INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS FLORIANO
CURSO TADS

JOÃO VITOR LEAL CRUZ

ANÁLISE DO ALGORITMO DE COMPRESSÃO DE IMAGEM UTILIZANDO SVD TRUNCADO

FLORIANO, PIAUÍ
2025

JOÃO VITOR LEAL CRUZ

ANÁLISE DO ALGORITMO DE COMPRESSÃO DE IMAGEM UTILIZANDO SVD TRUNCADO

Trabalho de Conclusão de Curso (artigo científico) apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Floriano.

Orientador: Profº. Me. André Francisco Coêlho Castro.

FLORIANO, PIAUÍ

2025

ANÁLISE DO ALGORITMO DE COMPRESSÃO DE IMAGEM UTILIZANDO SVD TRUNCADO

João Vitor Leal Cruz[1]
André Francisco Coêlho Castro[2]

RESUMO

Este trabalho aborda a compressão de imagens em tons de cinza por meio da decomposição de valores singulares truncada, com objetivo de quantificar o equilíbrio entre a redução do tamanho do arquivo e a preservação da qualidade visual. Para isso, foi montado um conjunto de vinte e nove imagens de diferentes resoluções e níveis de complexidade de textura. Cada imagem foi convertida para escala de cinza e submetida à decomposição de valores singulares truncada, sendo depois reconstruída variando o número de componentes significativos. Em cada configuração, calcularam-se a taxa de compressão, o erro quadrático médio e a relação sinal-ruído de pico em decibéis. Os experimentos foram realizados em ambiente Python, utilizando bibliotecas padrão para álgebra linear sem otimizações de hardware. Os resultados mostram como o aumento do número de componentes afeta simultaneamente a eficiência de compressão e a fidelidade visual, fornecendo um conjunto de tabelas e gráficos que ilustram a curva de desempenho do método. Como conclusão, verifica-se que a técnica se apresenta viável para aplicações acadêmicas, abrindo caminho para investigações futuras em imagens coloridas e implementações otimizadas.
Palavras-chave: compressão de imagem; svd truncado; python.

ABSTRACT

This study investigates grayscale image compression using truncated singular value decomposition, aiming to measure the balance between file size reduction and visual quality retention. A dataset of twenty-nine images with varying resolutions and texture complexities was assembled. Each image was converted to grayscale and processed through truncated singular value decomposition, then reconstructed by varying the number of significant singular values. For each configuration, compression ratio, mean squared error, and peak signal-to-noise ratio in decibels were computed. Experiments were conducted in a Python environment with standard linear algebra libraries and no hardware optimizations. The findings detail how the number of components impacts both compression effectiveness and visual fidelity, presenting tables and graphics that depict the performance curve of the method. The study concludes that truncated singular value decomposition is a viable approach for academic purposes, suggesting future work on color image applications and optimized implementations.
Keywords: image compression; truncated svd; python.

Data de aprovação: 15/07/2025

---

[1] Graduando em Tecnologia de Análise e Desenvolvimento de Sistemas. Instituto Federal do Piauí - Campus Floriano. jvitorlcruz@gmail.com.
[2] Mestre em Modelagem Matemática e Computacional pela UFPB. Instituto Federal do Piauí - Campus Floriano. andrecastro@ifpi.edu.br.

1 INTRODUÇÃO

Com o avanço e a normalização do uso da internet, houve um aumento expressivo na quantidade de informações que circulam pela rede. Além do volume, a qualidade dessas informações também evoluiu, como a resolução das imagens (quantidade de pixels), a frequência de reprodução de sons (em kHz ou mHz) e a fluidez dos vídeos (frames por segundo).

Esse crescimento, impulsionado tanto pelo aumento no número de usuários quanto pela evolução tecnológica, trouxe consigo um desafio significativo: a expansão exponencial da carga de dados em rede.

Para mitigar esse problema, diversas técnicas de otimização foram desenvolvidas, sendo a compressão de imagens o foco deste trabalho, que tem como objetivo a implementação de um algoritmo para compressão de imagens em escala de cinza com base na versão truncada do método de Decomposição de Valores Singulares (SVD).

As imagens digitais, fundamentais em diversas áreas como medicina, entretenimento e segurança, frequentemente contêm redundâncias que aumentam o consumo de espaço de armazenamento e largura de banda. Assim, a compressão de imagens torna-se uma ferramenta essencial para otimizar recursos computacionais e melhorar a experiência do usuário (Lotus e Viswanathan, 2014, p. 285).

A técnica explorada neste trabalho, o SVD, é uma abordagem matemática amplamente utilizada na análise e aproximação de matrizes. O método escolhido, o SVD truncado, permite obter uma versão comprimida da imagem original, preservando suas principais características visuais. O que será avaliado utilizando algumas métricas como Relação Sinal-Ruído de Pico (PSNR), Erro Quadrático Médio (MSE) e taxa de compressão.

O uso do SVD é justificado pois suas bases matemáticas são usadas nos mais diversos campos, como Dongarra *et al.* (2018, p. 810, tradução nossa[3]) explicou que “as aplicações são tão diversas quanto *squares data fitting*, compressão de imagem, reconhecimento facial, análise do componente principal,

[3]No original: Applications are as diverse as least squares data fitting [53], image compression [3], facial recognition [111], principal component analysis [92], latent semantic analysis [28], and computing the 2-norm, condition number, and numerical rank of a matrix.

análise latente semântica e o cálculo da norma 2, número de condição e posto numérico de uma matriz".

