



Affective  
Movie Evaluator

## **BIT304 FINAL YEAR PROJECT I**

B1301746 Ibrahim Mohamed Shaatha  
B1301511 Elebe Faith Alfred

Submitted to the

FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY  
(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)

In partial fulfilment of the requirements  
for the degree of

HUBIT  
HUBSC2

HELP UNIVERSITY  
APRIL 2019

# Final Report I: Affective Movie Evaluator

Ibrahim M. Shaatha, Faith Alfred

April 16, 2019

# Contents

<b>I</b>	<b>Proposal</b>	<b>1</b>
1	Introduction	2
2	Background	3
3	Issues	5
4	Benefits and Constraints	6
5	Project Aims and Objectives	7
6	Project Scope	8
7	Software and Hardware Requirements	9
7.1	Funding Sources . . . . .	9
7.2	Minimum hardware requirements . . . . .	9
7.3	Recommended hardware requirements . . . . .	9
8	Development Methodology	11
<b>II</b>	<b>Project Management Plan</b>	<b>14</b>
1	Introduction	15
1.1	Object of the Project . . . . .	15
1.1.1	To create metrics by which the system gives score. . . . .	15
1.1.2	To research on the requirements of the dataset and methodologies before March. . . . .	15
1.1.3	To Create and Collect Datasets for testing and development . . . . .	15

1.1.4	2.4 To determine accuracy of the project by comparing its output to existing scoring systems. . . . .	16
1.1.5	2.5 To produce accuracy reports for the analysis subsystems . . . . .	16
1.2	Output expected from the project . . . . .	16
<b>2</b>	<b>Work Breakdown Structure</b>	<b>18</b>
<b>3</b>	<b>Risk Management Plan</b>	<b>20</b>
<b>4</b>	<b>Project timeline</b>	<b>21</b>
<b>III</b>	<b>Requirement Analysis</b>	<b>22</b>
<b>1</b>	<b>Introduction</b>	<b>23</b>
<b>2</b>	<b>Requirement Summary</b>	<b>24</b>
2.1	Actors and Stakeholders . . . . .	24
2.1.1	Admin . . . . .	24
2.1.2	Subject . . . . .	24
2.2	Object represented in the system . . . . .	25
<b>3</b>	<b>Functional Requirements</b>	<b>26</b>
3.1	Use-Case Diagrams . . . . .	26
3.2	High-Level Use Cases . . . . .	27
<b>4</b>	<b>Non-Functional Requirements</b>	<b>28</b>
4.1	Technical Requirements . . . . .	28
4.1.1	Prototype 1 - BIT304 . . . . .	28
4.2	Usability Requirements . . . . .	29
4.3	Reliability Requirements . . . . .	29
4.4	Security Requirements . . . . .	29
<b>IV</b>	<b>Iterations</b>	<b>30</b>
<b>1</b>	<b>Iteration Plans</b>	<b>31</b>
1.1	Current Iteration . . . . .	31
1.2	Future Iterations . . . . .	32

<b>2</b>	<b>Iteration 1</b>	<b>33</b>
2.1	Introduction . . . . .	33
2.2	Purpose . . . . .	33
2.3	Context . . . . .	33
2.4	Schedule of Iteration Workflow . . . . .	34
2.5	Iteration Schedule Breakdown . . . . .	34
2.6	Resource Summary . . . . .	36
2.7	Evaluation Criterea . . . . .	37
2.8	Analysis and Design Artefacts . . . . .	38
2.8.1	Pipeline diagrams . . . . .	38
2.8.2	Class Diagrams . . . . .	38
<b>3</b>	<b>Iteration I: Implementation and Testing</b>	<b>40</b>
3.1	Implementation . . . . .	40
3.2	User Manual . . . . .	43
3.3	Testing . . . . .	43
3.4	Unit Testing . . . . .	44
3.4.1	UnitTest A1 . . . . .	44
3.4.2	UnitTest A2 . . . . .	46
3.4.3	ManualTest M1 . . . . .	48
3.4.4	ManualTest M2 . . . . .	50
3.5	Iteration Review and Evaluation . . . . .	51
<b>V</b>	<b>Conclusion</b>	<b>53</b>
<b>1</b>	<b>Current Status</b>	<b>54</b>



# List of Tables

7.1	Minimum Hardware Requirements . . . . .	9
7.2	Recommended Hardware Requirements . . . . .	10
2.1	Work breakdown tasks . . . . .	19
3.1	Risk Management Plan . . . . .	20
3.1	Analysis usecases . . . . .	27
4.1	Minimum Software Requirements . . . . .	28
1.1	Iteration Plan . . . . .	31
1.2	Future Iterations . . . . .	32
2.1	Schedule of Iterations Workflow . . . . .	34
2.2	Schedule of Tasks . . . . .	34
2.3	Hardware Resources used for iteration . . . . .	36
2.4	Software Resources used for iteration . . . . .	36
3.1	Table of Strategy . . . . .	43
3.2	Table of Strategy for Models . . . . .	43
3.3	Test Plan . . . . .	44
3.4	Test Plan . . . . .	44
3.5	Automated Unit Tests . . . . .	44
3.6	Manual Unit Tests . . . . .	47
3.8	Unit Test A2 . . . . .	47
3.9	Unit Test M1 . . . . .	48
3.10	Unit Test M2 . . . . .	50
3.11	Statistics . . . . .	52

# List of Figures

8.1	SEMMA (2018, Data Mining for Business Intelligence Series) . . . . .	12
2.1	Task in terms of complexity . . . . .	18
2.2	Work Breakdown Chart . . . . .	19
4.1	Overview of baseline gantt chart . . . . .	21
2.1	Objects in the system . . . . .	25
3.1	Iteration 1: Before integration . . . . .	26
3.2	Analyse usecase with subsystems . . . . .	26
3.1	The GUI Application . . . . .	41
3.2	Record Sample: 1. Select the movie file . . . . .	41
3.3	Record Sample: 2. Fill in the blanks . . . . .	42
3.4	Analyse Screen . . . . .	42
3.5	Burn down chart . . . . .	52
1.1	Expression Net . . . . .	55



## **Abstract**

For our final year project we are building an application that analyses videos of audience watching movie and gives the video a score called the Affective Movie Evaluator. We established the goals and objectives and we were able to achieve roughly half of our objectives. This report is a summary of the conception, developement and of the project. We also discuss the current progress of the project and future plans briefly.

## Declarations

I hereby declare that the report presented here as part of the requirement of BIT304 is original and no parts of this report had been plagiarised from any other resources unless those indicated with proper referencing. This report will be the property of HELP University and cannot be distributed in any form without the written consent of HELP University.

Student Names:

Mohamed Shaatha Ibrahim (B1301746)

Elebe Faith Alfred (B1301511)

Date: 12-04-2019

## **Acknowledgement**

Aknowledgements This project will not be possible without the help of a number of people. We would like to express our deepest gratitude to HELP's University Department of Information and Communication Technology for their support and provision to make this project a success and to our family for their unconditional love and support to make sure our education is successful.

Our gratitude goes to the following individuals who have been with us through this journey: Ms. Shu Mn for teaching and guiding us through the entire process of this project Mr. Abdul Quayoon for being patient with us as our supervisor, and for providing insight and expertise that has greatly helped us in implementing our project

And we also thank our Colleagues who have been of help to us in the best way that they can.

# Part I

## Proposal

# Chapter 1: Introduction

Movie production studios use test screening in order to determine whether their movies will succeed. Recently, internet movie distributors such as Netflix and Amazon have started taking more control of the production of their content, it is more beneficial for studios to re-edit their movie once they can predict that the product is going to fail. While there are automated sentiment analysis tools used to measure how well the audience reacts to test screening, most of the feedback is still taken from questionnaires and the audiences subjective evaluation.

We aim to provide an automated, movie evaluation system which is going to analyse the audience watching the movie and predict an overall score. Our system is intended to supplement the existing manual process of test screenings. We are not proposing a system to replace the existing subjective evaluations given by the audience.

The aim of this project is to develop a system that can evaluate how good a movie is, by analysing both face and body postures of the audience. In order for this to work we provide the system a video file consisting of audience reactions, and the system will tell us how good a movie is within a rating metrics, we will define later. It is important for us to state, that we are not proposing a real-time system here; the system accepts a recording of the raw data - which is currently defined as video file, and can be any other form of measurement - as the primary input.

A system like this can aid us develop further, a more general system that can evaluate how good a certain product or presentation is by analysing the audience reaction limited by a certain context. In our case the product is a movie, and how the audience reacts may depend on what genre of movie they are watching, or what kind of experience they were expecting beforehand. It is likely that our system may not work well without knowing these information. Systems like these are also related to computer vision with Emotion AI technologies, such as face based lie detectors, which uses the same techniques to work.

# Chapter 2: Background

For a long time movie industry have used statistical modeling and machine learning to predict, the success of movies based on high level data. While there have been several works in early to mid 2000s, which takes in multimedia information to form a representation for the users emotions (Zhang, Tian, Jiang, Huang, & Gao, 2008; Wu et al., 2008), the earliest work we could find that specifically focus on movie scenes was by Soleyman et. al.

Quite recently, there have been interest in using deep learning techniques to analyse audiences facial data such as (Saha, Navarathna, Helminger, & Weber, 2018), to predict whether an audience enjoyed the movie, most notably research work that has been funded by The Walt Disney Company (Deng et al., 2017). The purpose of their research is to use facial emotion recognition techniques, to analyse the emotion of the audience in order to improve the movies.

Research has shown that over 90% of our communication can be non verbal (Algorithmia, 2018). Non- verbal communication which includes facial expressions and body postures among others can be used to determine the emotional state of a human being. According to Paul Ekman (Ekman P, 1992), there are basically six types of human emotions namely: happiness, sadness, disgust, fear, surprise, and anger.

The system we are proposing is going to take in a video file of an audience reacting to a movie and attempts to determine whether the audience liked that movie. It will use emotion recognition and body pose to quantify the overall reaction in order to determine whether the viewer enjoyed the movie or not.

Although, idea of such a system is not new, we have not been able to find an existing system that considers both body pose and facial emotion data to create a model that can predict. Previous work such as (V Ramanarayanan et al., 2015), which evaluates a presenter by using a very similar technique that we are planning, performed very poorly compared to the human judges. However, we are a bit more optimistic, since movies are designed to elicit certain emotional responses and audiences in average exhibit similar reactions. Such a system, maybe useful in the film industry to analyse the reactions of the test audience as a supplement to the usual questionnaires. The problem we are trying to solve is that film studios screens movies to test audience, and audience writes back a subjective feedback usually on paper. Our software is intended to complement the existing evaluation procedure, by providing an

automated evaluation of audience's experience that is less subjective and provide a numerical score that represents audience's sentiment more accurately.

Affective Analysis is already used by marketing giants such as Coca-Cola to predict the effectiveness of their advertisement on target test audiences. Affective movie analysis could prove to be valuable to streaming services that distributes their own content and have more control over the productions such as NetFlix.

# Chapter 3: Issues

Our system is not real-time but still requires heavy computing power in order to run. This means the admin will often have to run multiple processes for analysis and come back later for the result. Our system does not support multiple processes yet, while it is possible for our system to run a process on a remote server or a cloud computer service via ssh and http, the admin still have to wait for the analysis to be completed.

In order to minimize the scale of the problem, the first few iterations of the problem deals with only a single audience member. The largest audience we will be designing our system will be for a group of two to four person, under controlled lighting environment. Unless we have access to infrared cameras and better computing facilities, we will not be able to overcome these limitations.

Our video can only have one person in front of the camera. The application is not able to give proper result if there are faces in the background when the audience member is reacting. Presence of another person in the room currently effects the resulting data, and it is very time consuming for a human to verify whether the recorded data does not have any background face. However, as we wrote our detection algorithm to choose the largest region of interest in the image, the fail cases we have observed usually consists of background posters and wallpapers with larger face than our subject.

One of the weakness we have is in the accuracy of the face detection algorithm. OpenCV's default facial detection algorithm, (Viola-Jones full frontal model) requires the audience member to face the camera directly. However, we plan to implement a more accurate face detection algorithm at the risk of increasing computing time.

Right now our system uses an external media player known as VLC media player, to play the movie while being recorded in use case 1. The system admin is usually required to install the software manually. However on Windows based systems, it is sometimes not possible to find the



# Chapter 4: Benefits and Constraints

For our current iteration the system can analyse only one audience member. The reason for this limitation is that we initially wanted to start the project with a smaller scope and also to reduce the amount of tasks we might have for the first iteration. Even in our future iterations, we will not be analysing an audience consisting of more than a handful of people, unless we get better camera equipment.

Another issue faced by us is that facial emotion recognition is not working that well, for now. In order to improve the accuracy of the FER model we plan to write separate test cases.

Since our system takes a long time to analyse, we had to modify our "Analyse Video" use case into two separate parts. Instead of directly providing a video file to run analysis and immediately getting the results and visualisations, our system actually requires the admin to run two separate instances of analysis, facial emotion analysis (FER) and pose analysis (BEGR). Running an instance of analysis will produce the required data file, which can then be visualised as a time-series and processed to be sent for our movie evaluation ML model.

