**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

# ANÁLISE DO POTENCIAL DE MODELOS CONTRASTIVOS EM TAREFAS DE IMAGEM-PARA-GRAFO

**EDUARDO MINORU TAKEDA**

**ORIENTADOR: PROF. DR. CESAR HENRIQUE COMIN**

São Carlos – SP

2025

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

# ANÁLISE DO POTENCIAL DE MODELOS CONTRASTIVOS EM TAREFAS DE IMAGEM-PARA-GRAFO

**EDUARDO MINORU TAKEDA**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Cesar Henrique Comin

São Carlos – SP

2025

*Dedico este trabalho a mim mesmo, como prova de minha resiliência, dedicação, esforço e foco quando necessário.*

# AGRADECIMENTOS

Agradeço primeiramente aos meus amigos que conheci durante a graduação, por todos os momentos e experiências compartilhadas nesta importante fase da vida e a companhia presente mesmo a quilômetros de distância.

Aos meus professores e professoras da Universidade Federal de São Carlos que construíram o conhecimento que tenho hoje. Principalmente ao meu orientador Dr. Cesar Henrique Comin, que me deu a oportunidade de realizar este projeto e me guiou de maneira exemplar. Uma menção honrosa ao Dr. Jander Moreira por ter sido o professor que me deu a primeira aula da minha graduação e agora está acompanhando a disciplina de Trabalho de Conclusão de Curso, sendo também o responsável pela minha última disciplina.

À minha família que me apoiou e me aconselhou em alguns momentos conturbados da minha vida universitária.

E por último um agradecimento especial à Maria Clara Katherine Rapelli por ter sido a pessoa que, mesmo que indiretamente, me fez escolher a UFSCar e sem ela eu provavelmente não estaria aqui.

A todos, minha gratidão.

*Cada falha é um passo ao sucesso.*
*(William Whewell)*

# RESUMO

Grandes Modelos de Visão e Linguagem (VLMs) têm demonstrado capacidade para diversas tarefas envolvendo imagens, como detecção, reconhecimento e contagem de objetos. Contudo, ainda apresentam dificuldades significativas ao lidar com imagens de grafos e diagramas. Diante desse cenário, torna-se relevante investigar tanto formas de adaptar esses modelos quanto alternativas metodológicas capazes de explorar suas reais capacidades. Entre as abordagens possíveis, o Aprendizado Contrastivo destaca-se por obter resultados superiores a outros modelos envolvendo múltiplos atributos textuais e dependências posicionais. Nesse contexto, a investigação das capacidades do modelo CLIP mostra-se pertinente, especialmente considerando o avanço de outros modelos multimodais em diferentes domínios. Este trabalho analisa o potencial do CLIP, aliado ao Aprendizado Contrastivo, para tarefas envolvendo a extração da similaridade entre imagens de grafos e a listagem de suas arestas, bem como a diferenciação de estruturas de grafos. Os resultados foram positivos no reconhecimento das listas de arestas, enquanto o agrupamento de diferentes estruturas e graus de regularidade apresentou potencial promissor, embora ainda dependente de maior exploração e ajustes finos. A análise das métricas e a ampliação das aplicações da metodologia adotada podem contribuir de forma significativa para o avanço do estado da arte na área.

**Palavras-chave**: modelos de visão e linguagem, grafos, aprendizado contrastivo, similaridade.

# ABSTRACT

Large Vision–Language Models (VLMs) have demonstrated strong capabilities across various image-related tasks, such as detection, recognition, and object counting. However, they still face significant challenges when dealing with graph and diagram images. In this context, it becomes essential to investigate both ways of adapting these models and alternative methodological approaches capable of uncovering their actual potential. Among the possible approaches, Contrastive Learning stands out by achieving superior results compared to other models that rely heavily on multiple textual attributes and positional dependencies. In this scenario, examining the capabilities of the CLIP model proves particularly relevant, especially considering the progress of other multimodal models in different domains. This work analyzes the potential of CLIP, combined with Contrastive Learning, for tasks involving graph images and the listing of their edges, as well as for differentiating structures and regularity levels for a fixed number of edges. The results were positive for edge-list recognition, while the clustering of different structures and regularity levels showed promising potential, though still dependent on further exploration and fine-tuning. The analysis of the metrics and the expansion of the applied methodology may significantly contribute to advancing the state of the art in this field.

