**PEITHO PROJECT**

A Capstone Project

by

**Finn Jensen and Anthony Calarese**

Submitted to the Faculty of the

Department of Software Technology

in partial fulfillment of the requirements for the degree of

**BACHELOR OF SCIENCE**

**in**

**COMPUTER SCIENCE and INNOVATION**

Champlain College 2019-2020

.

ABSTRACT

Crowds are not easy to monitor, especially when the presenter to that crowd is attempting to observe the overall emotional state of the crowd. The Peitho Project is aimed to assist presenters such as politicians and teachers in identifying the overall Happiness in a set crowd of users. That way heuristics on speeches can be analyzed and evaluated as either being significantly beneficial at certain focal points; or, points at which the crowd reacts negatively on the subject matter.

.

Table of Contents



.

List of Tables



.

# CHAPTER 1: INTRODUCTION

### *Problem Statement*

Crowds are hard to read in terms of gathering overall happiness. It is time consuming and resource heavy to go to each individual and ask how specific parts impacted them.

### *Purpose Statement*

To better identify the most powerful rhetoric, the most impactful topics during a speech; thus leading to drastically increased speech quality and understanding of crowd emotions for a given presentation. By creating an application to read the average happiness of the crowd, the ease of reading the crowds happiness will become easier to accomplish.

### *Context*

Facial Recognition has progressed over the years and few companies have started doing research on crowd emotions. For instance CrowdEmotion.co.uk is utilizing eye tracking and facial coding to understand the emotions of the crowd (CrowdEmotion, 2019). This is an overall very similar project that provides heuristical data on crowd emotions, however they seem to use this to perform focus group type analysis on pre-recorded videos. The general public's colloquial with facial detection and the emergence of FaceID will help to lower fear of our software. However, recognition requires us to store user information, the information about the face that will be recognized does not need the user to be identified for emotional detection.

.

Political polling is a time honored tradition and it has only grown more dominant in the political process. However no matter how many surveys people fill out after listening to speeches and no matter how much statistical analysis you run on that data it will always be tainted by the fact that people lie and change their mind. This can be seen in the way that political polls have come into question with recent elections failing to meet the polls (Bowman). People can lie on polling surveys, but it is significantly harder to hide one's emotions and as such it is easier to get a more accurate voting prediction. This, coupled with the fact that there are few if any real time polling solutions that do not require hardware for each person, makes our software extremely desirable.

Crowd analysis has been a field mostly focused on security applications such as detecting crime before it happens. Those projects have been on the forefront of the field given their obvious allure to governments and other security agencies. However, these projects have been hampered by the amount of detail needed for advanced facial recognition on so many faces at the same time. Our program subverts this by taking in less data given that we do not need every detail of every face inorder to determine emotion thus reducing the workload on the machine learning algorithm.

Overall our project takes the best aspects of several existing fields and projects only to improve their efficiency and apply them to a new niche. This makes our project unique and viable both from a programming perspective, but a business perspective as well. Using facial

.

detection will enable the users face to be individualized and then processed for emotional detection in the background.

*Significance of Project*

This project is critical for anyone presenting content to a group as it is a gateway to understanding an audience. By enabling live feedback to the user if they can easily tell one how they are doing during a speech. It may even tell a teacher when their students get excited about certain topics and when they do not like certain topics. Politicians can target rhetoric based on how the crowd historically acts on certain things; however, human beings are changing everyday and getting information in live time has become more accustomed in our growing world of technology. A user could potentially notify themselves of implications of danger by halting emphasis on topics that cause an audience distress. We believe that our application can be used as a tool for users to not only benefit the one giving the presentation, but even potentially helping the audience out as well.

With the implications of multi facial detection, we use data to note which individuals are struggling or are in distress. For instance, if a teacher is giving a presentation and a majority of the class is satisfied with the content then you will overall see more happiness. You may catch that one student that does not understand the material and is too afraid to speak up with our software. Thus providing the presenters with heuristics of happiness that may be used to understand audiences in all new ways.

.

We hope to provide the community with an application that can be used for detecting the happiness of an audience during presentations, speeches, or even lectures. By giving the user the power to not focus only one faces emotions at a time but many at once.

# CHAPTER 2: LITERATURE

*Literature Overview*

Dwivedi, P. (2019, April 4). Face detection, recognition and emotion detection in 8 lines of code! Retrieved October 11, 2019, from Medium website: https://towardsdatascience.com/face-detection-recognition-and-emotion-detection-in-8-lines-of-code-b2ce32d4d5de

Faces are easily distinguishable between people and with this source the process of identifying people has been simplified into three primary steps. Facial Detection, Facial Recognition, and then emotional recognition. The user breaks down the steps needed to follow these three steps via python using a face_recognition library for detecting people's faces. How they then Identify the face is that they encode the picture and set a tolerance to identify if it is a match with another picture. In other words they set a threshold for how low the accuracy of comparing two images together can be. If the provided image is above that threshold then the image counts and matches. Lastly, the process of Emotional Detection can be depicted by classifying the pictures that match within a threshold. In other words, you can portray two images of matching people to be

.

considered as two different emotions and run the same comparison algorithms for which the user classified the person to be matching.This source heavily relates to our topic as it not only gets the concept of facial recognition into the form of a utilizable library. But it also enables us to learn about how emotional detects are used in the same library. Giving us a gateway to work with in classifying a person as happy, angry, sad, neutral, surprise, disgust or in fear.

phy, lieng. (2018, August 17). Build your own Face Recognition using face_recognition library and K-Nearest Neighbors classifier. Retrieved October 11, 2019, from Medium website: https://medium.com/beesightsoft/build-your-own-face-recognition-using-face-recognition-library-and-k-nearest-neighbors-classifier-611ffc973d4b

This source information is more or less a tutorial on how a KNN (K Nearest Neighbors) machine learning algorithm is implemented into facial recognition. KNN is a classification algorithm for grouping like sets of data together. This source goes into detail about the steps of facial recognition. The First step of which being using an algorithm such as a HOG algorithm to detect the face. Implementing KNN to classify the facial features to that of the existing dataset to group the person into a classification to confirm which person they are. By utilizing the euclidean distance and analyzing to what indices that the object that is being classified, is in relation to that of the training data set. In other words, this algorithm will take an encoded set of data and get the euclidean distance of that to the rest of the dataset. To then return the indexes of the closest defined data in the amount of the variable K. If K is five, then five sets of

.

classifications will be returned and then the most frequent classification is used to classify the new piece of data. This is relevant to our work because of the grouping process of KNN. Because KNN is able to get a group of data and compare that large entity to that of a smaller testing data. Also the research insight given for learning about a HOG algorithm is very valuable as identifying faces is crucial for gathering a dataset. When we have a dataset we can then encode the data

Ekman, P., & Friesen, W. (2013). I can see it all over your face. In H. Roger , *40 Studies That Changed Psychology* (7th ed.). Pearson Education.

The sections of reading are about facial emotions and the study of them with interactions. With Ekman and Frisen concluding that there are specific facial expressions that are universal. The universal facial expressions can be classified as undergoing Happiness, Sadness, Anger, Surprise, Disgust, or Fear. These 6 core emotions were examined by translator teams of the highest profession to classify which of the users were experiencing which of the 6 emotions (Happiness, Sadness, Anger, Surprise, Disgust, or Fear). These 6 emotions originated from the participants receiving  sets of photos. These photos were then presented one time each to the participants where they could ensure that there were no repeating photos presented and that the prompters were not influencing the emotions of the individuals. These participants then had to choose the photos emotion based off of the 6 that were presented to them. With the results being

.

so high in identifying the emotions , it was determined that these 6 emotions (Happiness, Sadness, Anger, Surprise, Disgust, or Fear) are universally identifiable by people.

This study is utilizable for our Capstone mostly because of classification methods. We believe that if we narrow down the emotions to a basic list such as the 6 studied for in this study. That we will have a better chance to group our algorithms training process in identifying targets to the respective emotions. Also with the process of image manipulation, we could potentially utilize a similar study but in the opposite prospective. Whereas instead of users identifying which image goes to a place, we could capture their emotions based off of images we present to them.

Face detection concepts overview | mobile vision. (n.d.). Retrieved October 11, 2019, from Google Developers website: https://developers.google.com/vision/face-detection-concepts

This scholarly source is going to be a primary source for configuring our application to detect faces in the audience. This source goes in depth about Facial Detection which is not to be mistaken for Facial Identification. This documentation provides us with the general layout and functionality of Google’s Facial Detection API that can be utilized for Android Devices. This source is crucial to our work as it will help with multi facial recognition and let us do our classifications on the faces. This will not only enable us to start developing with a blank slate, but also give the option to move to iOS if Android doesn’t go as planned. We hope to implement an android application but if that does not work out well with what we have planned then with this giving us knowledge of an iOS api. It will ultimately give us options to work with in the end.

This Api serves as a gateway to understand and implement cross-platformed facial detection as it describes the concept of facial detection as from machine learning datasets. Datasets that have gathered images of faces and compare pixel to pixel to give an answer whether a mass presented them is a face or not.

Introduction to facial emotion recognition. (2018, February 28). Retrieved October 11, 2019, from Algorithmia Blog website: https://blog.algorithmia.com/introduction-to-emotion-recognition

This Source is a paraphrased reference of the parent source that is labeled within. Quoting that "Facial emotion recognition is the process of detecting human emotions from facial expressions." (Introduction to facial emotion recognition) Giving us the information that emotional recognition is used in areas such as the TSA and in Audience Engagement. This source goes into depth about how the implementations of Machine Learning algorithms are only as good as the data that you enter into it. It uses the analogy that if you put garbage in then you will get garbage out. But how to look for Human emotions, by either taking the categorical approach where emotions are grouped together in classes. Or the Dimensional approach where the emotions are measured on a spectrum instead of groups. With that said the studies of emotions are either generalized into grouped and labeled according to that group, or they are measured in parallel to all classifications and prioritized based on the emotions that are strongly presented.

.

This source gives us an insight as to what we could utilize in our algorithm when attempting to classify users. This gives us the option to either class data or to go on a more spectral approach. Also by giving us information about the Facial Action Coding System and how the emotions of happiness, sadness, surprise, fear, anger, disgust, and contempt are all classes of emotions. I believe this sets us up for taking a more categorical approach in terms of defining a face to the corresponding emotions.

Cassino, D. (2016, August 1). How today's political polling works. *Harvard Business Review*. Retrieved from https://hbr.org/2016/08/how-todays-political-polling-works

This magazine article talks about the failings in the modern polling system that have crept up in the past few years. These issues include rising costs, as it is illegal to call cell phones through automation, drastically decreased response rate for calls, and sample bias. There is also the problem of low quality polls flooding the market and being seen to be just as performable as high cost and high quality polls. While the invention of online polling has been helpful it presents an oversampling of men and the unemployed which can skew results. There is also the problem of upweighting and downweighting where one person of a hard to reach demographic, such as African American males, is treated as too representative of a whole or the opposite with an overrepresented demographic is treated as less important. The author Dan Cassino, is an associate professor of political science at Fairleigh Dickinson University and has written several articles for the Harvard Business Review; he possesses a left leaning political bias.  The relevance of this is that our program may be a better polling method given a diverse enough crowd thus providing better data to people.

McDuff, D., & al, E. (2016, October 27). Measuring Voter's Candidate Preference Based on Affective Responses to Election Debates. Retrieved from https://www.affectiva.com/wp-content/uploads/2017/03/Measuring-Voters-Candidate-Preference-Based-on-Affective-Responses-to-Election-Debate.pdf.

A study that used the Affectiva software to predict how a person would vote when shown clips from the 2012 Presidential debate. Politics can evoke an emotional response from many people because it can have a large effect on their life thus making a good predictor for voting. This included breaking down the debate into its five main talking points and gathering an individual's response to said point as opposed to doing it in a row.  The emotional responses to the speakers points were then used to determine the viewer's voting preference.  The model was 73% accurate when it came to predicting the viewer's vote.  This paper validates the idea of the planned application and its viability given how similar it is to this paper.  Our application plans on performing this same operation just in real time and on a crowd and this paper proves that the basis for this is both feasible and will provide applicable results.

McDuff, D., & al, E. (2015, September 7). AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit. Retrieved from https://www.affectiva.com/wp-content/uploads/2017/03/McDuff_2016_Affdex.pdf

An article that describes a facial recognition SDK. The Facial Action Coding System (FACS), a widely used taxonomy, is used to codify facial behavior into the 7 primary emotions. This is further broken down into Facial Action units (AU) to predict facial expression given that FACS is very hard to code and time consuming to do so making it impractical for any real time and scaled application. The SDK uses Support Vector Machine Classifiers to perform the predictions, once fed the data. The Support Vector Machines 10,000s of facial images from around the world and then provides a score from 0 to 100 for each image on how strongly it displays each emotion. The emotions of Anger, Disgust, Fear, Joy, Sadness, Surprise and Contempt are the categories that emotions can be scaled into on a scale of 0 to 100 for each one. The relevance of this is that it shows a good implementation of a similar machine learning algorithm into an application and how to use it in a more programmatic sense.

Cohen, I., Garg, A., & Huang, T. (n.d.). Emotion Recognition from Facial Expressions using Multilevel HMM. *The University of Illinois*, 1–7. Retrieved from http://www.ifp.illinois.edu/~ashutosh/papers/NIPS_emotion.pdf

An article describing a way to perform emotional recognition specially on video instead of the more common picture form. The model used for this is Hidden Markov Models which is a

.

dynamic Bayesian Network. The benefit of the HMM is that the segmentation and recognition of facial expressions are done automatcally while increasing the discrimination power between different classes. The use of facial and emotion recognition is often done on pictures but this model is directly for video which is highly relevant for our project considering we are going to need to perform live emotional recognition on video. This coupled with a more effective model is important, given that we are restrained by the lower end performance of mobile devices.

Géron, Aurélien. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow* (2nd Edition). Retrieved from https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

This book is all about the fundamentals of machine learning in the Python language with focus on three tools: the Scikit Learn Library, Tensorflow, and Keras. Tensorflow is one of the main ways that people both create and train neural networks and being able to understand and create them ourselves will allow us greater understanding of the one we use. Keras is a wrapper on top of Tensorflow that makes it much easier to both read and use and frees up a lot of unnecessary bloat that can come from Tensorflow. Scikit Learn is a Python library that lets you control every aspect of an algorithm from beginning to end. Given both partners' familiarity with Python, our background with machine learning, and this book as a reference we will be able to fully manipulate any off-the-shelf algorithm to meet our exact needs for this project.

.

*Software*

Thoughtworksarts/emopy [Python]. (2019). Retrieved from https://github.com/thoughtworksarts/EmoPy (Original work published 2017)

Emopy is an open source project that can successfully predict a person's emotions from a presented picture. This relevant software can act as both a gateway and a piece of knowledge that we could use to help us develop a structure for our classification algorithm . However this project works with images and not live data.

This project is heavily related to our capstone as it explores the field of Facial Expression Recognition using public datasets to train a machine learning algorithm to predict results based off of pictures provided as data. Our project's aim is nearly the same as this project's; however, we intend to aim our project on crowds with multiple in real time.

Tensorflow lite. (n.d.). Retrieved October 11, 2019, from TensorFlow website: https://www.tensorflow.org/lite

Tensorflow lite is a software that translates machine learning models written in Python for mobile devices. Most machine learning is done on python which cannot run on mobile devices given that Android apps are written purely in Java, or Kotlin, and IOS apps are written Swift, or Objective C and do not support Python code. This poses a problem given that the Emopy software is written in Python with Tensorflow handling the backend. Tensorflow lite lets us port Emopy to Android which will improve latency, as there is no round trip to a server, and privacy, as no data would leave the device.  This side steps a major issue by having us not implement complicated machine learning in a language that lacks both the tools and support for it.

.

CHAPTER 3: METHODS

*Design*

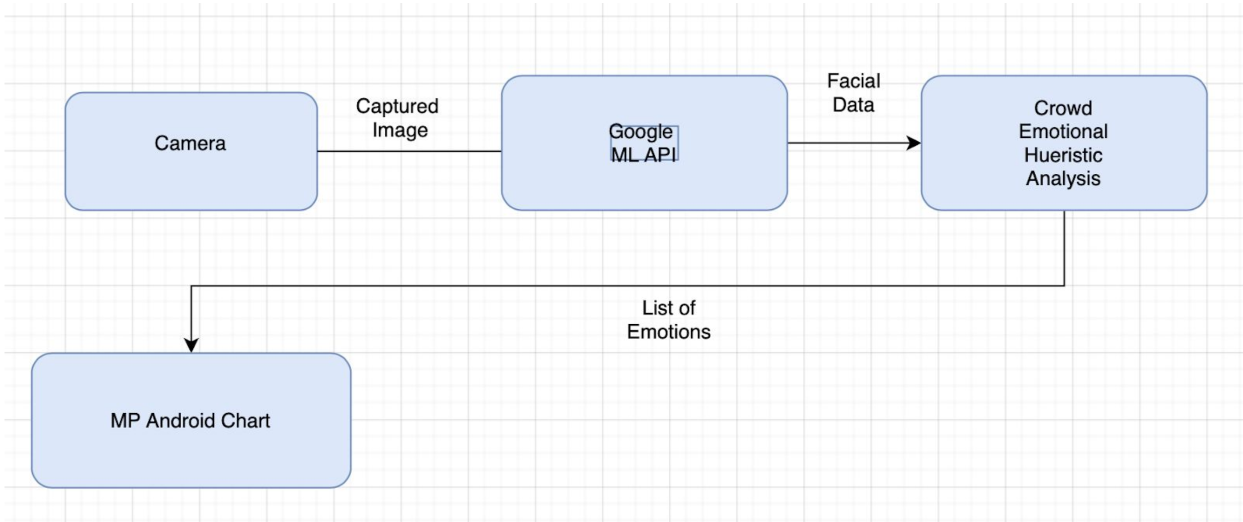



*Figure #1: General Internal Flow Diagram*

Figure One displays the general internal flow of data from its capture with the camera into the detection of each face in a crowd and isolating each individual face. Those faces are then fed into the Convolutional Neural Net for analysis returning all the emotions felt in the crowd at the given moment which is then performed heuristics on for ease of understanding for the user. This analyzed data is then displayed to the user.

This is the major engine which will drive the main functionality and what needs to be implemented at a basic level for normal function. We will harness this design to give back data to the users

On another note, Our final capstone is an Android Application that will be running on a Samsung A9 Tablet. The tablet is important to the structure of the project because of the size and specs that it provides to the user. Therefore, the screen size is the framework that will drive how our project looks and interacts with the user in the end. This way estimating the hardwares capacity will enable the scope of our design to target each emotion individually.

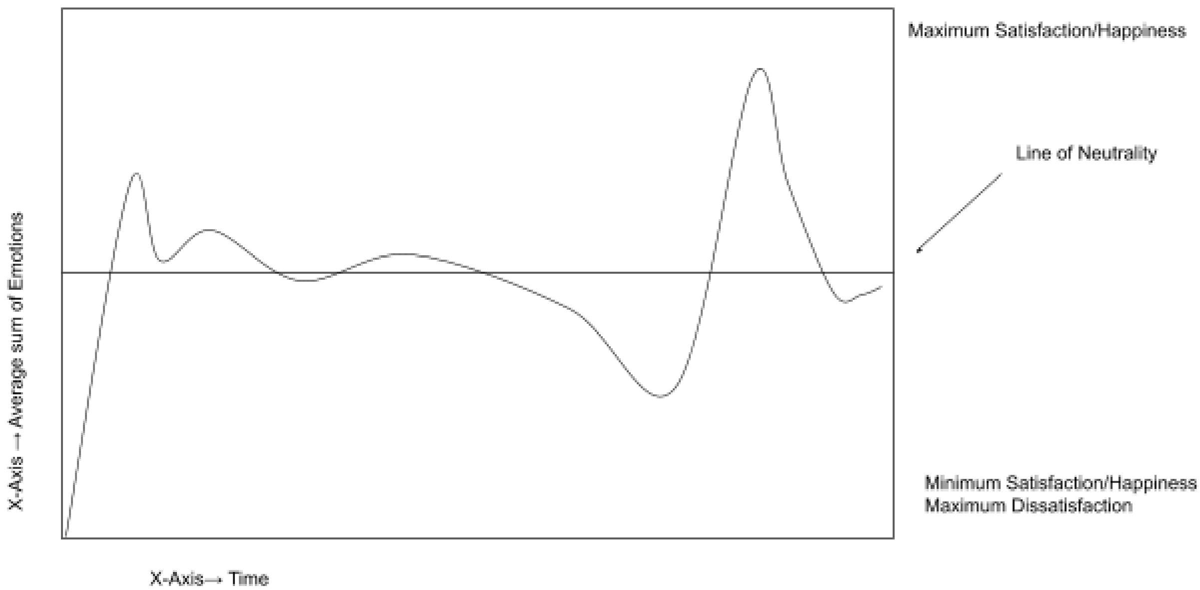



*Figure #2: Example of Charted Average Emotional Display for Happiness*

For displaying our average emotion we are planning to implement an average between happiness and unhappiness. This scaled average emotion will rely upon the emotions that are absorbed by the EmoPy/ Emotional Recognition Algorithm. This will enable the user to get a grasp of the averaged emotion during time slots of the given time frame amongst starting the main algorithm.In all, with this data we can get the user to read what the crowd is thinking at given time frames of a presentation.

.

*Frameworks*

The Framework Used for this Application via Android Studio and will be written in Java for the emotion detection process. (Changes may be made during progression of project if seen fit). The Hardware for the product will be running on a Samsung A9 tablet with 32GB of Memory.

Android Studio enables an android app to be made to host the Application, as well as give access to libraries in the Google and Android API. This provides us with the capacity to create a User Experience in a mobile matter, thus making the application proatable to the user. This framework was chosen due to both designers' familiarity with the design space and the dominant overall market share held by Android mobile devices over all other mobile operating systems.

Utilizing Tensorflow Lite to use an emotion recognition algorithm named EmoPy written in Python3 will enable smooth facial identification. Tensorflow Lite will be used to convert the pre-trained convolutional neural network of EmoPy into a format that is able to be used inside the Android SDK environment. To another extent OpenCV will be used to implement emotion recognition with the android application as well as photo manipulation such as isolating all the faces in a photo of a crowd into individual photos of faces for use in the neural network.

Other implemented API's include, Google's Multi-Facial recognition to pick out faces from a photo. As our Neural Networks are only able to decipher one face at a time, we will need to pipeline the faces captured to communicate with emotion recognition software. With the use

.

of a Handler, we can simulate photo taking in a pseudo-infinite loop to mimic the representation of recording a video.

Handler() → will act as infinite loop for the camera process that is stoppable by the user

scanHappiness() ---> will be responsible for scanning the happiness of the faces in the picture

updateChart() → will be responsible for charting the data accurately for the user

Descriptions and pseudocode of important algorithms in use, relevant, or to be implemented as part of your project.

*Above represents the core functions of the project and what it will do in the end*

*Analytical Methods*

The positive, or engagement, emotions will be total and weighed against the negative, or disinterested, emotions to generate the percentage of the audience that can be perceived as engaged in the speech. This main setting can be further extrapolated to show what individual emotions are rising and falling at a given time without tweaking them after detection depending on user preference. Regardless of whether the user uses the default engagement metric or some specific emotions, all data of each emotion's prevalence through a speech will be stored should a user wish to break down the engagement in a more detailed manner after the speech has concluded. Minimum and maximums of each emotion shall be identified and presented.

*Features*

The Features we intend to implement are:

- The ability to view crowd emotions in real time with specific percentages for each captured emotions.
- Have a user choose to view the heuristics of Happiness between saved speeches

.

*Test Plan*

Initial testing will be self-testing without crowd functionality to ensure both proper individual facial detection and correct emotional detection under ideal conditions. Once both of these functions have been proven to work to a satisfactory level testing on a crowd in general will happen with crowds of varying sizes and conditions, such as lighting. The main type of data to be gathered from this process will be general emotional detection accuracy as well as the speed of the application. In particular the test must determine the lag between a moment passing and the emotional data appearing before the user. This is measured for user use given that any significant delay between these two functions would significantly affect the general use of the program. Given the current world status of the COVID-19 virus, it is unsafe to perform testing on human subjects and as a result testing will be done via viewing television shows. Specifically one's at which the number of people on the screen are more than one person. This will be applied to make sure that the Faces are being detected and that the emotions are returning numerical results.

.

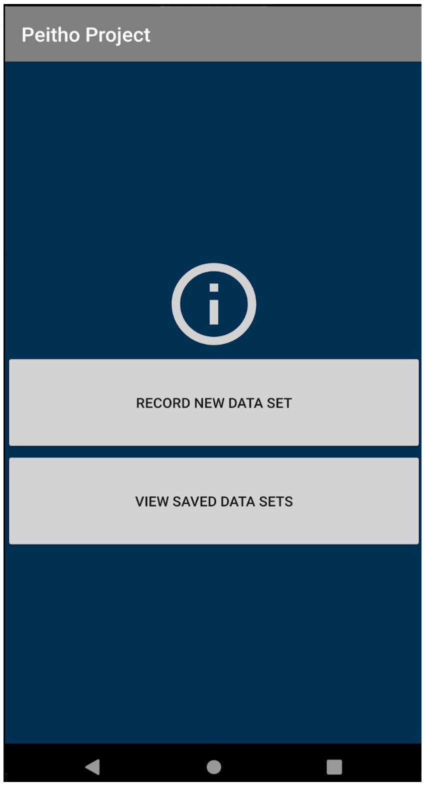
*Criteria and Constraints*

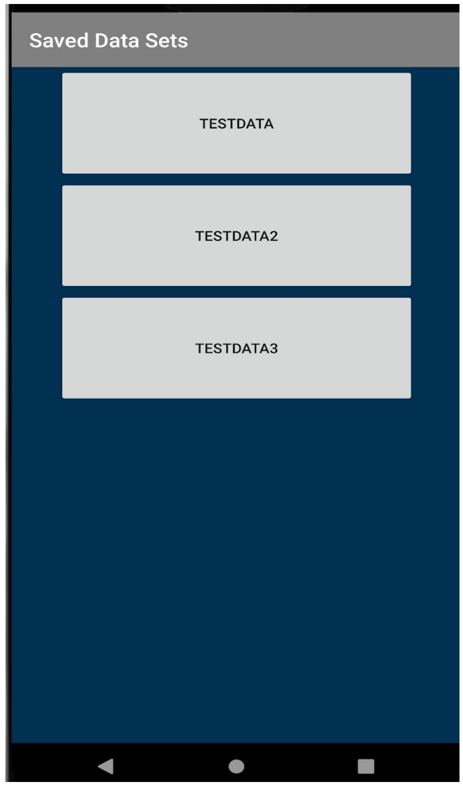
There are several constraints that are applicable to this project. For one there is the political constraint given that this was designed with improving speech effectiveness. This could be used by people with nefarious political views to spread their message in a more effective way. This leads to a problem regarding distribution. With wide scale distribution through an app store there would be no true control over who uses the program thus leaving it open to abuse. However should the distribution be directly controlled there would be a greatly reduced number of users as well as remove any casual user who doesn't plan on using it politically. This leads to the conclusion that if it is going to be released it should be widely done so in order to level any playing field by letting anyone use it thus negating any advantage a candidate might have over another. The second criteria that appears is the ethical issues of the program. Many people could be adverse to being recorded and having their emotions analyzed leaving the user in a precarious position. To solve this the terms of service will state that you must disclose the use of this software to people who are attending an event and inform them of its purpose. The disclosure of informing the participants of its use will allow them to opt out of the event should they feel uncomfortable with the program's use.

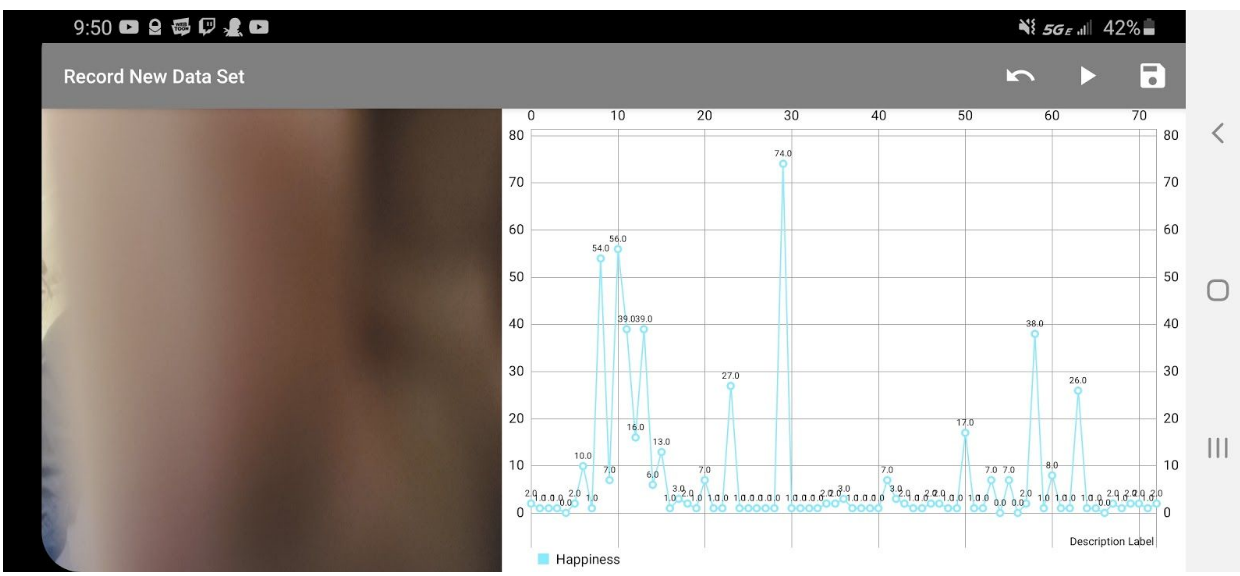
.

# CHAPTER 4: RESULTS (2-4 PAGES)

*Final User Interface*



*Testing Results*

During the recording the users were tested at various distances with 3 different devices with 3 different Cameras. A Google Pixel 3 with a 12MP/2160p (GSMARENA) Camera as well as a Samsung SM-T290 (Galaxy Tab 8.0 2019) with a 8MP Camera (Herbrich, Marcus) and on a Samsung Galaxy S20 Ultra. It has been noticed that the camera quality of the testing device's camera has a significant factor on the accuracy of Happiness that is recorded. Where the Samsung Galaxy Tablet would have missing factors (not detecting faces) to the testing data due to poor camera quality.. If the device does not detect any faces then the function will not update the chart. Therefore, from this testing we can conclude that the Firebase API for detecting faces is reliable on the quality of the image when expecting high accuracy.

The Happiness Detector is based off of the image captured in the moment at which the handler runs the task to capture the image. It is possible for the image to be captured in a pin-point specific moment of pure happiness of which the users look as if they are not showing happiness, thus resulting in a non-happy number to be recorded. This can be apparent in Testing 5 in results 3 and 4; as well as, testing 6 in results 6, 7, and 8. Where users showed signs of happiness and the result recorded did not reflect the actual portrayed results.

The overall functionality of the Algorithm is accurate as the happiness scale is based on the user's smile as well as details within the contour of their faces. Happiness is measured in a

.

number ranging between 0-100. With the lower boundaries reflecting unhappy results and the high boundaries reflecting happy results. Since the range is determined by smiling probability and facial contouring, it is fair to assume that the algorithm provides accurate representations of happiness.This is primarily due to the fact that when the users are not smiling they are showing a neutral emotion and not happiness. Because of this I would say that the Methodology and the construction of how the algorithm detects and portrays happiness is accurate and functioning to the best of its ability. Although I would have to say that the five second buffer between pictures may need to be adjusted for more accurate results.

Upon further analysis, a five second gap between facial capture is far to largeas many laughs and smiles may only last a few seconds and therefore be lost in time. Within the graph it is easy to see where the “joking” scenes of the show are detailed by the high volatility of certain time chucks such as 6-16 and 55-65.  This lines up with the most comedic part of the episode that was viewed.  Given the testers observations during this period the initial 5 second interval of photo capturing has been deemed too long.  This conclusion was reached based on the observation that many comedic and humorous moments were missed by the device because they occurred during the period between two captures.  A shorter capture interval, of at least one second, will provide clearer and more consistent data and also reduce the volatility seen in the graph as the fall in happiness will be captured.

.

| Test ID | Device filming emotions | Version of Android ran | Time Duration | Refresh Rate | Number of Reads In charting | Overall Average (For All reads) | Number of Test Subjects | Distance from subjects | Participants current actions |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Google Pixel 3 | 10/X | 45s | 5000 ms (5 seconds) | 9 | 18/100 - Not very Happy | 2 people | 6.5 Feet | Watching Schitts Creek S6 Ep 1 |
| 2 | Google Pixel 3 | 10/X | 45s | 5000 ms (5 seconds) | 9 | 29.3/100 - Somewhat Medium | 2 people | 5.5 Feet | Watching Schitts Creek S6 Ep 1 |
| 3 | Google Pixel 3 | 10/X | 45s | 5000 ms (5 seconds) | 8 | 21.5/ 100- Somewhat Medium | 2 people | 5.5 Feet | Watching Schitts Creek S6 Ep 2 |
| 4 | Google Pixel 3 | 10/X | 45s | 5000 ms (5 seconds) | 9 | 25.33/100 - Somewhat Medium | 2 people | 4.5 Feet | Watching Schitts Creek S6 Ep 2 |
| 5 | Samsung SM-T290 | 9/Pie | 45s | 5000 ms (5 seconds) | 7 | 18.85/ 100- Not happy | 2 people | 6.5 Feet | Watching Schitts Creek S6 Ep 3 |
| 6 | Samsung SM-T290 | 9/Pie | 45s | 5000 ms (5 seconds) | 8 | 13.125/100 - Not Happy | 2 people | 5.5 Feet | Watching Schitts Creek S6 Ep 3 |
| 7 | Samsung Galaxy S20 Ultra | 10 | 6 minutes | 5000 ms (5 seconds) | 72 | 5000ms (5 seconds) | 2 | 4 feet | Watching Will and Grace |

| Test ID | Reading 1 Result | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | Result 7 | Result 8 | Result 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 40 | 67 | 7 | 12 | 8 | 12 | 2 | 2 | 12 |
| | Personal Note | | | | | | | | |
| | Users are giggling | Users had a sudden laugh | Users expression is bland | | | | | | |

| Test ID | Reading 1 Result | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | Result 7 | Result 8 | Result 9 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 63 | 17 | 24 | 22 | 7 | 8 | 80 | 16 | 27 |
| | Personal Note | | | | | | | | |
| | Users started smiling - One is showing teeth | Users not smiling anymore | | | | Both users started Laughing hysterically after this number was recorded | Both Users are hysterically laughing | Users calmed down | |

| Test ID | Reading 1 Result | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | Result 7 | Result 8 |
|---|---|---|---|---|---|---|---|---|
| 3 | 10 | 20 | 20 | 22 | 23 | 40 | 22 | 15 |
| | Personal Note | | | | | | | |
| | Users not smiling but not frowning | | | | | Users giggled for a moment | Users emotions looked Generic | |

| Test ID | Reading 1 Result | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | Result 7 | Result 8 | Result 9 |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 22 | 33 | 26 | 48 | 30 | 21 | 30 | 23 | 25 |
| | Personal Note | | | | | | | | |
| | Users smiling without teeth | | Users Giggling to jokes | | | Users laughing | Users Neutral faced | | |

| Test ID | Reading 1 Result | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | Result 7 |
|---|---|---|---|---|---|---|---|
| 5 | 2 | 3 | 44 | 51 | 2 | 17 | 13 |
| | Personal Note | | | | | | |
| | Users are not expressing emotion | | Users are laughing hysterically | | Users are neutral faced | | |

| Test ID | Result 1 | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | Result 7 | Result 8 |
|---|---|---|---|---|---|---|---|---|
| 6 | 6 | 7 | 3 | 7 | 14 | 29 | 39 | 0 |
| | Personal Note | | | | | | | |
| | Users showing neutral Emotions | | | | | Users Starting giggling | | |

| Test ID | Result #1 | Result #2 | Result #3 | Result #4 | Result #5 | Result #6 | Result #7 | Result #8 | Result #9 | Result #10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 2 | 1 | 1 | 1 | 0 | 2 | 10 | 1 | 54 | 7 |

| Test ID | Result #11 | Result #12 | Result #13 | Result #14 | Result #15 | Result #16 | Result #17 | Result #18 | Result #19 | Result #20 |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 56 | 39 | 16 | 39 | 6 | 13 | 1 | 3 | 2 | 1 |

| Test ID | Result #21 | Result #22 | Result #23 | Result #24 | Result #25 | Result #26 | Result #27 | Result #28 | Result #29 | Result #30 |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 1 | 1 | 27 | 1 | 1 | 1 | 1 | 74 | 1 |

| Test ID | Result #31 | Result #32 | Result #33 | Result #34 | Result #35 | Result #36 | Result #37 | Result #38 | Result #39 | Result #40 |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 1 | 1 | 1 |

| Test ID | Result #41 | Result #42 | Result #43 | Result #44 | Result #45 | Result #46 | Result #47 | Result #48 | Result #49 | Result #50 |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 7 | 3 | 2 | 1 | 1 | 2 | 2 | 1 | 1 | 17 |

| Test ID | Result #51 | Result #52 | Result #53 | Result #54 | Result #55 | Result #56 | Result #57 | Result #58 | Result #59 | Result #60 |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 1 | 1 | 7 | 0 | 7 | 0 | 2 | 38 | 1 | 8 |

| Test ID | Result #61 | Result #62 | Result #63 | Result #64 | Result #65 | Result #66 | Result #67 | Result #68 | Result #69 | Result #70 |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 1 | 1 | 26 | 1 | 1 | 0 | 2 | 1 | 2 | 2 |

| Test ID | Result #71 | Result #72 |
|---|---|---|
| 6 | 1 | 2 |

# CHAPTER 5: CONCLUSION

*Challenges and Solutions*

There were a plethora of challenges that we faced during our Capstone experience.For instance, when attempting to utilize Tensorflow, many of the application's pieces would either break or fail to operate further. Data would get lost as too many threads are being generated. As a result, the solution that we had to implement was a downgrade in complexity from our original intention—we were forced to use the happiness detection API from Firebase instead of our original Tensorflow model. This challenge was due to the complexity of implementation being high and to the integration being unsuccessful. Our original Tensorflow-backed solution, EmoPy, could detect six emotions, whereas the final Firebase solution could only detect happiness.

This decision did not come easily and we wish to attempt Tensorflow implementation further in the future. However, due to the COVID-19 Pandemic causing many of our daily lives to halt and causing social distancing lifestyles temporarily during the semester, we felt like this was a necessary decision to make to bring our project to a conclusion during the semester given the time and productivity constraints.

Regardless, we have implemented Google's MLK Firebase API into our code and have decided to use the Vision Face API to detect faces which will enable us to throw back a number to the user. This number is the happiness of the face on a scale of zero to one hundred. By utilizing the happiness scale of people's faces in a crowd we believe that the results will be just as good as using the primary six emotions. This is because happiness is one of the only emotions

.

labeled as positive. Therefore, we can distinguish between positive and not-positive emotions by using happiness as the barrier. We think this solution is sufficient to give a sense of crowd emotion in a given that happiness is a good indicator of engagement and agreeableness both of which are useful metrics to a speech giver.

### *Future Work*

Our project aimed to go further during the semester, but due to world circumstances we were unable to successfully make our full goals; but, in the future with more time on our hands we intend to see if we can make our full vision come true with classifying the faces with the core six emotions. The final classifier will use a similar scale to the happiness scale, as we intend to classify the user on a scale of zero to one hundred.

We also intend for our user interface to get better as we will want to make the options available to the user to choose which emotions they want to capture. For example, the user may not want to detect the happiness of the crowd, but instead the anger of the crowd. We want to make that option available to the user.

### *Project Importance*

Our project is important because depicting the happiness of an audience is revolutionary in understanding crowds. For instance in the medical field you can utilize an application to get emotional data from a patient that is uncooperative in sharing their emotions. The user will more

.

often than not be able to use this tool to help classify patients as emotionally stable or not in circumstances of domestic violence or in cases of prescription medication. Emotion detection can be an especially potent tool when administered through a mobile, accessible, device.

This project is beneficial for the computing community because of the influence that it may have on others to inspire them to work on emotion recognition. Emotion recognition is a great machine learning concept that should not be underestimated. Emotion detection could be used as another factor on facial recognition security as it may be able to learn the common emotions of the face detected and depict more against dopplegangers, or even notify police of users who are mentally unstable.

There are those who suffer from the inability or difficulty in reading the emotions of others due to lack of social awareness or cognitive disability. This application could provide them an easy way to interpret the emotions of others in a format that is easier to understand without having to rely on social cues. This would lower the barrier of entry to many social environments that would be much harder for people with those problems to navigate; thus granting them a more robust social life and experience.

Anyone who has to do any form of public speaking will find some use for the application. To know what points or jokes are the most effective and knowing which are the worst can give people an easy idea of what they are doing both right and wrong. This gives them easier time when it comes to improving their speech quality and performance without having to rely on others.

.

REFERENCES


Crowdemotion. (n.d.). Retrieved September 20, 2019, from CrowdEmotion website:

https://www.crowdemotion.co.uk

Bowman, K. (n.d.). The trouble with polling. Retrieved September 20, 2019, from

https://nationalaffairs.com/publications/detail/the-trouble-with-polling

GSMARENA. “Google Pixel 3.” *Google Pixel 3 - Full Phone Specifications*, www.gsmarena.com/google_pixel_3-9256.php.

Herbrich, Marcus. “Samsung Galaxy Tab A 8.0 (2019) Tablet Review: A Budget Samsung Tablet with Great Deficiencies.” *Notebookcheck*, Notebookcheck, 2 Jan. 2020, www.notebookcheck.net/Samsung-Galaxy-Tab-A-8-0-2019-Tablet-Review-A-budget-Samsung-tablet-with-great-deficiencies.448922.0.html.


.