Portanto, a relevância deste trabalho está na avaliação da eficiência do método SVD truncado, ao se verificar estatisticamente os valores obtidos de sua taxa de compressão, erro quadrático médio e relação sinal-ruído de pico, o que tanto de forma direta fala sobre seu funcionamento na compressão de imagem, quanto pode indiretamente mostrar indícios sobre sua efetividade nos demais campos.

2 REFERENCIAL TEÓRICO

2.1 COMPRESSÃO DE DADOS

Todo algoritmo produz uma ação sobre dados e esses dados existem nas estruturas mais variada, desde textos, imagens, sons e etc, contudo nem sempre os dados são utilizados em sua forma pura, isso ocorre pela influência da Teoria da Informação, que segundo Cover e Thomas (2006, p. 1, tradução nossa[4]) "responde duas questões fundamentais na teoria da comunicação: qual a compressão de dados final (resposta: entropia H), e qual a taxa de transmissão de comunicação final (resposta: capacidade do canal C)."

Dessa forma, em busca de otimizar a manipulação dos dados, seja apresentando, modificando, armazenando e transmitindo, existe a compressão de dados, "a arte ou ciência de representar informação de forma compacta" (Sayood, 2012, p. 1, tradução nossa[5]). Tal representação compacta se refere à forma primária dos dados, ou seja, os bytes que os compõem, a fim de atingir o objetivo de compactá-lo, faz-se necessário reduzir a quantidade de bytes utilizada para armazenar a informação.

No entanto, esse seria somente um dos algoritmos, algoritmo de compressão, que fazem parte do processo de compressão, pois existe o algoritmo de reconstrução, que atua sobre o produto gerado pelo primeiro e produz uma reconstrução da informação comprimida. Durante esse segundo processo, há chance de uma perda de informação em referência à contida no original, que o leva

---

[4]No original: Information theory answers two fundamental questions in communication theory: What is the ultimate data compression (answer: the entropy H) and what is the ultimate transmission rate of communication (answer: the channel capacity C).

[5]No original: In brief, data compression is the art or science of representing information in a compact form.

às classificações dos processos de compressão em *Lossless* (sem perda) e *Lossy* (com perda) (Sayood, 2012, p. 3-4).

Logo, levando em consideração que o trabalho em questão trata de imagens, o algoritmo produzido nesse trabalho terá como seu fim a compressão de imagens.

2.2 COMPRESSÃO DE IMAGEM

Dentre as técnicas de compressão de dados, a compressão de imagem trabalha especificamente com imagens digitais, que são formadas por capturas do mundo real ou de uma imagem gerada previamente, que são chamados *pixels* ou *pels* (picture elements), originados de uma fonte contínua em espaço e amplitude (Jones e Rabbani, 1991, p. 4).

Isso irá resultar numa organização em forma de matriz, para que se possa representar todos os valores que compõem a imagem em sua altura e largura, até mesmo suas possíveis camadas. Ademais, tais técnicas até mesmo levando seu nome como a extensão da imagem que foi gerada, como *Joint Photografic Experts Group* (JPEG) e *Portable Network Graphics* (PNG).

Entretanto, as imagens possuem dentro de sua estrutura de representação uma considerável quantidade de informações desnecessárias. Tal fato é apontado de forma exímia por Jones e Rabbani (1991, p. 3, tradução nossa[6]) que explicam: “imagens digitais, em sua representação canônica, geralmente contém uma significativa quantidade de redundância.”.

Seguindo essa lógica, as imagens digitais podem ter seu conteúdo reduzido sem necessariamente ter um prejuízo visível à informação que é passada, pois a redundância serve apenas para ocupar o espaço de armazenamento sem que haja acréscimo de novas informações ao conteúdo. Por isso, mesmo que a técnica de compressão utilizada neste trabalho, Decomposição em Valores Singulares, seja classificada como *Lossy*, dependendo da forma de sua implementação o grau de perda da imagem é mínimo, sendo muitas vezes imperceptível para o observador.

2.3 JPEG

O formato JPEG é originado do método de mesmo nome, que é “projetado para comprimir imagens coloridas ou em tons de cinzas de cenas naturais, que são

---

[6]No original: Fortunately, digital images, in their canonical representation, generally contain a significant amount of redundancy.

do mundo real" (Dhawan, 2011, p. 23, tradução nossa[7]). Sua compressão é baseada na formulação matemática da Transformação Discreta de Cosseno (DCT), o que tem por função a conversão de um sinal em outro tipo, o qual pode ser aplicado às imagens, transformando elas, que são representações matriciais do espaço, em dados números, ou seja, coeficientes.

Seu processo começa pela segmentação da imagem em camadas de blocos 8x8, os quais são definidos assim pelo tempo de execução que cresce exponencialmente quanto maior for. Esses blocos sofrendo do uso do DCT são transformados para coeficientes dentro do domínio de frequência, os quais são rearranjados de acordo com sua magnitude de informação, para serem comprimidos posteriormente por estratégias como codificação *run length*, codificação aritmética ou codificação de Huffman (Dhawan, 2011, p. 23).

Sua fórmula para matrizes de duas dimensões é:

$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x,y)\, COS\left[\frac{(2x+1)i\pi}{2N}\right] COS\left[\frac{(2y-1)j\pi}{2N}\right]$$

$C(x) = \frac{1}{2N}$ se x = 0, senão 1 se x > 0.

