An *AUesome* Therapy Activity Invention:

Gensic: Generative Music Therapy Based on Physiological Properties for Individuals on the Autism Spectrum

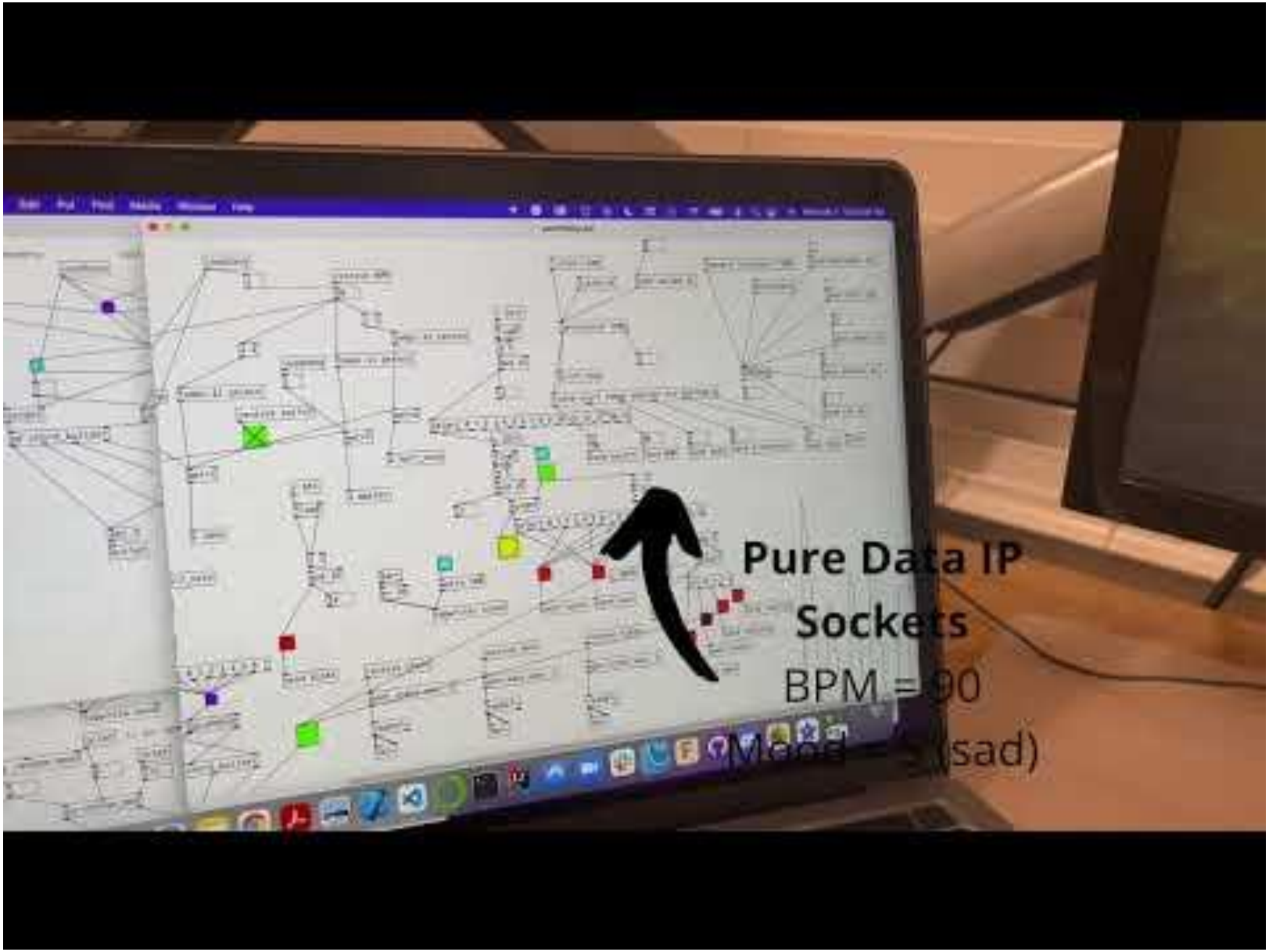
by Anshul Gupta

Awards Won:
Synopsys Science & Engineering Fair Championship:

- Synopsys Project Qualifier
- Honorable Mention
- Certificate of Achievement
- AIAA Intelligent Systems Award

California State Summer School for Mathematics & Science:

