

PROJETO 1 – ADAPTAÇÃO AO BRILHO E DISCRIMINAÇÃO

João Vitor R Baptista

15/0013329

Programa de Engenharia

Eletrônica

Faculdade Gama -

Universidade de Brasília

RESUMO

O projeto 1 de processamento digital de imagens tem como objetivo realizar um experimento clássico para observar a capacidade do sistema visual humano na discriminação de níveis de cinza de uma imagem.

1. INTRODUÇÃO

A habilidade do olho para discriminar mudanças de intensidade da luz em qualquer nível específico de adaptação também é de considerável interesse. Um experimento clássico, usado para determinar a capacidade do sistema visual humano na discriminação de brilho, consiste em fazer uma pessoa olhar para uma área plana e uniformemente iluminada que seja grande o bastante para ocupar todo o campo visual.[1]

Se a iluminação de fundo for mantida constante e a intensidade da outra fonte agora puder variar de forma incremental do imperceptível ao sempre perceptível, em vez de piscar como um flash, o observador típico poderá discernir um total de uma a duas dúzias de variações de intensidade. De uma forma geral, esse resultado está relacionado ao número de diferentes níveis de cinza que uma pessoa pode ver em qualquer ponto de uma imagem monocromática [1].

Dois fenômenos demonstram claramente que o brilho percebido não é uma simples função da intensidade [1]. Geralmente o sistema visual humano tende a subestimar ou superestimar a intensidade de brilho nas regiões de contorno que são descritas pelas chamadas *bandas de Mach*. Já no segundo fenômeno, conhecido como contraste simultâneo, descreve que dependendo do fundo do objeto que sendo avaliado interfere na percepção do brilho, ou seja, quanto mais o fundo fica claro, mas o objeto de interesse se torna escuro.

2. EXPERIMENTO

O objetivo deste experimento é simular um experimento clássico de discriminação de níveis de cinzas baseado na percepção humana. Primeiro foi feito um círculo de 1024x1024 pixel com nível de intensidade 0, ou seja, totalmente

preto. Usando a biblioteca *opencv2* do *python3*, foi feito uma matriz de 1023x1023x3, pois o *python* tem indexação que começa em 0, onde esse matriz é uma imagem BGR.

Em seguida, foi feito uma lógica dentro de um *while loop*, onde ao inserir “1” o *script* incrementa uma unidade de intensidade na matriz ao apertar “3” o *script* faz um *offset* vertical e horizontal na matriz de 7x7, iniciando assim a contagem no ponto de *offset*. Para salvar as imagens e sair do *loop* basta digitar “sair” assim a lógica sai do loop e salva a imagem da matriz.

Entre cada interação com o input do usuário é mostrado a imagem correspondente ao estado atual da matriz e seus níveis de intensidade.

Para plotar os gráficos foi utilizado uma biblioteca chamada *matplotlib* que em cada interação plotada e apagava a imagem no final da próxima interação.

Para plotar os gráficos de níveis de pixel e de incremento de pixel foi feito duas listas para armazenar os valores da interações, a cada interação de incremento era adicionado uma unidade para uma das listas quando era chamada a função de *offset* esses valores eram salvos nas respectivas listas e o valor do incremento era zero para iniciar uma nova contagem na interação subsequente. Essas duas listas foram transformadas em um *dataframe* para serem plotadas usando a biblioteca *pandas*.

Por fim, para fazer a aproximação polinomial e a normalização dos valores, foi usados as funções *polyfit* e *polyval* da biblioteca *numpy* e plotadas com *matplotlib*.

O experimento foi realizado em quatro condições diferentes: na primeira: foi utilizado um monitor Philips de 1280 x 1024 com um brilho fixo, na segunda, terceira e quarta respectivamente: um monitor Asus 1920 x 1080 com brilho médio, alto e baixo.

3. RESULTADOS

Dentro do ambiente do *spyder* foi feito o *script* com a lógica e os recursos citado previamente.