Existe essa diferenciação para a fórmula de duas dimensões, porque existe também a aplicação do DCT em matrizes de uma dimensão, a qual pode ser feita usando somente um passo a passo estruturado e tabelas, mas sem necessidade de formalização matemática. Além disso, existe o padrão JPEG que fornece um codec, o qual define como uma imagem deve ser comprimida em uma linha de *bytes* e descomprimida novamente em uma imagem. Tal tabela é utilizada para normalizar os coeficientes obtidos.

2.4 DECOMPOSIÇÃO EM VALORES SINGULARES

O método de Decomposição em Valores Singulares permite trabalhar abrangentemente com matrizes, pois ele é orientado a dados, o que possibilita efetuar suas operações puramente com o que é recebido como *input*, sem a necessidade de algum auxílio ou conhecimento externo ou intuitivo. Além disso, ele

[7]No original: JPEG is designed for compressing full-color or gray-scale images of natural, real-world scenes

é muito importante na computação dentre os métodos de aproximação de matriz existentes, pois fornece uma decomposição numericamente estável e que é garantido de existir (Brunton e Kutz, 2017, p. 4). Segue-se a sua fórmula:

$$X = U\Sigma V^{*}$$

Por meio da análise do posto de uma matriz, que corresponde aos valores singulares não nulos *r*, é possível entender a relação de dependência entre as colunas e linhas de uma dada matriz, essa relação de dependência revela a existência e, caso haja, uma quantificação da redundância dentro dessa matriz. O que é tratado por meio da aproximação de baixo posto, em que uma matriz $X_{n\times m}$ é representada por meio de duas matrizes unitárias ortogonais U e V^{*}, que descrevem respectivamente as linhas e colunas da matriz X, contudo enquanto X se utiliza de *m* linhas e *n* colunas, $U_{m\times m}$ descreve suas linhas por meio de *m* linhas e *m* colunas. De forma igual, é possível descrever a relação da matriz unitária ortogonal $V_{n\times n}$ com a matriz X, V sendo composta por *n* linhas e *n* colunas

Ambas podendo ser escrita conforme segue: V = $\{v_1, v_2, \ldots, v_n\}$; U = $\{u_1, u_2, \ldots, u_m\}$.

Enquanto que os elementos de U são definidos pela fórmula: $u_i = \frac{i}{\sigma_i} X v_i,\ i = 1, \ldots, m.$

O símbolo $(*)$ está aqui para representar uma transposta conjugada, que cumpre a função de manter a ortonormalidade das matrizes, assim garantindo, por consequência, a unitariedade de V, essa sendo uma característica amplamente explorada durante toda a execução dessa operação. Pode-se notar também, que tal característica contribui para a capacidade do método SVD, em que ele consegue operar sobre qualquer matriz, desde que ele consegue atuar sobre matrizes reais e complexas.

A matriz Σ é uma matriz diagonal, que tem como valores apenas números não negativos organizados de forma hierárquica, isso acontece dessa maneira, pois ela serve como um descritor da magnitude das informações que estão contidas na matriz X, ou seja, os valores dos elemento σ_{ij}, sendo i = j, são organizados tal que $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq 0$. Portanto, a matriz Σ é utilizada como o limitador de até

qual grau de informação seria relevante para ser mantido ou excluído do resultado desejado da aproximação.

É possível concluir, seguindo o exposto, que pode-se representar a matriz X em forma do produto de seus fatores ou fatorada: $X = u_1 \sigma_1 v_1^* + \ldots + u_r \sigma_r v_r^* = \sum_{i=1}^{r} u_i \sigma_i v_i^*$.

2.5 DECOMPOSIÇÃO EM VALORES SINGULARES TRUNCADA

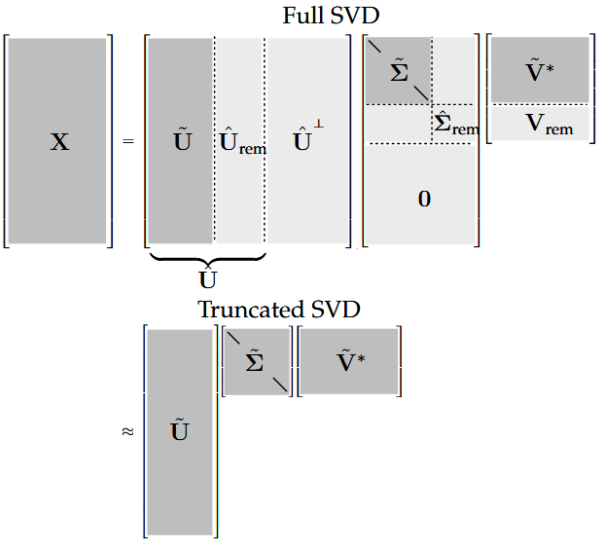
A aproximação que é feita usando o SVD ao se decidir qual a magnitude da informação a ser usada, pode ser feita de forma empírica, ao se testar repetidas vezes e alcançar um resultado desejado. Mas, “Schmidt (de Gram-Schmidt) generalizou o SVD para espaços de função e desenvolveu uma teorema de aproximação, estabelecendo o SVD truncado como uma ótima aproximação de baixo posto da matriz subjacente X.” (Brunton e Kutz, 2017, p. 8, tradução nossa[8])

Logo, ao se utilizar da fatorização que o SVD possibilita para fazer a aproximação de uma dada matriz decomposta, utiliza-se da técnica do SVD truncado, em que é feita a seleção dos *k* primeiros valores singulares, obtendo-se a representação da matriz original de forma consistente, pois a relevância da informação perdida pode ser considerada irrisória.