# Chapter 5: Project Aims and Objectives

The aim of our project is to develop a system that can evaluate how good a movie is, by analysing both face and body postures of the audience. In order for this to work we provide the system a video file consisting of audience reactions, and the system will tell us how good a movie is within a rating metrics, we will define later. It is important for us to state, that we are not proposing a real-time system here; the system accepts a recording of the raw data - which is currently defined as video file, and can be any other form of measurement - as the primary input.

- system like this is very useful for movie studios, which often relies on test audience in order to predict whether their movies are going to be successful and a system like this can aid in measuring the audience reaction more accurately.

The objectives we have set for our project are the following: Define the metrics by which the system is going to give the score before we create the evaluation model. (e.g: value between 0.0-1.0 or a discrete grading system, like A+, A,

- Research the requirements of the dataset, and methodologies before week 3.
- Find or create the required test datasets for in order to complete objective 6.
  - Dataset for the emotion recognition subsystem
  - Dataset for body pose subsystem.
  - Dataset for the whole system: Video recordings of audiences reacting to movies.
  - Determine accuracy of the overall project by comparing its output to existing scoring systems.
- To prove that we are using state of the art technologies in our system; produce:
  - Accuracy report for Emotion recognition
  - Accuracy report Body pose/action estimation

# Chapter 6: Project Scope

- For developement purpose, we will be dealing with short movies mostly.
- We will be recording atleast 12 subjects, watching multiple movies.

Before we proceed, we would like to borrow a very recent terminology from Andrej Karpathy that has been in use in the industry. *Software 2.0* refers to any software that is written without direct human involvement, with the help of another software. We chose to use this term in document to highlight the fact that the developement approach and the documentation practices used for the machine learning portions of the code is slightly different than the rest, and it will not be honest from our part to describe the process as the same. Though term originated in an article he wrote, we have seen the term used within a very few academic papers (Ratner, Hancock, Dunnmon, Goldman, & Ré, 2018). In contrast, *Software 1.0* refers to any type of software that a human is directly responsible for creating and we will be only using the term to distinguish it, if necessary.

A large portion of the project is in Software 2.0, therefore we do realize we have to spend a lot of time collecting data, analysing the data and evaluating our application. For this reason our primary methodology is described as SEMMA methodology, even though for iteration 1 we actually use a mixed method.

# Chapter 7: Software and Hardware Requirements

## 7.1 Funding Sources

As mentioned before we do not have any corporate and industry backing, therefore we are expecting HELP School of ICT to provide any possible funding we may require, if requested. Since the University already have most of the assets we require for our FYP1, we will require permission from the IT department to use the facilities, for our project. As for FYP2, the tools and resources we require for data collection and obtaining ground truth can only be determined after we complete the first part of our project.

## 7.2 Minimum hardware requirements

Although, our system can run with a laptop with a webcam we designed the system with the following requirements in mind.

Table 7.1: Minimum Hardware Requirements

No	Item	Usage	Qty
1	VGA USB Camera	Used to record audience	1
2	Computer Terminal (Desktop/Laptop /SmallFactor)	Our application need to run on an computer with a display terminal.	1

## 7.3 Recommended hardware requirements

Table 7.2: Recommended Hardware Requirements

No	Item	Usage	Qty
1	HD Camera	Used to record audience	1
2	High-end Desktop Computer	Our application need to run on an computer with a display terminal.	1
3	CUDA compatible Graphics Card - NVIDIA geforce gtx 1080 ti	Used to improve video analysis models	1

# Chapter 8: Development Methodology

To give you the context of the reasoning behind our decision, we would like to discuss a brief overview of commonly used development methodologies first. The most commonly followed methodologies for data-science projects are either SEMMA or CRISP-DR methodologies. SEMMA is a methodology developed by SAS institute, while CRISP-DR method was originally developed in a joint effort by five companies, and further improved by data mining industry, most notably by IBM.

The main difference between both methodologies are that CRISP-DR considers business understanding of the process and also considers model deployment and the results. SEMMA is mainly used for data mining projects, which does not consider business aspect. The methodology we chose for our project is SEMMA methodology.

The reason we chose SEMMA is that, we wanted to keep option generalise our problem scope later, for example we wanted to apply what the system does in other scenarios than movies. Also, business project timelines and research project timelines are usually incompatible, and this is a fact we need to consider while planning early on, since machine learning projects falls under the latter. So not considering the business aspects of the project, will be an advantage for us given the time we have.

SEMMA methodology consists of 5 phases (Sample, Explore, Modify, Model and Assess). SEMMA methodology gives us freedom to reiterate from any step, as we require. It does not have a definitive cycle for iteration, which makes it a more simple and less rigid framework to follow, compared to CRISP-DR (Palacios, 2017). Sample: First step of SEMMA involves collecting all data samples. Explore: Understand the problem through exploration of sample data. Exploration can be helped through visualisations and other data analytic tools. Modify: Next steps, involves selecting which variables/features can be used, how they can be transformed or processed in order to create a model. Model phase consists of selecting and creating a model. Assess: Finally we evaluate how well the model performed, and determine whether it is useful in real world situation. In order to properly use SEMMA methodology into our project we are going to collectively apply the methodology to three separate components of our system that we have previously identified; the facial emotion subsystem, body emotion subsystem and movie evaluation model. After we started planning, we realized given only two

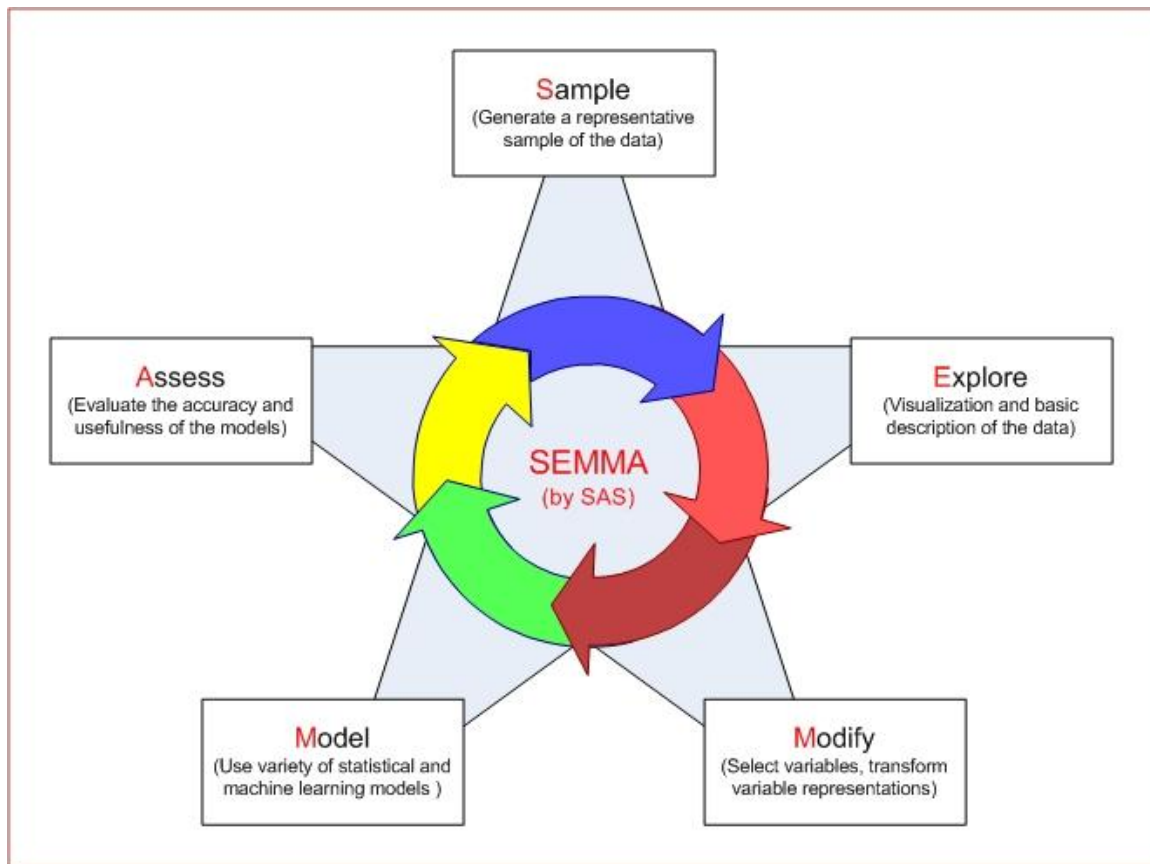


Figure 8.1: SEMMA (2018, Data Mining for Business Intelligence Series)

team members, it was actually impossible for us to go through the whole system with one iteration within one semester.

Therefore, we have split the project into two phases. Each phase is considered iteration by itself, because during the first phase, we build the first two components of our pipeline simultaneously (by two members), such that they both go through the SEMMA process at the same time. During the process, we also ensure both components are integrated before we start the next phase.

Then, we will start next iteration; go back to the sample stage to work on the final part of our project. For example, after the Assess phase of the emotion models we will be starting again from Sample phase again, and ensure everything in the pipeline is working well. This does not mean we define the term phase and iteration as being interchangeable.

Eventually, given the results it is possible we may plan a new iteration, if necessary. For example, if after assessing the Movie Evaluation Model, it performed poorly, we may go back to either Modify stage to pick better features to be passed into the model or we can even go back to Model stage to pick a different model.

The first planned iteration of our project we will be focused on building the facial emotion

and body pose emotion subsystem, which will be for simplicity hence referred to as Emotion Model and Pose Model, respectively. The second part of our project will be the combined system along with Movie Evaluation Model. Most of our work on Movie Evaluation Model, is currently planned to commence on the second part of our internship (FYP2), and the gantt chart we have proposed in the Appendix section might significantly change.

As our primary version control system we will be using gitlab as our github repository. Also, since SEMMA does not define a project management style, we will be using kanban/scrumban like board feature on gitlab, to organize the tasks that we will be doing. However, we will not be adhering to any strict project management style, which is very prevalent in the software industry.

As for the specific techniques we will use throughout the development, our idea is to analyse existing literatures, and compare notes and follow general data mining/modeling techniques with fewer brainstorm sessions. In order to obtain the ground truth while collecting the data, our current idea is to use questionnaires after recording a subjects reaction. Alternatively, its possible for us to measure blood pressure or any other means of observation if possible, and we will consider that for our FYP2.



## Part II

# Project Management Plan

# Chapter 1: Introduction

The aim of our project is to develop a system that can evaluate how good a movie is, by analysing both face and body postures of the audience. In order for this to work we provide the system a video file consisting of audience reactions, and the system will tell us how good a movie is within a rating metrics, we will define later. It is important for us to state, that we are not proposing a real-time system here; the system accepts a recording of the raw data - which is currently defined as video file, and can be any other form of measurement - as the primary input. A system like this is very useful for movie studios, which often relies on test audience in order to predict whether their movies are going to be successful and a system like this can aid in measuring the audience reaction more accurately.

## 1.1 Object of the Project

### 1.1.1 To create metrics by which the system gives score.

In order to accomplish our goal our system needs a valid output format. The output format maybe a discrete list of values, for example the letters A to C or a continuous number. The purpose of this metrics is to indicate the overall sentiment of the movie for a single audience.

### 1.1.2 To research on the requirements of the dataset and methodologies before March.

We decided to add research on dataset and methodologies as an objective since, we cannot start development of the system without it.

### 1.1.3 To Create and Collect Datasets for testing and development

The following datasets will be collected for testing and development of our system. The exact criteria and the requirements for the dataset will be decided after analysis in the work item.

1. Facial Dataset with labelled Emotion

2. Pose Dataset with body pose labelled
3. Video Recordings of audience reacting to Movie

#### **1.1.4 2.4 To determine accuracy of the project by comparing its output to existing scoring systems.**

This is essential because we need the means to determine how well our system is performing. The system cannot be considered complete without that.

#### **1.1.5 2.5 To produce accuracy reports for the analysis subsystems**

Analysis report for subsystems are used to tune the system and lets us know which part of the application needs improvement.

## **1.2 Output expected from the project**

The project will be divided into two phases; initially, we will be working on two separate system that capture the emotion and pose respectively, into meaningful data and then streamline our work into a single codebase. The second phase of our project will be working together to create the movie evaluation model, which takes in the processed data from our previous system and gives us a prediction of how well the audience liked that movie.

The main focus of our project will be developing the Movie Evaluation Model that we have discussed earlier, in our project proposal. However, since the movie evaluation model we have proposed requires all the emotion data, we need to use a working emotion recognition system first. We may use any python/tensorflow based off the shelf or open source system for that purpose.

The first part of our project is choosing and testing the preliminary subsystems of the pipeline. Note that since we use off the shelf code for the first phase of the project, there is no training dataset required, we expect that pre-trained models are provided by the library we choose. Also, in order to properly validate the system it is ideal to avoid using the original training dataset that was used to create the pre-trained model as our test datasets.