A. Primeiro foi feita na tela Philips de 1280 x 1024 com um brilho fixo onde foi obtido os seguintes resultados:

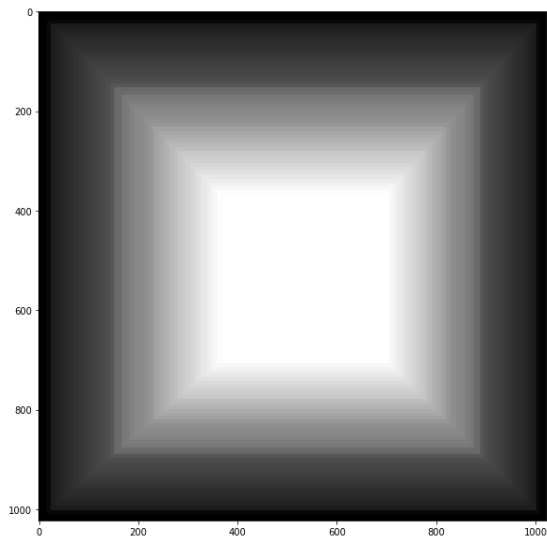


Figura I - Quadro de níveis de intensidade para tela d Philips de 1280 x 1024

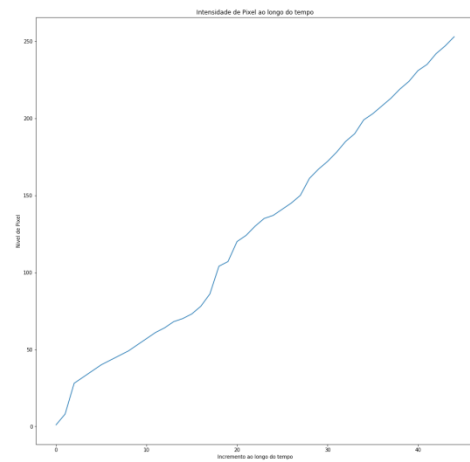


Figura II - Gráfico de intensidade de pixel por níveis de percepção

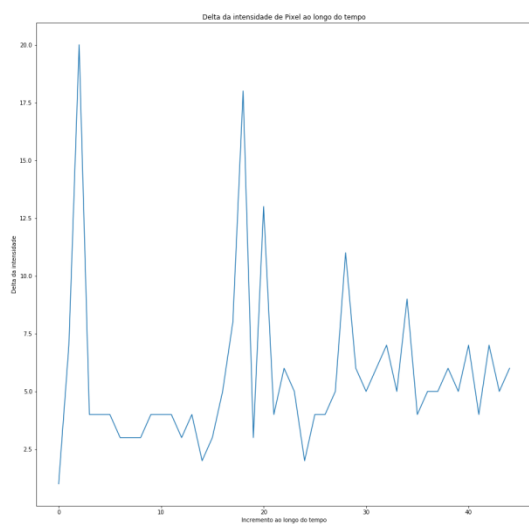


Figura III - Gráfico de incremento de pixel por nível de percepção

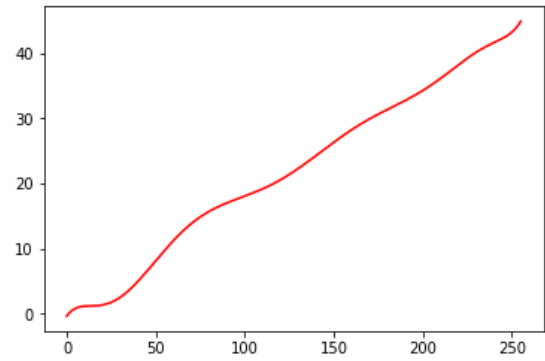


Figura IV - Aplicando polyfit e polyval nos valores de intensidade por nível de percepção usando 10 graus.

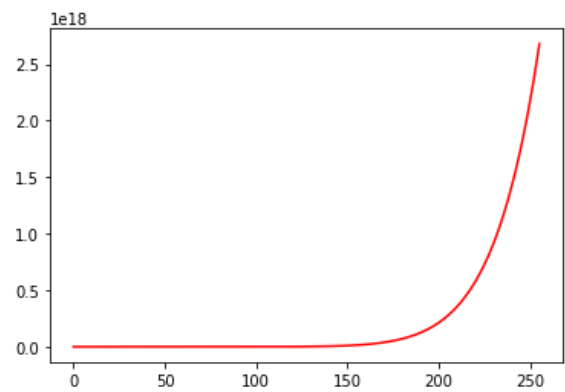


Figura V - Aplicando polyfit e polyval nos valores de incremento de intensidade por nível de percepção usando 10 graus.

