

# Using biometric data to detect the emotional state of social media user as a first step towards early anxiety detection and intervention

Emily Sam ([emilyhsam@gmail.com](mailto:emilyhsam@gmail.com)) and Zoey Mo ([mozhongyu@yahoo.com](mailto:mozhongyu@yahoo.com))

Flintridge Sacred Heart Academy

Submitted for Publication: 22 May 2021

## Abstract

Intense emotions often suppress reason, leading to potentially impulsive or harmful behaviors, such as sending negative messages to friends when not in a favorable mental state or causing physical or emotional harm. Since intense emotions are connected to physiological responses, this project aims to develop a software solution that utilizes AI image recognition to help prevent or reduce harmful behaviors by constantly measuring three physiological signals—heart rate, galvanic skin response, and skin temperature—during social media use, and providing alerts and suggestions to the user when a negative emotion is sensed. The target market for the AI is teenagers, as they are more prone to displaying impulsive behaviors. Using data acquired while watching movies, surfing social media, viewing short internet videos, various studies have proven the connections between the physiological data and emotional states. This project will use AI image recognition routines to make the connections between physiological data and emotional states.

## 1.0 Introduction

### 1.1 Intro

Is it possible to prevent people from exhibiting poor behavior on the internet? Why does social media contain so much non-factual information? Social media hate culture and cyberbullying is rooted deeply in the motives of social media users, more specifically teenagers. These behaviors ultimately lead to social issues surrounding cyberbullying and other mental health issues. Current research suggests that specific external physiological biomarkers, such as heart rate, galvanic skin response, and temperature, can indicate emotions. Apps have been developed specifically for therapy and meditation to help people address their emotional adversities, such as *Calm* and *Headspace*. Also, social media apps such as Instagram have recently been making their software intervene in their users' scrolling time for certain posts that are particularly triggering to see. The app adds a filter over the post with the words "Sensitive Content" and a warning that the post might contain graphic or violent content, making the user take an extra step to acknowledge that fact before proceeding to see the post, further utilizing intervention. However, there has not been an accessible app made to track a person's emotions by monitoring their physical responses towards their emotions while using social media directly and then intervening in their social media scroll because of their immediate emotion.

## 2.0 Background

### 2.1 Initial research

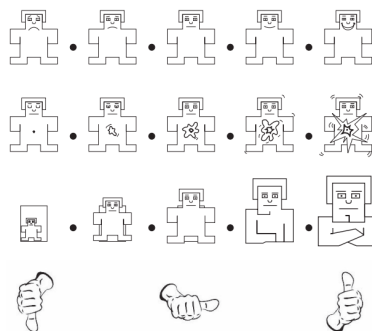
Multiple sources and technical papers have pointed towards simple everyday emotions being the main building blocks for a person’s mental health. These simple everyday emotions built up in most young people, especially teenagers, are caused by social media. As part of our initial research, we have decided to test this idea by conducting an experiment where we went on social media for about ten minutes to see what types of posts we would see: A CNN news report about Anti-Asian Hate, a comical internet meme about how Amazon obnoxiously recommends users variations of products that they’ve bought in the past (in that particular case, it was a wall of toilet lids), a cute baby goat, and a sarcastic yet impactful post about global climate change and carbon capture. Each of these Instagram posts were completely different, evoking completely different emotions. With seeing this much of a variety of instagram posts, it only made sense for disheveled, unsure, and other emotions to be felt by teens after ten minutes of social media use, let alone hours on end.

## 2.2 Physiological sensors and emotion

The physiological sensors used were a galvanic skin response (GSR) sensor, heart rate (HR) sensor, and a skin temperature (SKT) sensor. Studies of the interactions between emotions and physical biomarkers, as introduced in the lit review, have used these sensors and found that certain combinations of the sensors’ data can be used to identify different emotions (Lisetti & Nasoz, 2004). When reorganizing our data from collection to be better suited for the image formation to then be put into the AI, we had segmented our data or broken it into multiple pieces to provide near real time feedback and to multiply training opportunities. The images are made and are processed by the AI.

### 2.2.1 Valence, Arousal, Dominance.

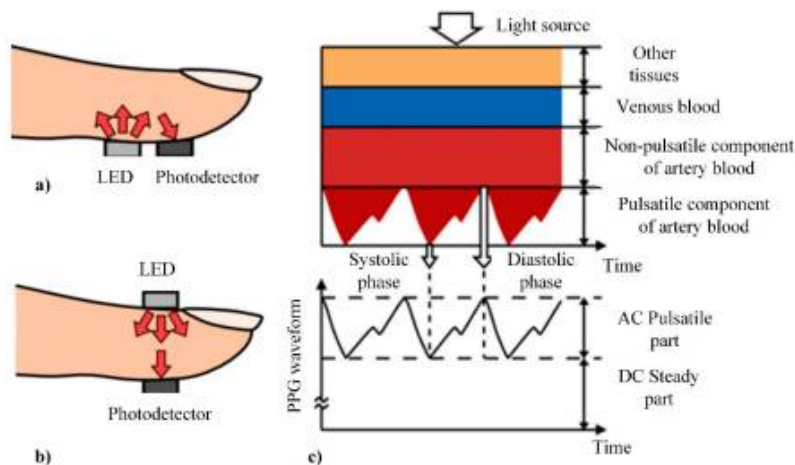
Figure 2.1 is a self-assessment used in the related study #2, illustrating valence, arousal, dominance, and liking. Valence levels indicate how pleasant or unpleasant an emotion is or how positive or negative the emotion is; arousal ranges from inactive to active, representing the intensity of the emotion; dominance ranges from feeling helpless to feeling in control of everything (Koelstra et al, 2012). This method of measurement helps us better understand the relationship between different emotions. We will either use this measurement directly as part of our experiment, or use what we learned from it to decide what emotions (ex. Anger, laughter, etc.) to identify.



