LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA
MESTRADO EM MODELAGEM COMPUTACIONAL

Lista 2 - GA023
Elementos de Processamento de Imagem

Lorran de Araújo Durães Soares

Petrópolis - RJ
2024

Lorran de Araújo Durães Soares

Lista 2 - GA023
Elementos de Processamento de Imagem

Trabalho apresentado como parte dos critérios de avaliação da disciplina GA023 - Elementos de Processamento de Imagem.

Professor(a): Gilson Antonio Giraldi

Petrópolis - RJ

2024

Sumário

1 Introdução **2**

2 Questão 1 **2**

3 Questão 2 **5**

4 Questão 3 **10**

5 Questão 4 **13**

6 Questão 5 **17**

1 Introdução

Este texto refere-se à lista realizada na disciplina **GA023 Elementos de Processamento de Imagem**, do curso de pós-graduação oferecido pelo **Laboratório Nacional de Computação Científica (LNCC)**, sob a orientação do professor **Gilson Antonio Giraldi**.

Neste documento, serão apresentadas as questões propostas pelo trabalho, seguidas de suas respectivas resoluções. Todas as questões foram implementadas na linguagem Python, utilizando notebooks do tipo `iPynb` e empregando bibliotecas como `Matplotlib` [7], `Numpy` [2], `Scikit-learn` [4], `OpenCV` [6], `Seaborn` [11], `Scipy` [10], `Scikit-Image` [9] e `PIL` [1].

As implementações das questões estão disponíveis clicando aqui.

2 Questão 1

Considere a distribuição normal bidimensional padrão (veja a página 32, referência [3]).

(a) *Usando versões discretas dessa função, construa um filtro passa-baixa.*

(b) *Agora, tome a derivada com respeito às variáveis x e y da distribuição normal bidimensional e repita o item anterior. Usando a transformada de Fourier de sequências e o respectivo teorema da convolução (teorema 3 do arquivo aula2.pdf), tente caracterizar o tipo de filtros obtidos.*

(c) *Aplique os filtros sobre uma imagem e analise os resultados.*

Resolução:

Para a realização desta questão, foi construído, através da biblioteca `Numpy`, um grid quadricular com dimensões 21×21, utilizando o método *meshgrid*. Esse tamanho foi definido empiricamente. Em seguida, foi definida a distribuição normal bidimensional com esse grid como domínio, a qual, de acordo com `aula1.pdf`, é dada pela seguinte equação:

$$\mathtt{G}(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right) \tag{1}$$

Onde σ também foi definido empiricamente. De acordo com `aula1.pdf`, temos que esse filtro é caracterizado por ser um passa-baixa, como veremos mais adiante. A Figura 1 apresenta a visualização do filtro no domínio do espaço.

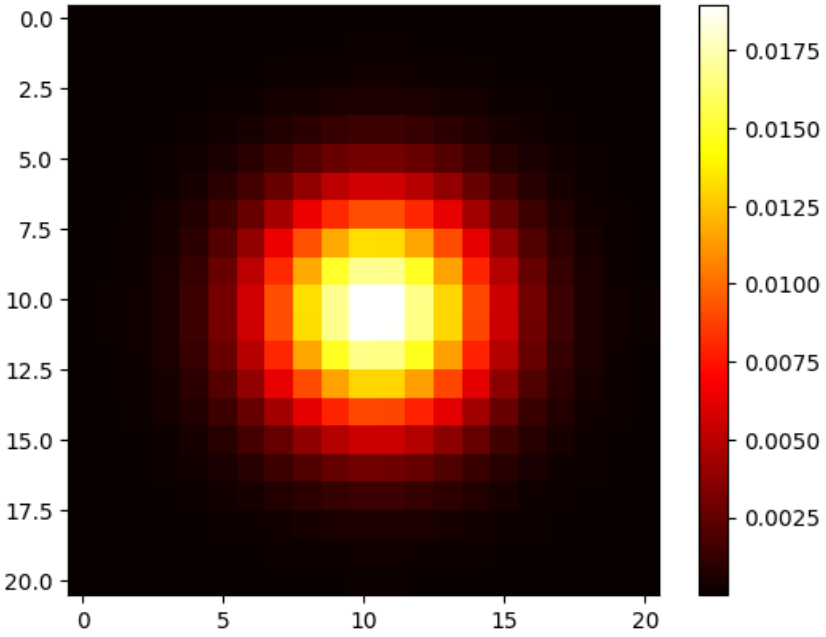

Figura 1: Visualização do filtro Gaussiano no domínio do espaço

Derivando a equação 1 em relação a x e a y, obtemos então as seguintes equações, também definidas sobre o mesmo grid criado anteriormente:

$$\frac{\partial G}{\partial x}(x,y) = -\frac{x}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2}$$

$$\frac{\partial G}{\partial y}(x,y) = -\frac{y}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3}$$

Então, os filtros foram normalizados com a normalização L2, que é uma técnica essencial para garantir que o filtro preserve a energia (ou magnitude geral) da imagem ao longo da transformação. Especificamente, a normalização pela norma L2 é definida como:

$$\text{filter} \leftarrow \frac{\text{filter}}{\sqrt{\sum(\text{filter})^2}}$$

Usando o método *fft* da biblioteca `Numpy`, os filtros discretos, determinados pelas equações 1, 2 e 3, foram transformados para o domínio da frequência através da Transformada de Fourier. Com isso, foi possível plotar, utilizando a biblioteca `Matplotlib`, uma visualização de cada um dos filtros no domínio da frequência, conforme apresentado na Figura 2.

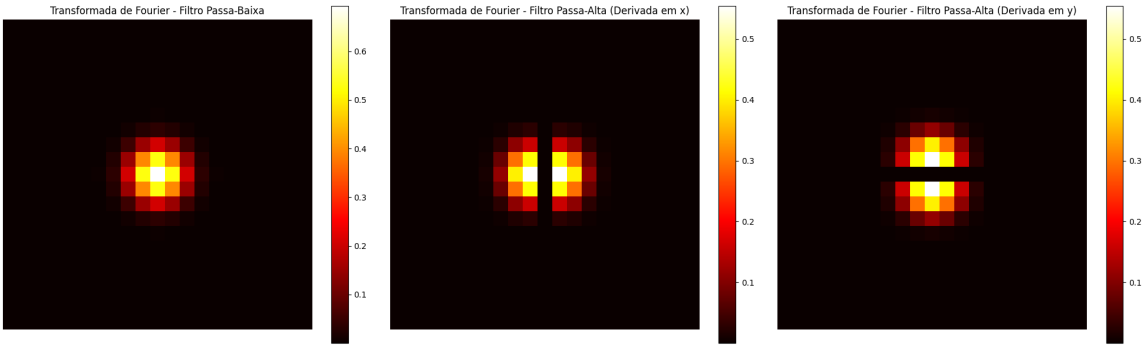

Figura 2: Visualização de cada filtro no domínio da frequência

Sabendo que, segundo o teorema da convolução, a multiplicação direta do filtro com a transformada da imagem é equivalente à realização da convolução no domínio do espaço, podemos concluir que a primeira imagem se trata de um filtro passa-baixa, pois as maiores intensidades estão no centro da imagem, correspondendo aos menores valores de frequência. Por outro lado, as outras duas imagens mostram que os filtros são do tipo *high-pass*, já que as frequências próximas de zero nos eixos x e y são anuladas pelos filtros gaussianos parcialmente derivados em relação a x e y, respectivamente.