Dessa forma, tomando que *k* < *r*, em que *r* representa o posto e os valores singulares não nulos da matriz original X, é possível reescrever a representação da matriz fatorada como: $X_k = \sum_{i=1}^{k} u_i \sigma_i v_i^*$ (Figura 2). Tendo-se, portanto, uma das melhores aproximações de posto *k* dessa dada matriz, pois essa aproximação mantém os valores singulares com uma maior magnitude de informação, a qual comumente consegue representar entre 90% a 99% da matriz original no truncamento. (Brunton e Kutz, 2017, p. 35)

[8]No original: Schmidt (of Gram-Schmidt) generalized the SVD to function spaces and developed an approximation theorem, establishing truncated SVD as the optimal low-rank approximation of the underlying matrix X.

Figura 2 - Demonstração do SVD truncado



Fonte: Brunton e Kutz (2022)

Observando a Figura 2, é possível observar a forma como é truncada a matriz original, a fim de diminuir seu tamanho, enquanto mantém-se a parte mais relevante da informação dentro da matriz truncada que é produzida.

2.6 ALGORITMO

Para o funcionamento de qualquer processo ou para a “ação”, como a citado acima, dentro de um dispositivo de computação existe a necessidade da criação e implementação de um algoritmo, especificamente um algoritmo computacional, que segundo Cormen (2014, p. 2) “é um conjunto de etapas para executar uma tarefa descrita com precisão suficiente para que um computador possa executá-la”.

Há a necessidade dessa driscriminação, pois a palavra “algoritmo” possui um significado de maior abrangência, como bem conceituado por Manzano (2000, p. 6) “algoritmos são regras formais para a obtenção de um resultado ou da solução de um problema, englobando fórmulas de expressões aritméticas”.

É possível perceber pela definição apresentada acima, que o dado conceito de algoritmo surge sob influência da área matemática, em que, embora, exija um rigor na sequenciação dos passos lógicos, sendo indispensável a coerência e coesão, na parte prática da implementação pelo matemático de algum algoritmo podem ocorrer saltos lógicos. Tal capacidade, de adaptar o algoritmo durante a sua

execução, encontra-se em falta nos dispositivos de computação nativamente e mesmo que haja a possibilidade disso, não se enquadra no escopo do atual projeto.

Portanto, a ação que o algoritmo computacional proposto irá permitir o computador realizar, se caracteriza como uma ação estática, que segundo Farrer (1999) “ação é um acontecimento que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido”.

3 METODOLOGIA

O presente trabalho tem como objetivo analisar a eficiência do método de compressão de imagem SVD truncado, por meio da aplicação do método em várias imagens e fazendo a avaliação de suas compressão se utilizando de métricas como erro quadrático médio (MSE), relação sinal-ruído de pico (PSNR) e taxa de compressão.

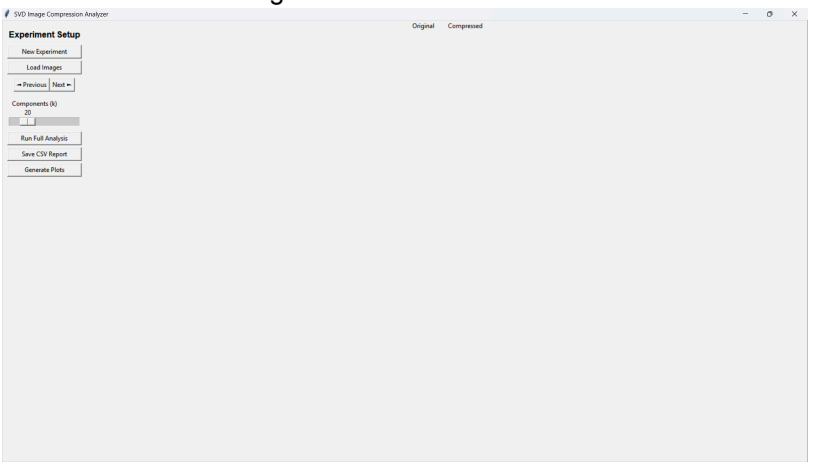
Tal forma de análise torna essa pesquisa de natureza experimental, por se propor a realizar simulações do método desenvolvido sob condições controladas, que seriam a do notebook pessoal do autor executando as instruções criadas também pelo autor do trabalho em questão. O ambiente de execução é composto por: Windows 11 (versão 10.0.26100), CPU Intel(R) Core(TM) I5-1235U, GPU Intel(R) Iris(R) Xe Graphics e memória RAM de 8GB.

Essas condições se enquadram na definição do Gil(2002, p. 47) sobre pesquisa experimental, que segundo ele “consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observações dos efeitos que a variável produz no objeto.”

Em relação ao objeto de estudo, aquele que foi selecionado é um processo de compressão de imagem baseado no método SVD truncado, que foi desenvolvido em linguagem de programação *python*, junto dele houve a seleção de um *dataset* de 1344 imagens, das quais apenas algumas foram selecionadas. As bibliotecas *python* utilizadas foram: *os* (versão do python 3.12.10), *numpy* (versão 1.26.4), *tkinter* (versão 8.6), *PIL* (versão 11.3.0), *sklearn* (versão 1.7.0), pandas (versão 1.5.2) e *matplotlib* (3.10.3).