The deliverables consists of:

- A Facial emotion recognition subsystem. (Faith)
  - Data collection: Facial dataset and emotion dataset.

- Evaluate facial detection library
  - Evaluate emotion recognition library
- Test script
  - Performance Report.
- Body pose emotion (Body language) subsystem. (Ibrahim)
  - Data Collection: Body pose data
  - Test script
  - Performance Report.
- Integrated codebase: Contains code to load and run the previous two modules on a frame of a video file, and prepare data to feed the movie evaluation model.
- Movie Evaluation model.
- Collect movie dataset: Record audience reacting to movies.
- Report

## Chapter 2: Work Breakdown Structure

During our analysis phase we broke down the project into largest completable tasks for each phase. In-order to identify the completable tasks, we thought of the project in terms of user-story/use-case perspective and also by identifying the independent components of the system, from our analysis. We thought of the first input data that will pass through the pipeline, and how it will be processed along to get the results we wanted. For example, if our initial data is video files captured by a camera device, the system must have to at somepoint interact with a camera object, therefore it is very likely during our design phase that we may need to create a camera object for a module.

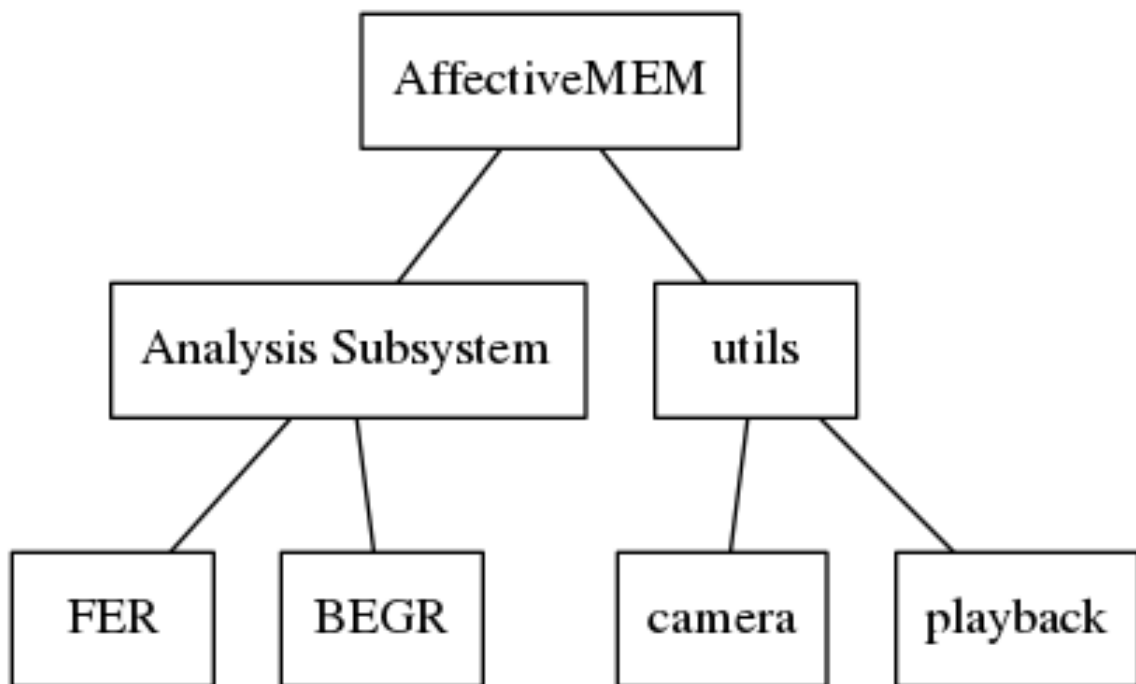


Figure 2.1: Task in terms of complexity

The figure 2.1 shows an approximate overview of the entire system. The picture is not intended to give you an idea of how those subsystems will interact. It is there to show you what components we will be creating and working on in terms of the workbreakdown structure. In contrast, figure ?? shows the actual work breakdown structure we planned for the only iteration in FYP 1.

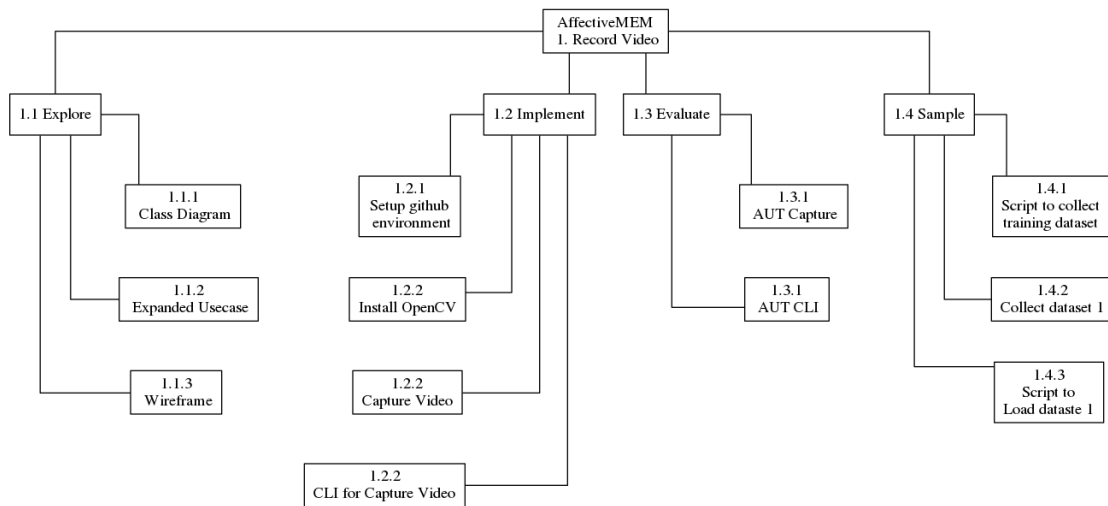


Figure 2.2: Work Breakdown Chart

Table 2.1: Work breakdown tasks

Phase	Task	Start	Finish	Size
1	1. Class Diagram	12/03/19	16/03/19	M
	2. Expanded Use-case	12/03/19	16/03/19	S
	3. Wireframe	12/03/19	16/03/19	S
2	1. Setup github environment			S
	2. Install OpenCV			S
	3. Capture Video in a folder			L
	4. CLI to Capture Video			M
3	1. AUT Capture			M
	2. AUT CLI			S
4	1. Script to collect training dataset			M
	2. Collect dataset 1			L
	3. Script to load dataset1			M

# Chapter 3: Risk Management Plan

Table 3.1: Risk Management Plan

#	Description	Probability	Impact	Mitigation Strategy
1	Cannot collect dataset to test movie evaluation system.	40%	4	Use online reaction videos for shorter movie clips.
2	Insufficient memory to train the model	20%	4	Ask the department for cloud computer access or more physical computer resources
3	Evaluation model is not performing as expected, because features extracted are not sufficient to represent properly	5%	2	Use another mean to take the measurements, use a thermal camera or heartbeat measuring instrument. If possible use new measurements to create a baseline to compare the evaluation model.

# Chapter 4: Project timeline

We were assigned our supervisor near the by of January, and we spent the first two months mostly on research, analysis and planning our approach. Our official timeline for the project started on March 11, though we were did analysis work few weeks prior. Since our earlier analysis work was presented on the interim project plan we submitted and the requirements were changed slightly after considering the advice we had on our own literature review and estimation of computer processing related resources we will need.

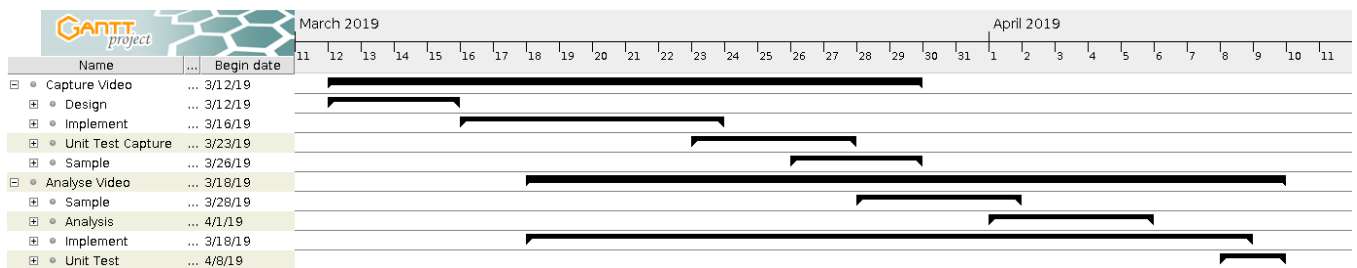


Figure 4.1: Overview of baseline gantt chart

We have mentioned a few times throughout this documentation how the project timeline was created from the WBS. After the time and priority was set what we are supposed to do was to pick the tasks for every week by ourselves, and assign daily hours allocated to spend for the task . After the first week we found out that we are able to finish the tasks really fast, and we were free without a task assigned to ourselves. Eventually we decided to assign the tasks evenly for the entire iteration, this was easy to achieve because early in part of the iteration we were both working on our own subsystem, which can be run independently. Later on more tasks were dependent on each other, especially the design and testing tasks.

You can find our baseline gantt chart for estimated project plan in Appendix A. We created the work breakdown structure during our project plan, and we grouped each task under a phase in our methodology. Then we estimated time for each task and prioritised them, finally creating a project timeline in a gantt chart format.



## Part III

# Requirement Analysis

# Chapter 1: Introduction

Since the conception of the project plan, our initial goal for the first iteration was to create a system that is going to analyse the facial emotion and body posture emotion independently and construct a data representation such as a time-series from the analysis. We delegated the primary goal of our project, which is to use that data to train and create a model that can score movies for later iterations intentionally, so it will give us time to reconsider our methods and also create dataset in between the downtime.

Like any other software development project, our approach was to first of all analyse the system and identify. Practically, most of the requirement analysis was done while we were creating the project plan, while we did not thoroughly document it in the proposal itself. This part of the document was later updated to reflect the change in number of use-cases in our iteration 1, after we realised that we are not able to fit the three initial use-cases we planned into the first iteration.

Along with the project's functional requirements and non-functional requirements, we will also try to give an overview of the requirements from the machine learning perspective. We will try to state the ML problem, and define the requirements and structure for the problem. It should be noted by now, our system will consist multiple ML algorithms interacting with each other and a portion of has to be designed by us.

We also wanted to complete all the software design and engineering intensive tasks as early as possible so we can focus on the machine learning aspects of our project. Thus, a large portion of this iteration is writing the code for user interaction, and designing processes within the system that can perform independent of the type of user interface.

Initially, we like we planned we worked on the facial analysis, and posture analysis portion of the code separately. Later on we refactored and integrated the code as separate modules into a single codebase with the help of object-oriented programming. Refactoring and Integration was done together on site with each others help.

# Chapter 2: Requirement Summary

The first thing we did during our first meeting was infact to identify all the stakeholders and actors who interacted with the system. While we discussed the naming conventions of the actors involved during the later meetings, it was early on established that only one user directly interacts with the system and can be considered to be the primary user.

Our systems consists of mainly this one user controlling the application directly. Ideally this user is a system operator, hired by the production company with a knowledgeable background in IT and system administration. Throughout this document we refer to this user, simply as *Admin*. We should note that the user *Admin* is not really a recognized object within the system and thus does not have a login function. *Admin* refers to whoever is operating the system.

## 2.1 Actors and Stakeholders

### 2.1.1 Admin

An *Admin* can use the system to record audiences watching a movie. Right now, in order to do that, the movie must exist within the computer system first. Admin is supposed to launch our application, and provide the details of the audience member and the movie, because the system does not yet store the metadata of the movies yet. The movie object is not considered as an object of the system in this iteration.

- Admin is able to Record a video of a *Subject* watching a movie.
- Admin is able to Analyse a video of a *subject*
  - Produce time-series.

### 2.1.2 Subject

Subject is not considered a primary actor in the system. The only interaction between the subject and the system occurs when the admin records the subject watching a movie in the first use-case. The admin initiates the use-case and subject passively sits infronts of the camera,

until the movie is completed, whole while the admin actually have the control of the system such that they can terminate the recording process or wait until the movie is finished.

While the UML for iteration 1 might not directly hint that our system considers *Subject* as a seperate entity for design purpose, it is going to be added as an Object in our future iteration. We did not include the *Subject* as an object that is in the system, yet.

## 2.2 Object represented in the system

For this iteration, all use-cases are going to be involved with only two object. Object Sample represents a sample recording of an audience video, while the object movie represents a Movie that an audience is watching. It is important that these two logical objects are well defined within the system, as we might later need to categorize and display the statistics for future use cases.