**Figure 2.1: Images used for self-assessment (Koelstra et al, 2012).** Top to bottom: Valence, arousal, dominance, liking.

### 2.2.2 Photo-plethysmography (PPG).

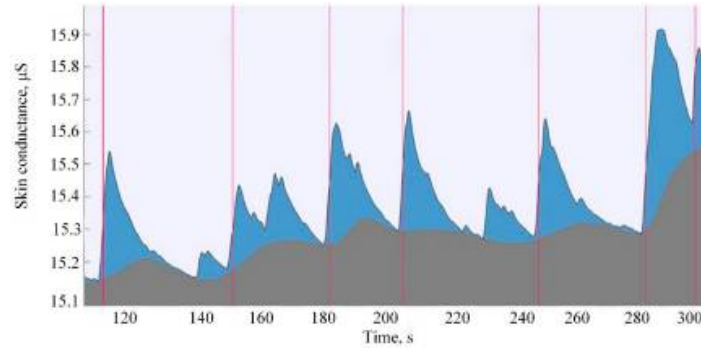
PPG is a relatively cheap and comfortable method that can be used to measure heart rate (Dzedzickis et al., 2020). Plethysmography measures volume changes. The volume change in blood can be converted into blood flow with the formula  $F=dV/dt$ , where  $dv$  is the change in volume and  $dt$  is the change in time (Langereis, 2010). The relative volume, with its information in the timing, indicates the pulsation rate of the heart (Langereis, 2010). A PPG sensor contains a light source from which a light is transmitted through or reflected back from the tissue and a photodetector that measures the small variation in the light to indicate the changes in perfusion, or passage of fluid through the the circulatory system or lymphatic system to an organ or a tissue, in the tissue (Dzedzickis et al., 2020)(Figure 2.2).



**Figure 2.2: Principle of Photo-plethysmography(PPG) (Tamura et al, 2014).**

### 2.2.3 Galvanic Skin Response (GSR).

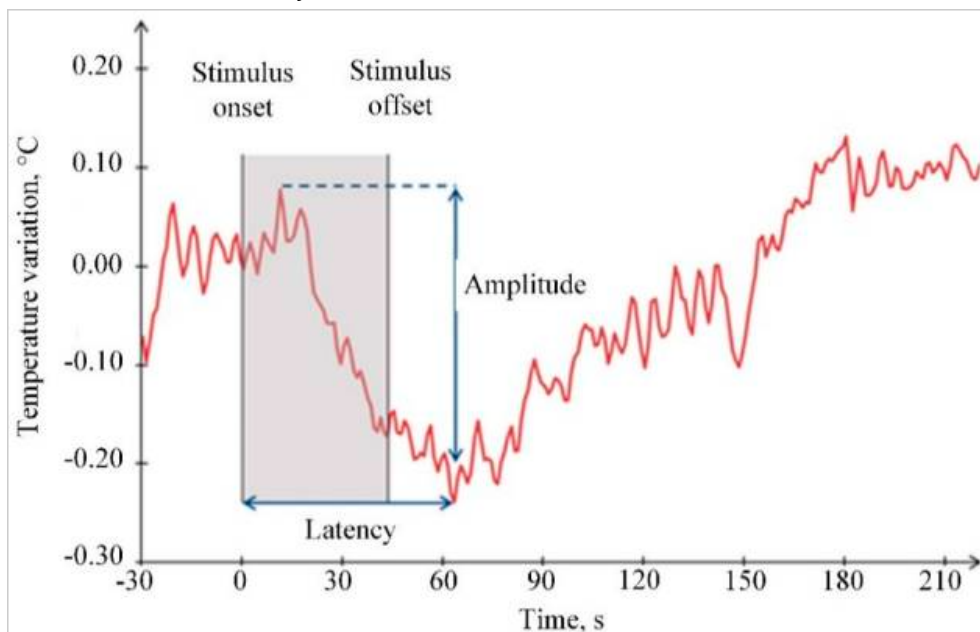
GSR is also known as skin conductance, which is based on and thus indicates the variation of sweat reaction. Sweat contains salt, and the variation of the amount of salt in human skin changes the electrical resistance of the skin. Since emotional changes induce sweat reactions, GSR can be used to indicate emotion. A raw GSR signal contains tonic and phasic components. The tonic component mainly depends on skin hydration level and environment factors, which change incrementally and individually in each usage. The phasic responses are short term peaks that are mostly independent of the tonic level, reflecting emotional changes. Skin conductance mainly indicates the arousal level, usually analyzed in time and frequency domains by features such as median, mean, standard deviation, minimum, maximum, and a ratio of minimum and maximum. The main drawback of GSR, in terms of emotion indication, is its lack of information about valence level, which can be solved by integrating with other emotion recognition methods. (Dzedzickis et al., 2020)



**Figure 2.3: Example of raw GSR signal (Gatti et al, 2018).** The blue area represents the phasic component (relatively rapid changes related to emotional triggers) of the signal; grey area represents the tonic component (relatively slower changes related to individual skin and environmental factors) of the signal; the red lines represent the occurrence of a stimulus.