**Keywords**: vision-language models, graphs, contrastive learning, similarity.

# LISTA DE FIGURAS

# LISTA DE SIGLAS

| | |
|---|---|
| VLM | Visual-Language Model |
| LVLM | Large Visual-Language Model |
| LLM | Large Language Model |
| LMM | Large Multimodal Model |
| CLIP | Contrastive Language Image Pre-training |
| CNN | Convolutional Neural Network |
| MLP | Multilayer Perceptron |
| PLN | Processamento de Linguagem Natural |
| NLP | Natural Language Processing |
| IA | Inteligência Artificial |
| AI | Artificial Inteligence |
| GenAI | Generative Artifical Inteligence |
| VQA | Visual Question Answering |
| DPR | Description-Program Reasoning |
| UMAP | Uniform Manifold Approximation and Projection |
| LDA | Linear Discriminant Analysis |
| PCA | Principal Component Analysis |
| VP | Verdadeiro Positivo |
| VN | Verdadeiro Negativo |
| FP | Falso Positivo |
| FN | Falso Negativo |

# SUMÁRIO

# Capítulo 1
# INTRODUÇÃO

## 1.1 Contexto e Motivação

As áreas de Visão Computacional e Saúde estão fortemente relacionadas no quesito da tecnologia, principalmente em relação ao diagnóstico de imagens médicas. As transformações de imagens são ferramentas poderosas para extrair mais informações de forma que ajudem o profissional a enxergar alguma anomalia, que podem estar obscurecidas pela qualidade ou a forma com que a imagem foi gerada.

A segmentação de imagens é uma dessas ferramentas que tem como função filtrar as informações mais úteis ou aquelas que precisam estar destacadas, como vasos sanguíneos, ossos, células, etc. Por exemplo, os autores (DAVID et al., 2022) e (COMIN; GALVÃO, 2025) segmentam imagens de vasos sanguíneos com alta confiabilidade para detectar alterações vasculares.

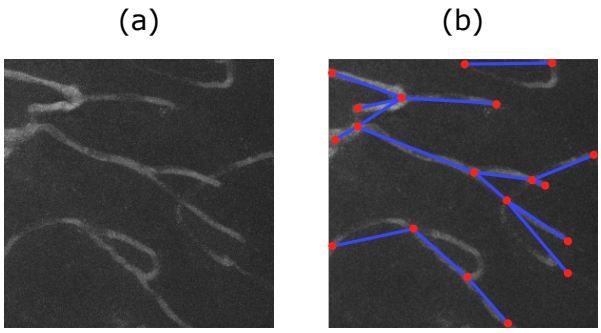
Com isso, as imagens podem receber transformações como a esqueletização para simplificar a representação dos vasos e, então, transformá-los em grafos mapeados, como (FREITAS-ANDRADE et al., 2022) o fez. A Figura 1 exemplifica um mapeamento em grafo para representar os vasos sanguíneos da imagem original, enquanto a Figura 2 demonstra os passos utilizados como um exemplo para alcançar tal resultado.

Uma tarefa ainda em exploração no estado da arte é o uso de Modelos de Aprendizado de Máquina para reconhecimento de grafos para diversas tarefas, como detecção de nós e arestas, cálculo de caminhos, pesos, direções, entre outras.

Os tipos de modelos mais testados foram as *Large Multimodal Models* (LMM) e *Visual-Language Models* (VLM), ou *Large Visual-Language Models* (LVLM), que tem como principal característica a possibilidade de receber uma imagem como entrada, para então ser analisada pelo modelo e em seguida extrair alguma informação em linguagem natural, como perguntas e respostas sobre o que o modelo detecta na imagem.

