a68cdb03d7e4168eae7e	51	51	341	341	105.89;195.22;110.47;233.55	3	-0.24452	0.140019

*Figure 5.4.3: Information Available in the CSV Provided*

The information of each data are provided in the CSV given. By utilizing them, the noises of data can be reduced. Thus the challenge here is to determine the method of utilizing these information to preprocess the data before using them to train the CNN model.

### 5.5 Concluding Remark

In conclusion, the success of the project relies heavily on accurate listing of hardware and software requirements, and the careful consideration of setting and configuration issues.

The hardware and software requirements must be carefully selected to ensure that they are suitable for the intended use of the modelling. The setting and configuration issues must also be taken into account, as they can greatly impact the functionality and effectiveness of the training CNN.

Implementation issues, such as folder classification and corrupted image, should not be overlooked. It is important to ensure that all issues faced during the implementation process are solved.

By taking a methodical and well-planned approach, these challenges can be overcome. With the suitable hardware and software ready, the model training process can be conducted efficiently, effectively, and with greater flexibility.

# Chapter 6: System Evaluation And Discussion

## 6.1 System Testing and Performance Metrics

In a classification task, there are several performance metrics that can be used to evaluate the performance of a deep learning model. The performance metrics applied in this projects are:

- **Training accuracy: the accuracy of the model on the training data during training.** It is the proportion of correctly classified samples in the training set. Training accuracy is used to monitor the performance of the model during training and to check if the model is overfitting or underfitting.
- **Training loss: the error of the model on the training data during training.** It is the difference between the predicted outputs and the actual outputs of the model for the training set. The goal of training is to minimize the training loss.
- **Testing accuracy: the accuracy of the model on the unseen testing data.** Testing accuracy is a measure of how well the model can generalize to new data. A high testing accuracy indicates that the model is able to perform well on new data.
- **Testing loss: the error of the model on the testing data.** It is the difference between the predicted outputs and the actual outputs of the model for the testing set. The goal of training is to minimize both the training loss and the testing loss.
- **Precision:** the proportion of true positive predictions out of all positive predictions. It measures **how well your model in correctly identify positive cases.**
- **Recall:** the proportion of true positive predictions out of all true positive cases. **It measures the ability of the model to identify all positive cases.**
- **F1-score:** the harmonic mean of precision and recall. It provides a **balance between precision and recall** and is often used as a single metric to evaluate classification performance.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 6.1.1: Confusion Matrix

A confusion matrix illustrates a model's prediction performance on a binary class dataset, which only has “true” (positive) or “false” (negative) classes. There are four fundamental components in confusion matrix:

- **True Positives (TP):** Number of predictions correctly predicted as “positive.”
- **False Positives (FP):** Number of predictions wrongly predicted as “positive.”
- **True Negatives (TN):** Number of predictions correctly predicted as “negative.”
- **False Negatives (FN):** Number of predictions wrongly predicted as “negative.”

The picture

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Figure 6.1.2: Equations of Precision Score, Recall Score (Left) and F1 Score (Right)

## 6.2 Testing Setup and Result

Name	Training			Testing		Dataset Used	No. of Classes	No. of Epoch	No. of Layers	Regularization	Learning Rate	Output Layer Activation	Transfer Learning	Source Domain
	Figure	Accuracy	Loss	Accuracy	Loss									
binary(2)	6.2.2	0.8038	0.4207	<b>0.8236</b>	0.4153	Balanced_Cropped_Manual	2	80	12	Dropout = 0.2	0.001	Sigmoid	NO	NO
binary	6.2.3	0.7861	0.4578	<b>0.7944</b>	0.4412	Balanced_Cropped_Manual	2	50	12	Dropout = 0.2	0.001	Sigmoid	NO	NO
5class	6.2.4	0.3379	1.5242	<b>0.366</b>	1.4776	Balanced_Cropped_Manual	5	50	11	Dropout = 0.5, l2 = 0.001	0.001	Softmax	NO	NO
vgg16	6.2.5	0.3373	2.0197	<b>0.3631</b>	1.9557	Balanced_Cropped_Manual	11	100	5	Dropout = 0.5, l2 = 0.01	0.00001	Softmax	YES	VGG16
vgg19	6.2.6	0.3207	2.0649	<b>0.3379</b>	1.9956	Balanced_Cropped_Manual	11	100	5	Dropout = 0.5, l2 = 0.01	0.00001	Softmax	YES	VGG19
vgg16(2)	6.2.7	0.6355	1.0933	<b>0.3165</b>	2.2608	Sorted	11	10	5	None	0.00001	Softmax	YES	VGG16
mobilenet	6.2.8	0.5508	1.3991	<b>0.2514</b>	2.5961	Sorted	11	30	5	None	0.00001	Sigmoid	YES	MobileNet
3 indicators	6.2.9	0.1811	2.1521	<b>0.2038</b>	2.1215	Balanced_Cropped_Manual	9	50	11	Dropout = 0.5, l2 = 0.001	0.001	Softmax	NO	NO
resnet50	6.2.10	0.1285	2.3656	<b>0.1409</b>	2.3514	Balanced_Cropped_Manual	11	100	5	Dropout = 0.5	0.00001	Softmax	YES	ResNet50
fyp2	6.2.11	0.1315	2.459	<b>0.1309</b>	2.4603	Balanced_Cropped_Manual	11	30	15	Dropout = 0.2, l2 = 0.001	0.00001	Softmax	NO	NO