Utilizando a biblioteca `Skimage` com as classes *io* e *color*, foi carregada uma imagem e convertida para tons de usando a biblioteca `OpenCV`, com o método *cvtColor*. Em seguida, com o método *convolve* da biblioteca `Scipy`, aplicamos os filtros a esta imagem. Por padrão, a biblioteca detecta automaticamente a complexidade do cálculo para decidir entre aplicar a convolução diretamente no domínio do espaço ou utilizar o teorema da convolução para operar no domínio da frequência e, posteriormente, retornar ao domínio do espaço. Após esses passos, foi obtido o resultado apresentado na Figura 3, que mostra a imagem original seguida das aplicações de cada filtro.

Figura 3: Visualização da aplicação dos filtros em uma imagem

Como podemos observar, os resultados obtidos são consistentes com as características de cada classe de filtro. O filtro passa-baixa realizou uma suavização na imagem, enquanto os filtros do tipo high-pass destacaram as frequências mais atenuantes, geralmente presentes nas bordas dos elementos da imagem. É interessante notar que, no filtro derivado em relação a x, houve uma detecção das bordas predominantemente no sentido do eixo x, enquanto no filtro derivado em relação a y, as bordas foram detectadas no sentido do eixo y.

3 Questão 2

Estude a teoria do PCA para problemas de poucas amostras, onde o número de dados é menor do que a dimensão do espaço de dados. Escolha um banco de dados de imagens, converta as imagens para escala de cinza e aplique a teoria de "PCA para problemas de poucas amostras"para redução de dimensionalidade.

(a) *Se $\bar{x}$ é a média amostral (centroide do conjunto de dados) e p_1 é o componente principal, visualize o resultado da expressão:*

$$x = \bar{x} + \alpha p_1,$$

onde $\alpha \in \{-\beta\lambda_1, 0, \beta\lambda_1\}$ com λ_1 sendo o autovalor associado a p_1 e β um fator escalar.

(b) *Estude o espectro da matriz $X^T X$ para realizar a redução de dimensionalidade. Visualize algumas imagens no espaço de dimensão reduzida.*

(c) *Construa um gerador de imagens usando os d componentes principais escolhidos no item (b).*

Resolução:

Para realizar a redução de dimensionalidade utilizando Análise de Componentes Principais (PCA) em um conjunto de dados onde o número de amostras é inferior à dimensão do espaço dos dados, foi escolhida a base de imagens *originalimages* da FEI Face Database [8]. Esta base é composta por 400 imagens faciais frontais, coloridas, de homens e mulheres, sérios e sorridentes, com resolução de 260×360 pixels. Todos os gráficos e visualizações de imagens foram realizados com a biblioteca `Matplotlib`. Para o carregamento das imagens, obtidas em [8], foram utilizadas as bibliotecas `OS` e `PIL`, que também realizaram a conversão das imagens para arrays `numpy`, o formato desejado para os cálculos subsequentes. Antes da conversão para arrays, através da biblioteca `OpenCV`, com o método *cvtColor*, as imagens foram convertidas para tom de cinza. A figura 4 mostra alguns exemplos de imagens presentes neste banco de dados já convertidas para tom de cinza.

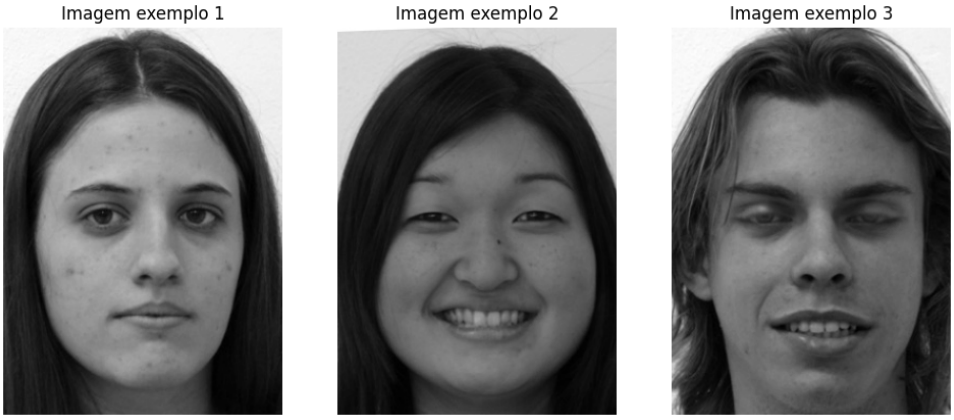

Figura 4: Imagens geradas para o item (a)

Como este é um problema com poucas amostras em relação à dimensionalidade dos dados, para o cálculo da matriz P_{PCA}, com o objetivo de minimizar a perda de informação das imagens ao reduzir sua dimensionalidade, seguimos os passos descritos em [5]. Aqui, X representa a matriz de dados já vetorizada:

1. Calculo da matriz $\tilde{X}$ de dados centralizados pela expressão

$$\tilde{X} = \begin{bmatrix} \tilde{x}_1^T \\ \tilde{x}_2^T \\ \vdots \\ \tilde{x}_N^T \end{bmatrix} \in R^{N \times n},$$

onde N é o número de amostras e n é a dimensão do espaço de dados (neste caso, $360 \cdot 260 = 93600$) e cada $\tilde{x}_i$ é centralizado da forma $\tilde{x}_i = x - \bar{x}$, onde $\bar{x}$ é a média dos dados.

2. Resolução do problema de autovalores e autovetores para a matriz $\frac{1}{N}\tilde{X}\tilde{X}^T \in R^{N \times N}$:

$$\frac{1}{N}\tilde{X}\tilde{X}^T v_i = \lambda_i v_i,$$

3. Calculo dos vetores $\omega_i = \tilde{X}^T v_i, \quad i = 1, 2, \ldots, N.$

4. Normalização dos vetores ω_i para obter os autovetores da matriz de covariância $S = \frac{1}{N}\tilde{X}^T\tilde{X}$:

$$a_i = \frac{\omega_i}{\|\omega_i\|} = \frac{\tilde{X}^T v_i}{\|\tilde{X}^T v_i\|}.$$

Os autovetores obtidos ao final desse processo, ordenados em ordem decrescente em relação aos respectivos autovalores, são as colunas da matriz P_{PCA} que queríamos.

O cálculo dos autovetores e autovalores da matriz foi realizada através da função `np.linalg` e a multiplicação de matrizes foi calculada por meio da função `np.dot`, ambas presentes na biblioteca científica `Numpy`. Foi criada uma classe para a criação de todos os métodos necessários para a realização desta questão.

Para o item (a), o resultado da expressão $\mathbf{x} = \bar{\mathbf{x}} + \alpha \mathbf{p}_1$, com $\alpha \in \{-\beta\sqrt{\lambda_1}, 0, \beta\sqrt{\lambda_1}\}$, com $\beta = 0.8$, resultou nas imagens presentes na figura , onde $\lambda_1 = 51139693.992$ é o autovalor correspondente à componente principal $\mathbf{p_1}$. A legenda coeficiente 1 se trata do valor $-\beta\sqrt{\lambda_1}$, o coeficiente 2 é 0 e o coeficiente 3 é $\beta\sqrt{\lambda_1}$.