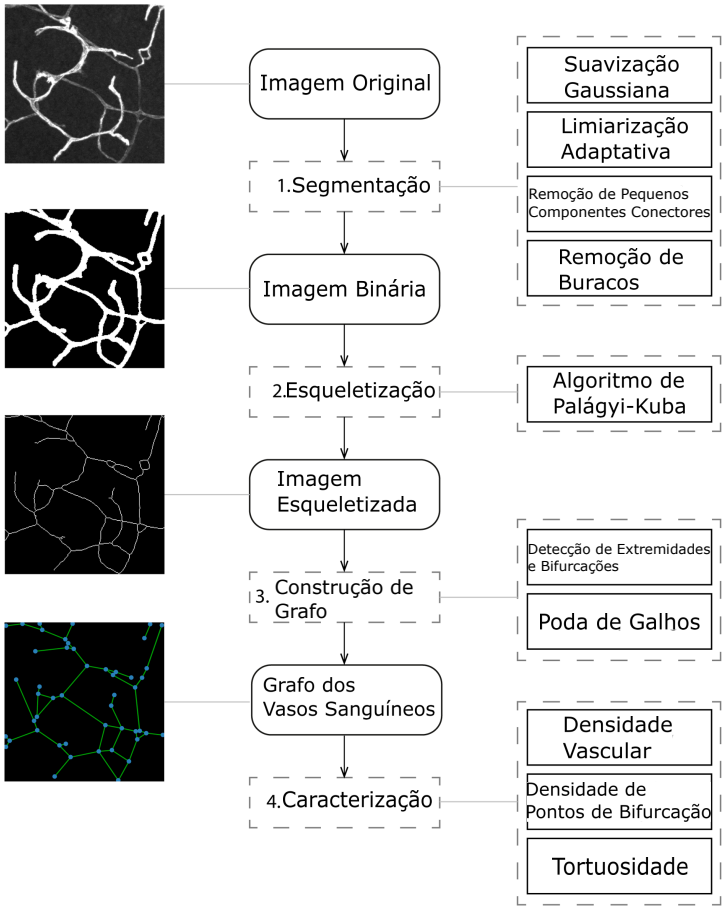
Contudo, pesquisas mostram a dificuldade que os grandes modelos de referência encontram ao se deparar com esse tipo de estrutura de imagem. (ZHU et al., 2025), por exemplo, chega a conclusão das limitações das LVLMs utilizadas. Porém, os autores mostram o potencial desses

**Figura 1 – Mapeamento de Vasos Sanguíneos em Grafos**




**(a) Corte de uma imagem de vasos sanguíneos em escala de cinza; (b) O mesmo corte com grafos mapeados sobre a imagem original. Fonte: Próprio autor**

**Figura 2 – Esquematização dos Passos Usados para Transformação de Vasos Sanguíneos em Grafos**




**Fonte: Adaptado de (FREITAS-ANDRADE et al., 2022)**

modelos se auxiliados com informações extra.

Portanto, o uso de LMMs e LVLMs para tarefas envolvendo grafos apresentam desafios significativos, considerando os resultados encontrados. Porém, (TERASHITA et al., 2025) utiliza o modelo CLIP e o treina especificamente para detectar setas em diagramas usando Aprendizado Contrastivo para melhorar a performance do modelo. O resultado da pesquisa se mostrou promissor e com potencial de exploração de mais usos em grafos.

Este trabalho leva em consideração tal potencial do modelo CLIP em realizar tarefas de detecção de nós e listagem de arestas, assim como diferenciação de tipos de estruturas de grafos e quantidade de permutações entre conexões.

## 1.2 Hipóteses e Objetivos

### 1.2.1 Hipóteses de Pesquisa

As hipóteses de pesquisa a serem questionadas e sanadas neste trabalho são:

1. É possível treinar um modelo contrastivo para mensuração da similaridade entre imagens de grafos e suas listas de arestas respectivas;
2. Dada uma imagem de um grafo e um grande conjunto de possíveis listas de arestas, um modelo contrastivo é capaz de identificar corretamente a lista associada ao grafo;
3. Considerando diferentes estruturas de grafos e um grande volume de listas de arestas, um modelo contrastivo consegue agrupar as diferentes estruturas com base apenas em suas respectivas listas de arestas.

### 1.2.2 Objetivo Geral

Verificar e analisar a aplicação de modelos contrastivos para representações visuais e textuais de grafos.

### 1.2.3 Específicos

Os objetivos específicos do trabalho são:

1. Verificar a viabilidade de modelos contrastivos para a detecção de diferentes estruturas de grafos.
2. Analisar a capacidade de modelos contrastivos de diferenciar variações de um mesmo modelo de estrutura de grafo.