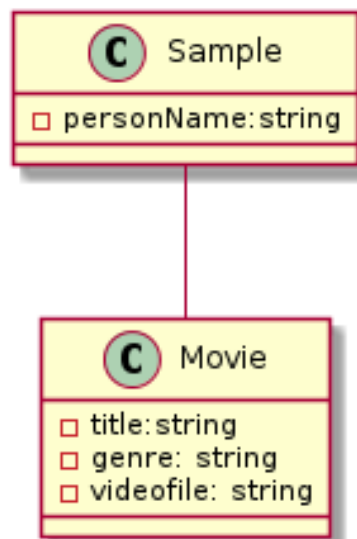


Figure 2.1: Objects in the system

# Chapter 3: Functional Requirements

## 3.1 Use-Case Diagrams

While designing of the use-cases we only thought from a logical/business perspective and not included the "technical" usecases, as our methodology and approach does not heavily rely on UML.

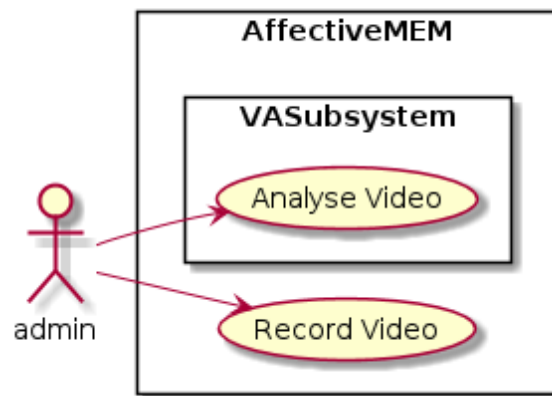


Figure 3.1: Iteration 1: Before integration

In figure 3.2, you can see that the Analyse Video is bound within a subsystem of Affective Movie Evaluator. The reason

While there is a weak implementation of other use cases, the two use-cases shown in the diagram 3.2, are the main goal of our current iteration until now.

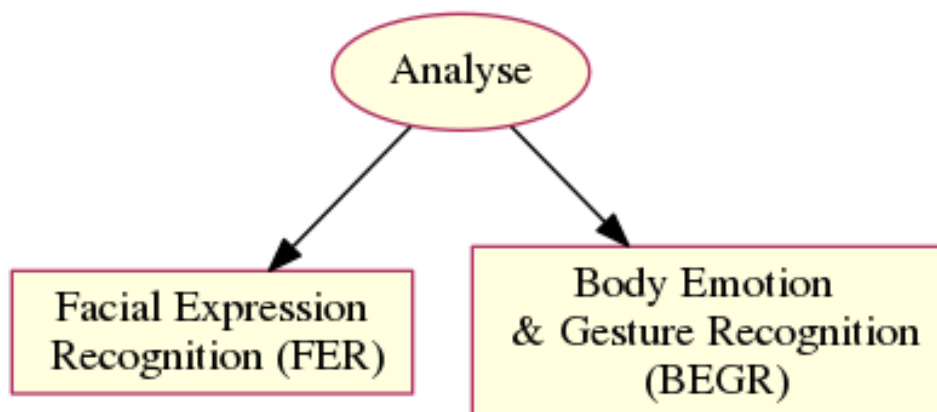


Figure 3.2: Analyse uscase with subsystems

It would have been ideal for the sake of this documentation for Analyse use case to include two "technical" use-cases, "Analyse FER" and "Analyse BEGR", so that it is easier to credit the two team members, but it would not have been accurate. Ideally a use case should show an added value to a system, and seperating them does not really contribute the value to from either logical or design perspective.

## 3.2 High-Level Use Cases

Table 3.1: Analysis usecases

<b>Use Case 1</b>	Record Video <sup>1</sup>
Goal in Context	To record an audience member's video and store it.
Primary Actors Secondary Actor	- Admin - Audience
Description	Use camera to record and store video session of an audience watching a movie screening.
<b>Use Case 2</b>	Analyse Video <sup>2</sup>
Goal in Context	To extract an audience's facial and body pose data from a video file and store it as readable data.
Primary Actors Secondary Actor	- Admin - Audience
Description	Use camera to record and store video session of an audience watching a movie screening.

<sup>1</sup>Use case was designed by faith and implemented by Ibrahim.

<sup>2</sup>FER portion of use-case was created by Faith and BEGR was created by Ibrahim.

# Chapter 4: Non-Functional Requirements

## 4.1 Technical Requirements

### 4.1.1 Prototype 1 - BIT304

We recommend to use a moderately powerful desktop computer to run the application. For video capture, it currently uses a webcam, though the application is currently compatible with any other type of live video capture stream. Ideally, a dual monitor system must be used one facing the audience and one facing the *Admin*, who will be interacting with the control panel. The system was actually tested on a laptop computer and we assume that it is also possible to setup and run this application on most standard laptops without any issue.

Almost all of our application is written using Python 3. We used the Anaconda distribution of python and it's environment and package management system because it reduced the compatibility issues associated with installing packages such as tensorflow across different systems.

Table 4.1: Minimum Software Requirements

Software
Windows 7/8

#### 1. Recommended Setup

- Linux based system, preferably Debian variant
- System Utilites
  - git
  - conda
- Intel based processor preferably greater than 5th Generation i5
- 8GB RAM

## 4.2 Usability Requirements

Our standard user documentation is written as a README file in markdown format. Since the project is currently available in GitHub, the documentation is the first thing the user can see of our project. The purpose of the user documentation is to describe the whole project briefly, and guide the user to install and setup the application.

The user is able to interact with the application through commandline and GUI.

## 4.3 Reliability Requirements

In our current iteration, all the files are stored in the filesystem within the project file. Our version control system (git) does not track the data files and models, since they are considered large and we do not think it will be effective to use VCS to track data files. Therefore, datasets and results must be shared and versioned manually, we will have to back it up separately and share it on internet with email and file sharing services.

One of the scripts we wrote that will run when a new user is going to setup the system, will actually download the pre-trained models used by subsystems from the internet. This seems to be a standard practice as even larger projects such as OpenPose often does not store their models in the github repository.

## 4.4 Security Requirements

By design, our system does not have an authentication system nor it is considered required. We consider that if deployed, most production studios have their own security systems to control access.

The only time the application uses network currently is to access the internet and download the required models and software.

One concern a production studio may have is regarding the security of test audience data that is stored. Depending on the contract they have with their own test audience, the test audience may be concerned with their privacy if the data is leaked or shared without their permission for other projects.



# Part IV

## Iterations

# Chapter 1: Iteration Plans

Early on we planned two iteration for FYP 1, but as mentioned in the requirement analysis we had to change that because of time constraint and risks of not completing the objectives of our FYP 1. We set out two meetings for iteration plans, which includes analysis, work break down structure and task assignment. Most of the work was already done during our analysis for the project plan, what we had to do was arrange our timeline so that we complete atleast two of the project objectives. Prior to analysis we already identified what is going to be two major subsystems in our system. Based on our research on multimodal emotion analysis, we decided to mention the two mode of emotion in our project goals. So it was a necessary to approach these subsystems seperately while we planned the iteration.

As shown in our project timeline, we combined SEMMA phases with SDLC phases in our project timeline. The SEMMA phase are not analogous to an SDLC phase, and we've learned that there may exist no standard for such analogy. For example most software engineers and feature engineers tend to write code during the Sample phase while ML engineers may work on something else. For later iterations, we are going to follow the SEMMA phases more rigidly as we will be working more on the machine learning aspects of the project.

## 1.1 Current Iteration

We only have one Iteration for this part of our final year project. Initially the work was divided between the two part. Use case one was primarily developed by Ibrahim and designed by Faith. Use-case two's main two sub-components were developed and tested independently and later on we re-integrated our work into a single codebase.

Table 1.1: Iteration Plan

<b>FYP</b>	<b>Iteration</b>	<b>Use Cases</b>	<b>Proposed Start Date</b>	<b>Proposed End Date</b>
1	1	1,2	11/03/19	09/04/19
2	2,3	2,3,4	22/08/19	18/11/19

## 1.2 Future Iterations

Iteration two consists of the following use cases. We have written them in the *user story* form instead of traditional UML use-case diagram, for better clarity.

1. Admin can manage the Sample, in order to detect and reduce the possible mistakes they make while data collection.
2. Admin can manage the Movies, so that the studio will be able evaluate their movie. From a developer's perspective, a wider library of movies can be used to create more datasets and that means a larger dataset can be used to develop and evaluate the system.
3. Admin can analyse the movie, so that they can get a score. This is use case 2 in our current iteration. We will be working on this use-case on every iteration.

Some of the use-cases we mentioned is currently possible without using our software. For example, *Admin* can manage the Samples or Movies directly by using the filesystem and editing metadata files, as we did for the first iteration. Iteration 3 will also continue use case two "Admin analyse sample", but we are going to really expand our our work on the machine learning algorithms. All tests for this iteration should be a global accuracy and performance tests, that we can use to evaluate the entire system.

Table 1.2: Future Iterations

Iteration	Use Cases <sup>1</sup>
2	3, 4
3	4, 5

---

<sup>3</sup>Future plans are also described further under the conclusion chapter.

# Chapter 2: Iteration 1

## 2.1 Introduction

Like we mentioned before this iteration implements only two use-cases, but it does not mean the amount of tasks we have is smaller than iteration 2, which we have planned that involves more use-cases. Our plan for this iteration was changed throughout the development, project. Initially we had 2 iterations and considered adding more use-cases that manages movies and samples within the system. However once we finished our work breakdown and started to sort out our tasks, we figured out it will be a risk considering all the tasks we have and we might not be able to complete our goals within timespan.

So iteration 1 was expanded to cover most of this semester and we focused on two use-cases from the user-perspective. From the system perspective, we started working on the two main analysis subsystems once our on use-case two commenced.

## 2.2 Purpose

The purpose of this iteration is mainly to make sure our application can record a video and extract the primary data, required for machine learning from the video. This means the two analysis subsystems must be designed tested, and they should be integrated into the system.

By the end of this iteration the system should be able to do the following:

1. **Capture Video:** The admin should be able to record video by using the command line interface and graphical user interface, to create a video file.
2. **Analyse Video:** The admin must be able to create time-series data from an existing recording of a subject.

## 2.3 Context

On a grand scale, by the end of this iteration we will have a codebase and an fixed project structure. It should be easier to navigate and design for our future iterations, since we will

be more familiar with how we are going to work and what exactly needs to be documented. Most importantly, the review of the first iteration is actually going to help us estimate and prioritize the tasks for future iterations because it is going to show how much we overestimate or under-estimate the size of a task.

## 2.4 Schedule of Iteration Workflow

Table 2.1: Schedule of Iterations Workflow

Workflow	Start	End	Duration
<b>1.Capture Video</b>	11/03/19	30/03/19	19 days
Design	11/03/19	16/03/19	
Implement	16/03/19	24/03/19	
Evaluate	22/03/19	28/03/19	
Sample	26/03/19	30/03/19	
<b>2. Analyse Video</b>	18/03/19	09/04/19	22 days
Sample	29/03/19	02/04/19	
Explore/Design	01/04/19	06/04/19	
Implement	18/03/19	09/04/19	
Modify	08/04/19	09/04/19	
Test	08/04/19	09/04/19	

## 2.5 Iteration Schedule Breakdown

Table 2.2: Schedule of Tasks

Use case	Task Name	Start	Finish	Est Time
1	<b>1.1 Design (Explore)</b>	11/03/2019	16/03/2019	5
	1.1.1 Class Diagram			
	1.1.2 Expanded Use-Case			
	1.1.3 Wireframes			
1	<b>1.2 Implement</b>	16/03/19	24/03/19	8

Continued on next page

Continued from previous page

Use case	Task Name	Start	Finish	Est Time
	1.2.1 Setup github environment 1.2.2 Install OpenCV 1.2.3 Capture video in a folder 1.2.4 CLI to capture video			
1	<b>1.3 Evaluate</b> 1.3.1 AUT capture 1.3.2 AUT CLI	22/03/19	28/03/19	6
1	<b>1.4 Sample</b> 1.4.1 Script to collect training dataset 1.4.2 Collect Test Dataset 1 1.4.3 Write script to load test dataset	26/03/19	30/03/19	4
2	<b>2.1 Sample</b> 2.1.1 Collect Test Dataset 2 2.1.1 Download FER dataset 2.1.2 Script to load FER dataset	29/03/19	02/04/19	4
2	<b>2.2 Explore/Design</b> 2.2.1 Split and label Test Dataset 2 2.2.2 Pipeline Diagram 2.2.3 Interaction Diagram (SSD) 2.2.4 Design Class Diagram 2.2.5 Wireframe	01/04/19	06/04/19	5
2	<b>Implement</b> 2.3.1 Setup FER library 2.3.2 Setup OpenPose Library 2.3.3 Generate FER time series 2.3.4 Generate Pose time series 2.3.5 Integrate FER analysis to UI 2.3.6 Integrate BEGR analysis to UI 2.3.7 Analyse Video CLI	18/03/19	09/04/19	22
2	<b>2.4 Modify</b> 2.4.1 Define combined time-series file format	07/04/19	09/04/19	2