A biblioteca *tkinter* foi utilizada para o desenvolvimento da Interface Gráfica do Usuário (GUI) (Figura 3).

Figura 3 - Interface *Tkinter*



Fonte: Elaborado pelo autor (2025)

Pode-se perceber pela visualização da Figura 3, que a interface possui alguns componentes, em sua maioria botões. *New Experiment* é o acionador de todo o processo, o qual irá permitir rodar as análises e fazer a geração dos arquivos de gráfico com *Generate Plots* e salvar os arquivos com as métricas em *Save CSV Report*.

Esses botões funcionam junto com o *slider* do componente k, que serve para gerar relatórios manuais, mas é possível rodar o algoritmo por 10 valores de k automaticamente acionando o botão *Run Full Analysis*, o qual já salva os gráficos.

Load Images tem como função selecionar a pasta das imagens utilizadas, fazendo com que sejam carregadas para dentro do programa. Junto a ele, têm-se os botões *Previous* e *Next*, que servem para navegar entre as imagens carregadas.

O código em si foi desenvolvido de forma modular, que significa a divisão em métodos, o que permite uma melhor leitura e facilita a manuseabilidade dele, pois ao alterar alguma funcionalidade, se faz necessário apenas modificar uma parte isolada, do que o conjunto como um todo. Há um total de 11 métodos construídos, mas para a execução do SVD trucado e mensuração de métricas se fazem necessários 2 deles (Figura 4).

Figura 4 - Métodos de Compressão e Avaliação

```python
    def process_image(self, k):
        if self.original_image is None:
            return None

        img_float = self.original_image.astype(np.float64) / 255.0
        svd = TruncatedSVD(n_components=k)
        reduced = svd.fit_transform(img_float)
        restored = svd.inverse_transform(reduced)
        return restored

    def calculate_metrics(self, original, compressed, k):
        h, w = original.shape

        # Storage calculations
        original_bytes = (h * w * 8) / 1024
        compressed_bytes = (k * (h + w + 1) * 8) / 1024

        # Quality metrics
        original_float = original.astype(np.float64) / 255.0
        compressed_clipped = np.clip(compressed, 0, 1)

        mse = mean_squared_error(original_float, compressed_clipped)
        psnr = peak_signal_noise_ratio(original_float, compressed_clipped, data_range=1.0)
        ratio = compressed_bytes / original_bytes

        return {
            'filename': os.path.basename(self.image_files[self.current_index]),
            'dimensions': f"{w}x{h}",
            'k': k,
            'max_k': self.max_k,
            'mse': mse,
            'psnr': psnr,
            'compression_ratio': ratio,
            'original_kb': original_bytes,
            'compressed_kb': compressed_bytes
        }
```

Fonte: Elaborado pelo autor (2025)

Como visível na Figura 4, para a compressão da imagem foi utilizado a classe *TruncatedSVD* disponibilizada pela biblioteca *sklearn* e por meio dela usa-se os métodos *fit_transform* e *inverse_transform*, os quais, respectivamente, fazem a redução da matriz com o parâmetro k passado, que vai de 5 até a dimensão de menor valor da matriz, o que seria min(altura, largura)-1. Posteriormente, a matriz comprimida passa pelo processo de reconstrução, para que seja possível a sua visualização, pois a função *fit_transform* retorna as matrizes que compõe o SVD separadas e já truncadas.

Abaixo da função de compressão, existe a função responsável pelo cálculo das métricas. As imagens foram todas equalizadas no seu *data type* para ficarem compatíveis com o tipo produzido pela compressão via SVD, que é o *float64.*

Então para saber o valor o da taxa de compressão é preciso calcular os tamanhos ocupados por cada imagem, multiplicando sua quantidade de *pixels* pelos *bytes* representando cada um desses *pixels*. Prosseguindo, usa-se funções prontas para calcular tanto o MSE quanto o PSNR. Ademais, todas as informações que foram utilizadas são retornadas para criação de uma linha no arquivo do relatório.

No que tange à seleção do *dataset* de imagens (Figura 5), ele teve suas imagens colocadas em tons de cinza, para reduzir a escala de trabalho computacional necessária. As imagens serão retiradas do *Super Resolution Benchmarks*(Agustsson e Timofte, 2017), que é um *dataset* voltado para trabalhos de Inteligência Artificial e Aprendizado de Máquina, mas que será útil para o proposto na pesquisa. Ele contém imagens das mais variadas formas (tabela 1), mas foram selecionadas as imagens mais coesas dentro do formato *bitmap*, somando 29.

Figura 5 - Exemplo de Imagens do *Dataset*

Fonte: Elaborado pelo autor (2025)

Tabela 1 - Quantidade de imagem por tamanho

dimensions	qtd_files
480x720	6
610x488	1
618x453	1
627x482	1
632x505	1
634x438	1
634x505	1
640x512	3
768x512	14

Fonte: Elaborado pelo autor (2025)

Vale ressaltar também que as variáveis estudadas e que são exibidas nos resultados são de natureza quantitativa, pois existem enquanto traduções numéricas

do fenômeno em estudo, que é a compressão de imagem. Por intermédio das métricas supracitadas, pode-se ter um entendimento estatístico de quão bem utilizável será o SVD truncado desenvolvido, especificamente e respectivamente, quão próxima a imagem comprimida está da original e quão bem comprimida está a imagem.