B. Segundo foi feita na tela Asus de 1920 x 1080 com um brilho médio onde foi obtido os seguintes resultados:

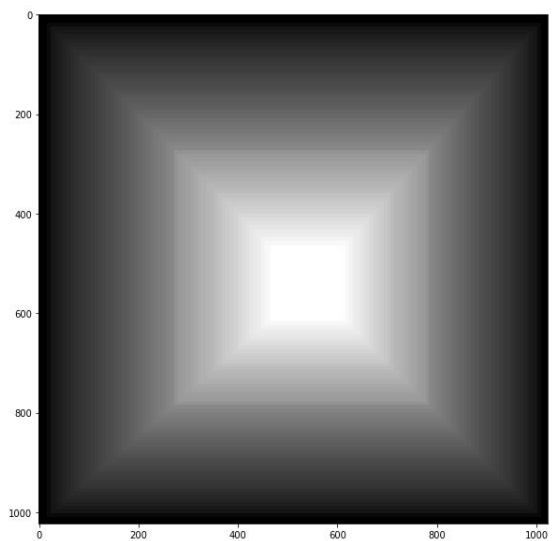


Figura VI - Quadro de níveis de intensidade para tela d Asus de 1920 x 1080 com brilho médio

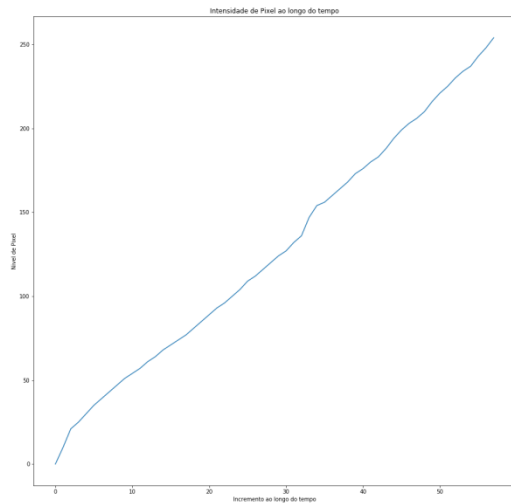


Figura VII - Gráfico de intensidade de pixel por níveis de percepção

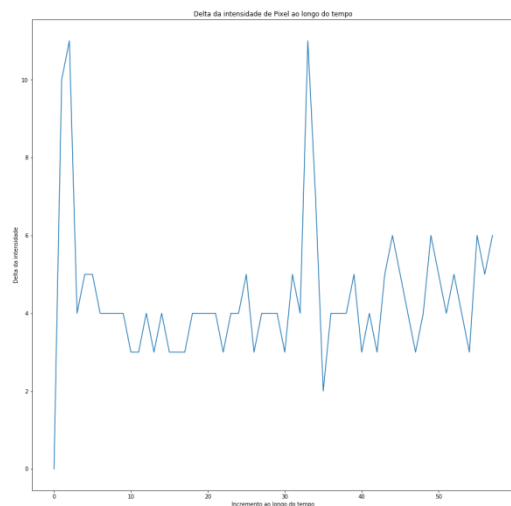


Figura VIII - Gráfico de incremento de pixel por nível de percepção

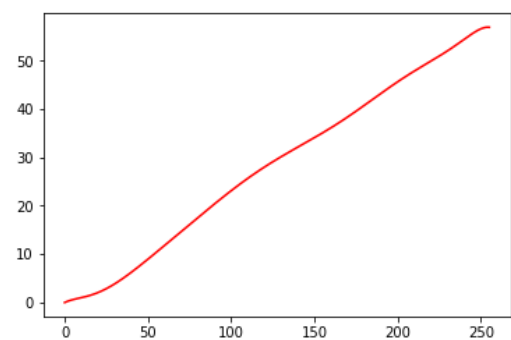


Figura IX - Aplicando polyfit e polyval nos valores de intensidade por nível de percepção usando 10 graus.

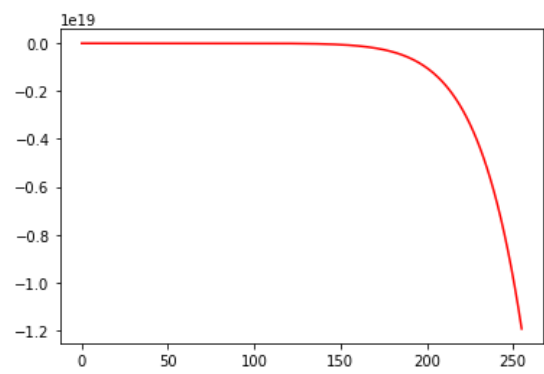


Figura X - Aplicando polyfit e polyval nos valores de incremento de intensidade por nível de percepção usando 10 graus.