Continued on next page

Continued from previous page

Use case	Task Name	Start	Finish	Est Time
	2.4.2 Code to Visualise Time Series 2.4.3 Refactor code			
2	<b>2.5 Evaluate</b> 2.5.1 Write automated unit test 2.5.2 FER Accuracy test 2.5.3 Pose accuracy test	07/04/19	09/04/19	2

## 2.6 Resource Summary

Resources used for this project

Table 2.3: Hardware Resources used for iteration

No	Hardware	Usage	Qty
1	Laptops with webcams	Testing and Developement	2
2	HD Webcam	Capture sample and testing	1
3	Microsoft Azure Cloud VM	To speed up analysis for testing purpose	1

Software Resources used for this project

Table 2.4: Software Resources used for iteration

No	Software	Usage	Qty
1	Python 3 (conda distribution)	Programming language and runtime environment	1
2	Tensorflow	A lower level machine learning API for python which interacts with hardware, used to train, load and predict ML/DNN models.	1

Continued on next page

Continued from previous page

No	Software	Usage	Qty
3	Keras	A higher level python library which can be used as an abstraction over tensorflow, or other lower level API. Like tensorflow optimized for deep deep learning and general ML tasks, but more simple to use.	1
4	Click	Python library used to create the command line interface	1
5	wxPython	Python implementation of the popular wx widgets toolkit. Used to create our GUI	1
6	pytube	Enables python script to interact with youtube. We use this in data collection scrips and also by scripts to download the test case movies.	1
7	OpenCV python	Python API for OpenCV library. OpenCV is the standard toolkit used to manipulate images and computer vision tasks. We use it for variety of purpose including reading from webcam and performing image manipulations.	1

## 2.7 Evaluation Criterea

The criteria for evaluation of the whole system is divided into three parts. We intend to test the software

1. FER Evaluation
2. BEGR Evaluation

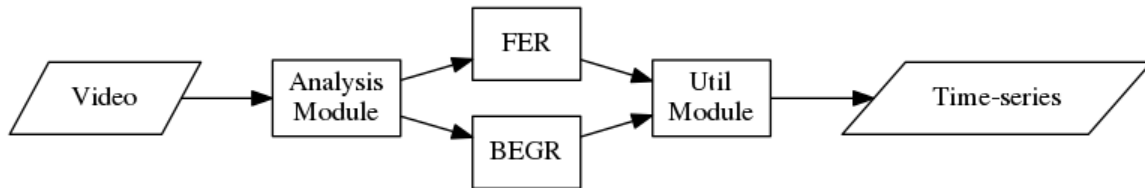
The approach to test the *Software 1.0* parts and the *Software 2.0* (machine learning) parts of the system are very different. The software engineering aspects and machine learning parts of the code must be tested using very serpe



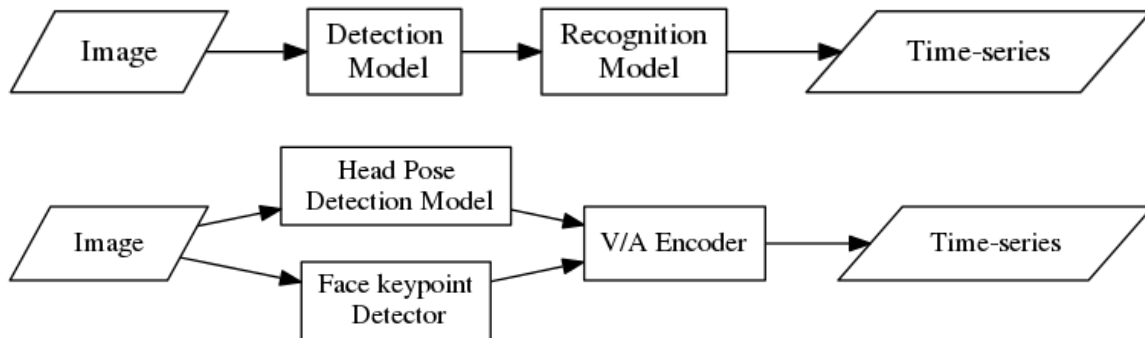
In this iteration we do not evaluate the speed and accuracy performance of the entire pipeline, from the beginning, something we intend to do in all the future iterations.

## 2.8 Analysis and Design Artefacts

### 2.8.1 Pipeline diagrams



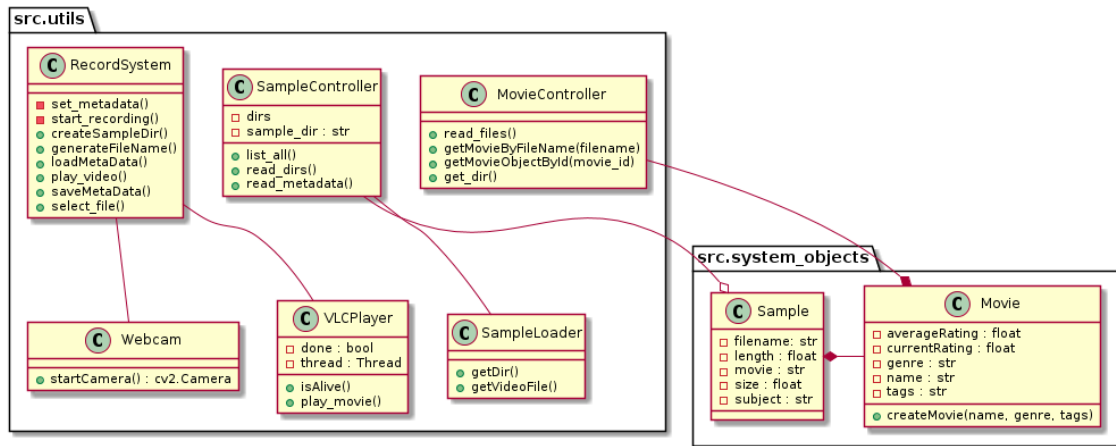
Even though it is possible to run *FER* and *BEGR* simultaneous and in parallel by design, during the implementation phase we realised our development systems are not capable of running the subsystems in parallel without running out of memory. Therefore, we decided that the video will pass through the analysis pipeline twice to extract the time-series data.



The valence arousal encoder is currently not built, instead we use a hard coded module as a replacement. Currently all the V/A encoder returns is a 1 dimensional value that we call 'interest'. If the subject is facing the camera the algorithm returns a higher value, otherwise we return a lower value.

### 2.8.2 Class Diagrams

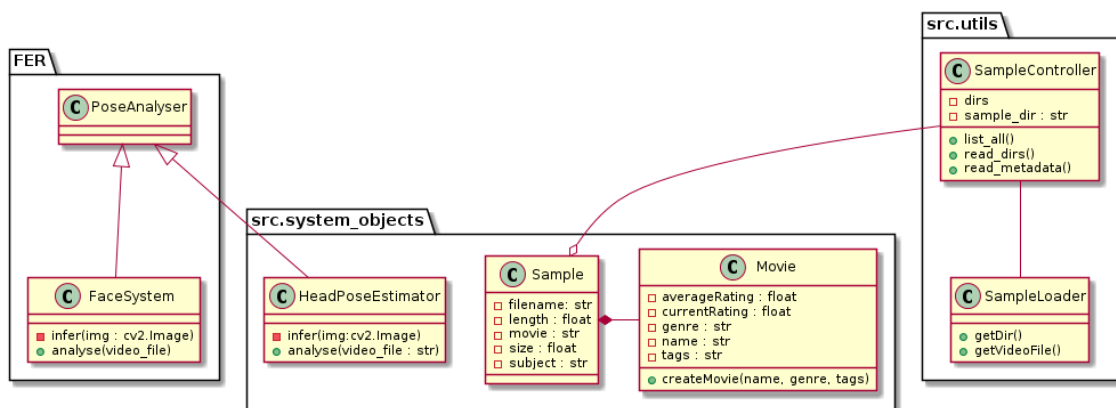
The design class diagram shown here has been updated to represent the actual implementation faithfully. The actual deliverables we made before the change can be found in the design doc folder of along with the source code. By glancing at the class diagram we can say that the structure is not consistent, since we followed the agile principle of "document what you need", the requirements were kept brief before we implemented each work item.



The classes `RecordSystem`, `SampleController` and `MovieController` are respectively used by the User interface code for the use case. We do not have a presentation layer abstraction between the actual user interface code and the *Controller* classes.

One design principle we tried to follow loosely is the Single-Responsible principle, as you can see from the diagram. A class should have one responsibility, but we did not adhere strictly to it.

Webcam module helps setting up the camera.



Note since the current BEGR subsystem's full body pose recognizer is too slow, we replaced it with another model only measuring head pose and renamed the package *begr* as *head\_pose*. Package *head\_pose* is just a placeholder for *begr* until we find a better replacement.

Also, the fact that BEGR code is not in a separate package and FER code is in a separate package, is because of a miscommunication and oversight between the two programmers who worked on this during the development and we decided not to refactor the mistake until, next iteration due to lack of time. BEGR should be refactored into its own package by in the future.

# Chapter 3: Iteration I: Implementation and Testing

Our manual tests corresponds to the use-cases, for demonstration purpose we isolate the components associated with the use-case. We do this manual process only for the sake of documentation, while most other smaller systems are tested with the help of automated python's automated unit testing framework.

## 3.1 Implementation

Our application is mostly written in python with the exception of few OS specific shell scripts used to setup and configure. For development we used conda distribution of python, along with its package and environment manager. The main reason we chose conda was it's ease of use, when installing certain packages such as tensorflow for various platforms compared to python's default package manager.

```
~/affective_movie_evaluator/$ conda create --name affmem_env \  
    pip tensorflow python=3.5  
~/affective_movie_evaluator/$ conda activate affmem_env  
~/affective_movie_evaluator/$ conda activate  
(affmem_env) ~/affective_movie_evaluator/$
```

The shell command snippet above shows an example scenario of creating a conda environment and activating the environment for the project. Right now we recommend all *Admins* to work using conda environment to avoid common multiple dependency version issues users face when they use the global python environment.

All of the data was stored directly on disk drive, and the system identifies the objects with the help of metadata written in JSON format. Since we are not expecting search functions, that will traverse through the disk drive and most machine learning algorithms either loads datasets in batches or in entirety we do not think a database is necessary. A search with the current system will perform linearly  $O(n)$  and an insertion will cost constant time.

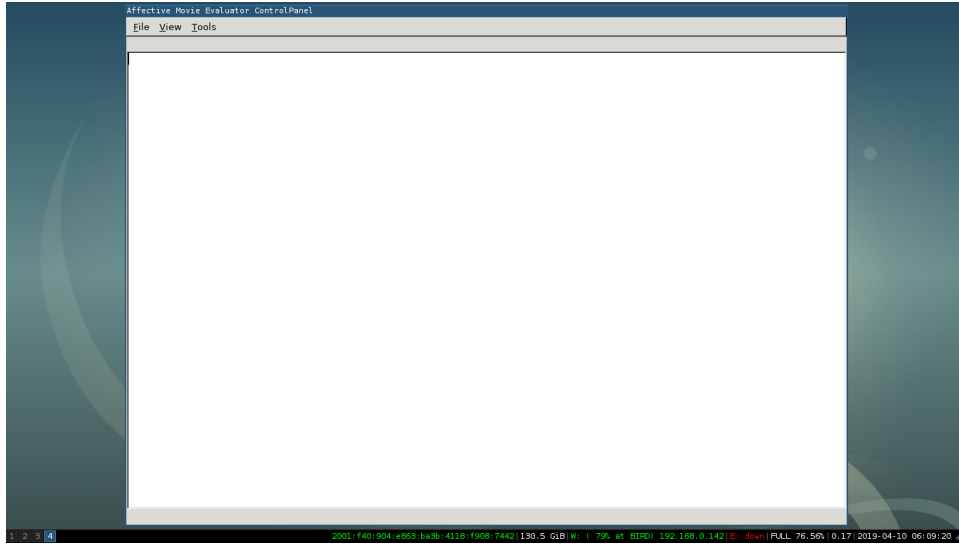


Figure 3.1: The GUI Application

Figure 3.1 shows the initial screen the admin will see once they start the application. The empty text area shown in the screenshot is going to display error messages for the admin, and was also used as the console log by us while we were developing the application.

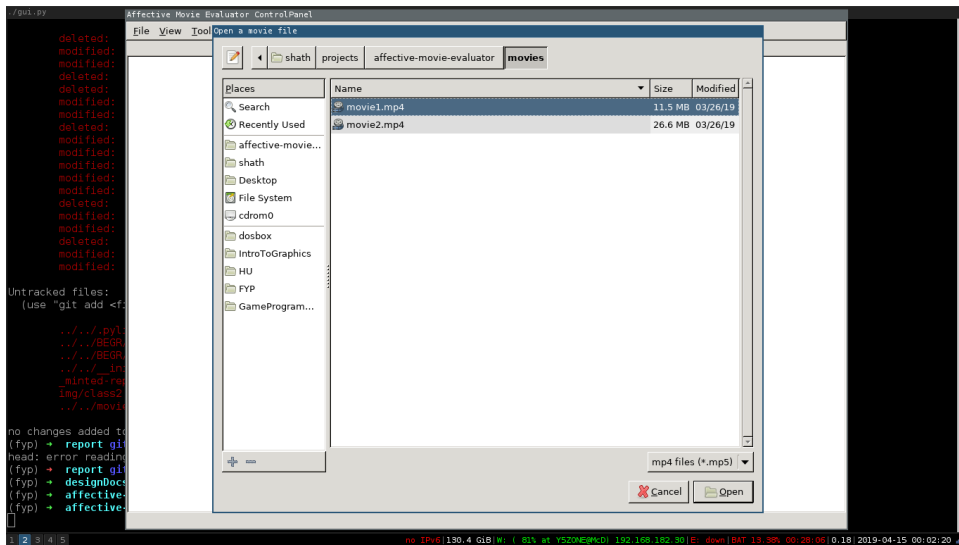


Figure 3.2: Record Sample: 1. Select the movie file

If the *Admin* wants to record a new sample, they will have to select a movie stored from the disk drives. The movie files are stored in a the 'movies' directory as shown in Figure 3.2, along with the metadata file which describes each movie.

Our overall strategy for implementation can be describe as a test driven bottom up approach. For modules that uses machine learning we have a different approach, instead of writing tests for accuracy or simple integration we write a procedural test file, then convert into Object Oriented code which is compatible with our Analyse interface.

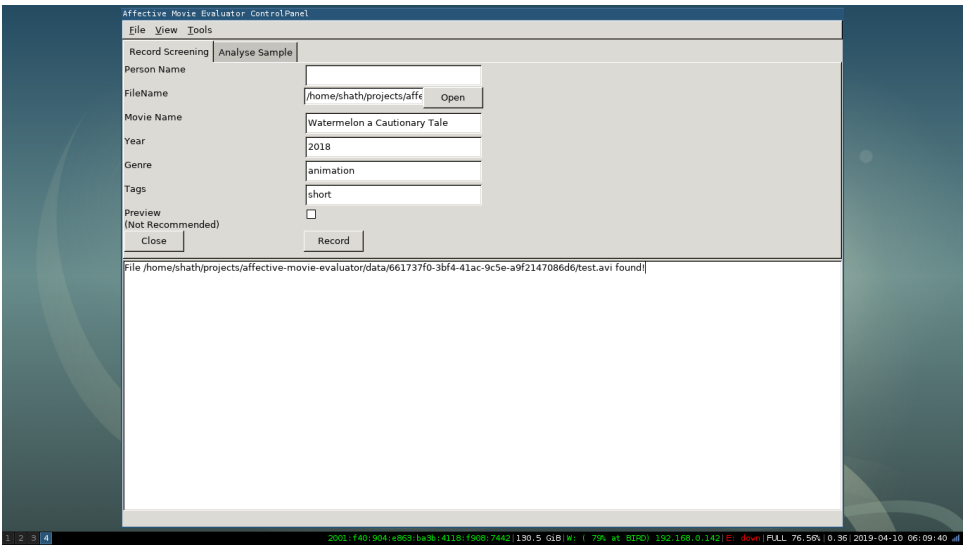


Figure 3.3: Record Sample: 2. Fill in the blanks

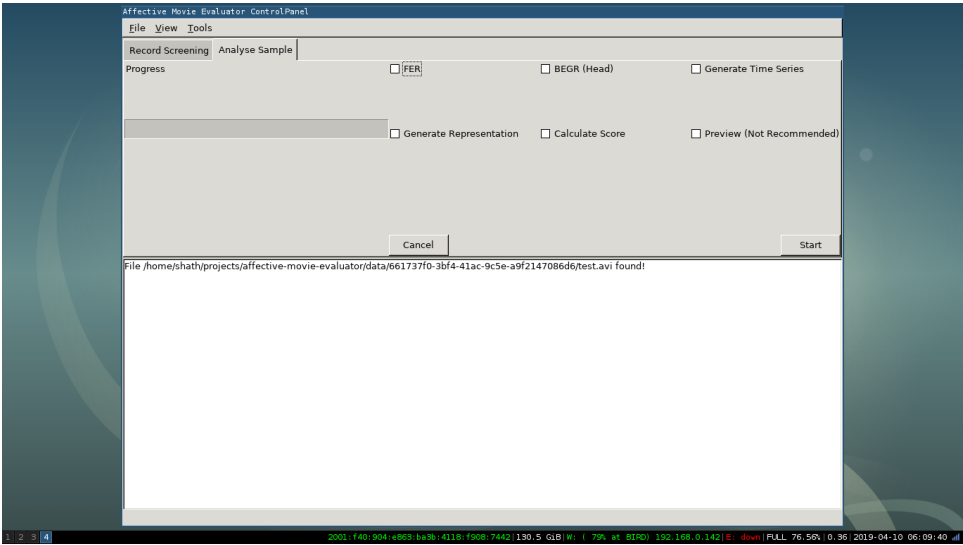


Figure 3.4: Analyse Screen

## 3.2 User Manual

The user manual is supposed to be at the project directory folder, in markdown format. We do intent to write a more detailed user documentation on the final iteration, on how to operate and trouble shoot. The user documentation right now consists of how to setup and install, then the only two use cases. The system does have extra features that need to be documented so that the user can effectively operate the system.

In order to use the system the user must, setup python and installed required libraries listed in requirements.txt then run the scripts to download and install the models. Once the models are installed, the user can run the `"/gui.py"` to open the user interface. A brief overview of how it is used is described in the implementation section.

## 3.3 Testing

Table 3.1: Table of Strategy

No.	Strategy
1	Create the smallest possible class
2	Write the unit test for the class and improve the class until it passes all the basic test conditions
3	Write the user interface code

Table 3.2: Table of Strategy for Models

No.	Strategy
1	Write driver code for used for testing purpose
2	Write the class which can import the model and perform predictions
3	Keep on improving the class until the realtime tests pass

We did not use the main dataset and movie diretory for testing purpose, but instead created mock data for that purpose. For example, in order to test the SampleLoader which loads the Sample for other module given the parameter, we actually created a test directory without actual samples but metadata files.

Table 3.3: Test Plan

Type of testing	Approach	Subject
Automated Unit Testing	White	Unit: RecordSystem
Integration Testing	Black Box	Usecase 1, Usecase 2

Table 3.4: Test Plan

Type of testing	Approach	Use Case	Task
Manual Unit Testing	White	1. Record Video	Webcam Test

All automated unit tests are prefixed with the letter A to distinguish it from manual tests, which are prefixed with the letter M. Manual unit testing requires user to run and observe the results for each test case, and tests may require user input unlike automated unit tests.

## 3.4 Unit Testing

Our goal of unit testing is to test the smallest possible software components within the system separately and manually without where it is possible. For more internal components that mainly interacts with other components, we do not rigidly test all possible fail cases as some fail cases may be triggered by a programming mistakes.

Unit testing was implemented under the task, "Test CLI".

Table 3.5: Automated Unit Tests

#	Module	Package	Usecase
A1	SampleLoader	src.util	1,2
A2	SampleController	src.util	1,2

### 3.4.1 UnitTest A1

Unit Test Plan	
<b>Module Name:</b> SampleLoader	<b>File:</b> ./test/path_test.py
<b>1. Module Overview</b>	
The purpose of the SampleLoader is to aid the SampleController load the Sample files from the directory. In spite of its name the actual responsibility of the class is to resolve the directory path of a sample folder and nothing else.	

Continued on next page

Continued from previous page

Unit Test Plan
<b>1.1 Inpselt to Module</b>
Input 'param1' will passed as argument 1 of the constructor. sys = SampleLoader(param1)
<b>1.2 Output from Module</b>
<b>1.3 Logic Flow</b>
<pre> class SampleLoader(object):     """ SampleLoader is responsible for aiding the load process of individual     sample data from our directory.     """      def __init__(self, filename):         #self._dir = "./data/"+filename+"/"         f = filename         if f[-1] != '/':             f = f + "/"         self._dir = f      def getVideoFile(self):         return self._dir+"test.avi"      def getDir(self):         return self._dir </pre>
<b>2. Test Data</b>

Continued on next page



Continued from previous page

Unit Test Plan
<ul style="list-style-type: none"> <li>• ./data/test/</li> <li>• ./data/test/</li> </ul>
2.1 Positive Test Cases
<ul style="list-style-type: none"> <li>• CASE 1: testGetDir()</li> <li>• CASE 2: testGetVideoFile()</li> <li>• CASE 3: testDirSlash()</li> </ul>
1
2.2 Negative Test Cases
None
2.2 Interface Modules
Python unittest
2.3 Test Tools
Test Tools

```
(fyp) affective-movie-evaluator git:(master) python3 -m unittest tests.path_test.Test
...
```

```
-----
Ran 3 tests in 0.000s
```

```
OK
```

### 3.4.2 UnitTest A2

Table 3.6: Manual Unit Tests

#	Module	Package	Usecase
M1	Webcam	src.utils	1
M2	VLCPlayer	src.utils	1

Table 3.8: Unit Test A2

Unit Test Plan	
Module Name: <u>SampleController</u>	file: <code>"/test/path_test.py"</code>
<b>1. Module Overview</b>	
The purpose of the SampleController is to handle Create, Retrieve, Update and Delete operations of the Sample Objects within the system.	
<b>1.1 Input to Module</b>	
Input 'param1' will passed as argument 1 of the constructor. sys = SampleLoader(param1)	
<b>1.2 Outputs from Module</b>	
expected_data = ['661737f0-3bf4-41ac-9c5e-a9f2147086d6', '1d989665-80ae-4bff-bf59-b5f7691fb3b9']  For every input.  - CASE 1: – assert sys.list_all() == expected_data  Expected data corresponds to valid sample folders manually created with valid, metadata.	
<b>1.3 LogicFlow / Source</b>	
class SampleLoader:  + read_dir() # scans the directories and creates dictionary for search  + list_all() # lists the sample_ids read by the last scan  + read_metadata(id) # retrieves sample metadata given and id	
<b>2. Test Data</b>	
- <code>"/data/test/"</code>	
<b>2.1 Positive Test Cases</b>	
- CASE 1: test1()  Purpose of test1 is to make sure that the list_all() method retrieves only valid Sample folders.	

Continued on next page

Continued from previous page

Unit Test Plan
<b>2.2 Negative Test Cases</b>
- None
<b>3. Interface Modules</b>
requires: none
<b>4. Test Tools</b>
Python

```
(fyp) affective-movie-evaluator git:(master) python3 -m unittest tests.path_test.Test
.
```

-----

Ran 1 test in 0.000s

OK

### 3.4.3 ManualTest M1

Table 3.9: Unit Test M1

Unit Test Plan
Module Name: <u>Webcam</u> file: <code>"./src/device.py"</code>
<b>1. Module Overview</b>
Purpose of the webcam module is to provide a standard camera interface through webcam for other modules.
<b>1.1 Input to Module</b>
- None
<b>1.2 Outputs from Module</b>

Continued on next page

Continued from previous page

## Unit Test Plan



### 1.3 LogicFlow / Source

- Create a new Webcam Object
- call runTest() method
- view the output

### 2. Test Data

Camera input

#### 2.1 Positive Test Cases

- Clear visual image displayed

#### 2.2 Negative Test Cases

- None

### 3. Interface Modules


requires: none

### 4. Test Tools

Python

### 3.4.4 ManualTest M2

Table 3.10: Unit Test M2

Unit Test Plan	
Module Name: <u>BEGR</u>	file: <code>"./src/head_pose.py"</code>
<b>1. Module Overview</b>	
Purpose of the BEGR subsystem is to estimate the upper body pose and measure the 3d keypoints of the face, which are used for emotion analysis.	
<b>1.1 Input to Module</b>	
- Camera feedback	
<b>1.2 Outputs from Module</b>	
	
<b>1.3 LogicFlow / Source</b>	
<ul style="list-style-type: none"> <li>- Create a new BEGR object</li> <li>- Create new Webcam Object</li> <li>- Open Webcam interface</li> <li>- For each frame that pass through the camera, call <code>BEGR.infer()</code>, then <code>BEGR.draw()</code> the inferred data to visualise</li> </ul>	

Continued on next page

Continued from previous page

<b>Unit Test Plan</b>
<b>2. Test Data</b>
- Live camera Feedback containing a face
<b>2.1 Positive Test Cases</b>
- Keypoints detected for clear face - 3-D Pose co-rdinate calculated - Interest value must approximate 1 when the point their head towards camera.
<b>2.2 Negative Test Cases</b>
- No keypoints detected for a clear face - No pose detected for a clear face
<b>3. Interface Modules</b>
- Webcam
<b>4. Test Tools</b>
Python

### 3.5 Iteration Review and Evaluation

The completion of this iteration makes us closer to our primary goal. While some objectives related to accuracy testing and data collection were not complete because of the way we organized the project we successfully completed all the work items prioritised by us for this iteration. We had two meetings as documented in the meetings reports, that were directly concerned with planning the iteration. Prioritising of the tasks and judging the duration for work time was at first very difficult to estimate. Because of that, we managed to over-estimate the duration for most of the items and were not able to properly manage what to do with the slack time.

In terms of usability we have implemented two modes of UI, for the *Admin* to use in order to interact with our system. While UX design was not part of our primary goals, because we expect the *Admin* user to have enough system administration skills to install and setup Unix like software. We instead focus on making sure our command line user interface is similar to existing tools available, so that the user can easily adapt into our system through familiarity. The GUI interface is in contrast designed for basic functionality, and mostly intended to demonstrate the

project for stakeholders. The graphical interface is also less optimized compared to the CLI, for example the preview mode of the analyse use-case may look slower on GUI if the progress bar is enabled, because the wx-widget library function are required to run on the main thread and some of the OpenCV's own functionalities require to be run on the main application thread. In contrast, the CLI can support multithreading without any issues and run OpenCV's gui functionalities on the main thread if the user enables live preview function.

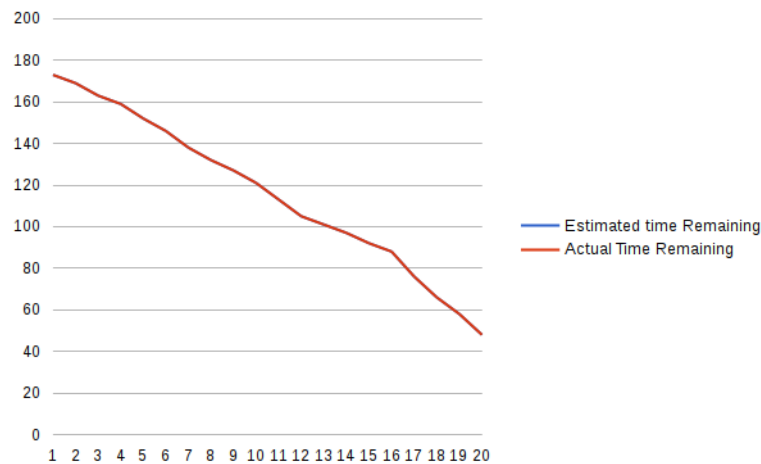


Figure 3.5: Burn down chart

On Overall, we do consider the achievement of the whole iteration as a success, but the planning and execution needs to be in future iterations. Looking at the individual work items and tasks, tasks related to designing and documentation was actually with more time than the programming tasks. If you look at the burdown chart (Figure and our evaluation) and the statistics shown, below it shows that we less time on testing and implementation per day than the other tasks.

Table 3.11: Statistics

Total Per Day	177
Faith	93
Ibrahim	70
Actual Time Spent	163
Estimated time Remaining	2530
Actual Time Remaining	2526

## Part V

## Conclusion



# Chapter 1: Current Status

So far we have been able to complete the most vital objectives. One of the failure of our FYP is that we prioritized system development work items such as GUI development over data collection and machine learning tasks. This is because we wanted to demonstrate the application as soon as possible. Throughout the development we were able to resolve most of the issues we faced with compromises which affects changes in the project. For example we prioritised speed, by often choosing less accurate model over models that can run in considerable time. This means often the models extract less features than we originally intended, and so far we cannot estimate how much of a difference that would make on our entire overall system.

So far, there are no clear results about how successful the project is going to be in terms of accuracy. Faith is planning to focus on improving the FER subsystem, and we might be able to detect and improve problems with the machine learning models in the pipeline once we the system is able to run the accuracy tests. Ibrahim will focus on building the BEGR and representation model, while both team members will contribute in other system development work items.

One of the deep learning model we used for was Expression Net (Chang et al., 2018), but we chose to use a faster and smaller pose estimation model for this iteration. If possible, what we plan to do is use similar techniques as used in a work we studied (Saha et al., 2018). Instead of analysing per frame we can analyse per key frame, try to detect changes in video on a static background using computer vision techniques, which are faster and apply the technique to detect changes in keyframes. Doing this might actually let us use Expression Net shown in Figure 1.1 because it will now only have to infer only a few frames from the whole video. Expression net, which produces a very accurate 3d vertices of the face which can be used as a keypoint more effectively than our current approach.

We also plan to possibly experiment integrating our own deep learning model into the pipeline, possibly to build representation for the time-series. Currently, our strategy for representation algorithm is to visualize and cluster the time series representation using a unsupervised ML algorithm before we sent for regression/classification.

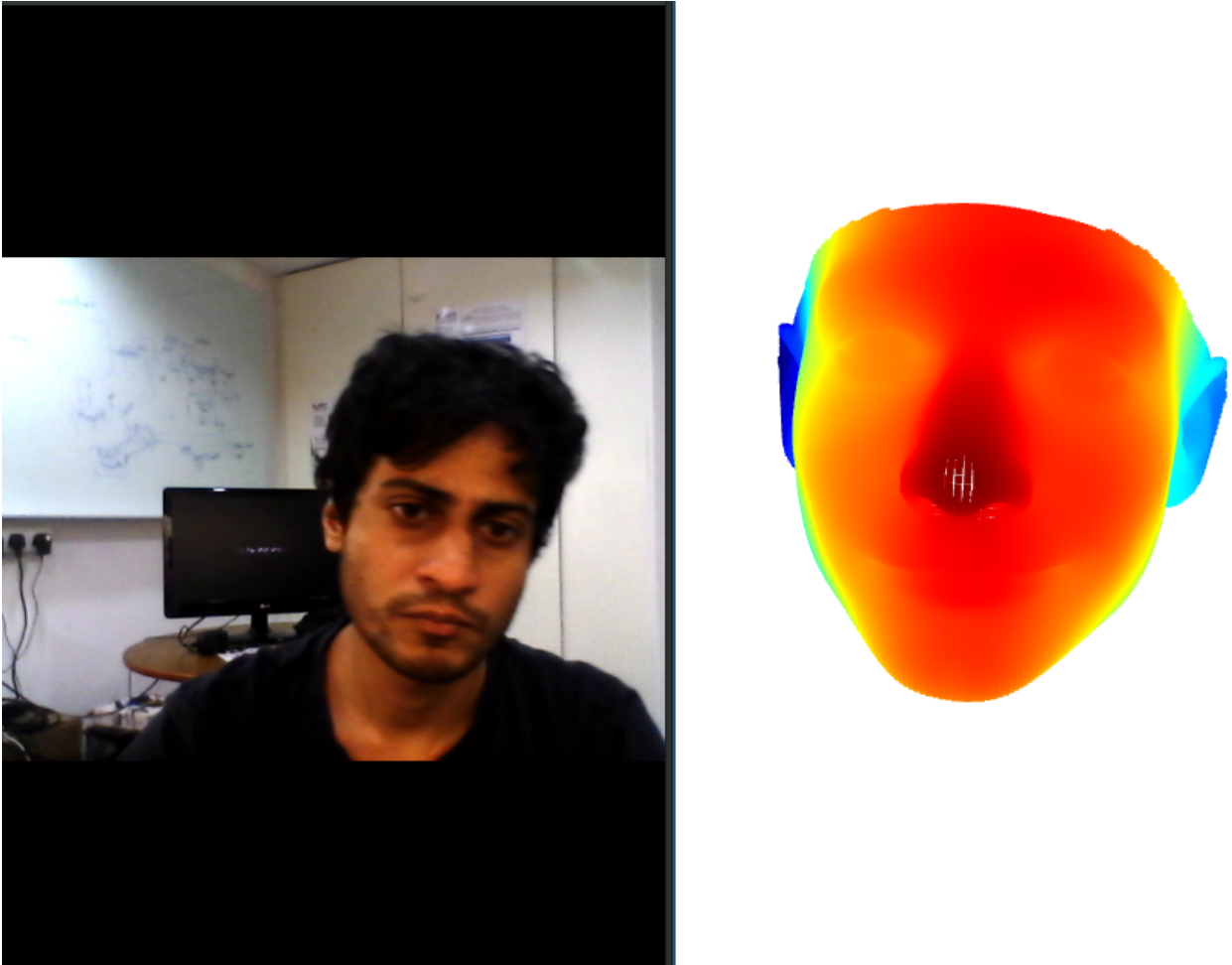


Figure 1.1: Expression Net

## Part VI

### Appendix A - Gantt Charts

<b>AffectiveMEM</b>		Apr 15, 2019
<b>HELP</b>		<a href="#">http://</a>
Project manager		
Project dates	Mar 12, 2019 - Apr 10, 2019	
Completion	99%	
Tasks	38	
Resources	2	

**AffectiveMEM**

Apr 15, 2019

**Tasks**

2

Name	End date	Begin date
Capture Video	3/29/19	3/12/19
Design	3/15/19	3/12/19
1.1.1 Class Diagram	3/13/19	3/12/19
1.1.3 Wireframe	3/12/19	3/12/19
1.1.2 Expanded Use-Case	3/15/19	3/14/19
Unit Test Capture	3/27/19	3/16/19
Unit Test Capture	3/27/19	3/23/19
Unit Test CLI	3/27/19	3/23/19
Implement	3/25/19	3/16/19
Setup githubInstall OpenCV Unit Test CLI CLI Capture Video Unit Test Capture	3/16/19	3/16/19
Setup github b		
Install OpenCV	3/18/19	3/18/19
Capture Video	3/25/19	3/18/19
CLI	3/23/19	3/18/19
Sample	3/29/19	3/26/19
Script to record dataset	3/29/19	3/26/19
Collect dataset 1	3/28/19	3/27/19
Script to load dataset	3/28/19	3/28/19
Analyse Video	4/9/19	3/18/19
Sample	4/1/19	3/28/19
Collect dataset 2	4/1/19	3/29/19
Script to load FER dataset	3/29/19	3/28/19
Analysis	4/5/19	4/1/19
SSD	4/5/19	4/1/19
Design Class Diagram	4/5/19	4/1/19
Wireframe	4/2/19	4/1/19
Implement	4/9/19	3/18/19
Setup FER Library	3/20/19	3/18/19

**AffectiveMEM**

Apr 15, 2019

**Tasks**

3

Name	End date	Begin date
Setup OpenPose	3/21/19	3/18/19
Generate FER time-series	4/3/19	4/2/19
Generate Pose TS	4/4/19	4/3/19
Integrate BEGR	4/6/19	4/5/19
Integrate FER	4/6/19	4/5/19
Analyse Video CLI	4/7/19	4/6/19
create combined time-series	4/9/19	4/7/19
Evaluate	4/9/19	4/8/19
Unit Test	4/8/19	4/8/19
Code to Visualize TS	4/9/19	4/8/19
	4/8/19	4/8/19

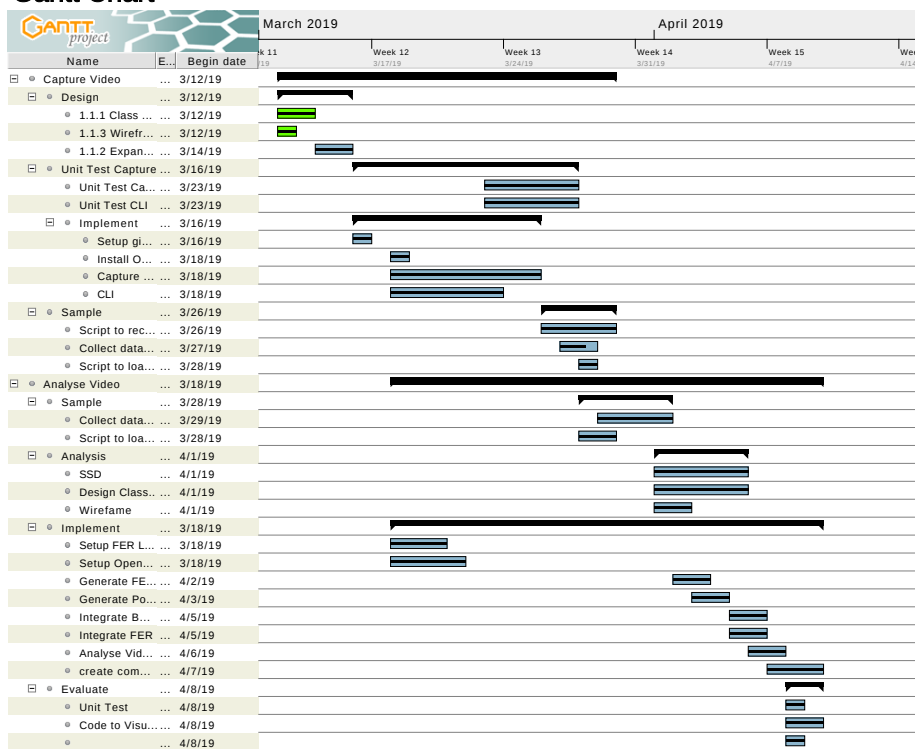
AffectiveMEM		Apr 15, 2019
Resources		4
Name	Default role	
Ibrahim	developer	
Faith	developer	

## AffectiveMEM

Apr 15, 2019

## Gantt Chart

5





**Apr 15, 2019**

## 6



## **Part VII**

### **Appendix B - Monthly Progress Reports**

## Part VIII

### Appendix C - Meeting Reports

## Part IX

### Appendix D - Minutes

## Part X

### Appendix E - Evaluation Report



## **Affective Movie Evaluator**

### **PROJECT EVALUATION**

**Ibrahim Mohamed Shaatha**

**Submitted to the**

**FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY  
(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)**

**In partial fulfilment of the requirements  
for the degree of**

**HUBSC2**

# HELP UNIVERSITY

## JANUARY - APRIL 2019

### **Project Evaluation**

#### **1 Introduction**

I am working on a system that is going to analyse an audience's reaction from a video and try to predict whether they are enjoying a movie they are watching. We spent the first half of our semester trying to analyse the problem, and design the system and we started working on the implementation on 11<sup>th</sup> of March.

So far, we were able to successfully complete most vital tasks we set out and planned for this iteration, though the goals and objectives for the iteration were kept as simple as possible to give us a timeframe to learn. The completion of this portion of the project means that, we are halfway towards achieving our primary goal, and also that we have completed roughly half of the project.

#### **2 Evaluation of Objectives**

In our project proposal we set out 5 different objectives. Our main goal has always been to create an application that can analyse a video file and give it a score that represents audiences sentiment for the movie. In order to achieve these goals we set out objectives in our proposal, which we still believe are simple, achievable, though in retrospect we should have re-written the objectives in a more clear and concise terms for clarity.

##### **2.1. To create metrics by which the system gives score.**

In order for our idea to work, we need to define how the system is going to rate the audience reaction. In other words we need to establish what exactly is our system going to output, once the analysis is complete.

We considered categorical rating and numerical ratings, while we were creating this objectives. What we didn't give much thought was about the fact, that this metrics is going to affect how the machine learning system was going to work. For example, if the metrics is categorical like A, B, and C the most suitable machine learning algorithm we choose for analysis is a "Classification" type of algorithm. If it is a continuous score like a percentage, we can treat the whole thing as a regression problem.

Building upon the existing works, we have considered to use a continuous scale between 0-1, but we have not finalised the metrics we will use and this objective is still not considered achieved.

## **2.2. To research on the requirements of the dataset and methodologies before March.**

Before we even started requirement analysis, we actually thought researchin on existing projects, datasets and methodologies will give us a good idea of how we are going to proceed. While it was beneficial for the project, in the end it took us back by one week, because we were getting used to how we are going to manage the project.

This object was achieved altogether, once we completed our literature review.

## **2.3. To Create and Collect Datasets for testing and developement**

To clarify all datasets with the exception of “Video Recordings” datasets, are supposed to be used in all iteration for testing. Information extracted from our “Video Recordings” will be used to train and develop our movie evaluation ML model.

Unfortunately, this objective is considered partially complete. It was really supposed to be completed by this iteration but we had several issue collecting the dataset, because most of the datasets we collected did not meet the testing requirements . Here is the status:

1. Facial Dataset with labelled Emotion: Pending
2. Pose Dataset with body pose labelled: Pending
3. Video Recordings of audience reacting to Movie: Completed

We are able to complete the objectives we set for dataset “video recordings” of audience reacting, because our criterea for this iteration was to record atleast 3 subjects watching more than two movies. We were able to record more than that.

One of the reason for failing to achieve this objective completely is that we actually put a really low priority on all data collection and during our planning stage we let the system development tasks take precedance over them.

## **2.4 To determine accuracy of the project by comparing its output to existing scoring systems.**

The purpose of this objective is to let us know that our system is working as we planned. A better accuracy score means the project is more successful than we anticipated, mediocre score means that we may still have to work on it until we get better results.

We did not set out to complete this objective for FYP 1. It is possible the project might meet this requirement be completed after the planned third iteration.

## **2.5 To produce accuracy reports for the analysis subsystems**

We have to produce the accuracy report for the two pre-trained models, FER and BEGR. We need to do this in order to evaluate the accuracy of the overall pipeline and choose better trained models if possible.



Accuracy report for FER was initially planned, but we decided to move it to the iteration two, because of lack of space and priority over the system development task. Our priority was to build a “Demo” app for the presentation, and we allocated most of our resources on that task.

### **3 Further Improvement**

We mixed SDLC phases and SEMMA terminologies, for iteration 1. Iteration 1 is mostly system development, while we expect iteration 2 to be more balanced and the final iteration to be mostly focused on the machine learning aspect. Therefore, we will likely not use SDLC names for phases later on during our project.

During analysis, we had difficulties measuring the time it will take for us to work on tasks and we actually ended up over-estimating time, except for three tasks, which gave us a lot of slack time. I believe a better approach would have been to consider the work items and measure the size and time relatively instead of by numbers. Then spend more time figuring out how long we actually take for the simplest task. Then try to estimate by comparing from there on.

### **4 Future Work**

In the next few iteration, I plan to spend more time on the Machine Learning algorithm itself. There are two main parts of the system, I plan to improve. One is the BEGR system needs to be trained to possibly give an emotion on a valence arousal scale. The other is that a movie evaluation system must be developed and trained by us. The system already have undocumented and extra features such as the ability to download movies from YouTube, I do also plan to add more interesting features as such so that the application may be a tool that holds the user within the systems as much as possible.

### **5 Conclusion**

Working on and developing this project gave me knowledge and experience on two different fronts. One is the python based software development and the other is the use of ML and Computer Vision technologies, such as pose estimation models. I learned more details on how deep learning models can be integrated with software designed with object oriented tools.



**AFFECTIVE MOVIE EVALUATOR  
PROJECT EVALUATION**

**B1301511 ELEBE FAITH ALFRED**

**Submitted to the**

**FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY  
(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)**

**In partial fulfilment of the requirements  
for the degree of**

**BACHELOR OF INFORMATION TECHNOLOGY (HONS)**

**HELP UNIVERSITY  
JANUARY - APRIL 2019**

## **Project Evaluation**

### **1 Introduction**

Sentiment Analysis is already widely used by different companies to gauge consumer mood towards their product or brand in the digital world. However, in the offline world, users are also interacting with the brands and products in retail stores, showrooms, etc., and solutions to measure users' reactions automatically under such settings has remained a challenging task. Emotion detection from facial expressions and body postures using AI can be a viable alternative to automatically measure consumers' engagement with their content and brands.

Affective Movie Evaluator is an automated system that helps Movie production studios to determine the success rate of their movies by analysing the emotions of an audience while watching the movie through their facial expressions and body postures. With a review of the project objectives and the quality of work done, I can say that the project is partially completed.

### **2 Evaluation of Objectives**

Below are the objectives listed in the project proposal:

**Define the metrics by which the system is going to give the score** (e.g.: value between 0.0-1.0 or a discrete grading system, like A+, A, B). This objective is not achieved. This is because it is to be achieved in iteration 2 of this project. At this phase we are completing iteration 1.

**Research the requirements of the dataset, and methodologies before week 3.**

This objective is successfully achieved. I learnt about a methodology slightly different from the traditional type used in most software engineering projects and I also learnt about the different libraries of dataset available today.

**Find or create the required test datasets for in order to complete objective 5**

**a. Facial dataset with labelled emotion:**

This task is not achieved. This is due to not having enough time to request for a more accurate and recent dataset. The dataset library we were able to gain access to was too old and the accuracy of the data was low as well.

**b. Dataset containing body pose labelled:**

This task is successfully achieved. Pose dataset is not freely available compared to facial dataset, so my project partner created pose dataset for the implementation of this project since this task is handled by him.

**c. Video recordings of audiences reacting to movies.**

This task is not achieved. This is to be accomplished in iteration 2 of this project.

**Determine accuracy of the project by comparing its output to existing scoring systems.**

This task is not achieved. This is because it is to be done in iteration 2, we are still in iteration 1.

**Produce accuracy reports with established methodologies for:**

- a. Emotion recognition
- b. Body pose/action estimation

This objective is not achieved. This is because it is to be done in iteration 2, we are still in iteration 1.

**Problems faced during project development**

**Project management**

We created our Work Item List, Work Breakdown Structure (WBS), Gantt Chart, Burn Down Chart etc. Before you can create a Work, Item List which will further enable you to create a Gantt Chart, you must have an idea of what is required in order to implement the project. This phase was difficult for me because I did not have a knowledge of what is required in projects like this. I had to read a lot of articles on existing projects that have been accomplished, ask questions during

meetings with my supervisor and get a lot of insight from my project partner who has an extensive knowledge in AI projects.

### **Time management**

Working on this project and trying to fulfil the criteria for other subjects I am taking this semester is difficult. There are few tasks I couldn't complete at the stipulated time, but in order to stop extending a task timeline which affects the completion of other tasks, I had to make sure I complete each task within the period allocated for it. Accomplishing this involves a lot of late nights and help from my project partner when I am stuck and need guidance.

### **Knowledge of Programming Language**

The programming language used in this project is python. I had to study python again from Data camp because it's been a long time since I worked on any project that requires the use of this programming knowledge.

### **Finding a current/recent facial dataset**

It was not easy to find and access a facial dataset that is recent and free at the same time. Most of the ones I found requires us to write and ask for permission before we could access it. But with continuous search we were finally able to find a recent and free dataset.

### **Communication Problem**

I had a minor communication problem with my partner, he was not always with his phone sometimes, but the problem was resolved quite early. If I sent him a message through social media and there was response, I will take it upon myself to give a call until he replies.

### **Skills in Analysis and Design**

Designing this system was a little challenging for me because it is both a traditional project and a machine learning project at the same time. I sat together with my project partner to create the Work Breakdown Structure, identify the tasks and

prioritise the tasks. One of the most challenging tasks was to decide on how long it will take to complete a task.

### **The interface code for external media player is not multiplatform.**

To solve this problem, we had to modify the existing code so the media file can play on both of our operating system.

### **3 Further Improvement**

Another issue is Emotion detection for facial expression is not accurate. The confidence level or probability level of the system fluctuates when analyzing peoples faces to detect the emotion they are expressing. One way to solve this problem is to have a large and more accurate facial dataset either by requesting for it from the available free libraries or by creating a dataset by myself.

One of the issues I faced during the design phase is the inability of the system to analyse more than a single audience member at a time. If more than one person is in a photo or video, the system can only detect the face of a single person.

### **4 Future Work**

I plan on working on the accuracy of the Facial Expression recognition because this is an important aspect of the project. The system must analyse emotions accurately as this is what will be used in scoring a movie, therefore determining if the movie will be successful or not. Another target of mine is for the system to be able to identify/detect more than one face at a time. The aim of our project is for the system to evaluate an audience of at most four people watching a movie, so it is very important for this aspect of the system to be implemented. For the body pose analysis, my partner plans to make sure the system can analyse a full body pose.

### **5 Conclusion**

The implementation of this project has enabled me to gain knowledge in the aspect related to AI. I have learnt about other methodologies that can be used in implementing a software project. In a quest to understand what the output of this

system will be and how to implement it, I have read numerous articles, books and published materials related to psychology and human emotions. The knowledge gained from these educative materials is not only limited to this project, but it's also helpful when interacting with individuals in real world. Through the implementation of this project, I have revised my knowledge of python programming, I understand what it means to train a model and I have also learnt that most classes that exist in object oriented does not have to exist in real life. In this project I have been able to develop a system that can analyse the different facial expressions of humans depicting the six basic human emotions – anger, sadness, surprise, happy, disgusted and fear.

## Part XI

### Bibliography



# Glossary

**FER** Facial Expression Recognition.. 6

# References

- Chang, F.-J., Tran, A., Hassner, T., Masi, I., Nevatia, R., & Medioni, G. (2018). Expnet: Landmark-free, deep, 3d facial expressions. In *13th ieee conference on automatic face and gesture recognition*.
- Deng, Z., Navarathna, R., Carr, P., Mandt, S., Yue, Y., Matthews, I., & Mori, G. (2017, Jul). Factorized variational autoencoders for modeling audience reactions to movies. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Retrieved from <http://dx.doi.org/10.1109/cvpr.2017.637> doi: 10.1109/cvpr.2017.637
- Ratner, A., Hancock, B., Dunnmon, J., Goldman, R., & Ré, C. (2018). Snorkel metal: Weak supervision for multi-task learning. In *Proceedings of the second workshop on data management for end-to-end machine learning* (p. 3).
- Saha, S., Navarathna, R., Helming, L., & Weber, R. M. (2018). Unsupervised deep representations for learning audience facial behaviors. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 1132–1137).
- Wu, T.-L., Wang, H.-K., Ho, C.-C., Lin, Y.-P., Hu, T.-T., Weng, M.-F., ... others (2008). Interactive content presentation based on expressed emotion and physiological feedback. In *Proceedings of the 16th acm international conference on multimedia* (pp. 1009–1010).
- Zhang, S., Tian, Q., Jiang, S., Huang, Q., & Gao, W. (2008, Jun). Affective mvt analysis based on arousal and valence features. *2008 IEEE International Conference on Multimedia and Expo*. Retrieved from <http://dx.doi.org/10.1109/icme.2008.4607698> doi: 10.1109/icme.2008.4607698