

# Visualização de obras pictóricas depois da análise de clusterização

Tiago Silva da Costa



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB

2024

Tiago Silva da Costa

# Visualização de obras pictóricas depois da análise de clusterização

Relatório técnico apresentado ao curso Ciência da Computação do Centro de  
Informática, da Universidade Federal da Paraíba,  
como requisito para a obtenção do grau de Bacharel

Orientador: Thaís Gaudencio do Rêgo

Setembro de 2024

C838v Costa, Tiago Silva da.

Visualização de obras pictóricas depois da análise de clusterização / Tiago Silva da Costa. - João Pessoa, 2024.

41 f. : il.

Orientação: Tháís Gaudencio do Rêgo.  
TCC (Graduação) - UFPB/CI.

1. Visualização de dados. 2. Clusterização. 3. Arte.  
4. Representação Visual. 5. Ferramentas de Visualização. I. Rêgo, Tháís Gaudencio do. II. Título.

UFPB/CI

CDU 004.92



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computação intitulado ***Visualização de obras pictóricas depois da análise de clusterização*** de autoria de Tiago Silva da Costa, aprovada pela banca examinadora constituída pelos seguintes professores:

---

Prof. Dr. Thaís Gaudencio do Rêgo  
Universidade Federal da Paraíba

---

Prof. Dr. Yuri de Almeida Malheiros Barbosa  
Universidade Federal da Paraíba

---

Prof. Dr. Gustavo Henrique Matos Bezerra Motta  
Universidade Federal da Paraíba

João Pessoa, 12 de setembro de 2024



## AGRADECIMENTOS

Primeiramente, agradeço a Deus por tudo, especialmente por ter me ajudado a chegar até aqui. Sou eternamente grato aos meus pais, senhora Severina e senhor Paulo, que dedicaram suas vidas inteiras para que eu pudesse alcançar este momento. Espero um dia conseguir retribuir ao menos parte do que fizeram por mim. Agradeço profundamente por acreditarem que eu era capaz de concluir minha graduação e por me apoiarem e darem suporte durante todo esse percurso. Sem eles, eu jamais teria conseguido.

Não poderia deixar de agradecer aos meus orientadores, Thais e Yuri, pelos puxões de orelha, pela ajuda inestimável neste trabalho e pela paciência comigo. Eles foram fundamentais para a realização deste trabalho.

Gostaria de expressar minha gratidão aos meus amigos da graduação, que estiveram comigo nesta jornada, ajudando nos estudos, cooperando ao longo do curso e sendo sempre unidos. Eles foram excelentes colegas de equipe.

Aproveito para agradecer ao professor Aleksandro Askiva, da Askiva Kickboxing. Foi com ele que aprendi a enfrentar os desafios, mesmo com medo. Quando comecei na Askiva Kickboxing, apanhava muito durante os treinos, mas com o tempo fui aprendendo a bater. Isso foi fundamental para aliviar o estresse que eu enfrentava na graduação, servindo como uma verdadeira válvula de escape para mim.

## RESUMO

Existe uma crescente necessidade de criação de ferramentas de visualização mais intuitiva para compreender os resultados de algoritmos de aprendizagem de máquina. O presente trabalho trata da implementação de visualização de clusterização de imagens, utilizando os resultados obtidos de um trabalho posterior, onde foi feito um estudo sobre a redução de dimensionalidade de 304 obras do impressionismo com PCA e t-SNE, e métodos de agrupamento, como *k-means* e agrupamento hierárquico. A aplicação conta com a plotagem de imagens no gráfico ao invés de pontos abstratos, assim como a aplicação de zoom quando há interação com os dados. O objetivo da aplicação é propor uma visualização para clusterização com imagens de maneira mais evidente. A visualização foi desenvolvida na linguagem de programação python e suas demais bibliotecas, matplotlib, annotationb-Box e Ipywidgets. Ao final é demonstrado os resultados das visualizações junto com a interação com as imagens.

**Palavras-chave:** Visualização de dados, Clusterização, Arte, Representação Visual, interatividade, Ferramentas de Visualização

## ABSTRACT

There is a growing need to develop more intuitive visualization tools to comprehend the results of machine learning algorithms. This study presents the implementation of image clustering visualization, using results obtained from a previous work that examined the dimensionality reduction of 304 Impressionist artworks with PCA and t-SNE, and clustering methods such as *k-means* and hierarchical clustering. The application features the plotting of images on the graph instead of abstract points, as well as the application of zoom when interacting with the data. The objective of the application is to propose a more evident visualization for image clustering. The visualization was developed using the Python programming language and its libraries, including matplotlib, annotationbBox, and Ipywidgets. Finally, the results of the visualizations are demonstrated along with interaction with the images.

**Key-words:** *Visualization of data, Clustering, Art, Visual Representation, Interactivity, Visualization Tools*

## LISTA DE FIGURAS

1	Exemplo de agrupamento hierárquico, onde à esquerda podemos observar um gráfico de dispersão, que mostra a relação entre a idade e salário de seis registros. À direita temos um dendrograma, uma representação visual do agrupamento hierárquico dos dados. A linha azul horizontal divide em dois agrupamentos baseado na distância escolhida (2,5 unidades de distância euclidiana). As linhas que conectam os registros aos agrupamentos é a árvore hierárquica, o eixo y representa a medida de dissimilaridade entre os registros ou grupos. Quanto maior a distância, menos semelhantes são os registros. O eixo x corresponde aos seis registros (1 a 6) mostrados no gráfico de dispersão à esquerda . . . . .	21
2	Exemplo de imagem . . . . .	25
3	Exemplo de imagem . . . . .	25
4	Visualização de imagens pictográficas realizado com T-SNE com $k=15$ . . .	27
5	Exemplo de interação com escolha dos dados e aplicação de zoom na imagem	28
6	Exemplo de interação com conjunto de dados diferente e zoom e tamanho da imagem distintas . . . . .	29
7	exemplo de interação com pca hierárquico . . . . .	33
8	exemplo de interação com tsne 10k . . . . .	34

## LISTA DE TABELAS

1	Experimentos e quantidades de imagens . . . . .	24
2	Caso de Teste 01 . . . . .	35
3	Caso de Teste 02 . . . . .	36
4	Caso de Teste 03 . . . . .	37
5	Caso de Teste 04 . . . . .	37
6	Caso de Teste 05 . . . . .	38
7	Caso de Teste 06 . . . . .	38
8	Caso de Teste 07 . . . . .	38
9	Caso de Teste 08 . . . . .	39
10	Caso de Teste 09 . . . . .	40

## LISTA DE ABREVIATURAS

PCA – *Principal component analysis* (Análise do componente principal)

TSNE – *T-distributed Stochastic Neighbor Embedding* (Implante Estocástico Disperso Vizinho )

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.0.1	Objetivo geral . . . . .	18
1.0.2	Objetivos específicos . . . . .	18
1.1	Estrutura do relatório . . . . .	18
<b>2</b>	<b>CONCEITOS GERAIS</b>	<b>19</b>
2.1	Python . . . . .	19
2.2	Matplotlib . . . . .	19
2.3	Ipywidgets . . . . .	19
2.4	Métodos de Redução dimensionalidade . . . . .	20
2.4.1	T-SNE . . . . .	20
2.4.2	PCA . . . . .	20
2.5	Métodos de Clusterização . . . . .	20
2.5.1	Algoritmos Hierárquicos . . . . .	20
2.5.2	<i>K-means</i> . . . . .	21
<b>3</b>	<b>METODOLOGIA</b>	<b>22</b>
3.1	Informações gerais . . . . .	22
3.2	Opencv . . . . .	22
3.3	Numpy . . . . .	22
3.4	Matplotlib . . . . .	23
3.5	T-SNE . . . . .	23
3.6	Ipywidgets . . . . .	23
3.7	Dados . . . . .	23
3.8	Redução de dimensionalidade . . . . .	25
3.9	Visualização . . . . .	26
3.10	Gráficos . . . . .	26
3.11	Interação do usuário com as imagens . . . . .	26
3.12	Requisitos funcionais . . . . .	30