C. Terceiro foi feita na tela Asus de 1920 x 1080 com um brilho no máximo, onde foi obtido os seguintes resultados:

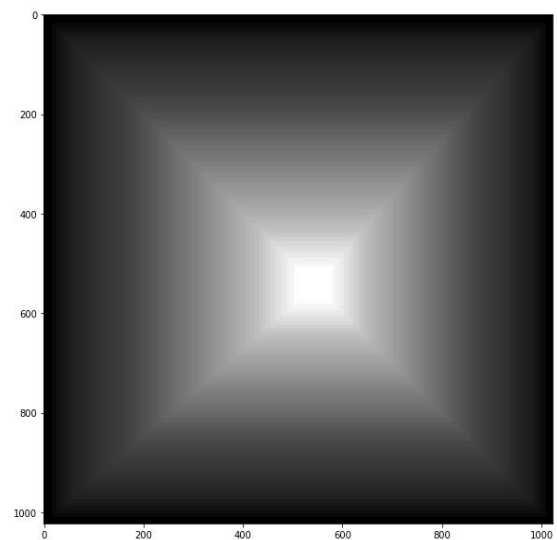


Figura XI - Quadro de níveis de intensidade para tela d Asus de 1920 x 1080 com brilho no máximo

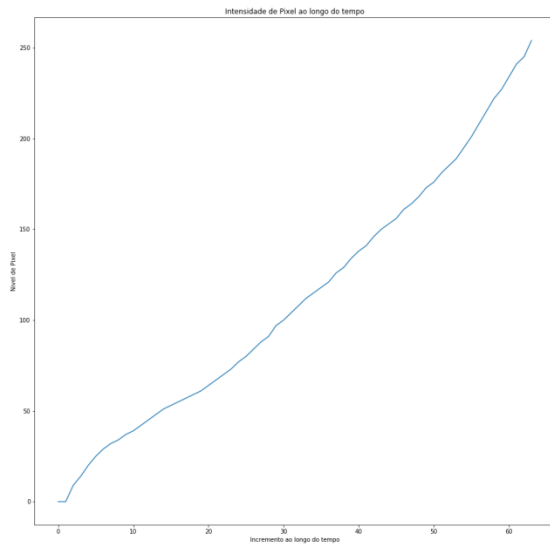


Figura XII - Gráfico de intensidade de pixel por níveis de percepção

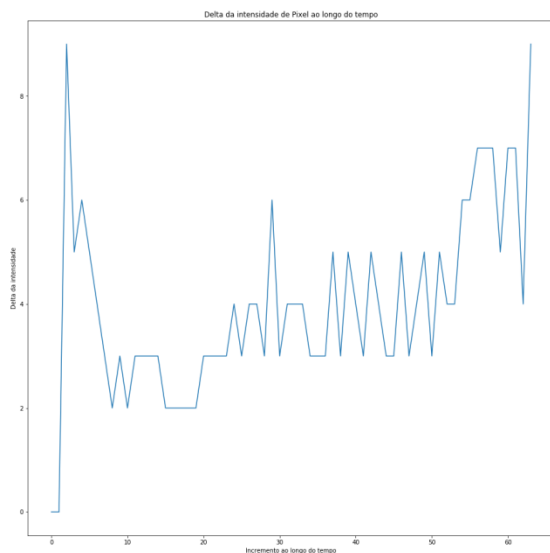


Figura XIII - Gráfico de incremento de pixel por nível de percepção

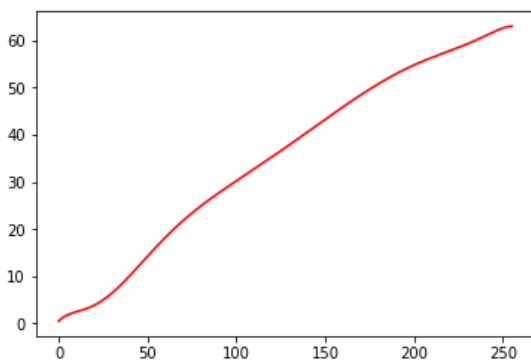


Figura XIV - Aplicando polyfit e polyval nos valores de intensidade por nível de percepção usando 10 graus.