Figura 5: Imagens geradas para o item (a)

As imagens revelaram que a principal componente capturou predominantemente o fundo das fotos, um elemento comum a todas elas. Isso resultou na aproximação das faces individuais à "face média"derivada do conjunto de imagens.

Para o item (b), foram calculados os autovalores da matriz $S = \frac{1}{N}\tilde{X}^T\tilde{X}$, que posteriormente foram ordenados em ordem decrescente. Com base nesses valores, foi plotado o gráfico da variância explicada, tanto individual quanto acumulada, com o objetivo de analisar a contribuição de cada autovalor na representação do conjunto de dados e identificar as principais componentes.

A variância acumulada foi obtida pela soma sequencial dos autovalores em ordem decrescente. Quando essa soma atingiu 95% do total dos autovalores, determinou-se que 95% da energia total estavam representados por esse subconjunto. Com base no gráfico da Figura 6, foi decidido reduzir a dimensionalidade para as 110 primeiras componentes, uma vez que essas já representavam mais de 95% da energia total dos autovalores.

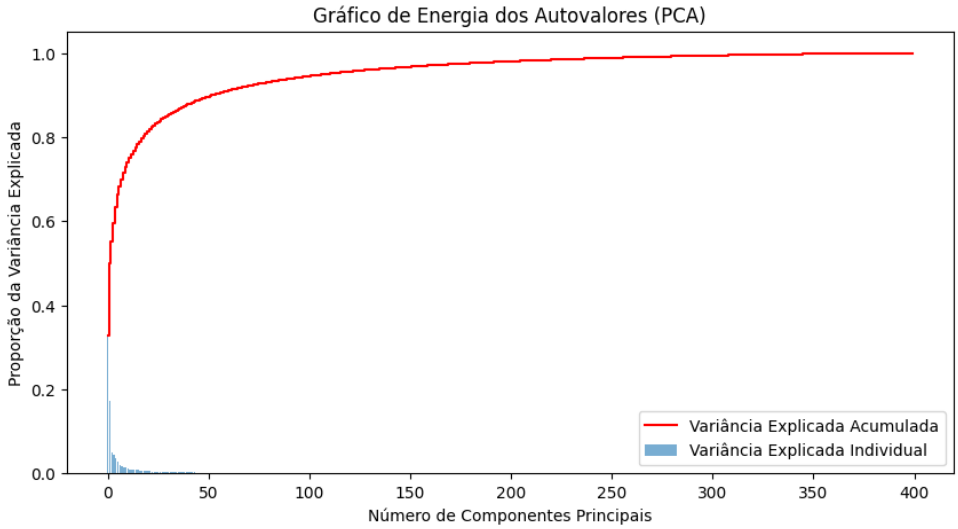

Figura 6: Gráfico de energia dos autovalores

Foi então aplicado o truncamento multiplicando a matriz P_{PCA} por I_{110}, onde I tem dimensão 400×400, que é o tamanho das colunas da matriz P_{PCA}, mas tem até os primeiros 110 números da diagonal preenchido por 1, e todo o resto por 0. Para visualizar então como ficaram as imagens após essa redução, foi realizado o cálculo $X \cdot P_{PCA} \cdot P_{PCA}^T$, obtendo então as imagens presentes na figura 7 com a comparação com suas originais.

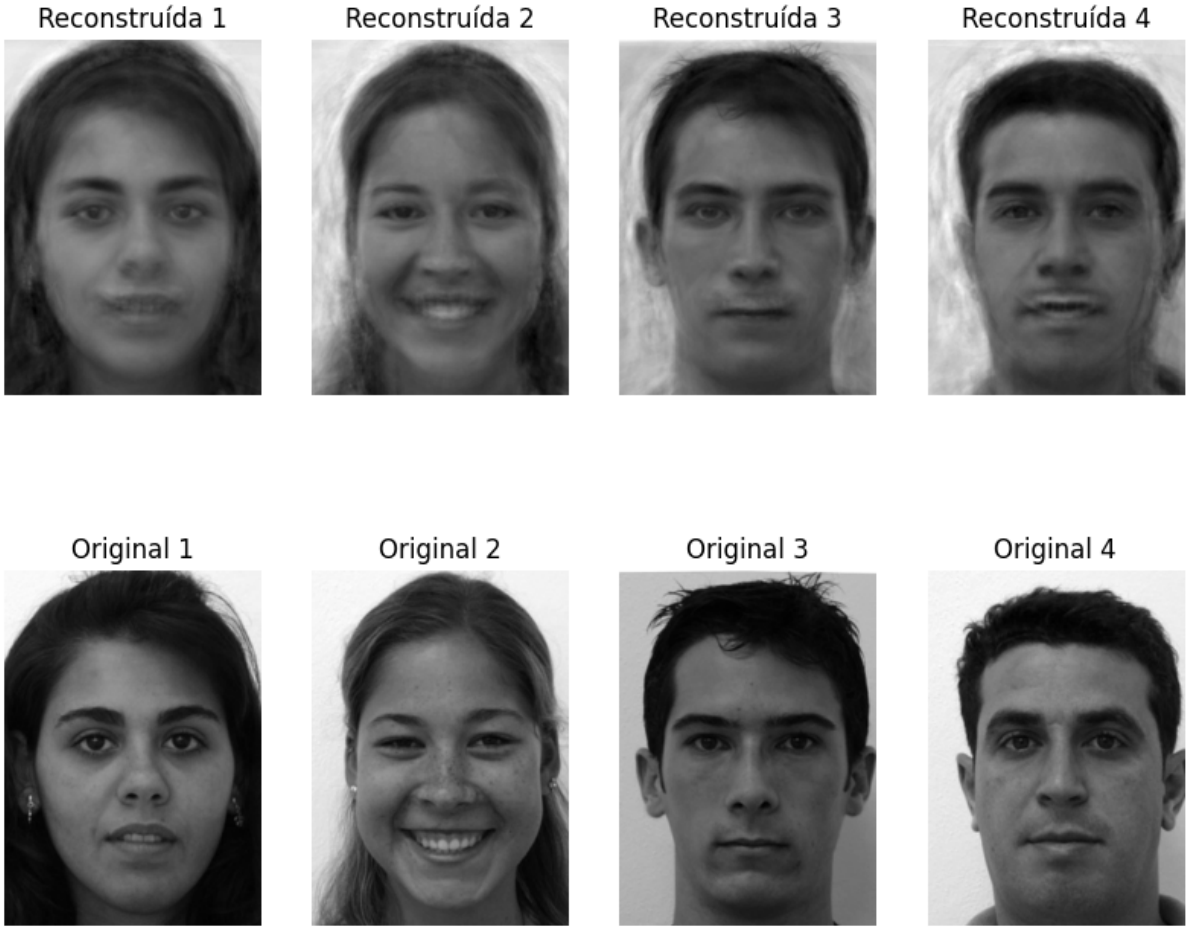

Figura 7: Comparação reconstruída e original

Nas imagens, é notada a redução de qualidade e nitidez em relação à imagem original, mas de forma que as faces ainda mantiveram seus traços e podem ser reconhecidas.