Constant Variable												
Hidden Layer Activation	Pooling	Kernal Size	Image Size	Batch Size	Rescale	Rotation Range	Width Shift Range	Height Shift Range	Shear Range	Zoom Range	Horizontal Flip	
ReLU	Max Pooling = (2,2)	3	224	32	1/255	20	0	0	0.2	0.2	TRUE	

Figure 6.2.1: Results of Each Model & Its Configurations

Several CNN models were trained for research purpose. All have them have different setting and configurations, such as number of epochs, number of layers, parameters, source domain of transfer learning, fine-tuned layers, and so on. The table below shows the results of each model and its configurations, sorted accordingly from best performance to worst performance. **Note: *binary(2)* is a model trained continued from *binary*.**

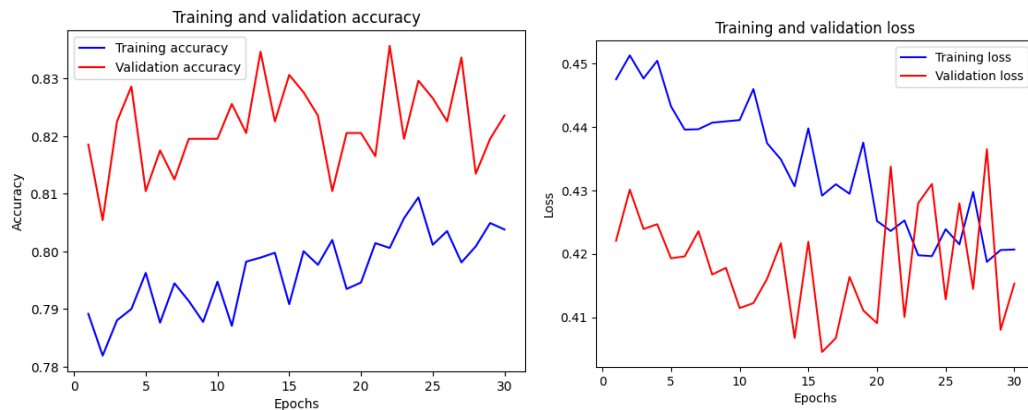


Figure 6.2.2: Results of *binary(2)* CNN

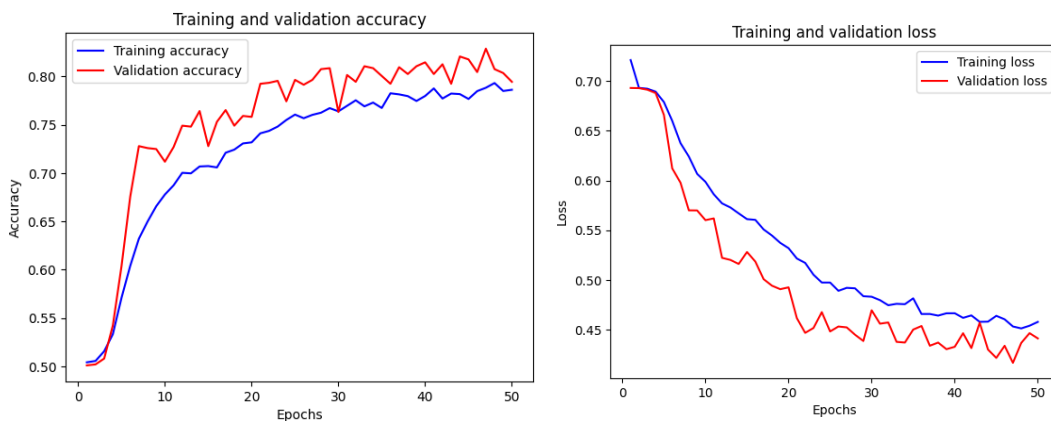
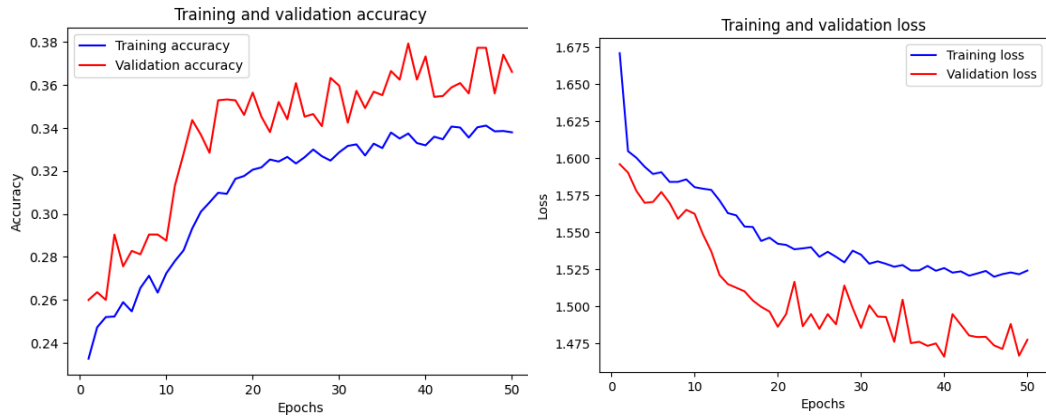
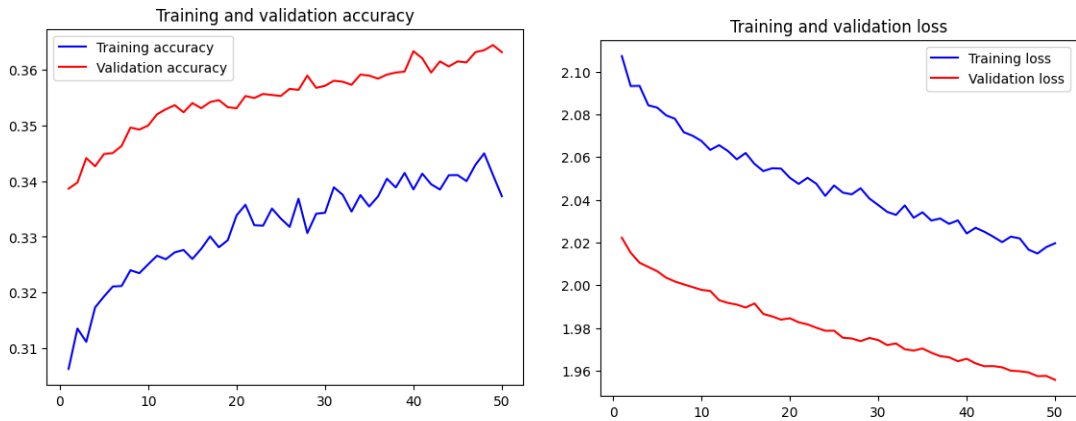