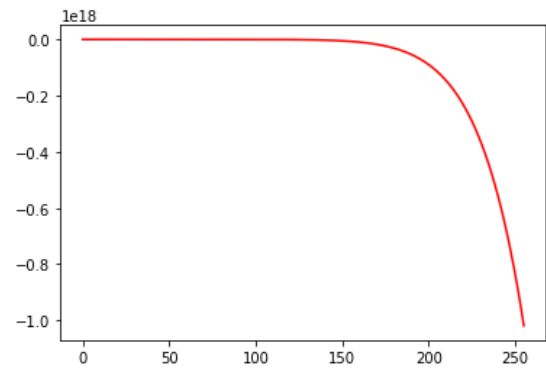


Figura XV - Aplicando polyfit e polyval nos valores de incremento de intensidade por nível de percepção usando 10 graus.

D. Quarto foi feita na tela Asus de 1920 x 1080 com um brilho próximo do mínimo, onde foi obtido os seguintes resultados:

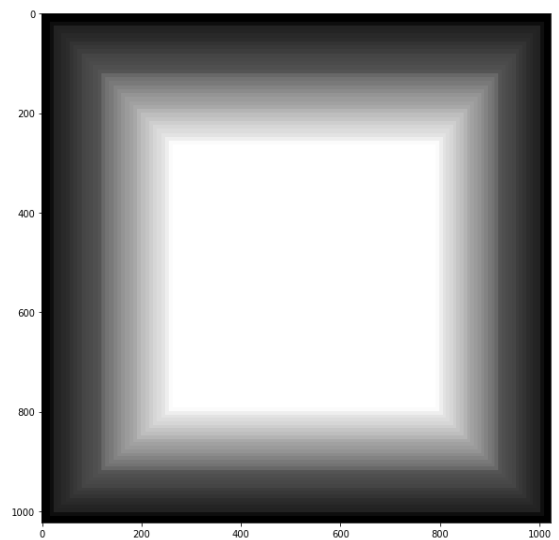


Figura XVI - Quadro de níveis de intensidade para tela d Asus de 1920 x 1080 com brilho no máximo

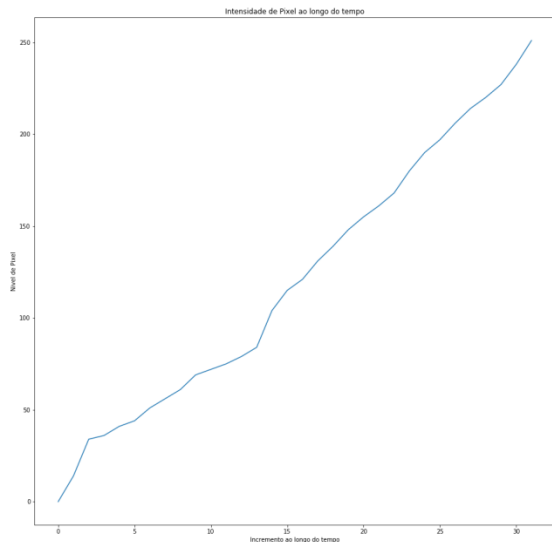


Figura XVII - Gráfico de intensidade de pixel por níveis de percepção

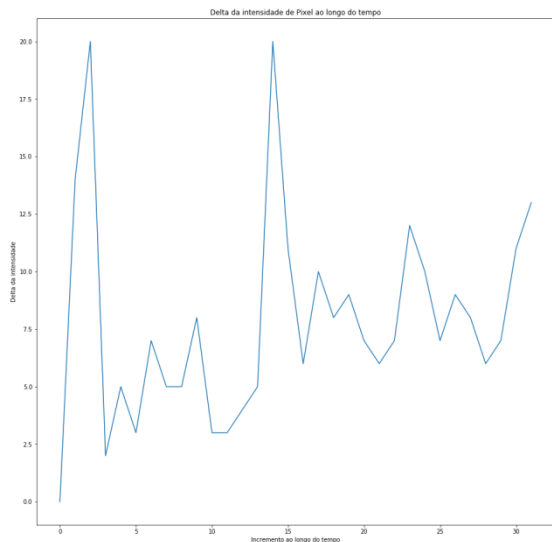


Figura XVIII - Gráfico de incremento de pixel por nível de percepção

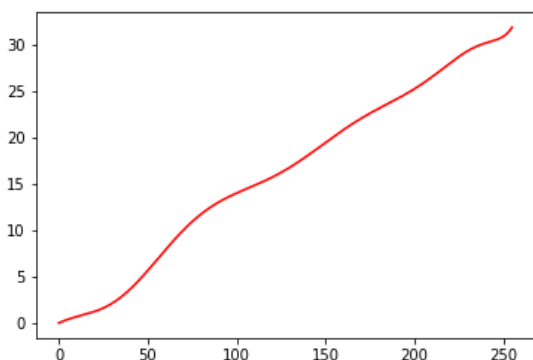


Figura XIX - Aplicando polyfit e polyval nos valores de intensidade por nível de percepção usando 10 graus.