Para o item (c), na construção do gerador de imagens, foi realizada a seguinte operação:

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{i=1}^{110} \beta_i \sqrt{\lambda_i} \mathbf{p}_i \tag{4}$$

Onde:

- λ_i é o i-ésimo autovalor;
- $\bar{\mathbf{x}}$ é a média amostral das imagens;
- $\mathbf{p}_i$ é o autovetor associado ao i-ésimo autovalor;
- β_i é o i-ésimo coeficiente, que foi sorteado entre 0 e 1, com o uso de uma distribuição de probabilidade uniforme disponibilizada pela `Numpy` denominada `np.random.uniform`.

Obtivemos, como resultado, as quatro imagens de novas faces ilustradas na Figura 8. As diferenças entre as imagens geradas devem-se ao fato de que os coeficientes β eram sorteados

aleatoriamente a cada iteração do cálculo da equação, resultando em variações nas características faciais.

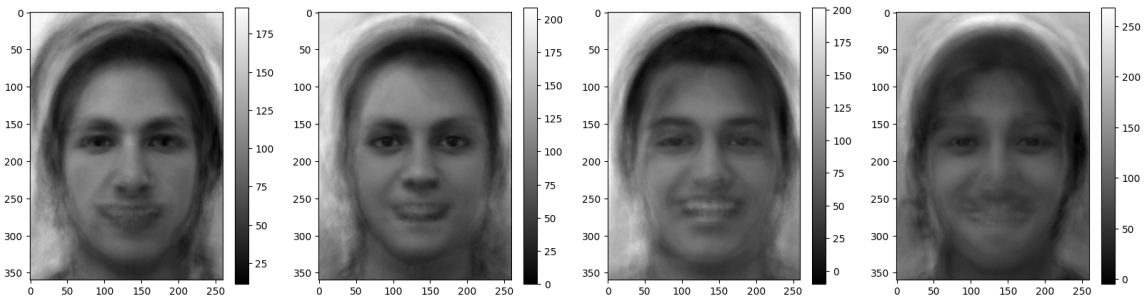

Figura 8: Imagem gerada pelas 110 primeiras componentes

Ao realizar várias gerações, foram observadas faces com características marcantes e variadas, como diferentes formatos de lábios, sobrancelhas e narizes. Em um dos testes, embora a imagem não esteja incluída neste trabalho, foi possível gerar uma pessoa usando óculos, enquanto outras não apresentavam esse acessório. Além disso, em comparação com o item (a), percebe-se que, quanto mais componentes principais são utilizadas, mais informações e detalhes são incorporados às imagens geradas.

4 Questão 3

Escolha uma base de imagens D, converta as imagens para tons de cinza. Separe D em dois subconjuntos disjuntos D_{tr} e D_{te}.

(a) *Calcule o* "PCA para problemas de tamanho de amostra pequeno" *em D_{tr}.*

(b) *Estude o espectro da matriz $R = \frac{1}{N} X^T X$ para realizar a redução de dimensionalidade.*

(c) *Aplique o PCA em D_{te} e compare a eficiência de compressão com o DCT.*

Resolução:

Para a realização desta questão, foi utilizada a mesma base de imagens *originalimages* da FEI Face Database [8]. As imagens foram novamente convertidas para tons de cinza e vetorizadas para aplicação do PCA.

Com o objetivo de dividir os dados em conjuntos de treino e teste, utilizou-se o método `train_test_split` da biblioteca `Sklearn`, com o parâmetro `test_size` configurado para 20%, garantindo que 80% das amostras fossem destinadas ao treinamento.

Utilizando a classe e as funções criadas no exercício anterior, o PCA foi treinado com o conjunto de treinamento, e o espectro da matriz $R = \frac{1}{N} X^T X$ foi analisado para realizar a

redução de dimensionalidade. Foram calculados os autovalores da matriz R, que foram ordenados em ordem decrescente. Com base nesses valores, foi plotado o gráfico da variância explicada acumulada, visando analisar a contribuição de cada autovalor na representação do conjunto de dados e identificar as principais componentes.

Os resultados mostraram que as 99 primeiras componentes já representavam 95% da variância dos dados, conforme ilustrado na Figura 9. Assim, foi realizado o truncamento, multiplicando a matriz P_{PCA} por I_{99}, utilizando as funções desenvolvidas na questão anterior.

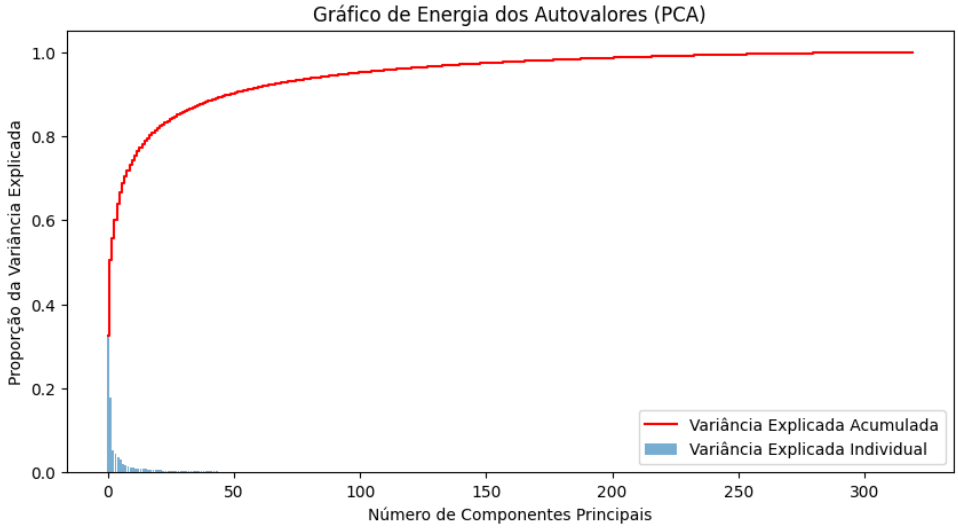

Figura 9: Gráfico de energia dos autovalores

Logo, foi aplicado o PCA no conjunto de testes e realizada sua reconstrução, utilizando a matriz P_{PCA} truncada e sua transposta.

Com a biblioteca `Scipy`, utilizando a classe *fftpack*, foi calculada a DCT (Transformada Discreta de Cosseno) deste conjunto de testes, usando a equação de acordo com a aula4.pdf:

$$c(k,n) = \begin{cases} \frac{1}{\sqrt{N}}, & \text{se } k = 0, \quad 0 \le n \le N-1, \\ \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(2n+1)k}{2N}\right), & \text{se } 1 \le k \le N-1, \quad 0 \le n \le N-1. \end{cases} \tag{5}$$

Essa equação é especificada como parâmetro type = 2 na função *dct* usada para a aplicação e *idct* para a reconstrução, na classe *fftpack*.

Para o truncamento, foram calculados os valores absolutos das intensidades dos pixels de cada imagem, ordenados em ordem decrescente, e realizada a soma cumulativa desses valores. Quando a soma acumulada atingiu uma determinada porcentagem da soma total, todas as coordenadas da DCT com valores abaixo desse limite foram zeradas, já que as componentes mais importantes na DCT correspondem às maiores intensidades.