- Brian Eno Award
- Pulse Prize Award



Project Video

youtu.be/i4RnfhCmmm0

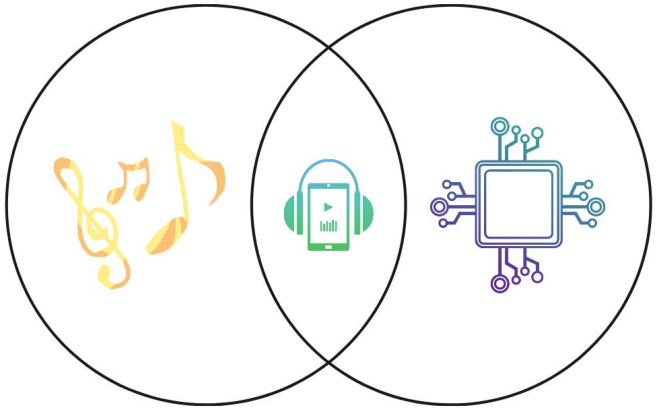
Purpose

Gensic is an internet of things (IoT) wearable device that provides music therapy to individuals with Autism Spectrum Disorder (ASD). It is a therapy activity made for AUesome LLC.

- Ameliorates physical & mental agitation
- Eases overstimulation
- Enabling emotional processing
- Helps with communication & expression

I was inspired to create this technology from my personal involvement in the neurodiverse community and my interest in music and technology. After leading a startup to make therapy more accessible for children with autism, assisting neurodivergent individuals, receiving licenses from UNC AFIRM on autism therapy training, meeting with therapists, and my own research, I recognized the gap in autism music therapy and the power technology has to transform it into an impactful resource.

Individuals with ASD benefit from generative music therapy through its simple and accessible process. Gensic even goes beyond these fields by leveraging a physiological factor – emotion – and a biometric one – heartbeat – to generate music in real-time reflective of users' statuses. This not only allows for music therapy, but also self-expression.



Music and Technology!

Introduction

Despite 1 in 54 children having autism spectrum disorder, 50% have access to only school-based treatment, and 17% have no access to any treatment. Existing systems require a deep foundation of musical knowledge and lack flexibility, while having a plethora of limitations:

- Geographic and physical barriers
- Expensive financial cost and strain
- Lacks Parental Implemented Interventions (PII)
    - Official process for parents to assist their child in therapy
- Non-continuous outside of clinics
- Lacks personalization

Expected to grow 7% by 2028, music therapy is an expanding field to assist with stimulation and expression through a “universal language”. It focuses on behavioral skills to aid cognitive functions via emotional expression.

Generative music

- Term coined by Brian Eno in 1996
- Systems compose their own music
- Music that is methodically generated to produce various harmonic sounds
- Always new & unique when produced, not controlled directly by a composer
- Can be applied to autism music therapy and greatly expand the field

Gensic dives into the interdisciplinary fields of electrical engineering, computer science, autism therapy, and music theory through its advanced music generation, while making therapy an accessible, enjoyable, and expressive process.

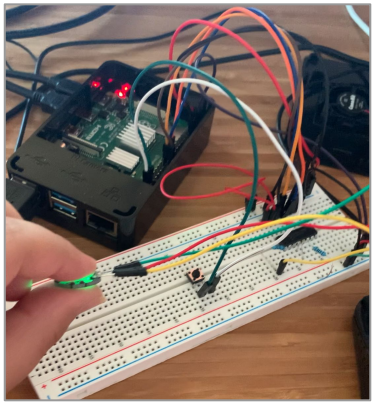
> “Music therapy uses musical experiences and the relationships that develop through them to enable communication and expression, thus attempting to address some of the core problems of people with ASD” (Geretsegger).