Aprofundando-se nas métricas, o MSE é utilizado para medir a distância entre a imagem comprimida e a original, fornecendo uma estimativa quantitativa do grau de distorção causado pela compressão. Embora o MSE seja amplamente utilizado devido à sua simplicidade, ele pode ser complementado por outras métricas, como o PSNR, que é mais intuitivo para avaliar a qualidade percebida de imagens comprimidas. Isso permite uma análise mais robusta e comparativa entre diferentes técnicas de compressão, ampliando a compreensão sobre a eficácia do método desenvolvido.

Sendo fórmulas complementares, ao se utilizar o MSE para fazer a comparação entre duas imagens O e C, respectivamente, a sua versão original e sua versão comprimida, ambas de tamanho M e N:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} ||O(i,j) - C(i,j)||^2,$$

obtêm-se assim o resultado que será utilizado no PSNR:

$$PSNR: 10 \cdot log_{10}(\frac{MAX_I^2}{MSE}) = 20 \cdot log_{10}(\frac{MAX_I}{\sqrt{MSE}})$$

$$PSNR: 20 \cdot log_{10}(MAX_I) - 10 \cdot log_{10}(MSE).$$

em que MAX indica o valor máximo de pixel possível em uma imagem. Por estar trabalhando com *bits* (B), use-se $MAX_I = 2^B - 1$.

Ademais, sobre a taxa de compressão, essa é uma métrica que está intrinsecamente relacionada ao processo de compressão, no caso concreto do SVD truncado, por causa da sua aproximação de baixo-posto, essa métrica vai depender de quantos *k* valores singulares foram selecionados, quanto menor *k*, menor a quantidade de informação necessária para representar a imagem, logo maior a taxa de compressão. A fim, de descobrir seu valor exato usa-se: $T = \frac{TmC}{TmO}$, em que TmO significa tamanho da imagem original e TmC significa tamanho da imagem comprimida.

3 RESULTADOS

Para uma análise dos resultados na primeira imagem do *dataset* pode-se observar na tabela 2, que contém a evolução das métricas.

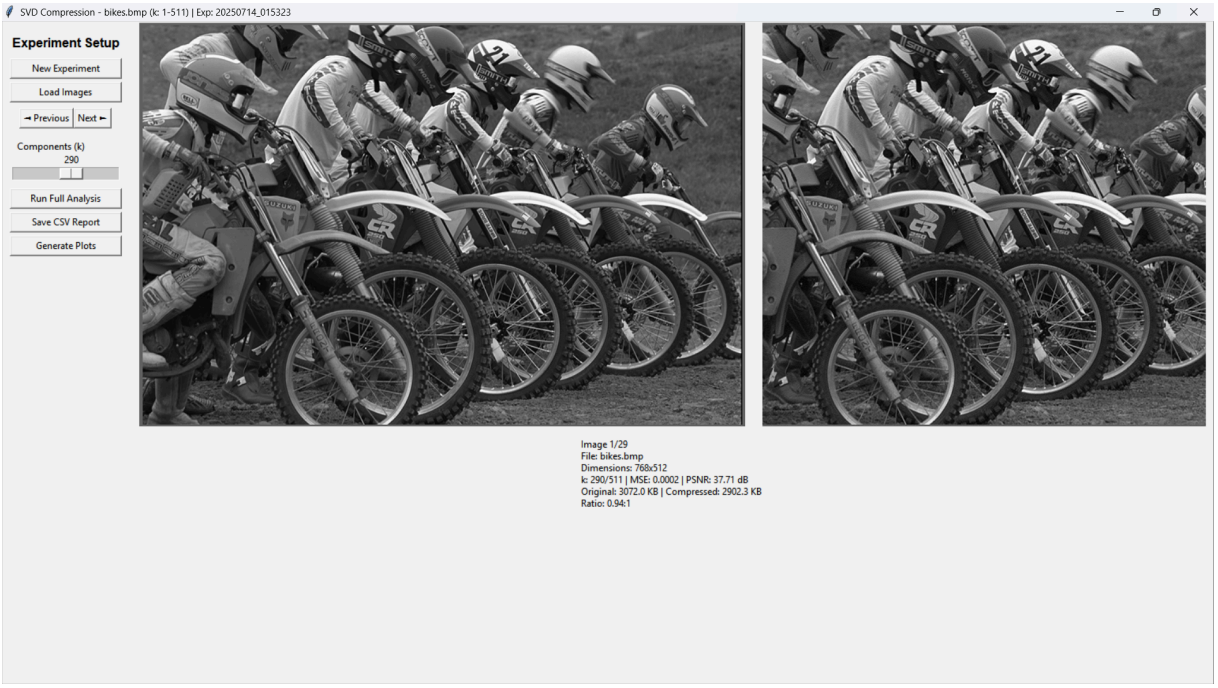
Tabela 2 - Evolução do MSE, PSNR, Taxa de Compressão por k