3.12.1	Carregar as imagens . . . . .	30
3.12.2	Interatividade . . . . .	30
3.12.3	Salvamento do gráfico da visualização . . . . .	30
3.12.4	Plotagem com imagens . . . . .	30
3.12.5	Todas as imagens presente no gráfico . . . . .	30
3.12.6	Detectar o mouse sobre as imagens . . . . .	30
3.12.7	Escolher bases de dados . . . . .	30
3.12.8	Interação com os tamanhos de imagem e níveis de zoom . . . . .	31
3.12.9	Restaura o tamanho original . . . . .	31
3.13	Requisitos não funcionais . . . . .	31
3.13.1	Permitir que aplicação seja simples para o usuário . . . . .	31
3.13.2	Eficiência . . . . .	31
3.13.3	Segurança . . . . .	31
3.13.4	Deve lidar com grandes volumes de dados . . . . .	31
<b>4</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS</b>	<b>32</b>
4.1	Comportamento da aplicação . . . . .	32
4.2	Resultados . . . . .	32
4.3	Plano de testes . . . . .	35
4.3.1	Casos de teste . . . . .	35
4.4	Limitações . . . . .	41
<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>42</b>
	<b>REFERÊNCIAS</b>	<b>43</b>

# 1 INTRODUÇÃO

A clusterização é uma técnica de aprendizagem de máquina não supervisionado que visa separar os dados em grupos distintos entre si [2]. Geralmente é usada em aplicações como em sistemas de segmentação de clientes, divisão de produtos, análise de redes sociais, visualização de dados e etc [3].

A clusterização também é utilizada para ajudar e automatizar tarefas no cotidiano, como por exemplo, o Google usa a clusterização de imagens para identificar e agrupar fotos de pessoas, com o intuito de automatizar o processo do usuário na busca de determinadas pessoas agrupando suas fotos em álbuns específicos [5]. Na área de marketing, é utilizada para segmentar grupos de pessoas, como por exemplo, grupo de pessoas de 30 a 50 anos, homens de mesma faixa etária ou grupos de moradores da capital paulista [3]. A visualização desses agrupamentos é muito importante, tanto para automatizar tarefas diárias, como para facilitar a tomada de decisão nas grandes empresas.

A visualização da clusterização de imagens possui um papel importante na compreensão de agrupamentos gerados e na identificação de padrões visuais [3]. Uma representação visual facilita a interpretação dos resultados, além de permitir que os usuários possam explorar e interagir com os dados com maior facilidade [3]. Diversas empresas usam a visualização para tomada de decisão, como em sistemas de *streaming* para separar em grupos de pessoas que vêm determinada categoria de filmes ou música. Por exemplo, a Netflix usa clusterização para recomendação dos filmes e séries de Tv [4].

No campo de análise de dados, a visualização de clusterização tem um papel fundamental para tomada de decisão e visualizações de situações de dados de larga escala. A exploração de grandes coleções de imagens é muito importante para uma compreensão mais aprofundada dos padrões presentes nos gráficos e entender melhor o seu comportamento [1]. Exemplos incluem a área médica, satélites e análise de imagens industriais, fotos pessoais, segurança e redes sociais [1].

Este estudo baseia-se nos resultados da clusterização obtidos no trabalho de conclusão de curso do graduado de Ciências da Computação, Pedro Lucas Rangel Felix, que fez a clusterização de obras pictóricas do impressionismo. O objetivo do trabalho é fazer a visualização de forma interativa dos resultados.

Devido a dificuldade de observar essas imagens do impressionismo utilizando pontos abstratos, ou outro método de visualização, este trabalho propõe um método de visualização, substituindo os pontos por imagem. Além disso, é incorporada a funcionalidade de zoom, permitindo que o usuário possa ver a imagem em um maior tamanho[23].

### **1.0.1 Objetivo geral**

Desenvolver uma proposta de ferramenta de visualização interativa, utilizando os resultados do trabalho de Rangel (2023), apresentando as imagens de forma a facilitar a interação e a visualização dos dados plotados no gráfico, tornando a experiência mais flexiva e intuitiva para o usuário.

### **1.0.2 Objetivos específicos**

- Mostrar as imagens no gráfico ao invés de pontos
- Permitir a opção de zoom ao passar o mouse por cima da imagem
- Permitir que o usuário escolha os conjuntos de dados a serem plotados
- Permitir a escolha dos tamanhos de imagem
- Definir os níveis de zoom quando o cursor estiver sobre a imagem
- Atender aos requisitos mínimos para que a aplicação funcione corretamente para o usuário final

## **1.1 Estrutura do relatório**

Este trabalho está estruturado em cinco capítulos. No primeiro capítulo é feita uma introdução ao tema, bem como a exposição do problema, e a apresentação do objetivo do trabalho. No segundo capítulo são tratados os conceitos gerais do trabalho, como as ferramentas utilizadas no estudo. No terceiro capítulo é descrita a metodologia para realizar a visualização dos dados, e seu processo de obtenção e organização, junto com as tecnologias utilizadas para aplicação. No quarto capítulo é feita a apresentação e análise dos resultados, onde são discutidos os resultados da aplicação, suas limitações e casos de testes. No último capítulo são discutidas as conclusões, limitações e são propostas as melhorias e trabalhos futuros.

## 2 CONCEITOS GERAIS

Neste capítulo são descritos os conceitos das ferramentas que foram utilizadas para o desenvolvimento da aplicação.

### 2.1 Python

Python é uma linguagem de programação de alto nível, interpretada, de tipagem dinâmica, orientada a objetos. Foi lançada por Guido Van Rossum em 1991, e atualmente possui um modelo de desenvolvimento de código aberto. A linguagem possui a filosofia de priorizar a legibilidade do programador sobre o esforço computacional [9].

A linguagem também se estabeleceu como uma das principais linguagens de programação para o desenvolvimento de soluções de inteligência artificial (IA) e ciência de dados, devido a sua versatilidade e facilidade, e por possuir diversas bibliotecas e *frameworks* para IA. Da mesma forma é geralmente usada para fazer visualização de dados pelo mesmo motivo de possuir diversos *frameworks* e bibliotecas próprias para essa tarefa.

### 2.2 Matplotlib

Outra ferramenta frequentemente utilizada para ciência de dados é a biblioteca *Matplotlib*. A ferramenta permite fazer plotagem em 2D, 3D, animadas e de forma interativa. A biblioteca foi criada em 2003 e é uma extensão de outra biblioteca do *Python*, *Numpy*, usada para resolver funções matemáticas. O *Matplotlib* permite criar histogramas, gráficos de linha, gráficos de dispersão, gráfico de barra, gráfico de pizza e diagrama de caixa [8].

### 2.3 Ipywidgets

Esta ferramenta permite a interação customizada com os dados da plotagem gráfica, possibilitando a personalização de botões, rótulos e seletores.

Sempre que um dado é modificado por meio da interface de alteração do gráfico, o gráfico se ajusta conforme essa modificação, refletindo imediatamente as mudanças efetuadas.