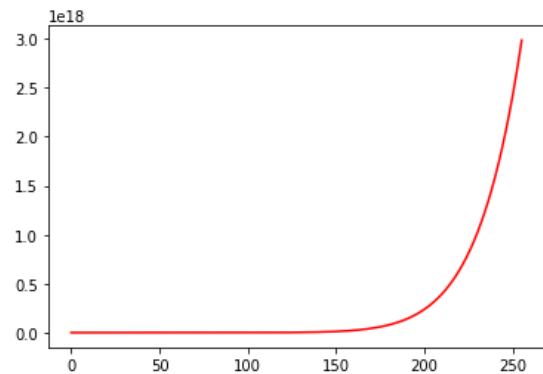


Figura XX - Aplicando polyfit e polyval nos valores de incremento de intensidade por nível de percepção usando 10 graus.

4. DISCUSSÃO E CONCLUSÕES

Baseado nos fenômenos estudados em sala é possível perceber um comportamento interessante nos resultados, quanto maior o brilho da tela, mais níveis são possíveis distinguir.

Observando que é descrito no fenômeno das *bandas de Mach* é possível verificar que nos níveis iniciais de preto é necessário mais incremento para perceber uma variação isso pode se dar devido a o fato da visão humana subestimar o incremento de níveis mais escuros e mais brancos, Esse comportamento é avaliado comparando os gráficos de incremento XXI, XXII, XXIII e XXIV. Onde o gráfico mais homogêneo e com mais níveis é o da tela com o brilho máximo e o menos homogêneo é o com brilho próximo do mínimo.

É possível notar que quanto maior o brilho da tela mais níveis foram possíveis de identificar isso pode estar relacionado com o fenômeno de quanto mais escuro o fundo mais variações claras pode-se notar, deste modo como os pixels estão mais fortes é possível identificar mais níveis.

Os gráficos de intensidade de pixel por níveis, XXV, XXVI, XXVII e XXVIII de percepção mostra que um comportamento linear entre o a intensidade do pixel com o número de níveis, porém o maior número de níveis foi da tela com brilho maior com aproximadamente 75 níveis e o menor número de níveis foi com a tela com pouco brilho com aproximadamente 40. Essa análise pode ser feita com a imagem dos quadros, XXIX, XXX, XXXI e XXXII, quanto menor o quadro branco mais níveis e quanto maior o quadro branco menos níveis foram identificados.

Para finalizar, o experimento feito para a tela com nível médio de brilho mostra que é mais difícil identificar as variações de nível pixel escuros e com o passar dos níveis vai ficando mais fácil de perceber as variações.

5. REFERÊNCIAS

- [1] Gonzalez, Rafael C. Processamento digital de imagens / Rafael C. Gonzalez e Richard C. Woods ; revisão técnica : Marcelo Vieira e Mauricio Escarpinati ; [tradução Cristina Yamagami e Leonardo Piamonte]. -- 3. ed. -- São Paulo : Pearson Prentice Hall, 2010..