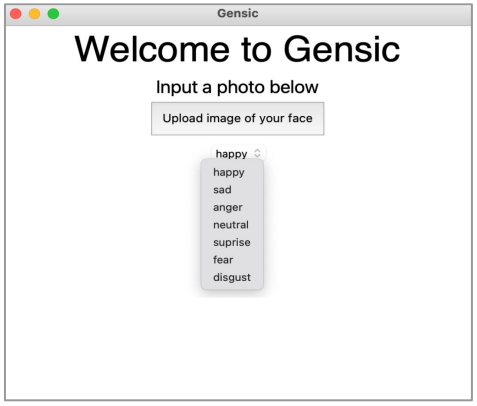
Methods

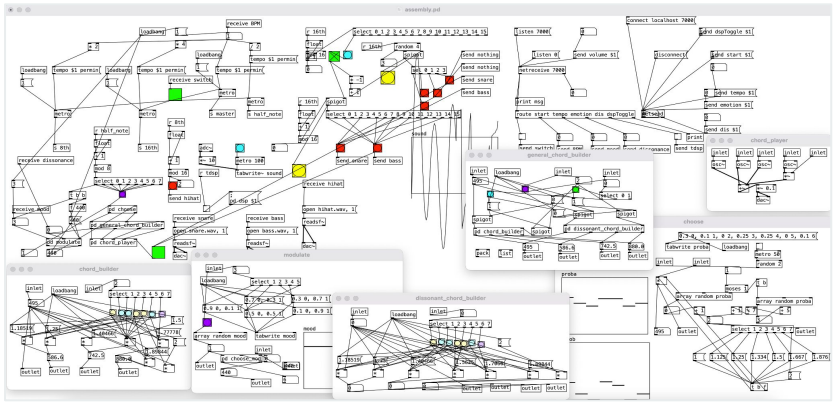
Gensic leverages generative music therapy using an Raspberry Pi, Arduino, Python, C, Machine Learning, Computer Vision, Internet Protocol, Sockets, and Pure Data to generate music and assist with music therapy for individuals with autism by leveraging their physiological properties.

It has three steps:

1. Using a pulse sensor and Raspberry Pi to determine heartbeat, continuously sending this value to Pure Data via IP Sockets.
2. Utilizing a GUI where users upload a photo of themselves, whose emotion is determined and sent to Pure Data via IP sockets.
3. Pure Data generating unique music based on these inputs using advanced music theory in real-time.



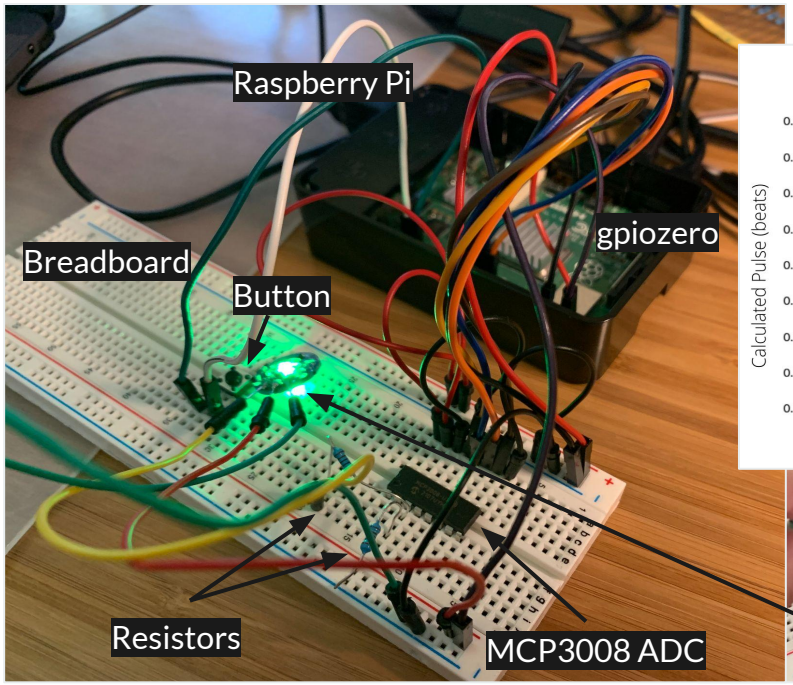


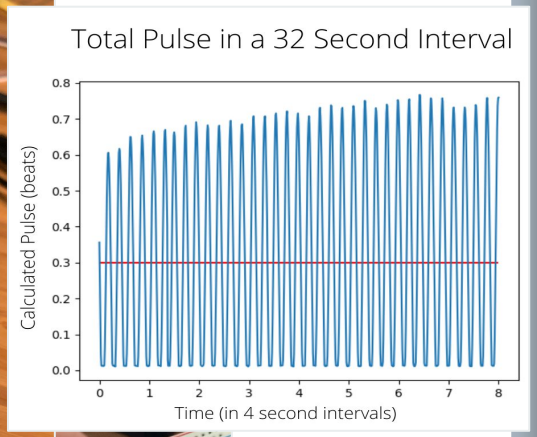


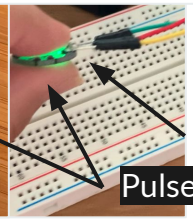
Hardware, Heartbeat, Sockets

Step 1: Determining Heartbeat

With a wearable pulse sensor wired to a breadboard and microcontroller (Arduino or Raspberry Pi) with an analog to digital converter, a Python and C script determine extremas of the users pulse to translate their heartbeat to beats per minute (bpm). BPM is sent over IP sockets to Pure Data every second.