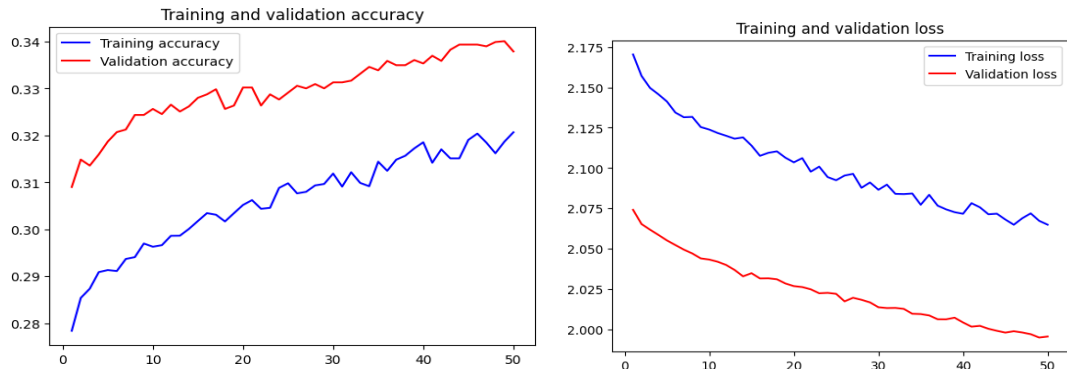
Figure 6.2.3: Results of *binary* CNN



*Figure 6.2.4: Results of 5 class CNN*



*Figure 6.2.5: Results of vgg16 CNN*



*Figure 6.2.6: Results of vgg19 CNN*

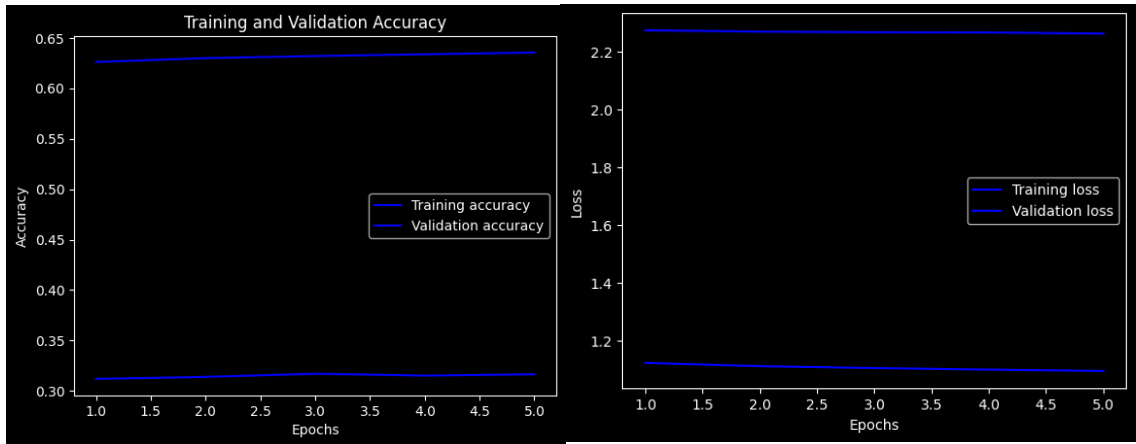


Figure 6.2.7: Results of vgg16(2) CNN

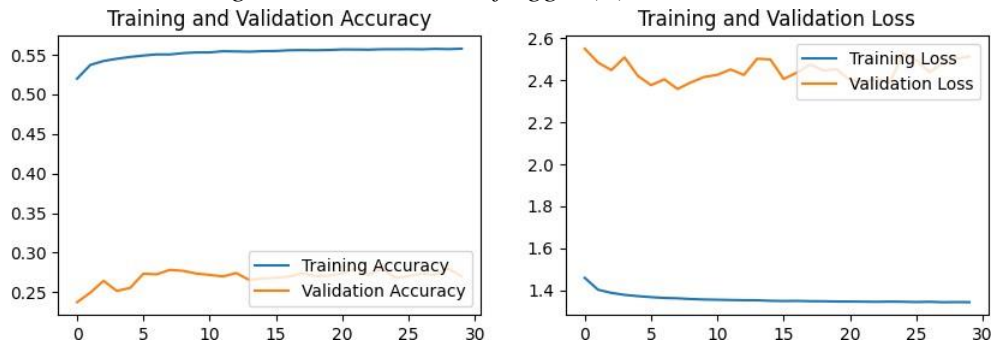


Figure 6.2.8: Results of mobilenet CNN