arquivo	dimensões	k	max_k	mse	psnr	taxa de compressão
bikes.bmp	768x512	5	511	0.021632704792209852	16.648891762200865	0.01628875732421875
bikes.bmp	768x512	61	511	0.005049603308275487	22.96742738276172	0.19872283935546875
bikes.bmp	768x512	117	511	0.002219421873175775 2	26.5376013815645	0.38115692138671875
bikes.bmp	768x512	173	511	0.001021388397601926 5	29.908090797936563	0.5635910034179688
bikes.bmp	768x512	229	511	0.000451265580342486 64	33.45567790427099	0.7460250854492188
bikes.bmp	768x512	286	511	0.000181631297659890 36	37.40809314240939	0.9317169189453125
bikes.bmp	768x512	342	511	6.515891840541811e-05	41.86026133302404	1.1141510009765625
bikes.bmp	768x512	398	511	1.900684920414055e-05	47.21089870860375	1.2965850830078125
bikes.bmp	768x512	454	511	3.815312143598176e-06	54.18469925154266	1.4790191650390625
bikes.bmp	768x512	511	511	2.159269592776668e-29	286.65693131011824	1.6647109985351562

Fonte: Elaborado pelo autor (2025)

De acordo com a tabela, é possível estabelecer uma relação inversamente proporcional entre as métricas MSE e PSNR, as quais demonstram por meios diferentes quão próxima da imagem original está a imagem comprimida. Além disso, o melhor valor de PSNR obtido durante os testes dessa imagem foi em torno de 37 dB, o que visualmente representa pouca diferença para a imagem original como pode ser observado na Figura 7 logo abaixo.

Figura 7 - Imagem Original vs Imagem Comprimida



Fonte: Elaborado pelo autor (2025)

Seguindo adiante, na tabela 3, em que há a seleção da média de compressão pelo tamanho da respectiva imagem, pode-se perceber que a taxa de compressão vai de 41% a 50%, o que mostra um ótimo grau de compressão para imagens do tipo *bitmap*, que um formato de imagem praticamente sem compressão nenhuma.

Tabela 3 - Taxa de Compressão por Tamanho de Imagem

dimensions	compression_ratio
480x720	0.473774112654321
610x488	0.41275262026337006
618x453	0.49332866589987406
627x482	0.5019621857359354
632x505	0.41147136232610604
634x438	0.4829991501375625
634x505	0.4108942124496361
640x512	0.4116851806640625
768x512	0.47291692097981763

Fonte: Elaborado pelo autor (2025)

Sabendo de todas as informações supracitadas, chega-se à questão sobre os melhores valores de k para a faixa de tamanho trabalhada. A resposta encontra-se na tabela 4, que traz os valores de k com seus respectivos números de MSE, PSNR, todos próximos ou dentro do padrão tolerável de variação visual da imagem, e da taxa de compressão, que se mantém na média já citada na tabela acima.

Tabela 4 - Média das Métricas para os Melhores Valores de k

	mse		psnr		compression_ratio	
	mean	max	mean	max	mean	max
k						
101	0.0004670938262668	0.0004670938262668	33.30595872887648	33.30595872887648	0.3902633133111505	0.3902633133111505
104	0.0020169187074708	0.0020169187074708	26.953116058072247	26.953116058072247	0.398236853197311	0.398236853197311
110	0.0010211029781184	0.0010211029781184	29.909304571556063	29.909304571556063	0.4040183446167285	0.4040183446167285
112	0.0006119314031901	0.0006119314031901	32.132972590402 69	32.132972590402 69	0.4134909970438054	0.4134909970438054
115	0.001426435453388	0.0020043722395304	28.846852281744702	30.713488355010764	0.40975754375741047	0.4100451184358942
117	0.0022265210299253334	0.003244832318011	27.16176076254156	30.728936034995733	0.4116851806640625	0.4116851806640625

Fonte: Elaborado pelo autor (2025)

3 CONSIDERAÇÕES FINAIS

Neste trabalho, investigou-se a aplicação do SVD truncado como técnica de compressão de imagens em tons de cinza. A análise de 29 amostras indicou que valores de *k* de 101 até 117 apresentam o melhor compromisso entre redução do tamanho do arquivo (taxa média de compressão entre 40%) e preservação da qualidade visual (PSNR médio por volta de 30 dB). Esses resultados corroboram achados de estudos anteriores, demonstrando que o SVD oferece uma estratégia viável quando busca-se compressão sem perdas drásticas de fidelidade.

Entretanto, algumas limitações do presente estudo merecem destaque. Primeiro, o experimento concentrou-se exclusivamente em imagens monocromáticas, o que restringe a aplicabilidade direta dos resultados a cenários de imagens coloridas. Além disso, a seleção das amostras, embora tenha considerado variações de resolução e conteúdo, não explorou exaustivamente casos de alta e baixíssima complexidade de textura. Por fim, a implementação em Python puro, sem otimizações via bibliotecas de baixo nível ou processamento paralelo, pode não refletir a performance real em aplicações de larga escala.

Como desdobramento para trabalhos futuros, propõe-se estender o estudo para imagens coloridas, seja tratando cada canal RGB separadamente ou aplicando versões do SVD a tensores tridimensionais e incorporar métricas perceptuais como SSIM e MS-SSIM para complementar o PSNR na avaliação de qualidade visual. Essas iniciativas pavimentam o caminho para um estudo futuro que investigará a integração do SVD truncado em fluxos de processamento ao vivo, avaliando sua viabilidade em cenários de vídeo em tempo real.