Considerando que o PCA reduziu as imagens, que originalmente tinham dimensão $260 \times 360 = 93.600$, para 400 componentes e, após o truncamento, para 99 componentes, foram realizadas duas abordagens para a seleção dessa porcentagem na DCT:

1. Taxa igual a $\frac{99}{400}$, correspondente à proporção de componentes retidas após o truncamento do PCA sobre as 400 componentes principais iniciais;
2. Taxa igual a $\frac{99}{93.600}$, que representa a redução total de componentes em relação à dimensão original das imagens.

O resultado da reconstrução das imagens pelo PCA e pela DCT está presente nas figuras 10 e 11, onde a primeira figura se trata da redução da DCT com a porcentagem $\frac{99}{400}$ e a segunda com a porcentagem $\frac{99}{93.600}$.

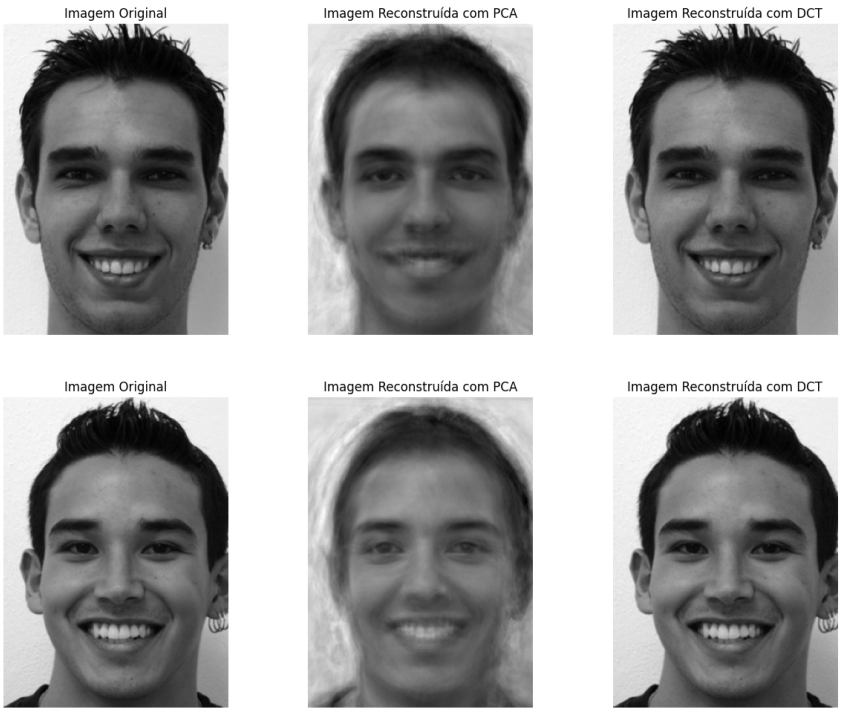

Figura 10: Imagens Reconstruídas - Primeira proporção

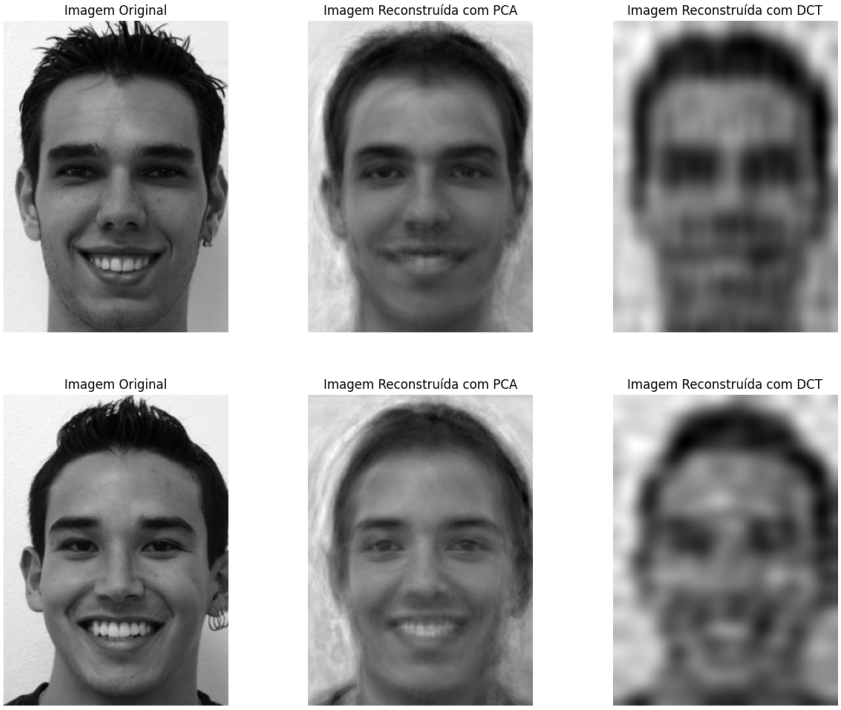

Figura 11: Imagens Reconstruídas - Segunda proporção

As reconstruções evidenciam que, para efeito de comparação de eficiência entre a reconstrução pelo PCA e pela DCT, considerando a primeira proporção, a DCT apresenta uma grande vantagem, gerando uma imagem muito mais próxima da original em comparação com o PCA. Já para a segunda proporção, embora seja possível identificar na reconstrução da DCT que se tratam de rostos, a reconstrução do PCA apresentou um desempenho superior. Logo, podemos concluir que, com apenas 99 componentes, o PCA conseguiu preservar melhor a informação original, superando a DCT em termos de eficiência na conservação de informações com menos componentes.

5 Questão 4

Escolha oito imagens de uma base de dados de imagens, por exemplo, $I_1, I_2, \ldots, I_8$*. Converta-as para escala de cinza (se necessário) e redimensione para* $N \times N$*. Calcule a transformada discreta do seno para cada imagem, denotada por* V_i*, onde* $i = 1, 2, \ldots, 8$*, e os valores absolutos correspondentes* $ABS(V_i)$*, para* $i = 1, 2, \ldots, 8$*.*

(a) *Concatenate as imagens em uma lista* $I = [I_1, I_2, \ldots, I_8]$ *e forme a lista* $L_{abs} = [ABS(V_1), ABS(V_2), \ldots, ABS(V_8)]$*. Calcule as características estatísticas para* L_{abs}*. Exemplos: valor mínimo, valor máximo, média, desvio padrão, histograma, etc. Discuta os resultados.*

(b) *Aplique o resultado do item 4(a) para projetar uma estratégia de redução de dimensionalidade. Visualize alguns resultados da reconstrução.*

Resolução:

Para a realização deste exercício, foram escolhidas 8 imagens da mesma base de imagens *originalimages* da FEI Face Database [8]. As imagens foram novamente convertidas para tons de cinza e vetorizadas. Além disso, elas foram transformadas em quadradas, simplesmente cortando as imagens com o uso do array `Numpy`, de modo que as dimensões se tornaram 260×260.