A ferramenta também permite redimensionar o tamanho da figura utilizando o cursor [17].

## 2.4 Métodos de Redução dimensionalidade

Os métodos de redução de dimensionalidade consistem em reduzir um conjunto de dados com múltiplas variáveis para uma dimensão menor, com o intuito de facilitar o aprendizado não supervisionado e a análise de dados. A seguir, são descritos os tipos de redução de dimensionalidade [20].

### 2.4.1 T-SNE

No contexto de grandes bases de dados, T-SNE (*T-distributed Stochastic Neighbor Embedding*) torna-se uma ferramenta útil para visualizar dados de alta dimensão. Ele converte similaridades entre pontos de dados em probabilidades conjuntas e tenta minimizar a divergência de Kullback-Leibler (medida não simétrica da diferença entre duas distribuições de probabilidade) entre as probabilidades conjuntas da incorporação de baixa dimensão e os dados de alta dimensão. Entretanto, a ferramenta não permite aplicar a opção de zoom nas imagens [11].

### 2.4.2 PCA

Uma outra forma de permitir a visualização de dados de alta dimensão é reduzindo esse valor. Para tanto, PCA (*Principal Component Analysis* - Análise do Componente Principal) é uma técnica de estatística para reduzir a dimensionalidade dos dados de um conjunto de dados, preservando ao máximo a sua variabilidade, sendo uma das técnicas mais populares em aprendizagem de máquina, devido à sua eficiência em simplificar a complexidade dos dados, facilitando a visualização, o processamento e a análise subsequente [16].

## 2.5 Métodos de Clusterização

Os métodos de clusterização são utilizados para identificar padrões nos dados e agrupar os objetos que apresentam características semelhantes. Em Inteligência Artificial existem diversos tipos de métodos de agrupamentos, cada um com suas especificidades e aplicações. Esses métodos permitem a organização de grandes volumes de dados de forma significativa, facilitando a descoberta de informações valiosas e a tomada de decisões informadas. A seguir são descritos alguns métodos de agrupamento comumente usados em Inteligência Artificial e análise de dados [3].

### 2.5.1 Algoritmos Hierárquicos

Em inteligência artificial, o termo hierárquico é geralmente utilizado quando há múltiplas análises em diferentes níveis de uma estrutura hierárquica de dados. Os agrupa-

mentos hierárquicos são organizados em uma estrutura de árvore, onde cada nó representa um agrupamento e podem ser aglomerativos ou divisivos. No método divisivo, por exemplo, cada grupo começa a partir de um único grupo e vai sendo dividido recursivamente em grupos menores na estrutura da árvore. Agrupamentos hierárquicos permitem descobrir relações hierárquicas nos dados e identificar diferentes níveis naturais de agrupamentos. A Figura 1 apresenta um exemplo de agrupamento hierárquico [19].

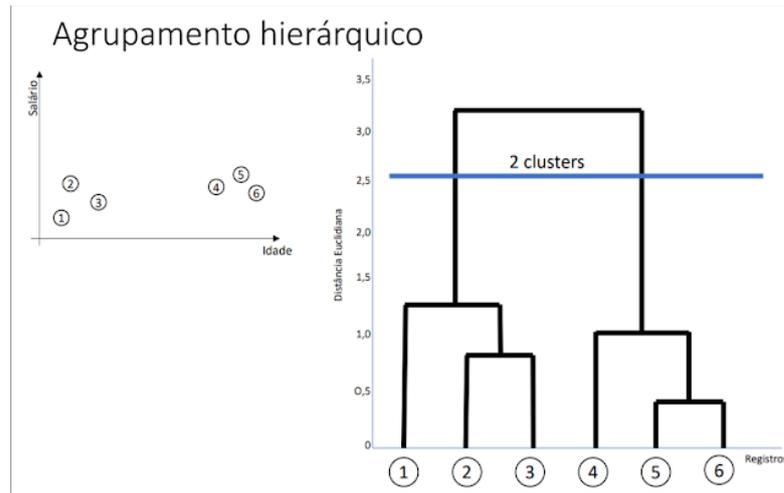


Figura 1: Exemplo de agrupamento hierárquico, onde à esquerda podemos observar um gráfico de dispersão, que mostra a relação entre a idade e salário de seis registros. À direita temos um dendrograma, uma representação visual do agrupamento hierárquico dos dados. A linha azul horizontal divide em dois agrupamentos baseado na distância escolhida (2,5 unidades de distância euclidiana). As linhas que conectam os registros aos agrupamentos é a árvore hierárquica, o eixo y representa a medida de dissimilaridade entre os registros ou grupos. Quanto maior a distância, menos semelhantes são os registros. O eixo x corresponde aos seis registros (1 a 6) mostrados no gráfico de dispersão à esquerda

Fonte: <https://www.computersciencemaster.com.br/como-funciona-o-algoritmo-de-agrupamento-hierarquico/>

### 2.5.2 *K-means*

*K-means* é um algoritmo de agrupamento utilizado para gerar conjuntos de dados não rotulados. A técnica permite criar quantidades de grupos, escolhendo o valor que deve ser atribuído no parâmetro  $k$  da função. Exemplo, suponhamos que utilizamos uma base de dados que contém número de 1 a 10 e queremos agrupar cada valor em 10 grupos, nesse caso o  $k$  receberia o valor 10 e criaria 10 grupos, ao redor de elementos selecionados como centróides e dos elementos mais similares a eles [15].

## 3 METODOLOGIA

Neste capítulo será abordado o processo de desenvolvimento da aplicação, para chegar ao objetivo proposto e a visualização dos dados de forma interativa. Neste tópico serão mencionadas as tecnologias, bibliotecas e os métodos utilizados para a construção da visualização e interatividade.

### 3.1 Informações gerais

O trabalho foi desenvolvido utilizando a linguagem de programação *Python 3*, versão 3.10.12. Esta linguagem foi utilizada para a obtenção das imagens nas pastas de resultados, e em seguida inseri-las em uma lista, contendo todas as imagens da pasta selecionada.

Para a visualização, foi utilizada a biblioteca *Matplotlib* (versão 3.8.3), uma biblioteca do *Python* que permite fazer visualizações de dados. Na interação com os dados do gráfico, foi utilizada a biblioteca *Ipywidgets* (versão 7.6.5), que possibilita a manipulação dos dados por meio da criação de interfaces interativas. Para a opção de zoom, quando o cursor estiver sobre a imagem, foi criada uma função que permite esse tipo de interação. O *script* para a realização do trabalho foi escrito no editor de texto *VSCode*, da Microsoft, utilizando as extensões do *JupyterLab*.

A aplicação foi construída em um computador equipado com 16GB de memória RAM, processador Ryzen 5600G com uma frequência de 3.9GHz, placa de vídeo RTX3060 de 12GB de memória de vídeo e disco rígido de 1TB.

### 3.2 Opencv

*Opencv* é uma biblioteca de visão computacional de código aberto, que permite que desenvolvedores possam fazer aplicações de processamento de imagens e vídeos. Neste trabalho o *Opencv* (versão: 4.9.0) foi destinado para fazer leitura e definição do tamanho da dimensão de cada imagem para um valor fixo igual a 100.

### 3.3 Numpy

A biblioteca *Numpy* (versão: 1.26.4) geralmente é destinada para realizar operações em *arrays* multidimensionais. Neste trabalho foi utilizada para normalizar as imagens, ou seja, transformar as imagens em números para serem utilizadas na função *T-SNE*, para aplicar a redução de dimensionalidade.