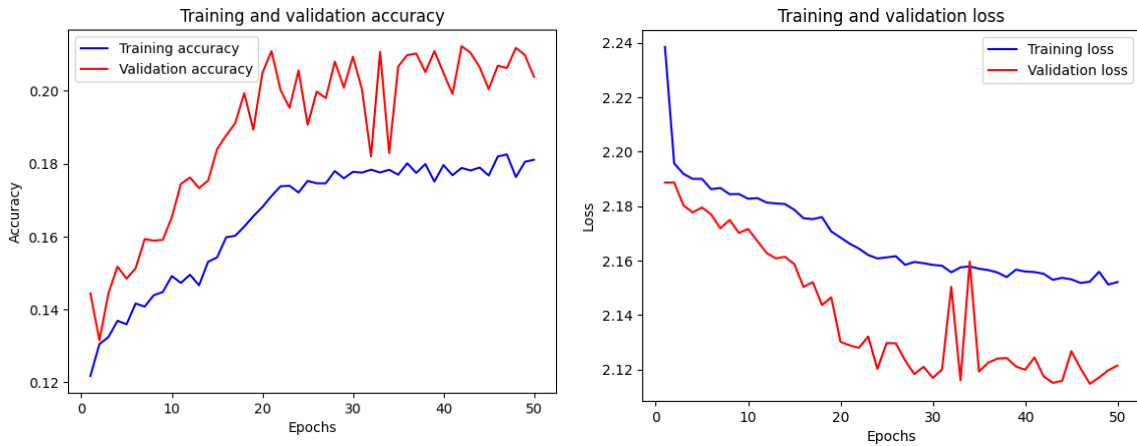


Figure 6.2.9: Results of 3 indicators CNN

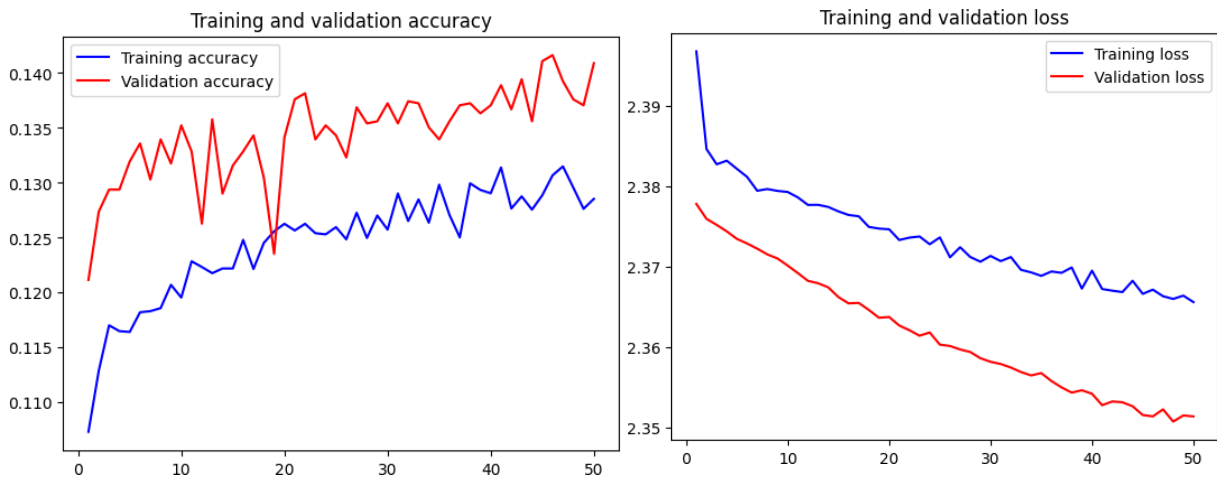


Figure 6.2.10: Results of resnet50 CNN

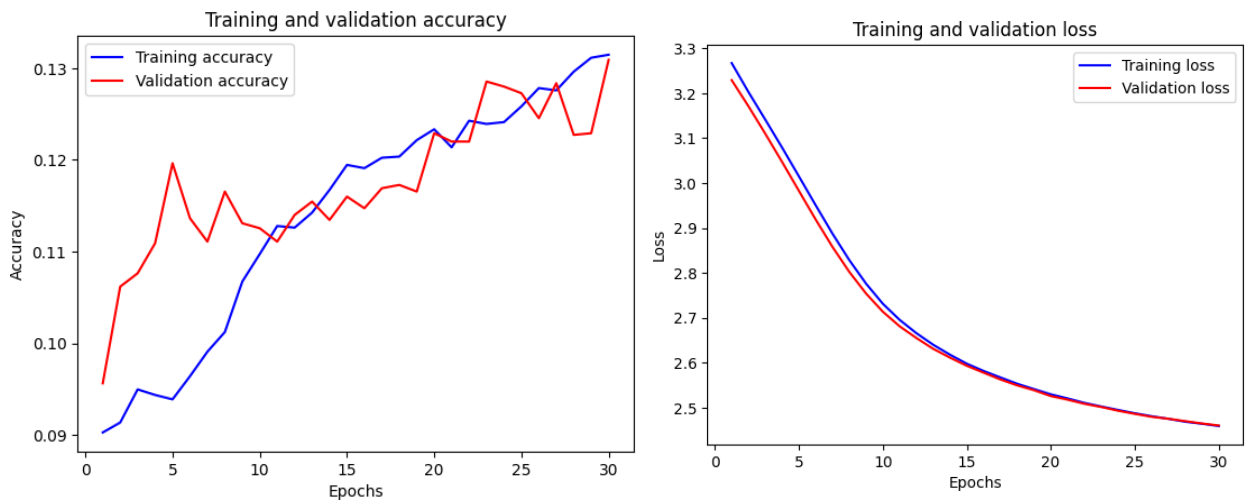


Figure 6.2.11: Results of fyp2 CNN



### 6.3 Project Challenges

Below are the list of challenges faced during working on this project:

- **Large Sized Dataset:** AffectNet dataset is too large in size that it takes more than 7.5 hours to train per epoch. The whole training time needed for a 10 epochs training can takes up to 3 days to complete. This is very time consuming when one have to train many epochs on a complicated CNN with multiple layers and various parameters.
- **Complexed Data Structure:** Multiple attributes are given in the annotation CSV. Time is spent on understanding the relationship among those attributes and analysis is conducted to test how they affect each other. One have to run multiple experimental training with different conditions (e.g. including some of the attributes) to explore and investigate the best method that secure satisfying results.

### 6.4 Objective Evaluation

Based on *sub-topic 6.2*, there are only qualified 2 models that is entitled for objective evaluation. As listed on *sub-topic 6.1*, these 2 models will be evaluated by applying precision score, recall score, and F1 score.

	precision	recall	f1-score	support
None	0.81	0.85	0.83	500
Happiness	0.84	0.79	0.82	500
accuracy			0.82	1000
macro avg	0.82	0.82	0.82	1000
weighted avg	0.82	0.82	0.82	1000

Figure 6.4.1: Objectives Evaluation Result of binary(2) CNN

	precision	recall	f1-score	support
None	0.80	0.79	0.79	500
Happiness	0.79	0.80	0.79	500
accuracy			0.79	1000
macro avg	0.79	0.79	0.79	1000
weighted avg	0.79	0.79	0.79	1000

Figure 6.4.2: Objectives Evaluation Result of binary CNN

Below is the instruction of understanding the result of the reports above, using *Figure 6.4.1* as example for explanation:

- Support is the number of instances in each class in the data. In *binary(2)* report (*Figure 6.4.1*), there are 500 instances in both "None" and "Happy" classes.
- The precision for "Happy" is 0.84, which means that out of all the instances the model predicted as "Happy", 84% of them were actually "Happy".
- The recall for "Happy" is 0.79, which means that out of all the "Happy" instances in the data, the model correctly identified 79% of them as "Happy".
- The F1-score for "Happy" is 0.82, which is a balanced measure of how well the model identifies "Happy" instances.
- The model achieved an overall accuracy of 0.82, which means that it correctly predicted the class of 82% of the instances in the data.
- Both macro and weighted averages of precision, recall, and F1-score are 0.82, which is the same as the accuracy, indicating that the model is equally good at predicting both classes.

## 6.5 Concluding Remark

As a continuous trained model of *binary CNN*, *binary(2)* is generally performing better than *binary CNN* and it is so far the best model among all the others. It has the best accuracy and loss while good fitting its test set, there is least overfitting and underfitting situation in this model.

Moreover, this model has scored well in the objective evaluation. Its precision score, recall score, F1 score is acceptable and can be considered as high. This model is proposed to be applied in further researches and development on related topics.

The concept of this model are as follow:

1. This model detects only “*Happiness*” and “*None*” classes
  - a. Note that “*None*” emotion in AffectNet dataset includes "confuse" expression[1]
2. When the student feels confused, he would show “*confused*” expression
3. When the student does not feel confused, he would show “*non-confused*” expression, and “*happiness*” expression is the most opposite expression to “*confused*” expression
4. Thus, if the model detected “*None*”, means the student is confused, else the model should detect “*Happiness*”

# Chapter 7: Conclusion and Recommendation

## 7.1 Conclusion

In conclusion, this project is aimed to solved the problem of low efficiency on progress of online academic studies due to shyness to ask for help when facing academic barriers. In order to build a program that will automatically trigger “raise hand” action on Microsoft Teams online meeting classes when the student looks confused, this project is aimed to develop a CNN model to detect and classify the facial expression of student. The motivations includes: boosting efficiency of online classes and allow lecturers to provide guidance to students and better understand their learning process.

On FYP1, the access permission of AffectNet dataset is gained with the help of supervisor. This dataset is super large on size and data availability and this helps to ensure the performance of trained model. It has to be downloaded and pre-processed before using it on model training.

After self-learning on the concept of AI and neuron networks, a CNN model is constructed based on the knowledge possessed and accessible information gained from the internet. Then this model is trained with the power of GPU. After solving the difficulties faced, the model is completely trained and evaluated. Several models are constructed with different configurations to experiment and explore the best setting that could assure the performance of model.

The result of each model is evaluated with their accuracy and loss during training and testing. The qualified models are then evaluated by objective evaluations. Lastly, the best model is proposed and reasons are listed with explanations.

This suggested model can be further trained and advanced in the future. More researches are needed in order to improve the performance of the model and build it in to more suitable to the task assigned.