#### 2.2.4 Skin Temperature (SKT).

Various studies have proven that negative emotions, such as stress, anxiety, anger, embarrassment, hostility, and fear, can lower the skin temperature. SKT measurements are often based on the face or full body. A main drawback of SKT is that a more persistent and intense stimulus is required for SKT to be useful in emotion detection. Therefore, SKT is not a good choice for short stimulus such as a picture. Besides, SKT cannot provide the exact emotions. Therefore, the technology is often integrated with other methods such as HR and GSR to provide a comprehensive biomarker analysis. (Dzedzickis et al., 2020)



**Figure 2.4: Example of SKT Data (Dzedzickis et al., 2020).**

#### 2.3 Deep Learning.

Deep learning is a technique by which computers learn to perform a specific task by extracting and transforming data using multiple layers of neural networks. The data are progressively refined while being passed down through each layer. Layers are trained by algorithms to maximize accuracy. After enough labeled data is fed, the algorithm finds out a pattern, and uses the pattern (or model) to predict the results of unlabeled data. One application of deep learning is image recognition, which is what we are doing. Our project is based on Fastai, a deep learning library which helps building accurate models quickly and easily.

### **3.0 Lit Review**

A study of (Lisetti & Nasoz, 2004) physiological signals including GSR, HR, and SKT (galvanic skin response, heart rate, and skin temperature) were utilized to detect emotions (sadness, anger, fear, surprise, frustration, and amusement). Emotions were elicited through the participants viewing 7 movie clips and solving math problems. Signals were collected using BodyMedia SenseWear Armband, a body monitor developed by BodyMedia. Collected data were normalized using the data taken during relaxation periods, and stored in 3D array of numbers: (1) subjects participated, (2) emotion classes, (3) data signal (minimum, maximum, mean, and variance). Three different machine learning algorithms were used to compare and categorize the measurements in terms of emotions: k-nearest neighbor algorithm (72.3% accuracy), Discriminant function analysis (75.0% accuracy), and Marquardt backpropagation algorithm (84.1% accuracy). The accuracy reported suggests that the reliability of emotion auto-detection via the three biomarkers (GSR, HR, and SKT) will also be sufficient for this project.

In a study of (Koelstra et al 2012), signals including EOG, electrooculographic signals, or facial expression around the eyes and forehead, EMG electromyographic signals, or a measure of the magnitude of muscle force, GSR, or galvanic skin response, blood volume pressure, temperature, and respiration, were measured with various sensors. Emotions valence, arousal, dominance, and liking were elicited through the participants viewing music videos. Thirty-two participants aged between 19 and 27 with an average age of 26.9. For each of the 120 videos, a one-minute segment with maximum emotional content was identified and extracted by an algorithm, based on features such as loudness and energy of audio signals, motion component, visual excitement, and short duration. Based on the 120 one-minute music video segments, 40 videos were chosen by volunteer ratings to be used in the experiment. The experiment started with a two-minute baseline recording, when the participants relaxed. The physiological signs of each participant were recorded while they watched each music video. At the end of each trial, a self-assessment of arousal, valence, liking, and dominance was performed by the participants. The signals were then related to the subjective ratings. Three modalities were used for classification: EEG, peripheral physiological signals, and MCA. Decision fusion, or a potential approach and is defined as the concept of combining information from different data sources, after each source has been classified individually, was used to integrate the three modalities, with the results compared to the expected values. Compared to feature fusion, in which data of different modalities are combined and processed, decision fusion processes each modality independently and combines the individual outputs to form the final result. EEG had the highest accuracy for arousal, peripheral for valence, and MCA for liking. In the end, the team presented a database to the public,

based on the 32 participants, for future analysis of spontaneous emotions. The study supports the close relationship between the peripheral signals and the valence level.

Since neither our two most influential studies (Lisetti, 2004) nor (Koelstra, et al 2012) focus on teenagers, this study will aim to fill that gap.

## 4.0 Methods

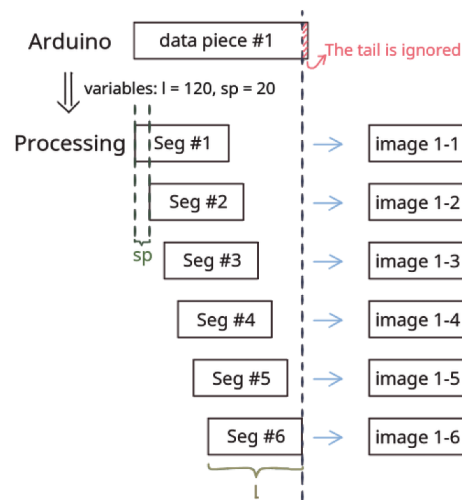
### 4.1 Data Collection

An Arduino microcontroller was used to collect Heart Rate (HR), Galvanic Skin Resistance (GSR), and Skin Temperature (SKT) data from users while they watched selected emotional movies. During the data collection process, users manually identified timestamps where intense emotions, positive or negative, were felt. The intense emotion timestamps were tagged for future use. Data processing utilized Arduino code, Google Sheets, Processing, Fastai, C++ code, command line, and .NET. All code is available in Appendix 7.0.