Soldered Cables

Pulse Sensor

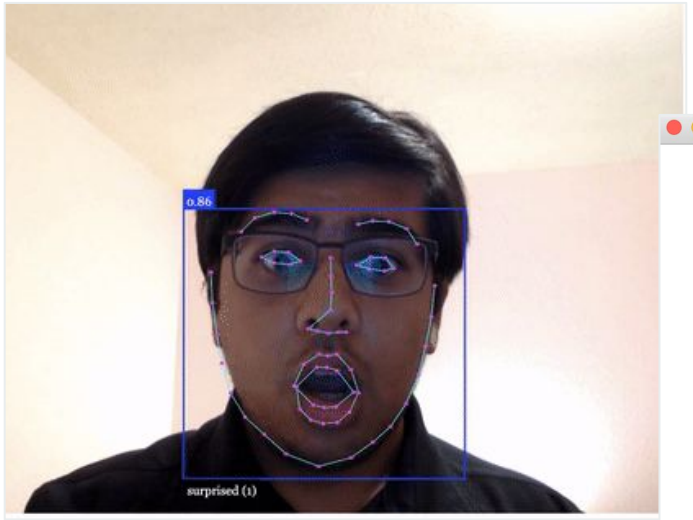
```python
import warnings
import socket
from time import sleep
import math
from gpiozero import Button
from time import time
from secret import IP

def addtime(times):
    if type(times) != list:
        raise(TypeError)
    t = time()
    if len(times) == 0:
        tdiff = 0  # initial seed
    else:
        tdiff = t - times[-1][0]
    return (t, tdiff)

def averagetimes(times):
    averagetime = sum([row[1] for row in times])/float(len(times))
    bpm = (1.0/(averagetime/60.0))
    return (averagetime, bpm)


class PDSocket:
    """PD-PI Socket"""

    def __init__(self, sock=None):
        if sock is None:
            self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        else:
            self.sock = sock

    def connect(self, host, port):
        self.sock.connect((host, port))

    def close(self):
        self.sock.close()

    def send(self, msg):
        pdmsg = ';'.join([f'{k} {v}' for k, v in msg.items()])
        sent = self.sock.sendall(bytes(pdmsg + ';', 'ascii'))
        if sent == 0:
            raise RuntimeError("socket connection broken")

    def receive(self):
        msg = str(self.sock.recv(1024), 'ascii')
        if msg == b'':
            raise RuntimeError("socket connection broken")
        return msg[:-2].split(';\n')


def main(host=IP, port=7000):

    times = []

    def got_pressed():
        times.append(addtime(times))
        if len(times) > 1:
            if times[0][1] == 0 or len(times) > 16:
                del times[0]
            (averagetime, bpm) = averagetimes(times)
            bpmValue = math.floor(bpm)
            print(bpm, bpmValue)
            socket.send({'tempo': bpmValue})


    print(f"> Connecting to '{host}' at port '{port}'")
    socket = PDSocket()
    socket.connect(host, port)

    bpin = 26
    button = Button(bpin)

    button.when_pressed = got_pressed

    # ?inputs
    start = 1  # switch
    emotion = 3  # mood
    dis = 0  # dissonance
    dspToggle = 1  # dsp

    #! killswitch
    #start = 0
    #dsp = 0

    values = {'start': start, 'emotion': emotion, 'dis': dis, 'dspToggle': dspToggle}

    try:
        print("init values")
        socket.send(values)

        while True:
            sleep(10)

        print("done")
    except KeyboardInterrupt:
        print("> Closing socket")
        socket.close()

if __name__ == '__main__':
    main()
```