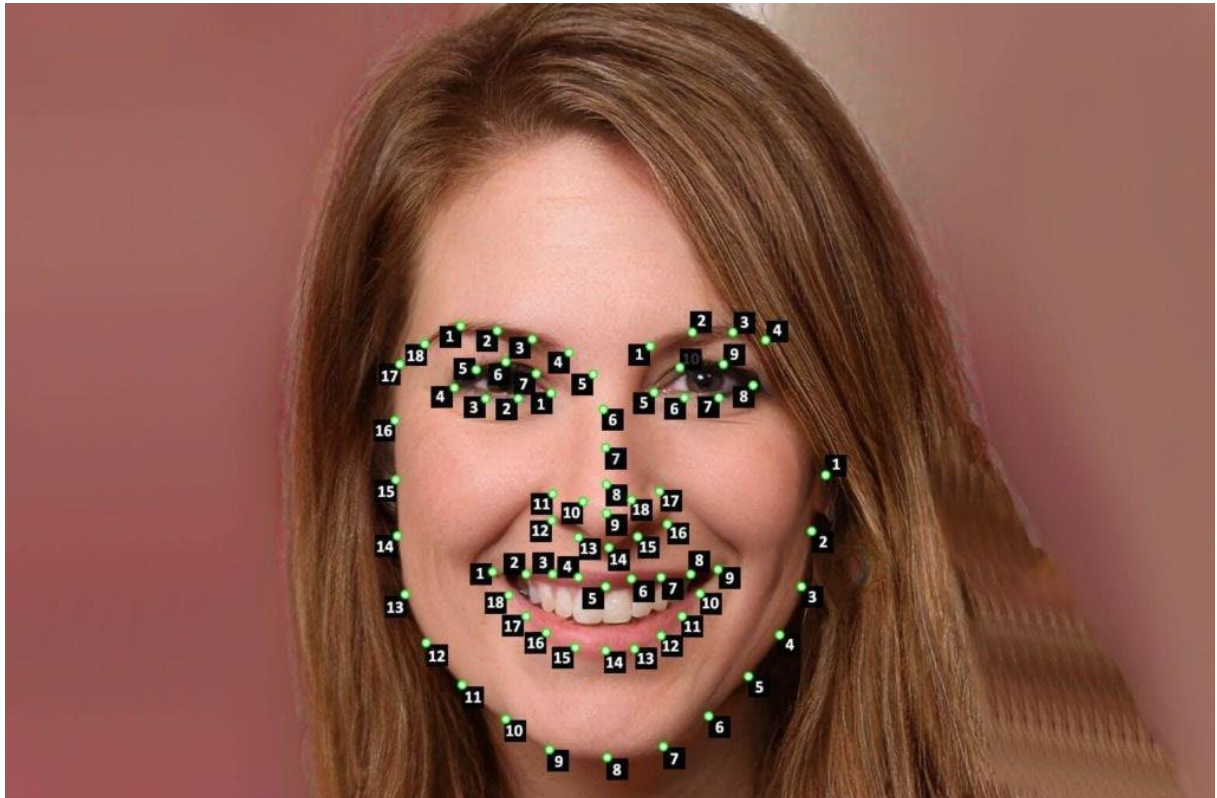
## 7.2 Recommendation

There are several improvements that can be done on this project. Yet, they are not performed in this project due to the time restriction.

### **Data Pre-processing - Image Cropping (advanced)**

In this project, the images are cropped based on the boundary boxes provided. The aim of performing this step is to remove unwanted parts from the images. This helps to reduce the noise of data so that the prediction made by the model is more accurate. In fact, each image

can be further cropped to minimize the unwanted parts on that image. This is because the landmarks of faces are provided in the CSV. Face landmarks plot out the exact face shape, includes only facial key landmark points such as eyebrows, eyes, nose, and lips on an image of a human's face. This can further reduce the noise of the data since only necessary key points are included in each data.



*Figure 7.2.1: Example of Face Landmarks*

### **Selection of Dataset**

A dataset called “**EmoReact**” is found at the moment I was finalizing my this project. EmoReact has a separated “confused” class, which is one of the most important criteria in this project. It is better to use EmoReact dataset to train the *binary* model I created as it is more suitable to be applied compared to AffectNet dataset, which does not has a separated “confused” class. Same like AffectNet, EmoReact is a private dataset that is not open to the public. One has to apply for the access of the dataset if he wishes to utilize this dataset.

The application google form link:

[https://docs.google.com/forms/d/1Pk7I9mIUBItVKrhZDM0grbucS0wnJ33zXbyisyNds08/vi/ewform?edit\\_requested=true](https://docs.google.com/forms/d/1Pk7I9mIUBItVKrhZDM0grbucS0wnJ33zXbyisyNds08/vi/ewform?edit_requested=true)

## REFERENCES

- [1] A. Mollahosseini; B. Hasani; M. H. Mahoor, "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild," in *IEEE Transactions on Affective Computing*, 2017.
- [2] Dhawan, S. (2020). Online Learning: A Panacea in the Time of COVID-19 Crisis. *Journal of Educational Technology Systems*, 49(1), 5–22.
- [3] Kim, K. J., & Bonk, C. J. (2006). The future of online teaching and learning in higher education. *Educause quarterly*, 29(4), 22-30.
- [4] Lapitan, L. D., Tiangco, C. E., Sumalinog, D. A. G., Sabarillo, N. S., & Diaz, J. M. (2021). An effective blended online teaching and learning strategy during the COVID-19 pandemic. *Education for Chemical Engineers*, 35, 116–131.
- [5] Li, X., Pfister, T., Huang, X., Zhao, G., & Pietikäinen, M. (2013, April). A spontaneous micro-expression database: Inducement, collection and baseline. In 2013 10th IEEE International Conference and Workshops on Automatic face and gesture recognition (fg) (pp. 1-6). IEEE.
- [6] Mohammadmahoor.com. 2022. *AffectNet – Mohammad H. Mahoor, PhD*. [online] Available at: <<http://mohammadmahoor.com/affectnet/>> [Accessed 29 August 2022].
- [7] Mollahosseini, A., Hasani, B. and Mahoor, M., 2019. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, 10(1), pp.18-31.
- [8] Numpy.org. 2022. NumPy Reference — NumPy v1.23 Manual. [online] Available at: <<https://numpy.org/doc/stable/reference/>> [Accessed 31 August 2022].
- [9] Pandas.pydata.org. 2022. API reference — pandas 1.4.3 documentation. [online] Available at: <<https://pandas.pydata.org/docs/reference/index.html>> [Accessed 23 August 2022].