### 4.2 Data Processing: Segmentation

Biosensor data was recorded at a rate of 2 lines of readings per second. As demonstrated in figure 4.1, each data session was divided, using Processing codes, into overlapping 1 minute segments with each successive segment offset from the previous by 10 seconds. Segment length (l) and non-repeating space (sp) are adjustable variables that can be used for future optimization. After segmentation, each segment (1 minute of data) is converted to a graphical image.

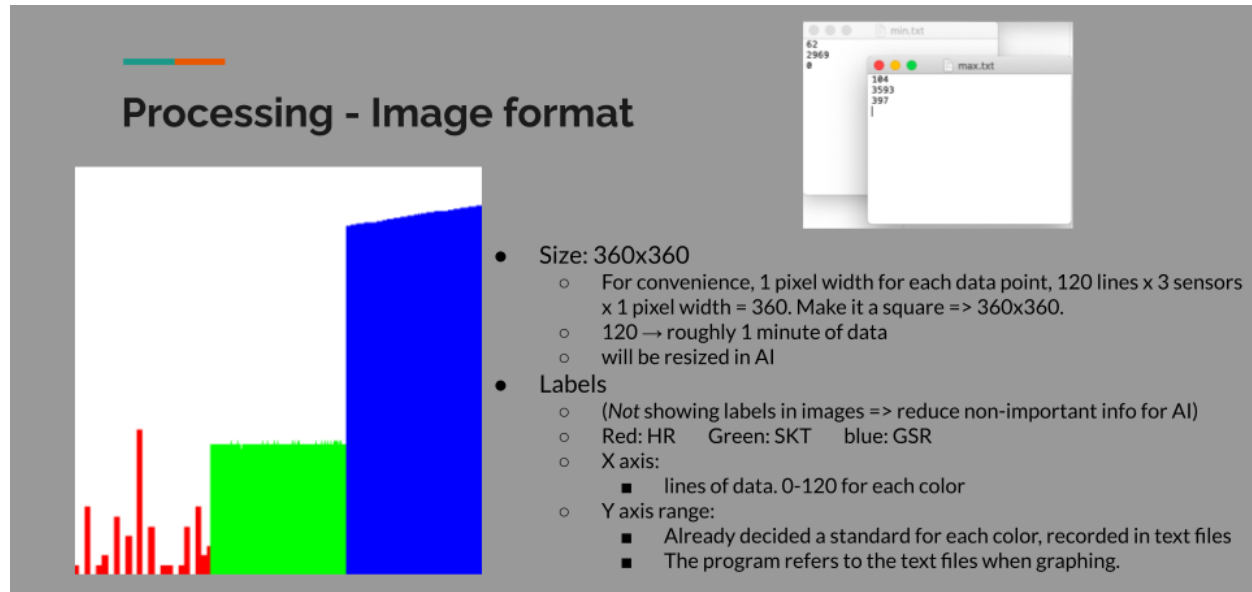
There are two reasons for using overlapping segments. First, while training the AI, partially repeating the data segments means more data segments for training without actually taking more measurements. In this example in the animation, out of a roughly 2 minute piece, we got 7 overlapping 1 minute segments, offset by 10 seconds each. Second, for the final prototype, partially repeating segments means that the software would still have relatively longer segments - 1 minute of data, to consider, making the decision more accurate, while taking shorter time between each decision - roughly 10 seconds, making the product more sensitive. The users do not need to wait for a whole another minute before the software makes another decision.



**Figure 4.1: Segmentation.**

### 4.3 Data to image conversion

The Processing program then organizes the segments of data into a visual form, a 360 by 360 square image (120 lines x 3 sensors x 1 pixel in width) where each pixel represents each data point. The program takes the integral of the point, or the area underneath the point, to create different colored blocks; the red represents heart rate, the blue represents galvanic skin response, and the lime green represents the skin temperature. Labels and axes are omitted to reduce meaningless or repeating information for the AI. The X-axis is lines of data, 0-120 for each color, or sensor. The Y-axis range is pre-determined for each sensor and recorded in text files ("max.txt" and "min.txt") — the Processing program refers to the same text files, or the same y range, each time it runs.



**Figure 4.2: Image format.**

#### 4.4 AI training

Systematic, comprehensive training in how to use an AI is what Fastai provides. The biggest challenge was to find the most effective format of input for our project and to take enough data for training - the rest was simply following the instructions.

In training the current prototype, only two pieces of unstable emotion were intense enough to be selected, each of which were split into multiple partially-repeating segments (explained in 2.2) for data augmentation. The first unstable piece was recorded when our researcher was feeling generally annoyed and was yelling to one of her parents for no reasonable excuse - she got even more annoyed when trying to assemble the device that was separated during travel. The second unstable piece was recorded after an intense argument with her parents - our researcher walked away from her parents, refusing to continue dinner or to talk to her parents. Weeping quietly, she was considering hurting herself to make her parents regret. Then, she happened to see the device not far from her and thus decided to put it on to record the emotion.

We created a notebook called training.ipynb by selectively copying, pasting, and editing the example code from the fastai tutorial. In the code, we specified that our input were images, and our output is categories (instead of continuous numbers); the inputs are stored in the folder

"Emotion4"; 20% of the images in Emotion4 are randomly selected to be in the validation set (still involved in the training process), and the selection is the same for the same dataset each time we run the codes (eg. Suppose we have i1, i2, i3...i10 in the database. If we know that i2 and i4 happened to be selected, we know that i2 and i4 will always be selected for this database. ); the labels are the names of the folders ("relax" and "unstable"); images are resized to 224x224.

The model is tuned 4 times and exported as "model\_Mar4.pkl", which can be used in another notebook we named "Prototype\_Test.ipynb". Later, in order to use the trained models offline without a GPU, or graphing processing unit (using .NET), we exported the model in .onnx format (Apr15.onnx) and exported the labels in .txt format (Apr15.vocab.txt).

#### 4.5 Validation

Again, selectively copying, pasting, and editing the example code from the fastai tutorials, we created another notebook called "Prototype\_Test.ipynb". After importing a trained model (model\_Mar4.pkl), we can directly upload images from the local desktop and see what it is categorized to.

In order to test out our current prototype, we gathered four pieces of data that were not in any way used in the training process of model\_Mar4.pkl - two relaxed piece and two unstable piece. Each piece was splitted into multiple segments and converted to images with Processing (Graphing\_Individual\_V3). The images are stored in a folder called "validation\_May24". We uploaded the images one-by-one into fastai without labels (Prototype\_Test.ipynb)(Figure 4.3). The results were manually recorded in test\_result\_P2.txt (an overall accuracy of 84.7%, with 77.8% correct for unstable#1, 85.7% correct for unstable#2, 77.3% correct for relax#1, 100% correct for relax#2) (Figure 4.4).

```
In [3]: path = Path('')
learn_inf = load_learner(path/'model_Mar4.pkl')
```

```
In [4]: #learn_inf.predict('test_images/annoyed.png')
```

```
In [5]: #hide
#testing if I understood it correctly
btn_upload = widgets.FileUpload()
btn_run = widgets.Button(description='Classify')
out_pl = widgets.Output()
lbl_pred = widgets.Label()

def on_click_classify(change):
    img = PILImage.create(btn_upload.data[-1])
    out_pl.clear_output()
    with out_pl: display(img.to_thumb(128,128))
    pred,pred_idx,probs = learn_inf.predict(img)
    lbl_pred.value = f'Prediction: {pred}; Probability: {probs[pred_idx]:.04f}'

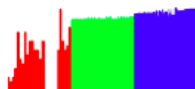
btn_run.on_click(on_click_classify)

VBox([widgets.Label('Upload your images!'),
      btn_upload, btn_run, out_pl, lbl_pred])
```

Upload your images!

Upload (1)

Classify



Prediction: unstable; Probability: 0.9736

**Figure 4.3:** Part of the codes in Prototype\_Test.ipynb. The widget below the codes shows the result from inserting the 9th frame of the unstable piece