## 1.3 Organização

Este trabalho está organizado da seguinte forma:

- Capítulo 2: são apresentados a fundamentação teórica e conceitos necessários para compreensão dos assuntos mencionados no decorrer do trabalho, incluindo: (I) Aprendizado Profundo e Redes Neurais Convolucionais, (II) Modelos Multimodais e de Visão e Linguagem, (III) Aprendizado Contrastivo e (IV) Métricas de Avaliação.
- Capítulo 3: neste capítulo é realizada uma revisão bibliográfica de trabalhos relacionados, incluindo: (I) Modelos de Visão e Linguagem Com Dados Espaciais, (II) Modelos de Imagens com Grafos e (III) Aprendizado Contrastivo.
- Capítulo 4: o capítulo descreve as ferramentas, metodologias e técnicas utilizadas para atingir o objetivo proposto.
- Capítulo 5: são descritos os experimentos e análises dos resultados.
- Capítulo 6: apresenta as considerações finais e propostas de trabalhos futuros.

# Capítulo 2
# FUNDAMENTAÇÃO TEÓRICA

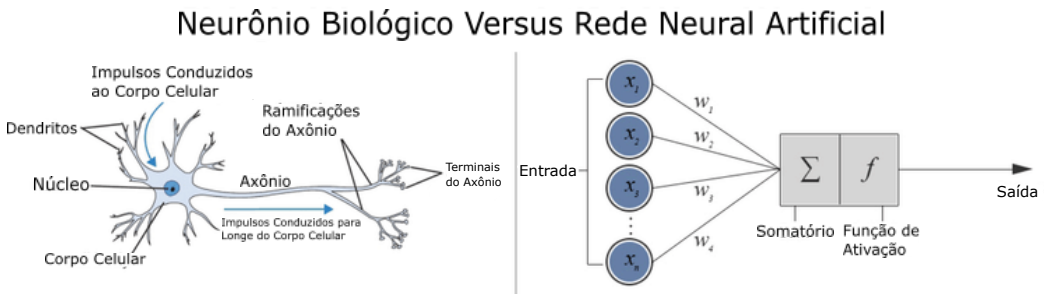
## 2.1 Aprendizado Profundo e Redes Neurais Convolucionais

Em Aprendizado de Máquina (*Machine Learning*) existe uma subcategoria de técnicas chamada de Aprendizado Profundo (*Deep Learning*), cuja principal característica se dá pelo uso de Redes Neurais Profundas capazes de exercer mais funções e funções mais complexas que as Redes Neurais mais simples.

A Rede Neural Profunda foi inspirada pela capacidade do cérebro humano de trabalhar com inúmeras sequências de neurônios para resultar em uma decisão (GOODFELLOW et al., 2016). Diferentemente das mais simples com uma ou duas camadas computacionais, a Profunda geralmente contém centenas a milhares de camadas para poder treinar um modelo.

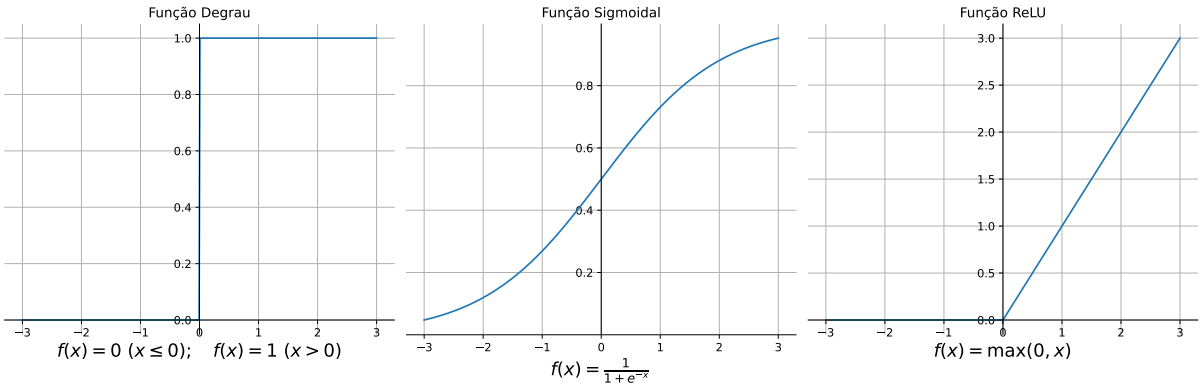
Cada camada é constituída por neurônios artificiais (MCCULLOCH; PITTS, 1943) que recebem um valor de entrada, são somados ponderadamente e geram um valor de saída, como mostrado na Figura 3. A Função de Ativação mais básica é a de Degrau (Heaviside), que retorna um valor fixo, geralmente 1, se a entrada for maior que 0. Em seguida, uma comumente usada é