### 3.4 Matplotlib

*Matplotlib* (versão 3.8.3) é uma biblioteca comumente utilizada para a visualização de dados. Neste trabalho, foi utilizada para realizar a plotagem do gráfico com as imagens, junto com a ferramenta *OffsetImage* e *AnnotationBox*, que estão embutidas no *Matplotlib*, para substituir os pontos do gráfico por imagens.

### 3.5 T-SNE

Os parâmetros definidos para o *t-SNE*, da biblioteca *scikit-learn* (versão 1.4.1), foram *n\_components* com valor igual a 2, para reduzir a dimensionalidade das imagens para duas dimensões, e *random\_state*, definido arbitrariamente como 42. O uso dessa biblioteca foi necessário para obter as coordenadas das imagens e realizar a plotagem no gráfico.

### 3.6 Ipywidgets

A ferramenta *Ipywidgets* (versão 7.6.5) permite a interação com os dados no gráfico. Neste trabalho, a ferramenta foi utilizada para permitir a seleção dos dados a serem plotados, a escolha do tamanho das imagens no gráfico e o nível de zoom, quando o cursor estiver sobre a imagem [17].

### 3.7 Dados

Os dados utilizados neste estudo foram originalmente obtidos da plataforma Kaggle e tratados por Pedro Rangel em seu trabalho de conclusão de curso, onde foram realizadas a clusterização das imagens.

A coleta dos dados foi realizada através de uma pasta compartilhada no Google Drive. Após o download dos dados para o computador, foi escrito um código em Python para coletar as imagens de cada pasta e adicioná-las a uma lista em Python. Caso a pasta contivesse subpastas com imagens, o script percorria essas subpastas, coletava as imagens, inseria-as na lista e prosseguia para a próxima subpasta. Ao final da coleta, a lista estava preenchida com todas as imagens, denominadas “imagens”.

Para cada imagem, foi definido um tamanho padrão de 100 pixels, escolhido arbitrariamente. Este ajuste foi necessário porque o algoritmo T-SNE não aceita imagens com tamanhos variados, exigindo assim a atribuição de um tamanho fixo para todas elas. As pastas em questão contêm um total de 1493 imagens do movimento impressionista. As imagens foram convertidas em um array numpy para possibilitar sua utilização na função T-SNE e para facilitar a manipulação dos dados.

Na Tabela 1 observa-se o número de imagens para cada um dos experimentos realizados por Pedro Rangel em seu TCC, onde foram avaliados os métodos T-SNE com *k-means* e *k* igual a 10, 15 e 20, e o TSNE com o método hierárquico, além dos experimentos avaliando o PCA hierárquico e com *k* igual a 10, 15 e 20

**Tabela 1: Experimentos e quantidades de imagens**

Experimento	Quantidade de imagens
T-SNE com <i>k-means</i> e <i>k=10</i>	258
T-SNE com <i>k-means</i> e <i>k=15</i>	150
T-SNE com <i>k-means</i> e <i>k=20</i>	284
T-SNE com hierárquico	185
PCA com hierárquico	11
PCA com <i>k-means</i> e <i>k=10</i>	300
PCA com <i>k-means</i> e <i>k=15</i>	133
PCA com <i>k-means</i> e <i>k=20</i>	172

Nas Figuras 2 e 3, pode-se observar exemplos de imagens avaliadas na clusterização e utilizadas neste trabalho.



**Figura 2: Exemplo de imagem**



**Figura 3: Exemplo de imagem**

### **3.8 Redução de dimensionalidade**

Após a coleta e o processamento das imagens, foi utilizado o algoritmo T-SNE para realizar a redução de dimensionalidade, configurando o parâmetro `n_components` com valor igual a 2, para representar duas dimensões, e o parâmetro `random_state` com valor 42, para garantir a reprodutibilidade dos resultados. Esse procedimento foi necessário para permitir a plotagem dos dados em um gráfico.

Os resultados do T-SNE foram armazenados em uma lista denominada `T-SNE_result`. Utilizando essa lista, que contém todos os resultados da redução de dimensionalidade e as

coordenadas das imagens, foi possível realizar a visualização dos dados. Para a redução de dimensionalidade, foi utilizado apenas o algoritmo *t-SNE*, enquanto o *PCA* foi empregado apenas para os testes da aplicação.

### 3.9 Visualização

A visualização foi desenvolvida utilizando o Matplotlib (versão: 3.8.3) e suas funções *Offsetimage* e *AnnotationBox*. O tamanho da figura foi definido de forma arbitrária com valor de 10 e 10. Para a plotagem em 2D, utilizando o modelo de aprendizagem de máquina T-SNE, foi primeiramente realizada a redução de dimensionalidade, para obter os valores de  $x$  e  $y$ , para a realização da plotagem, depois disso, um *Offsetimage* é criado para cada imagem, com um valor de nível de zoom escolhido arbitrariamente de 0,12, sendo assim a imagem vai parecer 12% menor do que a imagem original. O *AnnotationBox* determina os pontos específicos na plotagem, com os resultados do T-SNE.