Logo após, foi aplicada a transformação seno através do método *dst* da biblioteca `Scipy`, com o parâmetro `type=1`, o que significa que a transformação aplicada será condizente com a apresentada na aula4.pdf, dada por:

$$\psi(k,n) = \sqrt{\frac{2}{N+1}} \sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right), \quad 0 \leq k, n \leq N-1, \tag{6}$$

onde $N = 260$ neste caso, gerando o conjunto das imagens transformadas, denotadas por V.

Após isso, utilizando o método *abs* da biblioteca `Numpy`, foram calculados os valores absolutos de todas as imagens, denotados por $\text{ABS}(V)$.

Em seguida, com os métodos *mean*, *median*, *std*, *min* e *max* da biblioteca `Numpy`, foram calculadas as médias, medianas, desvios padrão, valores mínimo e máximo das intensidades dos pixels de cada imagem do conjunto $\text{ABS}(V_i)$, gerando os dados presentes na Tabela 1.

Imagem	**Média**	**Mediana**	**Desvio Padrão**	**Min**	**Max**
Imagem 1	8.7748	1.0253	87.3371	1.4401×10^{-5}	14926.2605
Imagem 2	8.9512	1.5361	97.2831	3.2046×10^{-5}	17381.0742
Imagem 3	8.2167	1.3179	83.0330	2.7641×10^{-5}	17053.3512
Imagem 4	8.5232	1.3155	97.3798	2.5473×10^{-5}	15056.8867
Imagem 5	7.5279	0.9699	79.0755	1.7497×10^{-5}	14301.8414
Imagem 6	8.5386	1.0917	90.3626	2.0046×10^{-5}	15674.7976
Imagem 7	6.5690	0.7710	83.5264	9.1508×10^{-6}	11966.9477
Imagem 8	8.6343	1.5474	95.8161	1.7381×10^{-5}	14666.8876

Tabela 1: Estatísticas das imagens.

Além disso, a fim de visualizar melhor a distribuição das intensidades no domínio da frequência, foi plotado o histograma do valor absoluto do logaritmo dessas intensidades. Para isso, foi utilizada a biblioteca `Matplotlib` com o método *hist*, configurado com o parâmetro *density* igual a true, para que fosse exibida a frequência relativa das intensidades. Os resultados estão ilustrados na figura 12, que mostra o gráfico para cada uma das imagens. Em imagens transformadas para o domínio da frequência (como usando a DST), há picos muito altos em baixas frequências e valores muito baixos em altas frequências. O uso do logarítmico reduz a diferença entre esses

extremos, tornando os detalhes nas regiões de menor amplitude mais visíveis. Logo, valores pequenos são destacados sem serem "esmagados"pelos valores mais altos. Isso é útil, por exemplo, para identificar ruídos ou padrões em altas frequências que poderiam ser negligenciados em uma escala linear. Por isso a adoção da escala logarítima para plotar os histogramas.

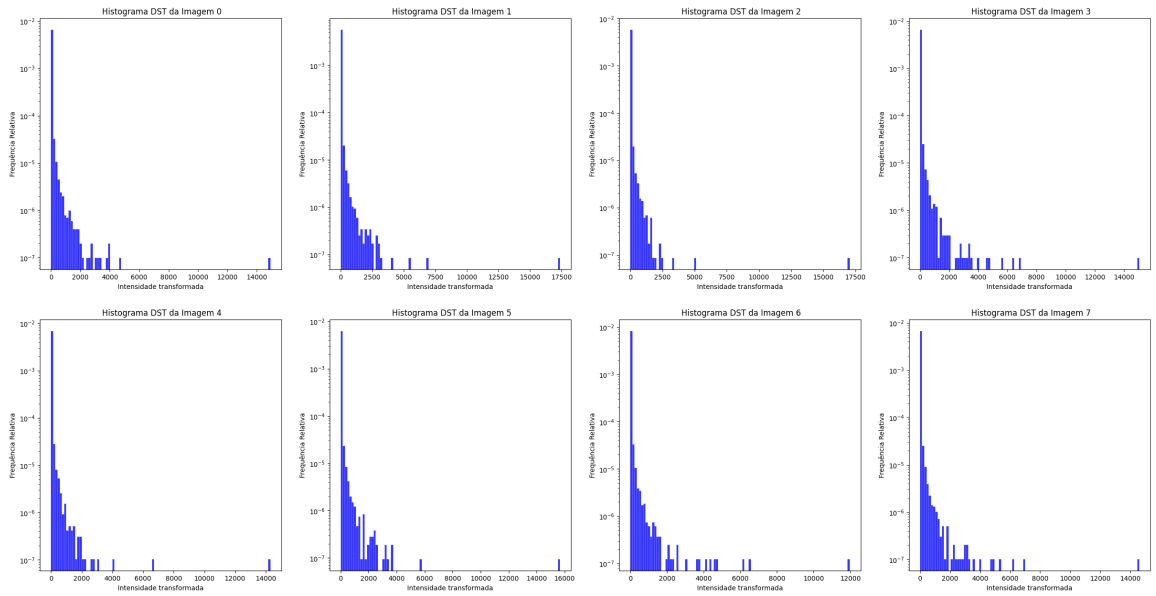

Figura 12: Histograma do logaritmo das intensidades após a transformação seno (DST).

De acordo com as estatísticas obtidas e considerando que, na transformada seno, as intensidades mais altas são as mais importantes, foi desenvolvido o seguinte critério para a redução de dimensionalidade: zerar todas as componentes com valor absoluto fora do intervalo entre a média e o valor máximo, ou seja, abaixo da média. Essa técnica foi adotada também levando em consideração a "grande concentração"de valores próximos à intensidade 0, visto que a média está próximo do valor mínimo e bem distante do valor máximo. Isso pode resultar em uma redução significativa do número de componentes, priorizando as maiores intensidades. Ao realizar essa abordagem, se resultou em média na redução de 87.52% das componentes.

Como forma de comparação, foi utilizado outro método para redução de dimensionalidade na DST. Esse método seleciona a mesma porcentagem de intensidades restantes, mas preservando preferencialmente as intensidades presentes no quadrante superior esquerdo da matriz, onde normalmente estão localizadas a maioria das intensidades mais altas.

Todas as operações necessárias para a realização dessas estratégias foram realizadas com a manipulação dos arrays `Numpy`.

Os resultados obtidos com as técnicas descritas acima estão apresentados na figura 13, onde "Reconstruída Q1"refere-se ao método que mantém o quadrante superior esquerdo das intensidades para comparação, e "Min-média"é o método desenvolvido com base na análise estatística.