6. CODIGO FONTE

```
"""
Spyder Editor

Auto: Joao Vitor Rodrigues Baptista.

"""

import cv2

print('Versão da OpenCV: ', cv2.__version__, end='\n\n')

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

qtdeLinhas = 1023

qtdeColunas = 1023

W = np.zeros((qtdeLinhas, qtdeColunas, 3))

intensidade_pixel = 0

deltaintensidade_pixel = 0

OffsetLinha = 1

OffsetColuna = 1

niveis_cinza_lista = []

deltaintensidade_list = []

sair = 1

while (sair):

    plt.figure(figsize=(10, 10))

    plt.imshow(W)

    plt.pause(0.5)

    input_controle = input(

        '\nDigite 1 para incrementar e 0 para decrementar: \

        \nDigite 3 para ir para o proximo quadrado, \

        Digite 4 para voltar para o quadrado anterior\

        \n Digite sair para sair salvar e Digite 5 para salvar o \

        documento atual ')

    plt.close()

    if input_controle == "1":

        intensidade_pixel = intensidade_pixel + 1

        deltaintensidade_pixel = deltaintensidade_pixel + 1

        for i in range(8 * OffsetLinha, qtdeLinhas - 7 * OffsetLinha):

            for j in range(8 * OffsetColuna, qtdeColunas - 7 *

OffsetColuna):

                intensidade_pixel = intensidade_pixel - 1

                deltaintensidade_pixel = deltaintensidade_pixel + 1

                for i in range(8, qtdeLinhas - 7):

                    for j in range(8, qtdeColunas - 7):

                        W[i][j] = intensidade_pixel / 255

            elif input_controle == "0":

                intensidade_pixel = intensidade_pixel - 1

                deltaintensidade_pixel = deltaintensidade_pixel + 1

                for i in range(8, qtdeLinhas - 7):

                    for j in range(8, qtdeColunas - 7):

                        W[i][j] = intensidade_pixel / 255

            # plt.figure(figsize=(10, 10))

            # plt.imshow(W)

            # plt.pause(0.5)

        elif input_controle == "3":

            niveis_cinza_lista.append(intensidade_pixel)

            deltaintensidade_list.append(deltaintensidade_pixel)

            OffsetLinha = OffsetLinha + 1

            OffsetColuna = OffsetColuna + 1

            deltaintensidade_pixel = 0

            # plt.figure(figsize=(10, 10))

            # plt.imshow(W)

            # plt.pause(0.5)

        elif input_controle == "4":

            OffsetLinha = OffsetLinha - 1

            OffsetColuna = OffsetColuna - 1

            # plt.figure(figsize=(10, 10))

            # plt.imshow(W)

            # plt.pause(0.5)

        elif input_controle == "5":

            plt.close()

            plt.figure(figsize=(10, 10))

            plt.imshow(W)

            plt.show()

            plt.pause(0.5)

            plt.savefig("quadrado_niveis_cinza.png")

        elif input_controle == "sair":

            plt.close()

            plt.figure(figsize=(10, 10))

            plt.imshow(W)

            plt.show()

            plt.pause(0.5)

            plt.savefig("quadrado_niveis_cinza.png")

            plt.show()

            sair = 0

print('Lista de niveis de cinza', niveis_cinza_lista)

print('Lista de delta niveis de cinza', deltaintensidade_list)

dt = pd.DataFrame(niveis_cinza_lista)

fig = dt.plot(figsize=(16, 16), legend=False)

fig.set_ylabel('Nivel de Pixel')

fig.set_xlabel('Incremento ao longo do tempo')

fig.set_title('Intensidade de Pixel ao longo do tempo')
```

```

fig.get_figure()

fig.savefig('grafico_niveis_cinza', dpi=300)

indices = []

df = pd.DataFrame(deltaintensidade_list)

fig1 = df.plot(figsize=(16, 16), legend=False)

fig1.set_ylabel('Delta da intensidade')

fig1.set_xlabel('Incremento ao longo do tempo')

fig1.set_title('Delta da intensidade de Pixel ao longo do tempo')

fig1.get_figure()

fig1.savefig('grafico_delta_niveis_cinza', dpi=300)

x = 0

indices_1 = []

for i in enumerate(niveis_cinza_lista):

    indices_1.append(x)

    x = x + 1

poly_deg = 10 #degree of the polynomial fit

polynomial_fit_coeff_1 = np.polyfit(niveis_cinza_lista, indices_1,

poly_deg)

start = 0

stop = 255

num_points = 255

arbitrary_time = np.linspace(start, stop, num_points)

lon_intrp_2 = np.polyval(polynomial_fit_coeff_1,

```

```

arbitrary_time)

plt.plot(arbitrary_time, lon_intrp_2, 'r') #interpolated window as a red

curve

x = 0

indices_2 = []

for i in enumerate(deltaintensidade_list):

    indices_2.append(x)

    x = x + 1

polynomial_fit_coeff_2 = np.polyfit(deltaintensidade_list, indices_2,

poly_deg)

lon_intrp_2 = np.polyval(polynomial_fit_coeff_2,

arbitrary_time)

plt.plot(arbitrary_time, lon_intrp_2, 'r') #interpolated window as a red

curve

```