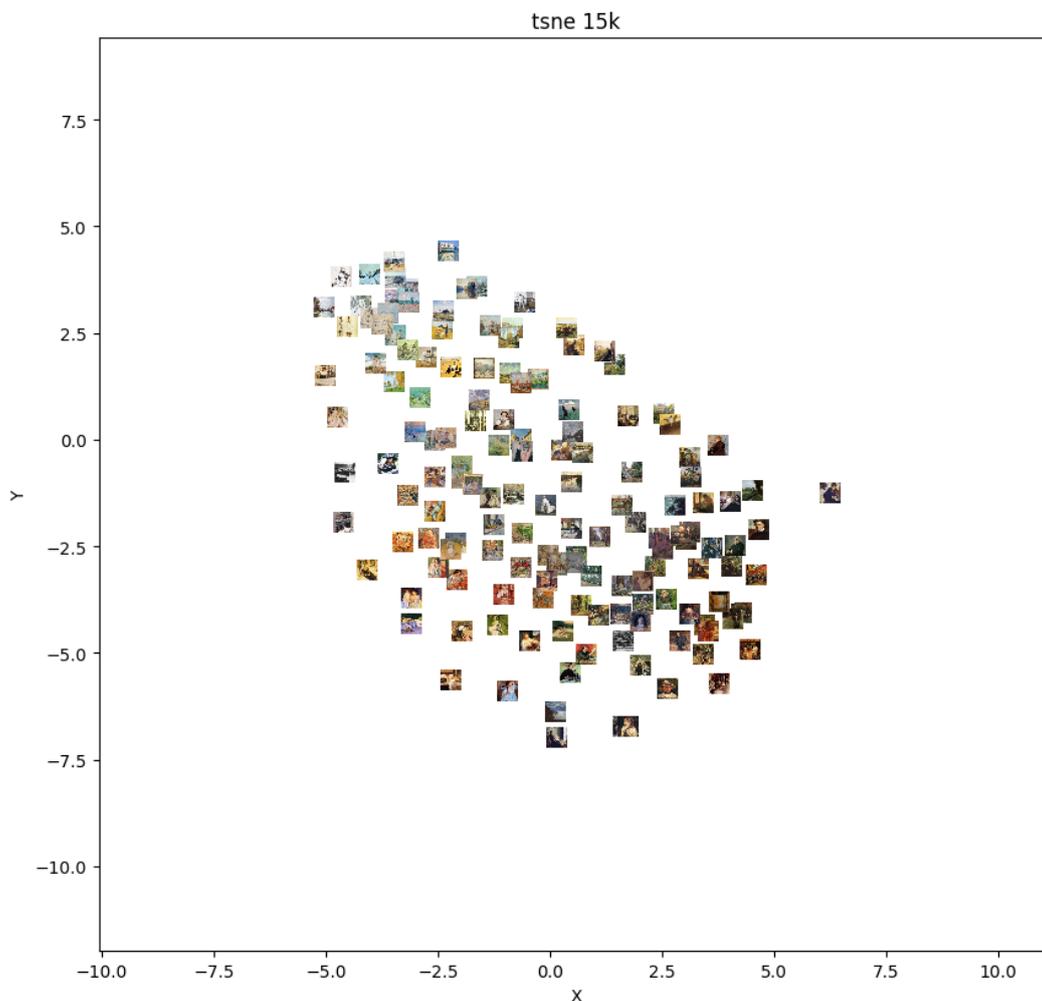
### 3.10 Gráficos

O tamanho do gráfico foi escolhido arbitrariamente, na função da biblioteca *Matplotlib*, com tamanho de 10 e 10 e o seus limites foram determinados, a fim de garantir que todas as imagens estejam visíveis na área dos gráficos. Os limites foram definidos da seguinte forma: para os valores  $x$  e  $y$  foram extraídos dos valores máximo e mínimo das coordenadas, após isso foi realizada a soma do valor máximo como 5, também definido arbitrariamente, e para o mínimo foi subtraído por 5. Esse processo foi aplicado para ambas coordenadas. Isso é útil para evitar que as imagens fiquem muito próximas das bordas do gráfico, como pode ser observado na Figura 4.

### 3.11 Interação do usuário com as imagens

Para permitir a interação do usuário com as imagens, foi criada uma função em Python que aplica zoom na imagem (Figura 5), quando o cursor do mouse passa sobre um dado. O valor do zoom é definido por padrão no parâmetro *FloatSlider* da biblioteca *Ipywidgets*. O tamanho das imagens também é definido como um parâmetro no *Ipywidgets*, sendo os valores de zoom e tamanho da imagem limitados entre 0,05 e 0,2, ambos escolhidos arbitrariamente. O comportamento principal da função é verificar se o mouse está próximo o suficiente da imagem para aplicar o zoom e ajustar os níveis de zoom e tamanho da imagem de acordo.

Um valor de proximidade é determinado, no qual o zoom é aplicado quando a distância entre o cursor e a imagem está abaixo desse valor ou exatamente sobre o dado. Caso esta condição seja atendida, um efeito de zoom é aplicado, permitindo a ampliação



**Figura 4: Visualização de imagens pictográficas realizado com T-SNE com  $k=15$**

da imagem, destacando-a para melhor visualização. A interação ocorre em tempo real, permitindo que o usuário selecione outra imagem imediatamente após interagir com a anterior.

O usuário pode selecionar as pastas que deseja visualizar na plotagem através de um seletor dropdown, que permite escolher a fonte de dados e visualizar os dados disponíveis para a plotagem. Por padrão, o conjunto de dados mostrado na visualização, após a execução do programa, é o T-SNE com  $k=10$ .

Nas Figuras 5 e 6, podemos observar o resultado da visualização e interação. A parte superior da figura mostra a seção de controle da plotagem, onde se encontram o seletor para escolha dos dados e dois *sliders* para ajustar o tamanho das imagens e o nível de zoom.

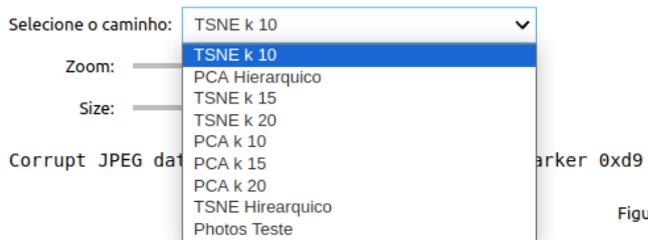


Figure 2

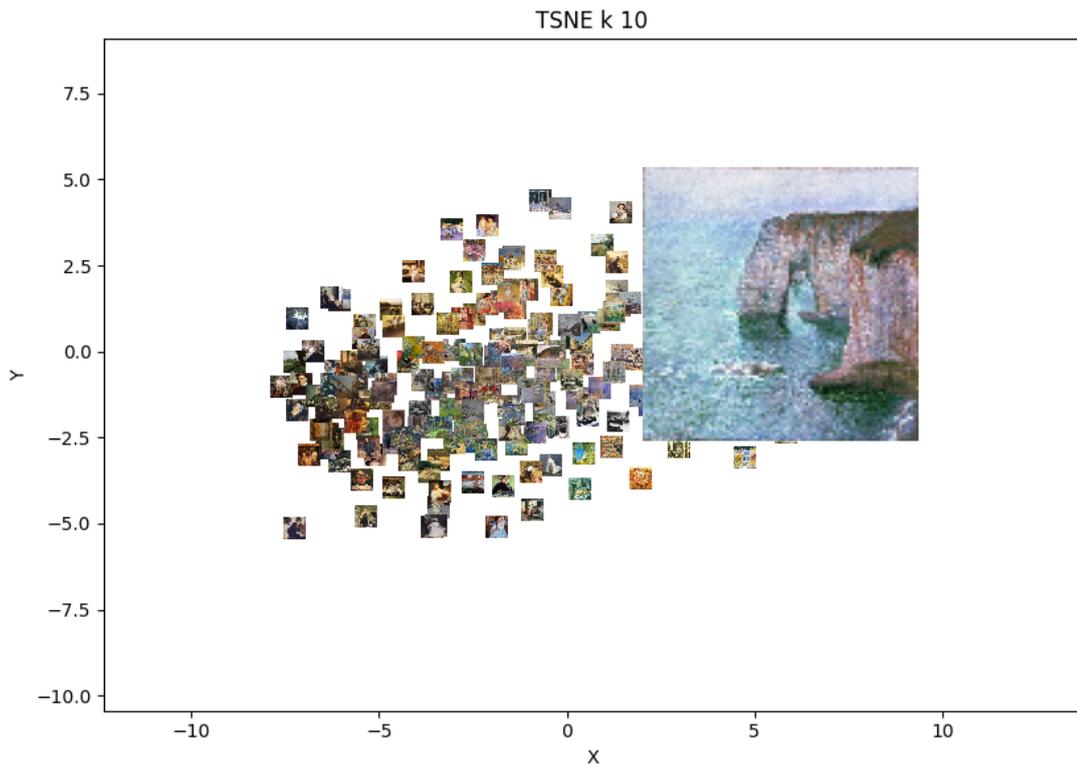


Figura 5: Exemplo de interação com escolha dos dados e aplicação de zoom na imagem