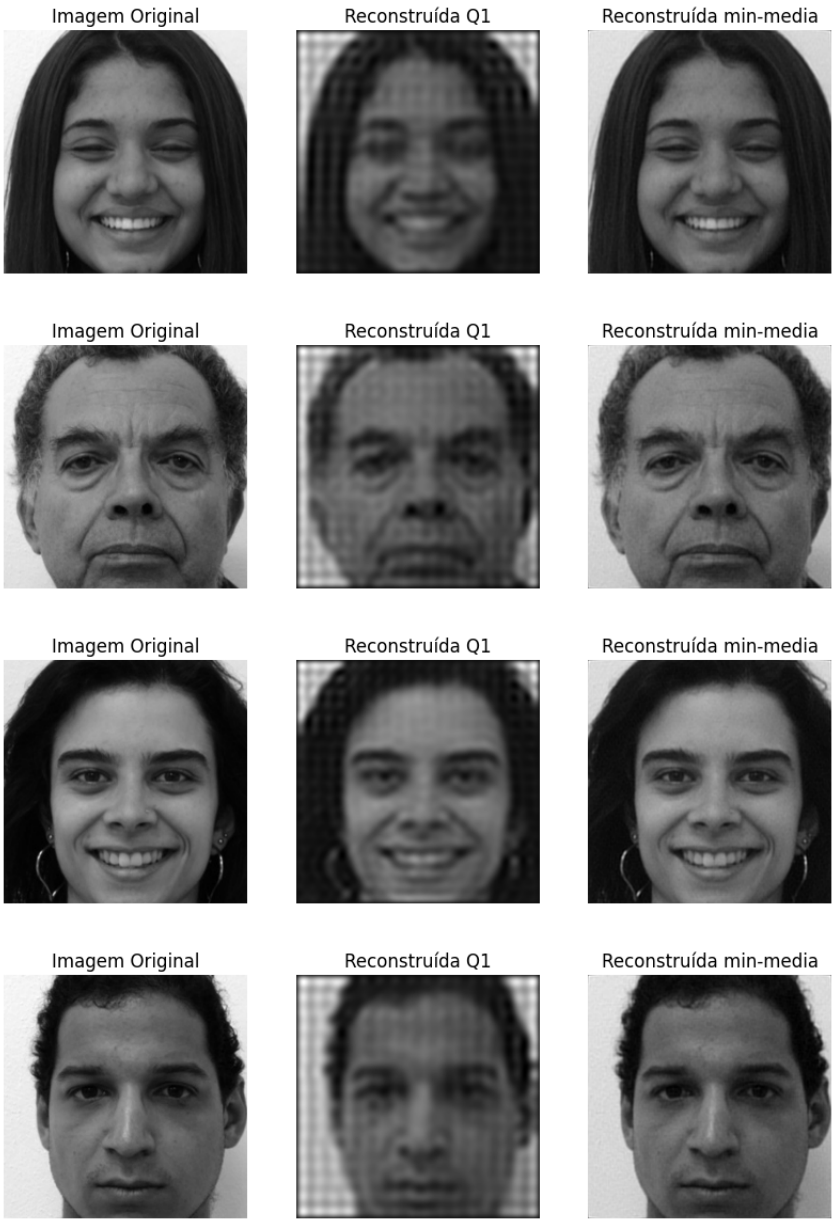

Figura 13: Imagens Reconstruídas - DST

Logo, a visualização evidencia como o critério desenvolvido preservou fortemente a qualidade das imagens, apresentando poucas diferenças em relação à imagem original, quando comparado ao outro método, no qual é possível perceber uma perda relevante. Além disso, considerando que, em média, apenas $12,48\%$ das intensidades foram utilizadas para a reconstrução com o método "Min-média", isso demonstra a eficiência do método visto à qualidade das imagens reconstruídas.

6 Questão 5

Agora, aplique o filtro passa-baixa do exercício 1 sobre as imagens na lista $I = [I_1, I_2, \ldots, I_8]$ *do exercício 4 para obter uma nova lista de* $I^\star = [I_1^\star, I_2^\star, \ldots, I_8^\star]$. *Calcule* $L_{abs}^\star$ *e suas estatísticas. Compare os resultados com as estatísticas de* L_{abs}. *Repita o mesmo para o filtro passa-alta do exercício 1.*

Resolução:

Utilizando o filtro passa-baixa Gaussiano construído na questão 1 e o método *convolve* da biblioteca `Scipy`, foi realizada a convolução no banco de imagens I definido na questão anterior. De forma análoga, aplicou-se o filtro passa-alta, que considera a derivada do filtro Gaussiano em relação à variável x.

De forma análoga ao procedimento realizado na questão anterior, os dois conjuntos de imagens correspondentes ao cálculo de cada filtro foram transformados para o domínio da frequência utilizando a Transformada Discreta do Seno (DST). Em seguida, foram calculados os valores de média, mediana, desvio padrão, mínimo e máximo para cada imagem. O resultado médio de cada conjunto, após a realização desses cálculos, está apresentado na Tabela 2.

Filtro	**Média**	**Mediana**	**Desvio Padrão**	**Mínimo**	**Máximo**
Original	8.2169	1.1968	89.2267	4.3522e-05	15121.0049
Passa-baixa	2.6108	5.4089e-06	80.3364	8.7297e-13	16088.6160
Passa-alta	16.5160	0.9477	179.9396	1.4909e-06	15394.3647

Tabela 2: Estatísticas médias dos conjuntos de imagens aplicados a diferentes filtros

Além disso, as figuras 14 e 15 ilustram como ficaram o histograma de frequências relativas de cada imagem em cada respectivo conjunto.

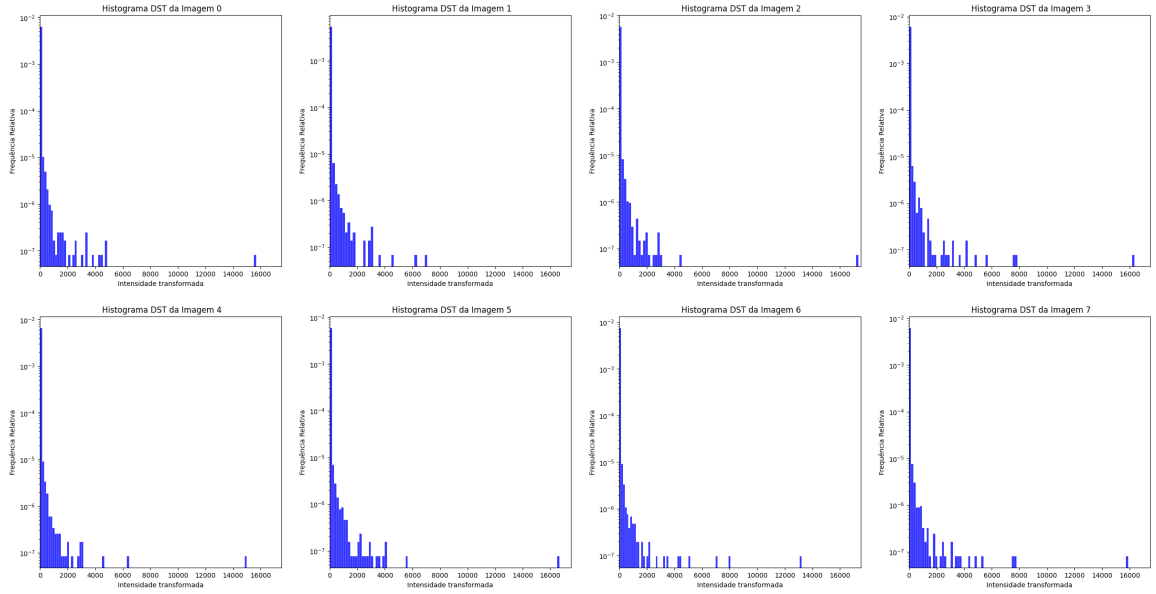

Figura 14: Histograma do logaritmo das intensidades após a transformação seno (Passa-baixa)