Socket Code to send BPM

GUI + Computer Vision and Deepface for Emotion

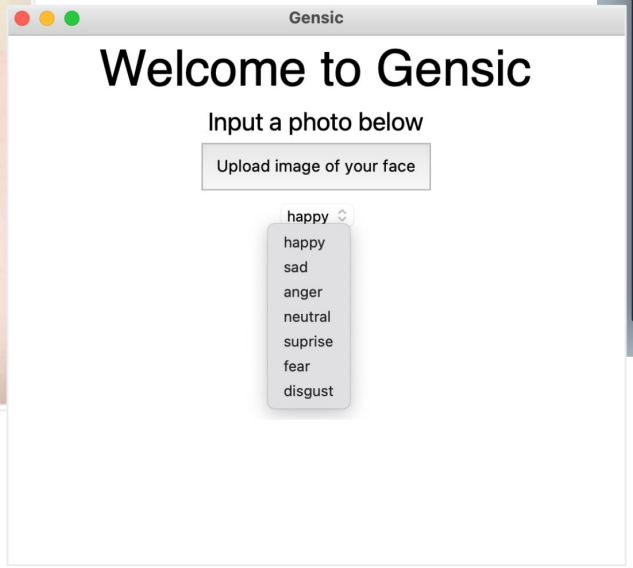
Step 2: Determining Emotion

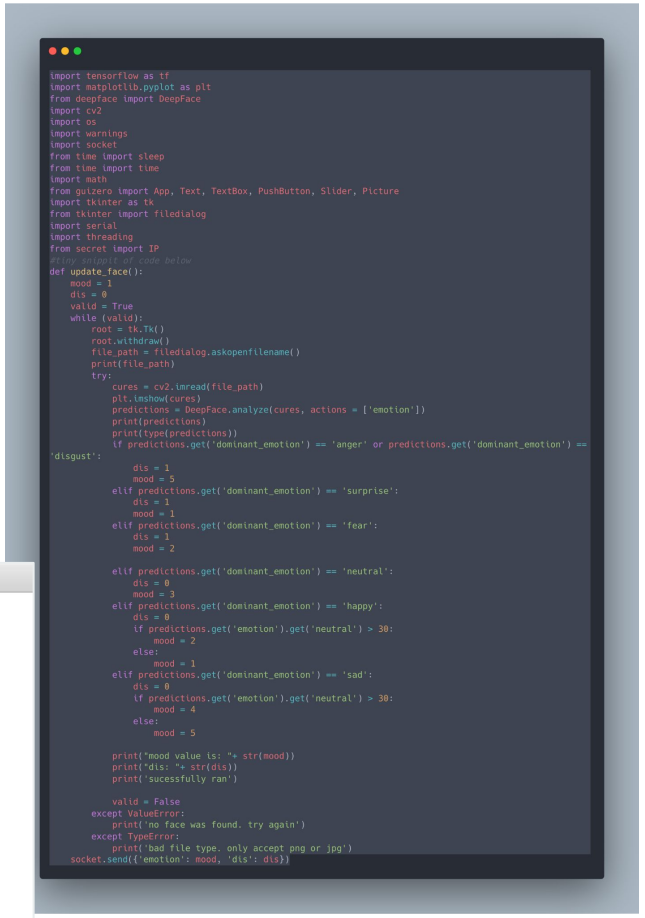
A Graphical User Interface (GUI) where user inputs photographic inputs the user that is analyzed with machine learning (ML) and computer vision (CV) using Tensorflow and DeepFace to determine their current emotional state – happy, sad, angry, disgusted, surprised, fearful, and neutral. Operates on Jupyter Notebook on the user's machine and sends the predicted emotion via IP Sockets to Pure Data once inputted.