Selecione o caminho: PCA k 10

Zoom: 1.72

Size: 0.09

Corrupt JPEG data: 10 extraneous bytes before marker 0xd9



Figure 2

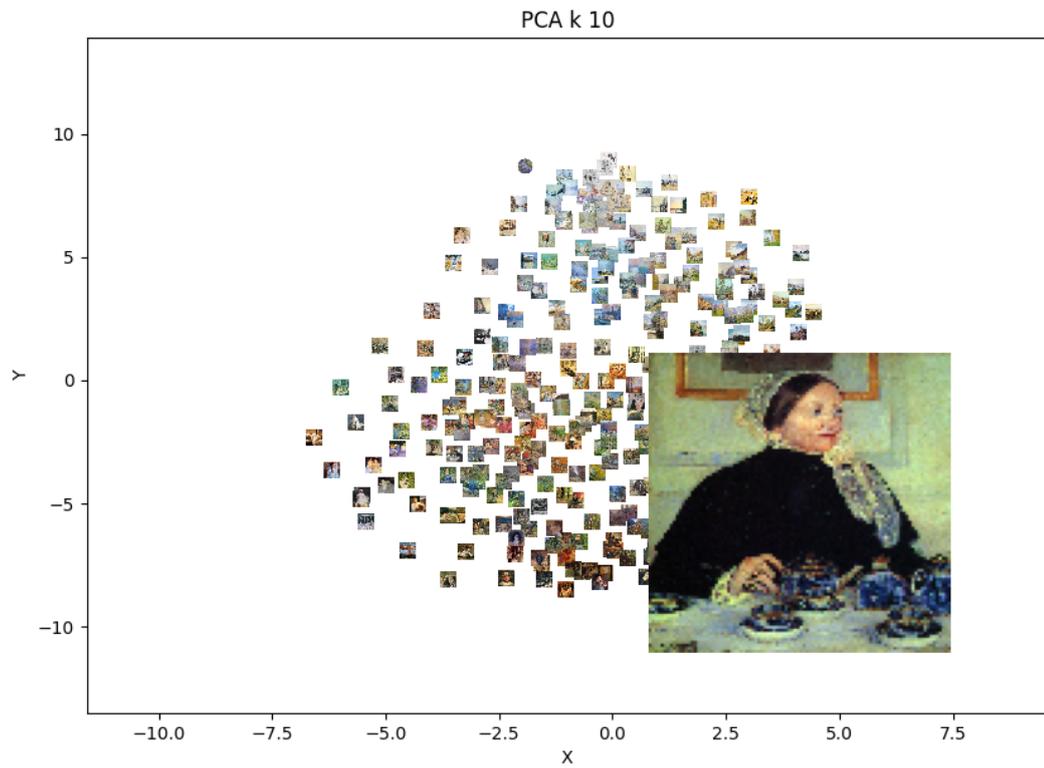


Figura 6: Exemplo de interação com conjunto de dados diferente e zoom e tamanho da imagem distintas

O efeito de grande escala de imagem acontece quando o mouse está sobre a imagem. Infelizmente não é possível reproduzir em forma de vídeos a interação consecutiva do usuário com as imagens neste documento.

### **3.12 Requisitos funcionais**

Os requisitos funcionais tratam das necessidades que devem ser atendidas e resolvidas pelo software por meio de serviços específicos. O principal objetivo dos requisitos funcionais é atender às necessidades do cliente, garantindo que o software funcione corretamente para o usuário final[21].

#### **3.12.1 Carregar as imagens**

A aplicação deve ser capaz de carregar as imagens.

#### **3.12.2 Interatividade**

A aplicação deve permitir que o usuário interaja com a visualização, por exemplo, ao passar o mouse sobre uma imagem para ver a imagem em um escala maior.

#### **3.12.3 Salvamento do gráfico da visualização**

A aplicação deve ser capaz de permitir salvar o gráfico do resultado.

#### **3.12.4 Plotagem com imagens**

A aplicação deve ser capaz de plotar as imagens ao invés de pontos abstratos.

#### **3.12.5 Todas as imagens presente no gráfico**

A aplicação deve ser capaz de mostrar todas as fotos no gráfico.

#### **3.12.6 Detectar o mouse sobre as imagens**

A aplicação deve ser capaz de detectar o evento de mouse (*hover*).

#### **3.12.7 Escolher bases de dados**

O usuário deve ser capaz de escolher os conjuntos de dados disponíveis que serão plotados no gráfico.

### **3.12.8 Interação com os tamanhos de imagem e níveis de zoom**

A aplicação deve permitir que o usuário possa escolher os níveis de aplicação de zoom quando o cursor estiver sobre o dado e o tamanho das imagens no gráfico.

### **3.12.9 Restaura o tamanho original**

A aplicação deve ser capaz de restaurar o tamanho original, quando o mouse não estiver sobre o dado.

## **3.13 Requisitos não funcionais**

Os requisitos não funcionais são tudo aquilo que o software tem que ter para funcionar de maneira eficaz e eficiente e não estão diretamente relacionados às funcionalidades que o sistema deve oferecer, mas sim às propriedades e comportamentos do sistema como um todo[22].

### **3.13.1 Permitir que aplicação seja simples para o usuário**

A aplicação deve ser capaz de gerar os gráficos com as imagens de forma intuitiva.

### **3.13.2 Eficiência**

A aplicação deve ser capaz de realizar a interação de forma eficiente ao colocar o mouse sobre as imagens

### **3.13.3 Segurança**

A aplicação não deve permitir vazamento de dados durante a aplicação.

### **3.13.4 Deve lidar com grandes volumes de dados**

A aplicação deve ser capaz de lidar com um grande número de imagens nas pastas.

## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Neste capítulo descreveremos os resultados, limitações e os casos de testes da ferramenta proposta.

### 4.1 Comportamento da aplicação

Todos os caminhos das pastas contendo os resultados foram atribuídos a variáveis. Para ter acesso aos dados, essas variáveis são passadas para um dicionário, que é atribuída ao parâmetro da biblioteca ipywidgets, para possibilitar a seleção das pastas durante a interação. Cada vez que uma pasta é trocada ou um dos valores dos floatSliders, uma nova figura é gerada com os dados da nova interação.

### 4.2 Resultados

A aplicação demonstrou um desempenho consistente em relação aos objetivos estabelecidos. Entretanto a perplexidade, que serve para controlar a quantidade de vizinhos próximos que cada ponto considera durante a construção da estrutura de vizinhança no espaço de alta dimensão na função T-SNE, teve que ser definida para o valor mínimo (5 - o número de amostras), nesse caso, o 5 foi escolhido arbitrariamente, mas também por ser um número pequeno.

A adição da perplexidade se fez necessária para que a aplicação funcionasse corretamente com os resultados dos conteúdos resultantes do experimento com PCA hierárquico, pois a perplexidade deve ser menor que o número de amostras.

Podemos observar, intuitivamente, nos resultados, que as imagens perdem qualidade quando recebem o efeito de zoom, isto acontece devido a aplicação de zoom sobre uma imagem muito pequena, pois o tamanho de cada imagem foi reduzido para um valor muito pequeno (12% do tamanho original), para que fosse possível plotar a imagem no gráfico. Na Figura 7 podemos observar o resultado com as imagens do experimento utilizando PCA e hierárquico e na Figura 8, o resultado com as imagens do experimento utilizando T-SNE com  $k=10$ .