[10] Pillow.readthedocs.io. 2022. Pillow. [online] Available at:

<<https://pillow.readthedocs.io/en/stable/>> [Accessed 31 August 2022].

[11] Seaborn.pydata.org. 2022. API reference — seaborn 0.11.2 documentation. [online]

Available at: <<https://seaborn.pydata.org/api.html>> [Accessed 31 August 2022].

[12] Scikit-learn.org. 2022. API Reference — scikit-learn 0.21.3 documentation. [online]

Available at: <<https://scikit-learn.org/0.21/modules/classes.html>> [Accessed 31 August 2022].

[13] Tani, M. (2005, January). Quiet, but only in class: Reviewing the in-class participation of Asian students. In Higher Education Research and Development Society of Australia Conference, Sydney, New South Wales. Retrieved September (Vol. 5, p. 2007).

## Appendix

### Data Pre-processing - Image Sorting

```
import shutil
import csv
import time
import multiprocessing

def copy_image_file(row):
    path_in_csv, express = row[0], int(row[6])
    pathto = [
        'Neutral', 'Happiness', 'Sadness', 'Surprise', 'Fear',
        'Disgust', 'Anger', 'Contempt', 'None', 'Uncertain', 'No-Face'
    ][express]
    src_path = f"D:/Dataset/Manually_train_cropped/{path_in_csv}"
    dest_path = f"D:/Dataset/Sorted_resize_manual/val/{pathto}"
    print(f"copy {src_path} ({express}) to {dest_path}")
    shutil.copy(src_path, dest_path)

if __name__ == '__main__':
    starttime = time.time()

    # read csv file
    with open('D:/Dataset/AffectNet_Folder_Sorting/validation.csv') as f:
        rows = list(csv.reader(f))

    # copy image files based on expression category in parallel
    with multiprocessing.Pool() as pool:
        pool.map(copy_image_file, rows)

    endtime = time.time()
    duration = endtime - starttime
    print(f"Finished copying {len(rows)} files in {duration:.2f} seconds.")
```

## Identification of Corrupted Image – Train Set

```
import os
from PIL import Image

folder_path = r'D:/Dataset/Sorted/train/test/'
extensions = []
corrupted_files = []

for fldr in os.listdir(folder_path):
    sub_folder_path = os.path.join(folder_path, fldr)
    for filee in os.listdir(sub_folder_path):
        file_path = os.path.join(sub_folder_path, filee)
        try:
            im = Image.open(file_path)
            rgb_im = im.convert('RGB')
            if filee.split('.')[1] not in extensions:
                extensions.append(filee.split('.')[1])
        except:
            print("Error opening file: ", file_path)
            corrupted_files.append(file_path)

print("Corrupted files: ", corrupted_files)
```



## Data Pre-processing – Image Cropping

```
import io
import os
import sys
import random
import cv2
import pandas as pd
from PIL import Image
import csv
import time
from multiprocessing import Pool

base = 'D:/Dataset/ori_manual/Manually_Annotated_Images'
done = 'D:/Dataset/Manually_train_cropped/'
csv_file = 'C:/Users/tehboonhin/Desktop/Degree/Y3S3/FYP2/validation.csv'

def process_image(row):
    fname = row['subDirectory_filePath']
    x = int(row['face_x'][0:])
    y = int(row['face_y'][0:])
    width = int(row['face_width'][0:])
    height = int(row['face_height'][0:])
    expression = int(row['expression'][0:])
    folder_dir = fname.split('/')[0]
    img = fname.split('/')[1]
    image = cv2.imread(os.path.join(base,fname))

    imgROI = image[x:x + width, y:y + height]

    imgROI = cv2.resize(imgROI, (224, 224), interpolation=cv2.INTER_AREA)

    if not os.path.isdir(done + folder_dir):
        os.mkdir(done + folder_dir)
```

```
cv2.imwrite(done + folder_dir + '/' + img, imgROI)
print(fname)

if __name__ == '__main__':
    num_processes = 4 # set the number of processes to run in parallel
    start_time = time.time()
    with open(csv_file, 'r') as csvfile:
        reader = csv.DictReader(csvfile)
        rows = list(reader)
        num = len(rows)
        with Pool(num_processes) as p:
            p.map(process_image, rows)
    end_time = time.time()
    print("Processing time:", end_time - start_time, "seconds")
```

## Under Sampling

```
import os
from multiprocessing import Pool

def process_subfolder(folder, subfolder, smallest):
    if subfolder != 'Contempt' and os.path.isdir(os.path.join(folder, subfolder)):
        subfolder_path = os.path.join(folder, subfolder)
        num_images = len(os.listdir(subfolder_path))
        if num_images > smallest:
            images_to_delete = num_images - smallest
            for i in range(images_to_delete):
                os.remove(os.path.join(subfolder_path, os.listdir(subfolder_path)[0]))

if __name__ == '__main__':
    # Step 1: Determine the number of images in Contempt
    folder = 'D:/Dataset/Sorted_resize_manual/train/'
    smallest = len(os.listdir(os.path.join(folder, 'Contempt')))

    # Step 2: Iterate over the other subfolders and delete images until they have the same
    number of images as Contempt in parallel
    subfolders = [subfolder for subfolder in os.listdir(folder) if subfolder != 'Contempt']
    with Pool() as pool:
        pool.starmap(process_subfolder, [(folder, subfolder, smallest) for subfolder in
subfolders])
```