Deepface, OpenCV, and Tensorflow for emotion recognition

tkinter with Python for GUI

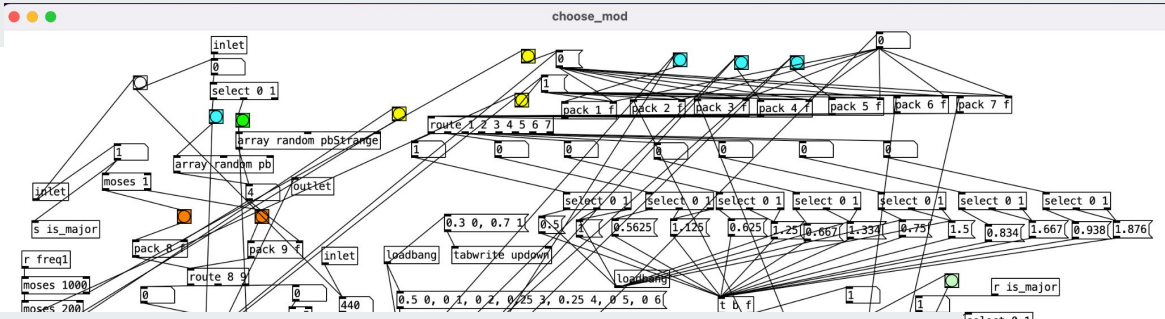




```
import tensorflow as tf
import matplotlib.pyplot as plt
from deepface import DeepFace
import cv2
import os
import warnings
import socket
from time import sleep
from time import time
import math
from guizero import App, Text, TextBox, PushButton, Slider, Picture
import tkinter as tk
from tkinter import filedialog
import serial
import threading
from secret import IP
#tiny snippit of code below
def update_face():
    mood = 1
    dis = 0
    valid = True
    while (valid):
        root = tk.Tk()
        root.withdraw()
        file_path = filedialog.askopenfilename()
        print(file_path)
        try:
            cures = cv2.imread(file_path)
            plt.imshow(cures)
            predictions = DeepFace.analyze(cures, actions = ['emotion'])
            print(predictions)
            print(type(predictions))
            if predictions.get('dominant_emotion') == 'anger' or predictions.get('dominant_emotion') ==
'disgust':
                dis = 1
                mood = 5
            elif predictions.get('dominant_emotion') == 'surprise':
                dis = 1
                mood = 1
            elif predictions.get('dominant_emotion') == 'fear':
                dis = 1
                mood = 2

            elif predictions.get('dominant_emotion') == 'neutral':
                dis = 0
                mood = 3
            elif predictions.get('dominant_emotion') == 'happy':
                dis = 0
                if predictions.get('emotion').get('neutral') > 30:
                    mood = 2
                else:
                    mood = 1
            elif predictions.get('dominant_emotion') == 'sad':
                dis = 0
                if predictions.get('emotion').get('neutral') > 30:
                    mood = 4
                else:
                    mood = 5

            print("mood value is: "+ str(mood))
            print("dis: "+ str(dis))
            print('sucessfully ran')

            valid = False
        except ValueError:
            print('no face was found. try again')
        except TypeError:
            print('bad file type. only accept png or jpg')
    socket.send({'emotion': mood, 'dis': dis})
```

Accessing inputs from GUI with Deepface to output values on sockets

Music Generation



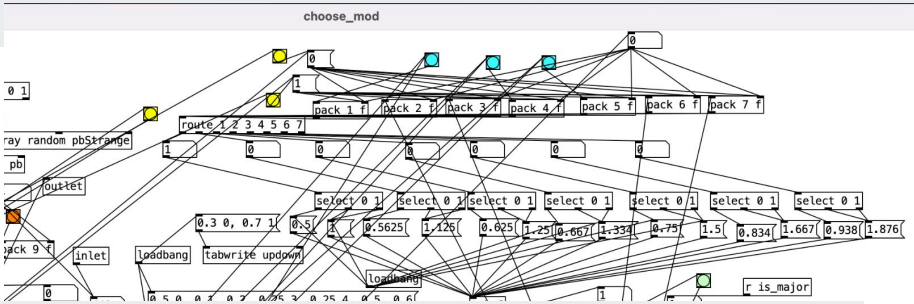
Step 3: Using emotion and heartbeat to generate music

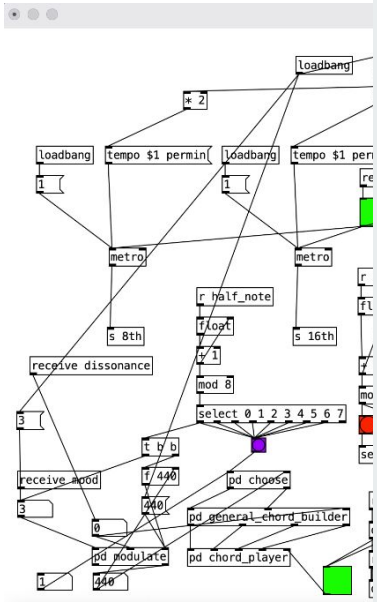
In the internet of things (IoT), heartbeat and emotion values are sent via network internet protocol (IP) sockets to Pure Data (Pd), which, coupled with music theory, generates a drum beat and alters chord progressions by modifying various pitches, in hertz (hz). Pure Data is a visual programming language used for music composition.