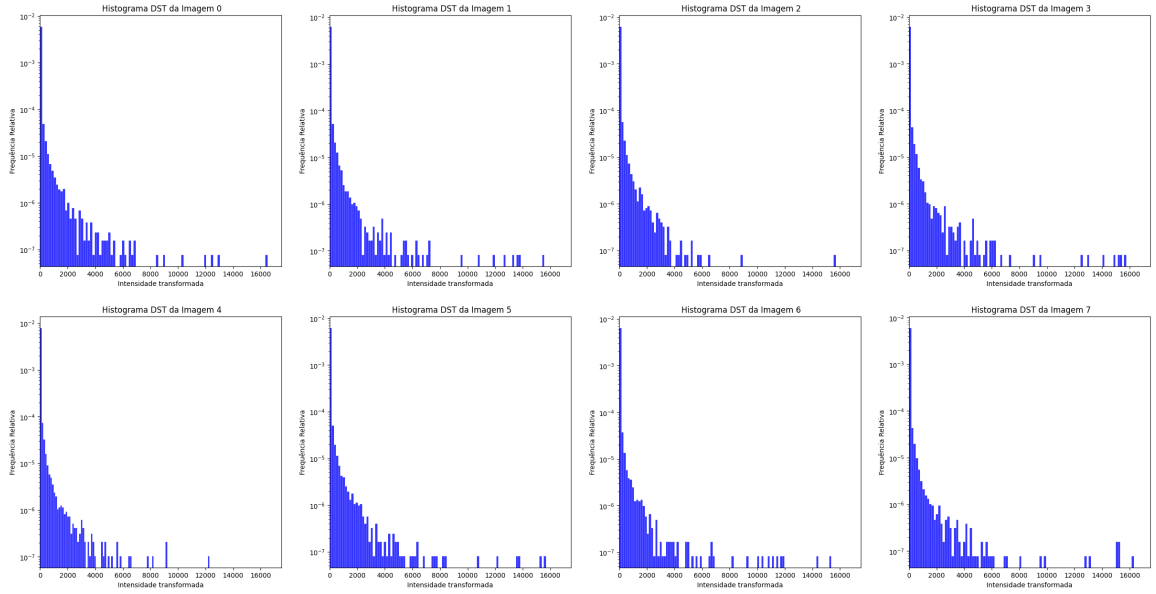

Figura 15: Histograma do logaritmo das intensidades após a transformação seno (Passa-alta)

Analisando os resultados apresentados na Tabela 2 e nas imagens 14 e 15, podemos observar os efeitos dos filtros passa-baixa e passa-alta aplicados às imagens.

O filtro *passa-baixa*, como esperado, causou uma redução significativa na média das imagens, que passou de 8.2169 no conjunto original para 2.6108. Isso reflete a atenuação das altas frequências, com o filtro suavizando a imagem e removendo detalhes finos de alta frequência. No histograma da imagem filtrada, podemos observar uma concentração de intensidades mais próximas de valores baixos, com uma distribuição de pixels mais comprimida. A mediana foi reduzida para 5.4089e-06, o que sugere que a maioria dos pixels da imagem ficou muito

próximo de um valor baixo. O desvio padrão também foi reduzido de 89.2267 para 80.3364, indicando uma diminuição da variação dos valores de intensidade, o que resulta em uma imagem mais homogênea. O valor máximo das intensidades aumentou levemente (de 15121.0049 para 16088.6160), refletindo um leve aumento das componentes de baixa frequência mais intensas, embora ainda dentro de uma faixa limitada.

Em contraste, o filtro *passa-alta* gerou uma mudança mais pronunciada nos dados. A média aumentou substancialmente, de 8.2169 para 16.5160, o que reflete o aumento das componentes de alta frequência e a ampliação dos detalhes da imagem. O histograma da imagem filtrada apresenta uma maior dispersão de intensidades, com picos mais acentuados em torno de valores mais altos, evidenciando o efeito do filtro na amplificação de detalhes agudos. A mediana, que era 1.1968 no original, foi reduzida para 0.9477, mostrando que, apesar do aumento da variabilidade, a distribuição dos pixels foi ligeiramente deslocada para a esquerda. O desvio padrão aumentou consideravelmente, de 89.2267 para 179.9396, como esperado, pois o filtro passa-alta acentua as variações locais na imagem, gerando maior contraste e detalhes. O valor máximo das imagens filtradas, que era 15121.0049, permaneceu relativamente estável (15394.3647), indicando que a intensidade máxima das regiões de alta frequência foi preservada, mas com a redistribuição dos valores dos pixels ao longo de um intervalo mais amplo.

Os histogramas das imagens filtradas são consistentes com os efeitos teóricos dos filtros. No caso do filtro passa-baixa, o histograma mostrou uma forte concentração de pixels em valores baixos, refletindo a suavização da imagem. Já o histograma do filtro passa-alta apresentou uma distribuição mais espalhada, com picos mais pronunciados em áreas de alta intensidade, evidenciando o aumento dos detalhes e a maior variabilidade das intensidades.

Em resumo, os resultados e histogramas confirmam que os efeitos dos filtros no domínio da frequência ocorreram conforme o esperado: o filtro passa-baixa suaviza a imagem, reduzindo a média, a variabilidade e a dispersão dos histogramas, enquanto o filtro passa-alta aumenta os detalhes e o contraste, evidenciado por uma maior dispersão e maior intensidade nos histogramas. A análise dessas métricas confirma que os efeitos dos filtros no domínio da frequência ocorreram conforme o esperado.

Referências

[1] Contributors, P. Pillow documentation. `https://pillow.readthedocs.io/`, 2024. Accessed: 2024-08-22.

[2] Developers, N. Numpy documentation. `https://numpy.org/doc/`, 2024. Accessed: 2024-08-22.

[3] Jain, A. K. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[4] learn Developers, S. Scikit-learn documentation. `https://scikit-learn.org/stable/`, 2024. Accessed: 2024-08-22.

[5] Miranda, L. C. P. Abordagens computacionais para calcular componentes principais ponderadas com aplicações em análise de imagens de faces humanas, 2023.

[6] OpenCV Team. Opencv documentation. `https://docs.opencv.org/`. Accessed: 2024-08-22.

[7] Team, M. D. Matplotlib documentation. `https://matplotlib.org/stable/contents.html`, 2024. Accessed: 2024-08-22.

[8] Thomaz, C. E. Fei face database. `https://fei.edu.br/~cet/facedatabase.html`, 2024. Accessed: 2024-08-14.

[9] van der Walt, S., Schönberger, J., Nunez-Iglesias, J., Boulogne, F., Warner, J., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors. *scikit-image: Image processing in Python*, 2014. Accessed: 2024-11-27.

[10] Virtanen, P., Gommers, R., Oliphant, T., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S., Brett, M., Wilson, J., Millman, K., Mayorov, N., Nelson, A., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E., Harris, C., Archibald, A., Ribeiro, A., Pedregosa, F., van Mulbregt, P., and Contributors, S. . *SciPy: Open source scientific tools for Python*, 2020. Accessed: 2024-11-27.

[11] Waskom, M., et al. Seaborn: statistical data visualization. `https://seaborn.pydata.org/`, 2024. Accessed: 2024-08-22.