Selezione o caminho: PCA Hierarquico

Zoom: 1.50

Size: 0.12



Figure 1

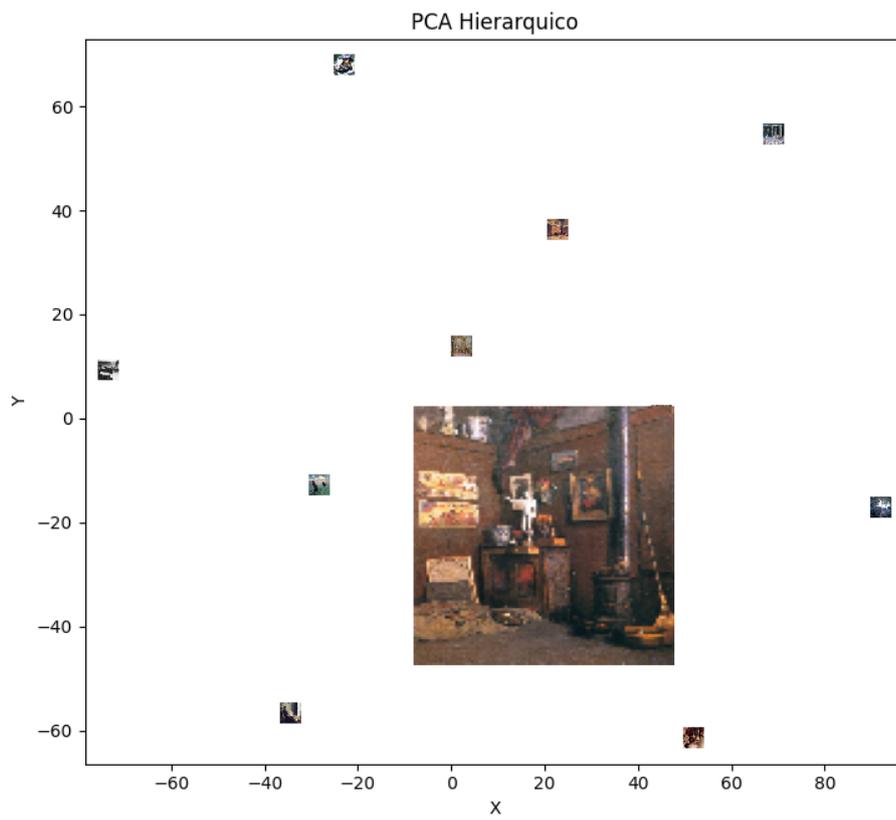


Figura 7: exemplo de interação com pca hierárquico

Selecione o caminho: TSNE k 10

Zoom: 1.50

Size: 0.12

Corrupt JPEG data: 10 extraneous bytes before marker 0xd9



Figure 1

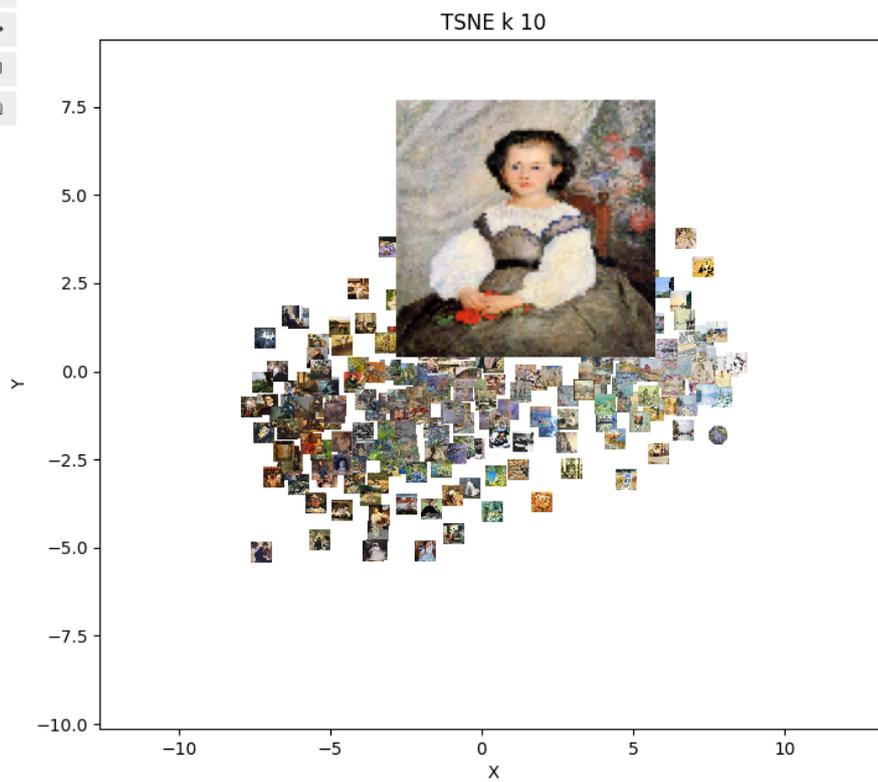


Figura 8: exemplo de interação com tsne 10k

### 4.3 Plano de testes

Nesta seção será descrito o planejamento para a realização dos testes. Os testes foram feitos com as imagens disponibilizadas via web, do site *pngwing*, no formato PNG. Cada teste foi realizado utilizando todas as imagens para garantir que a aplicação funcione corretamente. O objetivo destes testes é verificar se a aplicação funciona corretamente em cenários distintos. Os casos de testes foram elaborados e executados pelo autor deste trabalho.

#### 4.3.1 Casos de teste

A seguir serão descritos os casos de testes considerados de maior importância. Na linha de Descrição é descrito o objetivo da avaliação do teste; em Entradas, são descritas as entradas do programa; na linha de Resultados esperados, são descritos os resultados esperados após a execução do *script*. A linha de Resultados obtidos apresenta três possíveis situações, "Bem sucedido", caso a aplicação funcione corretamente sem erros, "Parcialmente Bem sucedido", caso a aplicação consiga funcionar, mesmo com alguns problemas não muito graves e "Falha", caso a aplicação falhe completamente para executar o objetivo proposto. As Tabelas 6-10 tratam dos casos de testes.

**Tabela 2: Caso de Teste 01**

<b>Caso de Teste 01</b>	<b>Carregamento de imagens</b>
<b>Descrição</b>	Avaliar o processo de carregamento das imagens
<b>Entradas</b>	Imagens na aplicação de cada pasta, realizada uma por vez
<b>Resultados esperados</b>	Carregar todas as imagens de cada pasta dentro de um array de string e mostrar a quantidade de imagens no terminal
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

**Tabela 3: Caso de Teste 02**

<b>Caso de Teste 02</b>	<b>Redução de dimensionalidade</b>
<b>Descrição</b>	Avaliar o processo de redução de dimensionalidade, utilizando o PCA e T-SNE
<b>Entradas</b>	Imagens nas funções PCA e T-SNE
<b>Resultados esperados</b>	Executar sem erros as funções de redução de dimensionalidade
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	No caso do PCA com hierárquico, por conter poucos dados na sua pasta, a perplexidade do T-SNE teve que ser definida

**Tabela 4: Caso de Teste 03**

<b>Caso de Teste 03</b>	<b>Plotagem das imagens</b>
<b>Descrição</b>	Avaliar a plotagem das imagens ao invés de pontos abstratos
<b>Entradas</b>	Os resultados da T-SNE e PCA
<b>Resultados esperados</b>	Mostrar as imagens no gráfico ao invés de pontos abstratos
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

**Tabela 5: Caso de Teste 04**

<b>Caso de Teste 04</b>	<b>Interação com as imagens na plotagem</b>
<b>Descrição</b>	Avaliar a interação com as imagens no gráfico
<b>Entradas</b>	Os resultados da T-SNE e PCA
<b>Resultados esperados</b>	Detectar o evento quando o mouse estiver sobre uma imagem.
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

**Tabela 6: Caso de Teste 05**

<b>Caso de Teste 05</b>	<b>Adição de zoom</b>
<b>Descrição</b>	Avaliar se a aplicação de zoom é realizado quando o mouse está sobre a imagem
<b>Entradas</b>	A entrada seria o curso sobre a imagem
<b>Resultados esperados</b>	Aplicar zoom quando o mouse estiver sobre a imagem
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