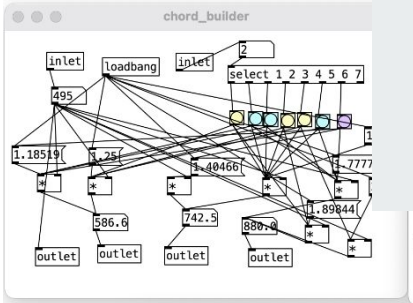
The system begins when the user begins measuring their heartbeat. The python script activities the digital signal processing via sockets to begin music generation, updating the Pd file every second with the new heartbeat, which is rounded down to the nearest 10 to prevent unnecessary fluctuation. The user can also input a photo of themselves on the GUI, which alters how the chords are generated.

a. **BPM:** determines the tempo of the music, while predicted emotion alters major and minor scales, modulates intervals, and adjusts the dissonance and tonality of the chords by altering the pitch of each note. This output of generated music reflects the individual's current physiological state.
b. Notes start at 440 hz and are modified in increments depending on the emotion

Music Generation (cont'd.)



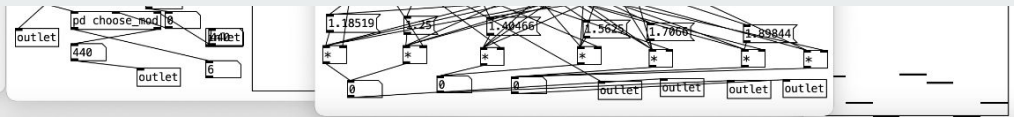




Chords: built from a root, in various scale degrees. To build chords, Gensic builds a 3rd, 5th, and 7th interval off the frequency of the root in hertz using *just intonation*, allowing for coherent and tonal chord progressions. To add harmonic interest, the keys modulate every few measures with a random probability of changing.

Western music has four primary chord types: Diminished, Minor, Major, and Augmented. Gensic generates diminished chords to sound crunchy and dissonant, while augmented ones to sound uncomfortable and wide. Major and minor sound the most pleasing, but major gives a brighter and more exuberant tone than minor.

a. **Emotion:** surprise and fear involve wide facial expressions, so the Pure Data file provides a higher probability for augmented chords to generate. Contrarily, anger and disgust, narrow expressions, are represented with a large ratio of diminished chords. Pure Data increases the probability of choosing minor chords and keys to make a song feel more sad with low energy and increases the probability of choosing major chords and keys to make a song more happy. Neutral has a mix of everything to ensure ambiguous tonality in the output.



Results

Gensic's prototype successfully met its goal to aid with music therapy by leveraging heartbeat and emotion through its accurate generation of music reflective of my current psychological statuses. As I was only able to validate it on myself, I was limited in my ability to publicly test the project, but found that it properly detected my heartbeat and emotion and sent those values in real-time to Pure Data via IP sockets. It then generated music with the massive Pure Data file using these inputs by modulating chords at various pitches with a proper harmonic melody; for every heart rate and emotion, it generates a unique set of frequencies. I personally found the output to be expressive of how I felt and relieving. Its potential therapeutic effect could be applied by a therapist or parent through its musical generation.

Actual vs Measured Result and Accuracy of BPM Detection by Pulse Sensor

TEST	ACTUAL HEARTRATE IN BEATS PER MINUTE	PREDICTED HEARTRATE IN BEATS PER MINUTE	BPM % ERROR
1	65	63	3.08%
2	83	78	6.02%
3	73	70.	4.11%
4	94	89	5.32%
5	75	77	2.67%
6	69	61	11.6%
7	74	71	4.05%
8	93	88	5.38%
9	77	78	1.30.%
10	84	81	3.57%
			AVERAGE: 4.71%

Inputted vs Predicted Result and Accuracy of Facial Emotion/Expression

TEST	INPUTTED EMOTION	PREDICTED EMOTION AND % ACCURACY
1	NEUTRAL	NEUTRAL (98%)
2	ANGER	ANGER (96%)
3	FEAR	FEAR (95%)
4	SURPRISE	SURPRISE (93%)
5	SAD	SAD (98%)
6	HAPPY	HAPPY (99%)
7	DISGUST	DISGUST (95%)
8	FEAR	FEAR (96%)
9	SURPRISE	SURPRISE (94%)
10	SAD	SAD (97%)

Pure Data Chord Root and Frequency (Hz)

TEST	CHORD ROOT	HERTZ (Hz)
1	C	260
2	G	780
3	D	590
4	A	440
5	E	260
6	B	490
7	F#	740
8	D♭	550
9	A♭	420
10	E♭	620

Discussion

Hardware: After designing the Raspberry Pi circuitry with the ADC and Python, I found that an Arduino could detect analog inputs and had a pulse sensor library. I built an alternative method with the Arduino and C, which uses serial communication instead of internet protocol sockets.