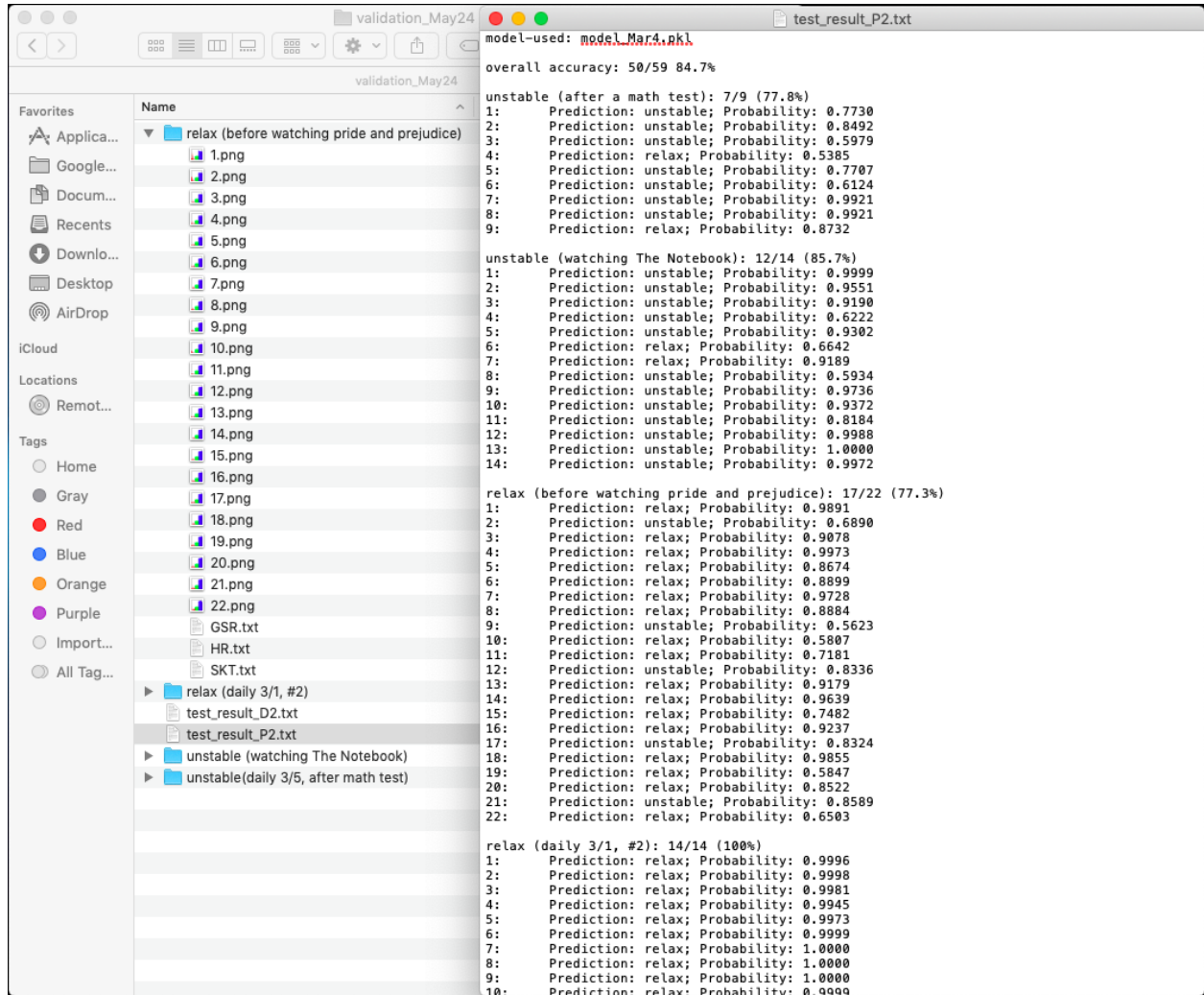


Figure 4.4: validation\_May24 and test\_result\_P2.txt

#### 4.6 Automation

In order to use the trained model offline and without a GPU, we used the example codes provided in the article “Cross-platform inference using fast.ai models” (tapesoftware). We created a new folder and named it “Emotion Classifier”. The edited example codes (Program.cs), the model (Apr15.onnx), the vocabulary (Apr15.vocab.txt), and any other files provided by .NET are stored in this folder. To classify an image, we simply need to run the line command “dotnet run [image path]”. This was much more convenient for automation compared to the Prototype\_Test.ipynb notebook, for which we must login to Paperspace, start a GPU, run the codes, and manually select and upload an image.

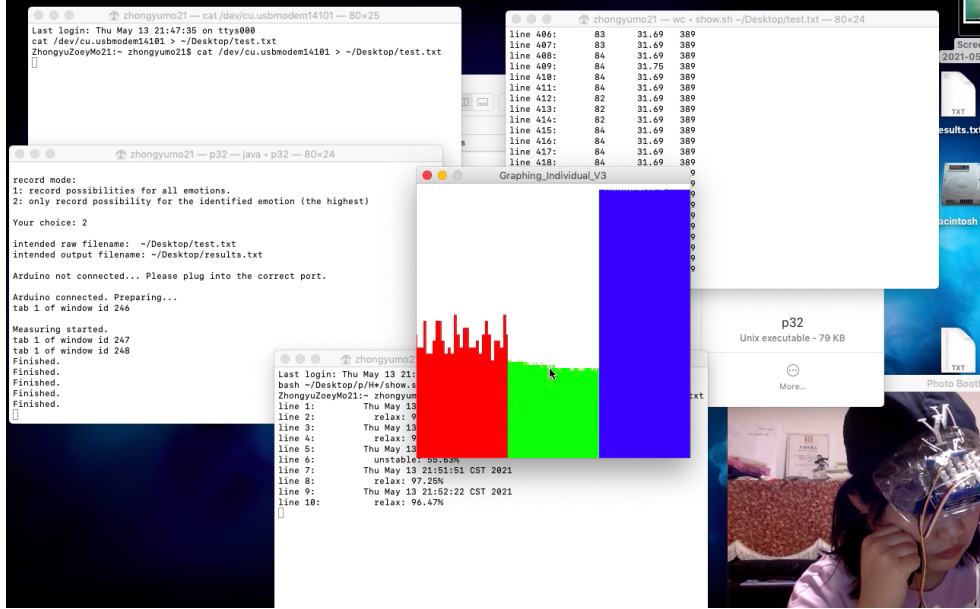
We wrote a C++ program (Prototype3\_3.cpp) to connect all three parts in our prototype: Arduino sensors that take the data, Processing program that processes and converts the data into images, and .NET project that uses the trained model to classify an image into either “unstable” or “relax”.

When the user run the compiled C++ program (p33), the user will be asked to choose from 1. Taking new data with the device and 2. Using an existing data file. Then, the user will be asked to choose a record mode: 1. for each segment, record the possibility for each emotion (which is the raw output from .NET) and 2. for each segment, record the time when its classification began and record the one emotion with highest possibility. Then, the user will be asked to enter the path for the intended raw data file, results file, and image folder. In order to avoid the confusion caused by repeated names, all related files from past running will be removed before new ones are generated.

If the user chose to use an existing data file, then what is left is simply to wait for the process to be done, after which the user will be asked whether to terminate the program or start another round, and the program will close all other terminal windows except for the main one.

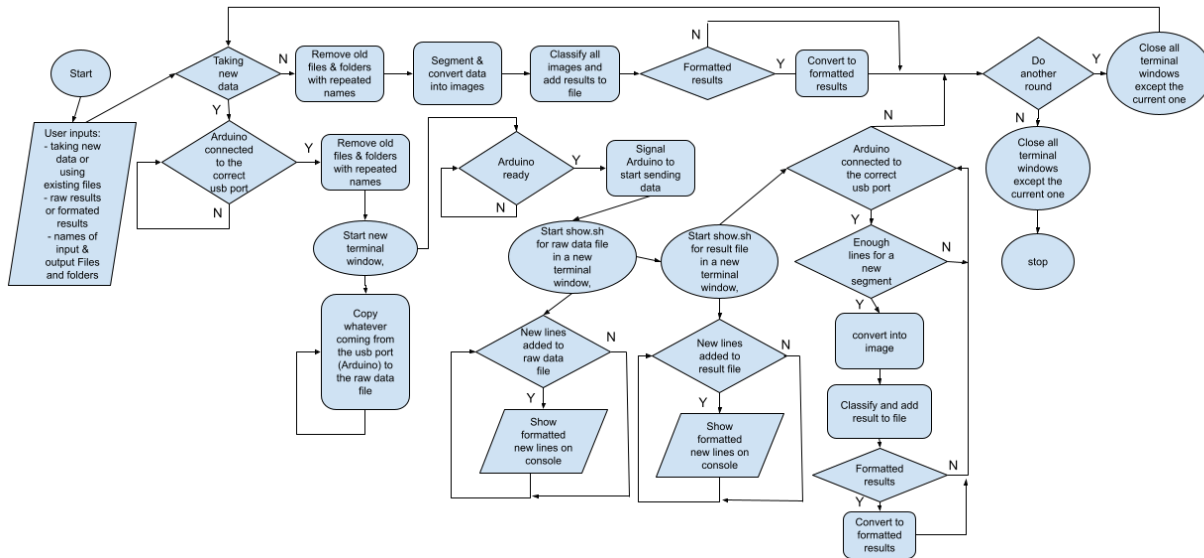
If the user chose to take new data, a loop will start when the device is correctly connected to the usb port on the computer (if not yet connected, a reminder will be sent to the user). After the device is set up and while the device is still correctly connected, there will be 4 terminal windows opened - a main window that interacts with the user and controls everything, showing the warnings and error messages if exists; a second window that continues to store the data from the device to the user-indicated raw data file; a third window that continues to check if new lines are added to the user-indicated raw data file and prints the new lines to the console in a more readable format (marked by line numbers); and a fourth window that does the same task as the third window except for targeting at the user-indicated result file instead of the raw data file. When there are at least 121 lines in the raw data file, the last 121 lines of the file will be copied to the processing program (Graphing\_Individual\_V3) as data.txt (after waiting for at least 20 lines of non-repeating space) to be converted into images and sent to the .NET project to be classified. The results are recorded in the user-indicated result file. The loop will end if the connection is broken, after which the user will be asked whether to terminate the program or start another round, and the program will close all other terminal windows except for the main one.

Command line is frequently used in the C++ program to control other applications and files. In order to use multiple terminal windows to execute multiple tasks synchronously, shell scripts ("show.sh", "kill.sh", "convert.sh") are written and used in the C++ program. "show.sh" continuously checks if new lines are added to the given file and prints the new lines to the console in a more readable format (marked by line numbers). "kill.sh" closes all the terminal windows except the current one. "convert.sh" converts the raw result file (record mode 1) to a processed result file (record mode 2).



**Figure 4.5:** A screenshot of p32 running.

The photo booth window is not related to the prototype and is open only to show what the user looks like while p32 runs. p32 and p33 look the same when they run - the only difference is that p33 stores all images even for continuous measurement of data, while p32 replaces the old images with new images for continuous measurement.



**Figure 4.6:** A flowchart for Prototype3\_3.cpp

## 5.0 Discussion

To further validate outside the AI, we hope to exercise our software's ability to be versatile across every single future user due to its individuality. Since differing intervals and tolerances are

what every person's emotions have, our software can continue to grow more and more intelligent the longer the users feed our software data.

We hope to build a more nuanced classification and alert system. Instead of only stating whether the user is unstable or not, we want to find a more acceptable way to remind the users to be careful with what they say and how they react or process their feelings. Also, we hope to further specialize our classifications and alerts: for instance, instead of simply categorizing an emotion as unstable, we want to further categorize it into the "don't hurt yourself" category and "don't hurt others" category. We also hope to expand the number of categories our AI can identify by making it differentiate what it feels like to be awkward rather than upset, and depressed rather than sad.

Although our original target market is teenage users of social media, the applications of our project can go beyond this range. If we make the data-taking device more portable, for instance, by using mature products from medical companies, as did in the related study #1(2004), the emotion detection and corresponding reminder system can continue helping users all day. Whenever an user's emotion becomes unstable, such as before starting an argument with friends or generally feeling annoyed to an extent of wanting to yell at anyone nearby, our device would be able to give advice and help users avoid regrettable behaviors.

If our product were to be widely used by other social media users, the internet can become a more reliable and better space to exchange ideas by decreasing the number of impulsive, feeling-based decisions people make. People will become more mindful of their actions online and choose to not like clickbaited content, or content whose main purpose is to attract attention and encourage visitors to click on a link to a particular web page, decreasing the circulation of that particular post, pulling it out of the social media platform's algorithm and therefore encouraging the circulation of much more reliable information sources on social media such as information from the CDC for coronavirus alerts or multinational news-based pay television channel such as *CNN* or *The New York Times* encouraging nearing-unbiased information and a more reliable place to find information on social media. In addition to this, social media will become a more pragmatic and reasonable space in terms of its information and types of opinions one might find, resulting in a more pragmatic and reasonable way of thinking for teenagers to develop and grow with a peace of mind, encouraging education, learning, emotional stability, creativity, and innovation.

## 6.0 References

Dzedzickis, A., Kaklauskas A., Bucinskas V. (2020). Human Emotion Recognition: Review of Sensors and Methods. NCBI.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7037130/>

Gatti E., Calzolari E., Maggioni E., Obrist M. Emotional ratings and skin conductance response to visual, auditory and haptic stimuli. *Sci. Data*. 2018;5:180120.  
doi: 10.1038/sdata.2018.120.

Jeremy et al. (2020). Fastai. GitHub.

<https://github.com/fastai/fastai>

Koelstra, S., Muhl, C., Soleymani, M., Jong-Seok Lee, Yazdani, A., Ebrahimi, T., ... Patras, I. (2012). DEAP: A Database for Emotion Analysis ;Using Physiological Signals. IEEE Transactions on Affective Computing, 3(1), 18–31. doi:10.1109/t-affc.2011.15

Langereis, G., (2010). Photoplethysmography (PPG) system. Seeedstudio. [https://github.com/SeeedDocument/Grove-Ear-clip\\_Heart\\_Rate\\_Sensor/raw/master/res/PPG%20Sensor%20System.pdf](https://github.com/SeeedDocument/Grove-Ear-clip_Heart_Rate_Sensor/raw/master/res/PPG%20Sensor%20System.pdf)

Lisetti, C.L., Nasoz, F. Using Noninvasive Wearable Computers to Recognize Human Emotions from Physiological Signals. EURASIP J. Adv. Signal Process. 2004, 929414 (2004). <https://doi.org/10.1155/S1110865704406192>

Tamura T., Maeda Y., Sekine M., Yoshida M. Wearable Photoplethysmographic Sensors—Past and Present. Electronics. 2014;3:282–302. doi: 10.3390/electronics3020282.

tape software. (2020, September 8). *Cross-platform inference using fast.ai models*. Cross-platform inference using fast.ai models - tape software. <https://tapesoftware.net/fastai-onnx/>

## 7.0 Appendices

- Appendices folder
  - <https://drive.google.com/drive/folders/1idEURFLuGtQaxXTXhQIYtJ1n5kkpR0oD>
- Data collection
  - Movie Data Collection:  
<https://docs.google.com/spreadsheets/d/1RzkT50k20-rVfM1vA60ICk7P6c6n2ae7AAGfraE4q6Q/edit?usp=sharing>
  - Daily Data Collection:  
<https://docs.google.com/spreadsheets/d/1MpG7qPAE9k2tVzXZZtxh0nnCycWDZQRlcUP2Dxpgme4/edit?usp=sharing>
- Jupyter notebooks
  - training.ipynb:  
<https://drive.google.com/file/d/1wKdGcWtM7kCFIwtq0Za-pmysUt3eds8z/view?usp=sharing>
  - Prototype\_Test.ipynb:  
<https://drive.google.com/file/d/1qNiFXL9Gc6GB7s6t11WVeYJrfzpFUywu/view?usp=sharing>
- Arduino codes
  - The code is different for us because we have different sensors to set up. If we keep them in one file as we did in the past, we will need to tell the prototype who is using

it each time we test it, or add new codes to record who the user is, which is not very convenient.

- Zoey: [https://drive.google.com/file/d/1mCV6XTXM6jRarG5jkJhqcBaf\\_vPhgUiy/view?usp=sharing](https://drive.google.com/file/d/1mCV6XTXM6jRarG5jkJhqcBaf_vPhgUiy/view?usp=sharing)
- Emily: [https://drive.google.com/file/d/1bY8mN2wU2YbdD9smDPhVKUmZ-0X9VK2\\_/view?usp=sharing](https://drive.google.com/file/d/1bY8mN2wU2YbdD9smDPhVKUmZ-0X9VK2_/view?usp=sharing)
- Processing code (Graphing\_Individual\_V3)
  - (need to be stored in Processing folder for the line commands in C++ code to run it)
  - [https://drive.google.com/file/d/1E6\\_\\_3q5ykXhGmJLEbC4MaqAwiGEsVc0f/view?usp=sharing](https://drive.google.com/file/d/1E6__3q5ykXhGmJLEbC4MaqAwiGEsVc0f/view?usp=sharing)
- Project folder
  - [https://drive.google.com/file/d/1YsJUatsZy-YNQSYPK20L8\\_FWewCTnun/view?usp=sharing](https://drive.google.com/file/d/1YsJUatsZy-YNQSYPK20L8_FWewCTnun/view?usp=sharing)
  - This folder contains:
    - Prototype3\_3.cpp
    - p33
    - convert.sh
    - kill.sh
    - show.sh
    - test.sh
    - EmotionClassifier
      - This is a project folder for .NET
      - It contains codes (Program.cs), the trained model (Apr15.onnx), the labels (Apr15.vocab.txt), and other files generated by .NET
    - GraphingV4\_2\_Apr\_20
      - GraphingV4\_2\_Apr\_20 is a folder we used during training.
      - It contains the processing code we used to process mass data (GraphingV4\_2\_Apr\_20.pde), and the data we used for training.
      - The data are stored in the numbered folder in GraphingV4\_2\_Apr\_20/relax and GraphingV4\_2\_Apr\_20/unstable.
        - The notes.txt in each numbered folder explains where each manually selected piece is from.
        - The complete sets of data can be found in the two Data Collection google spreadsheets linked earlier.
      - The images we used in Emotion4 are stored in GraphingV4\_2\_Apr\_20/images/AllThree
    - validation\_Apr15
    - validation\_May24