**Figura 3 – Neurônio Biológico e Neurônio Artificial**



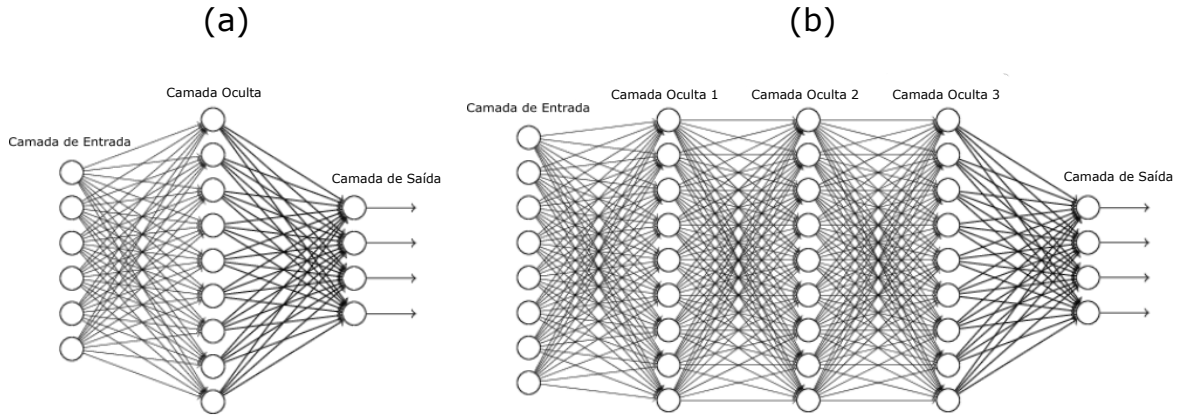

**Fonte: Adaptado de (ROUT et al., 2021)**

**Figura 4 – Funções de Ativação**




**Fonte: Próprio Autor**

**Figura 5 – Exemplos de Redes Neurais Simples e Profundas**




**(a) Rede Neural Simples (MLP); (b) Rede Neural Profunda. Fonte: Adaptado de (NIELSEN, 2015)**

a Sigmoidal, ou Logística, que tem formato de 'S' e é ativada suavemente conforme o valor de entrada aumenta. Em geral, a função de ativação mais utilizada é a ReLU (*Rectified Linear Unit*) que retorna 0 para valores menores ou iguais a zero e o mesmo valor de entrada para valores positivos. As três funções podem ser vistas na Figura 4.

Ao distribuir múltiplos neurônios em sequência se obtém o chamado Perceptron de Multicamadas (MLP) (ROSENBLATT et al., 1962). As camadas intermediárias de neurônios, também chamadas de "Camadas Ocultas", podem ser conectadas a outras camadas intermediárias em sucessão diversas vezes, formando assim uma Rede Neural Profunda. A Figura 5 exemplifica uma Rede Neural Simples e uma Rede Neural Profunda.

O Aprendizado de Máquina também pode ser dividido em dois métodos: Supervisionado e Não Supervisionado. Em Aprendizado Supervisionado, existem dois tipos de tarefas: Classificação e Regressão. Na Classificação, o modelo recebe um conjunto de dados rotulados/classificados

e aprende os diferentes padrões de característica de cada rótulo, para então receber um conjunto não rotulado e retornar as prováveis classes de cada objeto. Já a Regressão é usada para prever o valor de uma variável de acordo com os dados passados, formando uma projeção de uma função capaz de determinar o comportamento de crescimento ou decrescimento.

Para o modelo reconhecer os padrões, a informação inicial é introduzida na camada de entrada e então passada e distribuída por todas as camadas intermediárias, até a última, onde será verificado se o resultado calculado se iguala ao esperado. Esse fluxo de informação da introdução até a camada de saída é chamada de *Fowardpropagation*.