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.:4
Student Name & ID: Teh Boon Hin 19ACB03289	
Supervisor: Dr. Aun Yichiet	
Project Title: Auto Raise Hand in Microsoft Teams (API/Extension)	

## 1. WORK DONE

- FYP 1 progress follow up

## 2. WORK TO BE DONE

- Transfer learning
- Develop, train, and evaluate multiple CNN models
- Construct FYP report

## 3. PROBLEMS ENCOUNTERED

- Overfitting
- Huge imbalanced dataset

## 4. SELF EVALUATION OF THE PROGRESS

Conduct self-learning and make sure understand the concepts of transfer learning



\_\_\_\_\_  
Supervisor's signature



\_\_\_\_\_  
Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.:7
Student Name & ID: Teh Boon Hin 19ACB03289	
Supervisor: Dr. Aun Yichiet	
Project Title: Auto Raise Hand in Microsoft Teams (API/Extension)	

## 1. WORK DONE

- Transfer learning
- Overfitting handling
- Construct multiple models

## 2. WORK TO BE DONE

- Balance dataset
- Develop, train, and evaluate multiple CNN models
- Construct FYP report

## 3. PROBLEMS ENCOUNTERED

- Time constraint
- Inability

## 4. SELF EVALUATION OF THE PROGRESS

Conduct self-learning and make sure understand the concepts of transfer learning




\_\_\_\_\_  
Supervisor's signature




\_\_\_\_\_  
Student's signature

## POSTER



UTAR  
UNIVERSITI TUNKU ABDUL RAHMAN

# AUTO RAISE HAND IN MICROSOFT TEAMS (API/EXTENSION)




Students are too shy to ask for help upon facing difficulties during online class. A facial expression recognition CNN is constructed to develop an auto raise hand extension in Microsoft Teams.

### Proposed Method

- 1 Data collection and pre-processing
- 2 Construct multiple models
- 3 Transfer learning

Several models are created with different configurations. They are all evaluated and shortlisted models have gone through objective evaluation. Eventually the model with best performance is suggested.



### Suggestion

- Data pre-processing
  - Advanced image cropping
- Dataset selection
  - EmoReact

Project Developer: Teh Boon Hin      Project Supervisor: Dr. Aun Yichiet

# PLAGIARISM CHECK RESULT



## Turnitin Originality Report

fyp2 by Boon Hin Teh

From FYP2 (FYP202301)

Processed on 28-Apr-2023 01:44 +08

ID: 2077392052

Word Count: 7827

Similarity Index	Similarity by Source
12%	Internet Sources: 6% Publications: 5% Student Papers: 9%

### sources:

- 1 1% match (student papers from 22-Apr-2022)  
[Submitted to Universiti Tunku Abdul Rahman on 2022-04-22](#)
- 2 1% match (student papers from 07-Sep-2022)  
[Submitted to Universiti Tunku Abdul Rahman on 2022-09-07](#)
- 3 1% match (student papers from 22-Mar-2023)  
[Submitted to Middle East College of Information Technology on 2023-03-22](#)
- 4 1% match (Internet from 16-Nov-2022)  
<https://riunet.upv.es/bitstream/handle/10251/109776/Valero%20-%20Clasificador%20de%20expresiones%20faciales%20basados%20en%20algoritmos%20de%20aprendizaje%20autom%c3%a1tico..pdf?isAllowed=y&sequence=1>
- 5 < 1% match (student papers from 15-Apr-2022)  
[Submitted to Universiti Tunku Abdul Rahman on 2022-04-15](#)
- 6 < 1% match (student papers from 17-Apr-2023)  
[Submitted to Universiti Tunku Abdul Rahman on 2023-04-17](#)
- 7 < 1% match (student papers from 05-Sep-2022)  
[Submitted to Universiti Tunku Abdul Rahman on 2022-09-05](#)

<b>Universiti Tunku Abdul Rahman</b>			
<b>Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)</b>			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 27/4/2023	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

<b>Full Name(s) of Candidate(s)</b>	Teh Boon Hin
<b>ID Number(s)</b>	19ACB03289
<b>Programme / Course</b>	BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION SYSTEMS
<b>Title of Final Year Project</b>	Auto Raise Hand in Microsoft Teams (API/Extension)

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)</b>
<b>Overall similarity index: <u>12</u> %</b>  <b>Similarity by source</b> Internet Sources: <u>6</u> % Publications: <u>5</u> % Student Papers: <u>9</u> %	
<b>Number of individual sources listed of more than 3% similarity: <u>0</u></b>	
<b>Parameters of originality required and limits approved by UTAR are as Follows:</b> (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

**Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.**

Signature of Supervisor

Signature of Co-Supervisor

Name: Dr. Aun Yichiet

Name: \_\_\_\_\_

Date: 28/4/2023

Date: \_\_\_\_\_





**UNIVERSITI TUNKU ABDUL RAHMAN**

**FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY  
(KAMPAR CAMPUS)**

**CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	19ACB03289
Student Name	Teh Boon Hin
Supervisor Name	Dr. Aun Yichiet

<b>TICK (√)</b>	<b>DOCUMENT ITEMS</b>
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

\*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

\_\_\_\_\_ T B H \_\_\_\_\_

(Signature of Student)

Date: 28/4/2023