REFERÊNCIAS

BRUNTON, Steven L.; KUTZ, J. Nathan. ***Data-Driven Science and Engineering:: Machine Learning, Dynamical Systems, and Control***. 2. ed. [S. l.]: Cambridge University Press, 2022. 614 p. ISBN 978-1009098489.

CORMEN, Thomas H. ***Algorithms Unlocked***. [S. l.]: The MIT Press, 2013. 188 p. ISBN 978-0-262-51880-2.

COVER, Thomas M.; THOMAS, Joy A. ***Elements of Information Theory***. 2. ed. [S. l.]: Wiley-Interscience, 2006. 748 p. ISBN 978-0-471-24195-9.

CRUZ, João Vitor L; **Código de Compressão de Imagem com SVD truncado**, Código-fonte, Instituto Federal do Piauí - Campus Floriano, Junho de 2025; Disponível em: <https://github.com/FileQueTal/tcc_svd_truncated>. Acesso em: 14 jul. 2025

DHAWAN, Sachin. ***A Review of Image Compression and Comparison of its Algorithms***. IJECT 2(1), [s. l.], v. 2, p. 22-26, março 2011.

DONGARRA, Jack et al. ***The Singular Value Decomposition: Anatomy of Optimizing an Algorithm for Extreme Scale***. SIAM Review, [S. l.], v. 60, n. 4, p. 808 - 865, jan. 2018. Recebido pelos editores em 21 de fevereiro de 2017; aceito

para publicação (em forma revista) 22 de março de 2018; publicado eletronicamente em 8 de novembro de 2018.

EIRIKUR AGUSTSSON; RADU TIMOFTE. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. ***Computer Vision and Pattern Recognition***, 21 jul. 2017.

FARRER, H. **Pascal estruturado**. 3. ed. Rio de Janeiro: LTC, 1999.

GIL, Antonio Carlos. **Como Elaborar Projetos De Pesquisa**. 4. ed. [S. l.]: EDITORA ATLAS, 2002. 175 p. ISBN 85-224-3169-8.

JONES, Paul W.; RABBANI, Majid. ***Digital Image Compression Techniques***. [S. l.]: SPIE--The International Society for Optical Engineering, 1991. 221 p. ISBN 0-8194-0648-1.

LOTUS, Rayappan; VISWANATHAN, Gomathi K. ***DIGITAL IMAGE COMPRESSION TECHNIQUES***. International journal of research in engineering and technology, v. 3, n. 10, p. 285-190, 25 out. 2014.

MANZANO, José Augusto N. G., **Estudo Dirigido: ALGORITMOS** - Editora Érica, 2000.

SAYOOD, Khalid. ***Introduction to Data Compression***. 4. ed. [S. l.]: Morgan Kaufmann, 2012. 768 p. ISBN 978-0-12-415796-5

APÊNDICE A –Tabela de Evolução do MSE, PSNR, Taxa de Compressão por k

arquivo	dimensões	k	max_k	mse	psnr	taxa de compressão
bikes.bmp	768x512	5	511	0.021632704792209852	16.648891762200865	0.01628875732421875
bikes.bmp	768x512	61	511	0.005049603308275487	22.96742738276172	0.19872283935546875
bikes.bmp	768x512	117	511	0.002219421873175775	26.5376013815645	0.38115692138671875
bikes.bmp	768x512	173	511	0.001021388397601926	29.908090797936563	0.5635910034179688
bikes.bmp	768x512	229	511	0.000451265580342486	33.45567790427099	0.7460250854492188
bikes.bmp	768x512	286	511	0.000181631297659890	37.40809314240939	0.9317169189453125
bikes.bmp	768x512	342	511	6.515891840541811e-05	41.86026133302404	1.1141510009765625
bikes.bmp	768x512	398	511	1.900684920414055e-05	47.21089870860375	1.2965850830078125
bikes.bmp	768x512	454	511	3.815312143598176e-06	54.18469925154266	1.4790191650390625
bikes.bmp	768x512	511	511	2.159269592776668e-29	286.65693131011824	1.6647109985351562

APÊNDICE B –Tabela de Taxa de Compressão por Tamanho de Imagem

dimensions	compression_ratio
480x720	0.473774112654321
610x488	0.41275262026337006
618x453	0.49332866589987406
627x482	0.5019621857359354
632x505	0.41147136232610604
634x438	0.4829991501375625
634x505	0.4108942124496361
640x512	0.4116851806640625
768x512	0.47291692097981763

APÊNDICE C –Tabela de Média das Métricas para os Melhores Valores de k

	mse		psnr		compression_ratio	
	mean	max	mean	max	mean	max
k						
101	0.0004670938262668	0.0004670938262668	33.30595872887648	33.30595872887648	0.3902633133111505	0.3902633133111505
104	0.0020169187074708	0.0020169187074708	26.953116058072247	26.953116058072247	0.398236853197311	0.398236853197311
110	0.0010211029781184	0.0010211029781184	29.909304571556063	29.909304571556063	0.4040183446167285	0.4040183446167285
112	0.0006119314031901	0.0006119314031901	32.13297259040269	32.13297259040269	0.4134909970438054	0.4134909970438054
115	0.0014264354533888	0.0020043722395304	28.846852281744702	30.713488355010764	0.4097575437574104 7	0.4100451184358942
117	0.0022265210299253334	0.003244832318011	27.161760762541 56	30.728936034995733	0.4116851806640625	0.4116851806640625