Após o cálculo de probabilidade na camada de saída, é aplicada a técnica de *Backpropagation* (RUMELHART et al., 1986) que usa a diferença entre o resultado obtido e o esperado para calcular o erro, ou a *loss*, e ajustar os parâmetros de pesos e *biases* das camadas anteriores. Com isso a rede é capaz de aprender com seus erros e convergir até uma taxa aceitável, dependendo da tarefa.

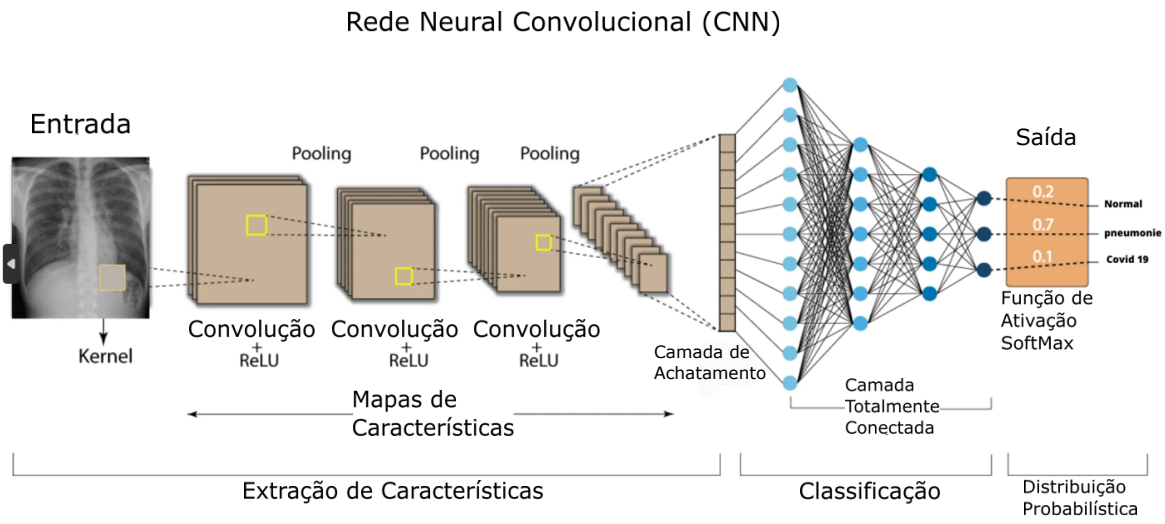
Uma das formas de otimizar o aprendizado do modelo é utilizando o algoritmo de Descida de Gradiente (CAUCHY et al., 1847), que minimiza o erro de forma iterativa ajustando os parâmetros do modelo na direção oposta ao gradiente da *loss* em relação a esses parâmetros, buscando o mínimo local ou global.

Em Aprendizado Não Supervisionado, o modelo recebe apenas um conjunto de dados sem rotulação e recebe as tarefas de Agrupamento e Regras de Associação. No Agrupamento, o modelo distribui cada objeto em um plano multidimensional de acordo com as variáveis que o representam e tenta formar grupos de objetos que mais se assemelham, normalmente pela proximidade entre eles. Existem vários métodos de se calcular a forma com que os grupos serão formados, como K-Means (MACQUEEN, 1967), DBSCAN (ESTER et al., 1996) e por Hierarquia (KAUFMAN; ROUSSEEUW, 2009; JR, 1963). A tarefa de Regras de Associação serve para identificar possíveis relações entre variáveis, sendo muito usada em análise de mercado como perfil de consumidor e venda de produtos.

No campo de Visão Computacional, foram introduzidas as Redes Neurais Convolucionais (CNN), que podem receber como entrada imagens para serem processadas e extraídas informações de interesse do usuário. As CNNs (FUKUSHIMA, 1980; LECUN et al., 2002) possuem camadas de convolução formadas por *kernels*, que são filtros que deslizam sobre a imagem e extraem uma versão compactada da informação contida naquela região da imagem. Essas características podem ser bordas, texturas ou quinas e formam um mapa de ativação.

As camadas de convolução também podem ser conectadas em sequência para evidenciar melhor alguma informação contida na imagem. Entre as camadas convolucionais é comum adicionar camadas de *pooling*, ou *downsampling*, que diminuem a dimensionalidade dos mapas de ativação e reduzem a quantidade de parâmetros de entrada da próxima camada, compactando cada vez mais em dimensões menores, mas aumentando o número de cortes da imagem. Isso

**Figura 6 – Esquema de uma CNN para Classificação**




**Fonte: Adaptado de (RGUIBI et al., 2022)**

permite que o modelo aprenda características de alto nível e que generalize melhor para diferentes escalas de imagem.

Após extrair as informações de interesse, os mapas de ativação passam por uma Camada Totalmente Conectada, que gera um vetor de probabilidades para poder classificar ou reconhecer um objeto. A Figura 6 esquematiza todo o processo de classificação de imagens com CNNs.

## 2.2 Modelos Multimodais

A chegada dos Modelos de Linguagem (LLM) revolucionou a forma com que computadores se comunicam com o usuário ao aprender como o ser humano se comunica naturalmente. Os LLMs são grandes modelos de Processamento de Linguagem Natural (PLN, ou NLP em inglês) especializados em retornar texto ao receberem uma entrada textual que descreva uma tarefa. Sua arquitetura é baseada em *Transformers* (VASWANI et al., 2017), uma Rede Profunda capaz de conceder pesos diferentes a palavras ou *tokens* que contenham maior relevância ou significância fazendo com que o modelo seja capaz de interpretar contextos e relacionar palavras. A *tokenização* é o processo de quebra das sentenças, seja em palavras, subpalavras ou até caracteres, pontuações e números. Isso serve para facilitar o processo de aprendizado da linguagem pelo modelo ao interpretar pedaço a pedaço da sentença. Com isso, modelos treinados são capazes de interpretar e responder textos de diversas formas, idiomas, períodos históricos da linguagem, usando gírias e vícios de linguagem.

Empresas como a OpenAI (OpenAI, 2025), Google (Google, 2025) e Meta (Meta AI, 2024) são globalmente reconhecidas por popularizarem o acesso a Inteligência Artificial Generativa (GenAI) e LLMs e continuarem a aprimorar os modelos.

A partir da GenAI e LLMs, foram desenvolvidos também modelos (GOODFELLOW et al., 2014) capazes de gerar imagens a partir de ruídos de acordo com o que foi pedido em forma de texto. Para isso, o modelo é treinado a reconhecer objetos em imagens que passam por filtros de ruídos diversas vezes, para que seja ensinado a gerar tal objeto no sentido oposto ao treinado. Com isso, a partir de *pixels* iniciais aleatórios, o modelo é capaz de desenhar livremente a forma e descrição passada por texto. Esse processo é chamado de Difusão (SOHL-DICKSTEIN et al., 2015; HO et al., 2020).

Com o avanço das áreas de aprendizado textual e de imagem, foram criados os Modelos Multimodais (LMM) capazes de receber e gerar tanto texto quanto imagens para suas tarefas, além de vídeos e sons em modelos mais recentes.

### 2.2.1 Modelos de Visão e Linguagem

Enquanto que os LLMs são focados em textos e LMMs em mídia em geral, os Modelos de Visão e Linguagem (VLM) (VINYALS et al., 2015) e Grandes Modelos de Visão e Linguagem (LVLM) são especializados em integrar imagem e texto permitindo aos modelos a aprenderem tanto o significado da sentença quanto sua representação visual. Com essa função é possível realizar aa tarefas como legendar imagens, responder perguntas sobre o conteúdo da imagem (*Visual Question Answering*) e a recuperação de correspondência entre imagens e textos.

## 2.3 Aprendizado Contrastivo

O Aprendizado Contrastivo (OORD et al., 2018) é uma técnica de Aprendizado de Máquina capaz de ensinar o modelo a separar os objetos mais diferentes e unir os mais similares no espaço de *embeddings*. Os *embeddings* (MIKOLOV et al., 2013) são vetores numéricos que representam objetos num espaço vetorial contínuo e são utilizados principalmente em PLN para representar numericamente o significado de palavras.