Heartbeat: Often, the heartbeat was slightly less than the expected input. There could have been several errors, including calculating an inaccurate input pulse, not tightly gripping the sensor, having a fluctuating heartbeat in between measurement and calculation, or having a faulty sensor.

Dataset: The MTCNN model was not completely exact as the dataset is not accustomed to individuals with autism; to expand this project, I could implement a more tailored dataset.

Generation: The Pure Data file was so massive that it was unable to run on the Raspberry Pi, instead operating on a personal computer. Creating objects for chord modulation and dissonance, which manipulated every note, proved to be helpful for adding harmonic interest for the music.

Applicability: Compared to existing systems, Gensic creates an affordable at-home process to music therapy using technology and generative music, while helping with self-expression by composing unique music tailored to an individual. The numeric results signify the accuracy of determining physiological characteristics, but hearing and feeling the audio shift to my internal feelings without me making it do so felt soothing, personalized, expressive.

Conclusion and Further Research

Gensic was able to consistently and accurately generate music that assists with therapy. Originally made to help individuals with autism, it can further be expanded to assist neurotypical individuals and people with other cognitive disabilities. Through my discussions with individuals with autism and music therapists, the following are several factors to consider in the future engineering of Gensic:

- Apply more physiological characteristics
  - Sweat, touch, energy, time of day, etc.
- Train dataset to more accurately detect individuals with autism
- Pulse detection in wearable smart watches to determine music pace
  - Condensing hardware into a smaller package, limited with budget
  - Physiological personalization in music streaming services
- Expand machine learning in generating music to create arpeggios, shift octaves, etc.

The World Federation of Music Therapy (WFMT) states that generative music has an untapped **potential** in the therapeutic field:

> "Generative music systems [were] initially developed for 'on hold' music services and public places and [should be] developed[ed]... for the domestic environment and health services where music is used for therapy" (Brown).

Endless **applicability** for generative music, that is often unexplored in the therapeutic field but could shift how music therapy is conducted and how music is interpreted by patients as a whole.

GENSIC

Join me in making music therapy make accessible and impactful for individuals on the autism spectrum, one rhythm at a time. **Thank you for your time.**

References

Brown, Paul. “Is the Future of Music Generative? .” Music Therapy Today, World Federation of Music Therapy, Apr. 2005, https://www.wfmt.info/Musictherapyworld/modules/mmmagazine/issues/20050411083630/20050411095852/MTT6_2_Brown2005.pdf. Accessed 12 Dec. 2021.

Cervellin, Gianfranco, and Giuseppe Lippi. “From Music-Beat to Heart-Beat: A Journey in the Complex Interactions between Music, Brain and Heart.” European Journal of Internal Medicine, Elsevier, 22 Mar. 2011, https://www.sciencedirect.com/science/article/abs/pii/S0953620511000513. Accessed 3 Jan. 2022.

Geretsegger, Monika, et al. “Music Therapy for People with Autism Spectrum Disorder.” U.S. National Library of Medicine, Cochrane Database System Review, 17 June 2014, https://pubmed.ncbi.nlm.nih.gov/24936966/. Accessed 12 Dec. 2021.

Hosken, Dan. Introduction to Music Technology. 2nd ed., Routledge, 2014. https://www.taylorfrancis.com/books/mono/10.4324/9780203539149/introduction-music-technology-dan-hosken. Accessed 10 Nov. 2021.

Irawan, Yuda, et al. “Detecting Heart Rate Using Pulse Sensor as Alternative Knowing Heart Condition.” Journal of Applied Engineering and Technological Science (JAETS), 1 Dec. 2019, https://journal.yrpipku.com/index.php/jaets/article/view/16. Accessed 8 Jan. 2022.

Mungan, R. Yagiz. “Introduction to Algorithmic Composition and Pure Data.” Interactive Art | XR | Emerging Technologies, Pure Data, 22 Mar. 2014, http://yagizmungan.com/ACinPD.pdf. Accessed 11 Nov. 2021.

Rump, Keiran M., et al. “The Development of Emotion Recognition in Individuals with Autism.” Society for Research in Child Development, John Wiley & Sons, Ltd, 14 Sept. 2009, https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8624.2009.01343.x. Accessed 10 Nov. 2021.

Straus, Joseph N. “Music Therapy and Autism: A View from Disability Studies.” CUNY Academic Works, CUNY Graduate Center, 2014, https://academicworks.cuny.edu/gc_pubs/415/. Accessed 4 Oct. 2021.