**Tabela 7: Caso de Teste 06**

<b>Caso de Teste 06</b>	<b>Funcionamento com outros tipos de imagem</b>
<b>Descrição</b>	Verificar se a aplicação funciona com outro tipo, como PNG, JPEG/JPG
<b>Entradas</b>	Imagens em formato PNG/JPEG/JPG
<b>Resultados esperados</b>	Funcionamento normal com tipos diferentes de imagem
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

**Tabela 8: Caso de Teste 07**

<b>Caso de Teste 07</b>	<b>Restauração do tamanho original da imagem</b>
<b>Descrição</b>	Verificar se a imagem retorna para o tamanho original quando o mouse não estiver sobre ela
<b>Entradas</b>	Imagens contidas nas pastas.
<b>Resultados esperados</b>	Restaurar a imagem para o tamanho original quando o mouse não estiver sobre o dado
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

**Tabela 9: Caso de Teste 08**

<b>Caso de Teste 08</b>	<b>Escolha das pastas no botão <i>dropdown</i></b>
<b>Descrição</b>	Verificar se a plotagem funciona ao trocar de pastas na opção <i>dropdown</i>
<b>Entradas</b>	Pastas .
<b>Resultados esperados</b>	Realização da plotagem de determinada pasta ao escolhê-la
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

**Tabela 10: Caso de Teste 09**

<b>Caso de Teste 09</b>	<b>Escolha dos tamanhos das imagens e níveis de zoom</b>
<b>Descrição</b>	Verificar se a plotagem funciona com a escolha dos tamanhos de imagem e níveis de zoom
<b>Entradas</b>	Níveis de zoom e tamanho das imagens
<b>Resultados esperados</b>	Gráfico com comportamento de acordo com os dados escolhidos
<b>Resultados obtidos</b>	Bem sucedido
<b>Análise dos resultados</b>	Funcionou corretamente
<b>Observações</b>	Nenhuma

#### 4.4 Limitações

Até o presente momento deste trabalho, a aplicação está limitada apenas a plotar as imagens e permitir que o usuário possa observar a imagem ao passar o mouse sob o dado, definir o tamanho das imagens, níveis de zoom e determinar o tamanho da figura. A aplicação não está contida em um website ou aplicativo, apenas um código escrito em python em um editor de texto. O script somente utilizou os dados disponibilizados e as imagens utilizadas para teste, o seu funcionamento com outras quantidades de imagens é desconhecido. No entanto, o código pode ser modificado para tratar deste propósito.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

O trabalho apresentou uma proposta de visualização com imagens de forma interativa. O estudo revelou uma alternativa de visualização de clusterização com imagens, que permite que o visualizador possa observar os dados de forma mais detalhada, sendo possível selecionar a imagem desejada em meio aos pontos no gráfico.

O foco principal do estudo foi tornar a visualização de imagens mais intuitiva. A aplicação ainda pode receber melhorias, como selecionar a imagem e abri-la, ou criar uma flecha que mostra a imagem em questão, quando o curso está sob o dado.

Apesar de ser uma visualização simples, ainda é possível criar melhorias ou servir de base para aplicações mais sofisticadas no futuro. Um dos problemas encontradas, foi a falta de tecnologias para este tipo de visualização.

Algumas propostas de aplicação e melhorias para trabalhos futuros são:

- Abrir a imagem quando o cursor estiver sobre o dado;
- Criar uma forma de visualizar o grupos por meio de contornos ou bolhas;
- Criar um website ou aplicativo que possa usar essa aplicação;
- Adicionar um filtro que pesquise a imagem em questão;
- Criação de biblioteca para desenvolver visualizações de clusterização com imagens.

## REFERÊNCIAS

- [1] Pogorelov, K., Riegler, M., Halvorsen, P., & Griwodz, C. (2017, June 6). Cluster-Tag: Interactive visualization, clustering and tagging tool for big image collections. **ICMR 2017 - Proceedings of the 2017 ACM International Conference on Multimedia Retrieval**, 112–116. <https://doi.org/10.1145/3078971.3079018>
- [2] Singh, S., & Kumar, K. (2021). A study of lean construction and visual management tools through cluster analysis. **Ain Shams Engineering Journal**, 12(1), 1153–1162. <https://doi.org/10.1016/J.ASEJ.2020.04.019>
- [3] Tera Blog. Clusterização de dados: entenda o conceito e formas de uso. Disponível em: <<http://blog.somostera.com/data-science/clusterização-de-dados>>. Acesso em: 20 de abril de 2024.
- [4] Clustering Algorithm. Disponível em: <<https://medium.com/@chitrark/clustering-algorithm-89f79c456336>>. Acesso em: 20 de abril de 2024
- [5] Google fotos vai agrupar imagens parecidas no android e IOS . Disponível em: <<https://www.terra.com.br/byte/google-fotos-vai-agrupar-imagens-parecidas-no-android-e-no-ios,2811a090e937d1a8b4c484191230808cy2g0hclv.html>>. Acesso em: 20 de abril de 2024
- [6] Clusterização: o que é, importância e aplicações . Disponível em: <<https://www.fiveacts.com.br/clusterizacao>>. Acesso em: 20 de abril de 2024
- [7] Cluster visualization. Disponível em: <<https://help.relativity.com/RelativityOne/Content/Rela>>. Acesso em: 20 de abril de 2024
- [8] Matplotlib Documentation. Disponível em: <<https://matplotlib.org/stable/index.html>>. Acesso em: 20 de abril de 2024
- [9] Python Documentation. Disponível em: <<https://docs.python.org/3/>>. Acesso em: 20 de abril de 2024
- [10] Python show image upon hovering over a point. Disponível em: <<https://stackoverflow.com/questions/42867400/python-show-image-upon-hovering-over-a-point>>. Acesso em: 20 de abril de 2024
- [11] Tsne Documentation. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>>. Acesso em: 20 de abril de 2024
- [12] Böhm, J. N., Berens, P., & Kobak, D. (n.d.). UNSUPERVISED VISUALIZATION OF IMAGE DATASETS USING CONTRASTIVE LEARNING.

- [13] Opencv Documentation. Disponível em: <<https://opencv.org/>>. Acesso em: 20 de abril de 2024
- [14] Imagens png. Disponível em: <<https://www.pngwing.com/>>. Acesso em: 27 de abril de 2024
- [15] Kmeans Documentation. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>>. Acesso em: 27 de abril de 2024
- [16] Pca Documentation. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>>. Acesso em: 4 de junho de 2024
- [17] Ipywidgets Documentation. Disponível em: <<https://ipywidgets.readthedocs.io/en/stable/>>. Acesso em: 4 de junho de 2024
- [18] Jupyter Documentation. Disponível em: <<https://jupyter.org/>>. Acesso em: 4 de junho de 2024
- [19] Clusterização hierárquica. Disponível em: <<https://www.escoladnc.com.br/blog/clusterizacao-hierarquica-uma-introducao-detalhada/>>. Acesso em: 20 de junho de 2024
- [20] Redução de Dimensionalidade. Disponível em: <<https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-dimensionality-reduction/>>. Acesso em: 27 de junho de 2024
- [21] Requisitos funcionais e não funcionais. Disponível em: <<https://www.mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>>. Acesso em: 08 de julho de 2024
- [22] Requisitos não funcionais. Disponível em: <<https://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525>>. Acesso em: 08 de julho de 2024
- [23] video da aplicação. Disponível em: <<https://youtu.be/0c1rQvAR0qc?si=jZQtwPPBK19etOIr>>. Acesso em: 25 de julho de 2024
- [24] código da aplicação. Disponível em: <<https://github.com/samfisherholian/imagesView>>. Acesso em: 25 de julho de 2024