Em VLMs, o Aprendizado Contrastivo permite que os *embeddings* de textos e imagens compartilhem o mesmo espaço, fazendo com que seja possível analisar as relações entre imagens e textos e possivelmente treinar o modelo para que possa associar os pares corretos. Dessa forma, é possível calcular a distância geométrica entre cada par de objetos, a fim de ensinar a relação e correlação entre imagens e palavras e assim designar tarefas como classificação, agrupamento, reconhecimento, detecção ou geração de resposta.

O modelo Contrastive Language-Image Pre-training, ou CLIP, (RADFORD et al., 2021) desenvolvido pela OpenAI, é um dos VLMs contrastivos mais utilizados para servir como base a outros LMMs e VLMs pela sua versatilidade e facilidade de uso. O CLIP calcula a similaridade cosseno entre os *embeddings* de imagem e texto de treinamento e com isso gera o erro (*loss*) InfoNCE, uma função de custo fundamental para ajustar os parâmetros de treinamento do modelo.

## 2.4 Métricas de Avaliação

Nesta seção são apresentadas as métricas e medidas utilizadas no trabalho, bem como seus cálculos matemáticos e definições técnicas.

### 2.4.1 Similaridade Cosseno

A Similaridade Cosseno é uma métrica que mede o cosseno do ângulo $\theta$ entre dois vetores para determinar o quão similares eles são a partir de sua orientação, independentemente de seus tamanhos:

$$cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|} \tag{2.1}$$

Seus valores podem variar de $-1$ a $1$, considerando próximos a $1$ como muito similar, $0$ sem relação e $-1$ opostos. Em Aprendizado Contrastivo os vetores usados são os *embeddings*.

### 2.4.2 Perda ou *Loss*

O cálculo da função de perda mede o quão errado o modelo está em relação a resposta correta. Existem várias formas de se calcular, mas a mais comum é usando a Entropia Cruzada. A perda por Entropia Cruzada penaliza mais o modelo quando este concede uma probabilidade baixa para a classe correta.

$$CrossEntropy(p, y) = -\sum_{i=1}^{C} y_i log(p_i) \tag{2.2}$$

Sendo na equação 2.2:

- C = Número de classes;
- $y_i$ = Classe correta;
- $p_i$ = Probabilidade prevista

Porém, quando a saída do modelo gera *logits*, que são valores brutos antes de serem transformados em probabilidades, é preciso aplicar a função *SoftMax*, dada pela fórmula:

$$SoftMax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.3}$$

Sendo, na equação 2.3:

- $z_i$ = vetor de *logits* da classe $i$ que se deseja calcular a probabilidade;

- K = Número de classes;
- $z_j$ = vetores de *logits* da somatória

#### 2.4.2.1 InfoNCE

A *loss* InfoNCE (OORD et al., 2018) é um tipo de cálculo de erro para modelos contrastivos que busca aproximar os pares positivos e afastar os negativos usando a Similaridade Cosseno, a Perda por Entropia Cruzada e *SoftMax*:

$$InfoNCE = CrossEntropy(SoftMax(S), y) \tag{2.4}$$

Sendo $S$ a matriz de similaridade entre os pares de elementos em um lote de dados. As funções *SoftMax* e Entropia Cruzada são aplicadas a cada linha da matriz $S$ de forma independente. $y$ representa o par correto correspondente.

### 2.4.3 Acurácia

A Acurácia é uma métrica simples de Aprendizado de Máquina que mede a taxa de acertos do modelo em relação ao número total de predições. Considere as seguintes definições:

1. Verdadeiro Positivo (VP): valor positivo predito corretamente;
2. Verdadeiro Negativo (VN): valor negativo predito corretamente;
3. Falso Positivo (FP): valor positivo predito quando o esperado era negativo;
4. Falso Negativo (FN): valor negativo predito quando o esperado era positivo.

A acurácia é calculada como

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.5}$$

#### 2.4.3.1 Acurácia-5

Para situações em que as classes possam ser muito parecidas, se utiliza a Acurácia-5, ou *Top*-5, como forma de garantir uma margem de erro. Isso permite que o modelo tenha uma confiabilidade sem causar *overfitting*. Para isso a Acurácia-5 é medida da seguinte forma:

1. Para cada item do conjunto de dados: as probabilidades de predição são calculadas;
2. É verificado se a classe correta está entre as cinco maiores probabilidades;
3. Se sim, conta como Acerto;

4. Se não, conta como Erro;

5. