# Resultados Modelos Frustracion

## Virtual Heart Rate

A continuación, se muestran las gráficas obtenidas del siguiente modelo, utilizando un video almacenado:

from pyVHR.analysis.pipeline import Pipeline

import matplotlib.pyplot as plt

from pyVHR.analysis.multi\_method\_suite import TestResult

import heartpy as hp

import matplotlib.pyplot as plt

from filteringHR import filter\_signal

import time

inicio = time.time()

pipe = Pipeline()

times, BPM, uncertainty = pipe.run\_on\_video(r"Video el 26-4-22 poca luz.mov",

roi\_approach="patches",

roi\_method="convexhull",method='cpu\_CHROM',bpm\_type='welch',

pre\_filt=False,verb=1)

#bpm welch , psd\_clustering #roi\_approach hol , patches

#roi\_method faceparsing , convexhull

# pre\_filt True , False

#post\_filt True , False

plt.figure()

plt.plot(times, BPM)

plt.fill\_between(times, BPM-uncertainty, BPM+uncertainty, alpha=0.2)

plt.show()

fin = float(time.time())

print(fin-inicio)

Gráfico, Gráfico de líneas

Descripción generada automáticamente

A partir del código anterior, obtenemos un array llamado BPM con los valores obtenidos por segundo del video

Texto

Descripción generada automáticamente

Del cual, luego calculamos los valores de R-R (Ritmo Regular) que utilizaremos para calcular la frecuencia de respiración (Breath Rate), para ello es necesario aplicar la siguiente formula: R-R=60000/BPM

Gráfico, Gráfico de líneas

Descripción generada automáticamente

### Breath Rate

A continuación, definimos la función que usaremos para calcular el breath rate extraída de la librería heartpy, específicamente del ejecutable analisis.py

def calc\_breathing(rrlist, method='welch', filter\_breathing=True,

bw\_cutoff=[0.1, 0.4], measures={}, working\_data={}):

#resample RR-list to 1000Hz

x = np.linspace(0, len(rrlist), len(rrlist))

x\_new = np.linspace(0, len(rrlist), np.sum(rrlist, dtype=np.int32))

interp = UnivariateSpline(x, rrlist, k=3)

breathing = interp(x\_new)

#print('this is x',x)

#print('this is x\_new:',x\_new)

#print('this is interp:', interp)

#print('this is breathing:', breathing)

if filter\_breathing:

breathing = hp.filtering.filter\_signal(breathing, cutoff=bw\_cutoff,

sample\_rate = 1000.0, filtertype='bandpass')

if method.lower() == 'fft':

datalen = len(breathing)

frq = np.fft.fftfreq(datalen, d=((1/1000.0)))

frq = frq[range(int(datalen/2))]

Y = np.fft.fft(breathing)/datalen

Y = Y[range(int(datalen/2))]

psd = np.power(np.abs(Y), 2)

elif method.lower() == 'welch':

if len(breathing) < 30000:

frq, psd = welch(breathing, fs=1000, nperseg=len(breathing))

else:

frq, psd = welch(breathing, fs=1000, nperseg=np.clip(len(breathing) // 10,

a\_min=30000, a\_max=None))

elif method.lower() == 'periodogram':

frq, psd = periodogram(breathing, fs=1000.0, nfft=30000)

else:

raise ValueError('Breathing rate extraction method not understood! Must be \'welch\' or \'fft\'!')

#find max

measures['breathingrate'] = frq[np.argmax(psd)]

working\_data['breathing\_signal'] = breathing

working\_data['breathing\_psd'] = psd

working\_data['breathing\_frq'] = frq

return measures, working\_data

Luego ejecutamos la función para los distintos métodos utilizados para la extracción de los BPM, y además los compararemos con los métodos "welch" y "periodogram" para calcular el breath rate

import numpy as np

import pandas as pd

from scipy.interpolate import UnivariateSpline

from scipy.signal import welch, periodogram

measures, working\_data=calc\_breathing(rr\_list\_cpu\_CHROM, method='welch', filter\_breathing=True,

bw\_cutoff=[0.1, 0.4], measures={}, working\_data={})

Gráfico, Gráfico de líneas

Descripción generada automáticamente

# Conclusiones

Actualmente de estas librerías se obtienen valores bastantes cercanos a ala realidad, tanto para el vitrual heart rate como para el breath rate. La desventaja de momento es que solo pueden utilizarse videos almacenados y no contenido en vivo como una web cam.

### ¿Cuáles son las futuras mejoras para realizar?

Una mejora que podría implementarse es ver la forma de realizar el análisis de VHR y BR con una web cam, además se hicieron varios benchmark entre los distintos métodos para calcular tanto el VHR y BR, pero aún no se define cual de ellos seria mejor. Aunque cabe mencionar que la diferencia de valores entre los métodos es pequeña.

### Premisas aplicadas

1. El virtual heart rate utiliza el método rPPG , el cual mide pequeños cambios de color en la piel para obtener el volumen de sangre por minuto (BVP) y a partir de ello se calculan los latidos por minuto (BPM)
2. En el caso del breath rate no se pudo conseguir una librería útil que lo midiera, por lo que buscaron formular para obtener la frecuencia cardiaca a partir de los BPM, que en caso de una persona sana debería ser igual.
3. Luego con la frecuencia cardiaca se obtiene los valores de intervalos regulares (R-R) utilizados para luego medir la frecuencia de respiración con la función antes mencionada.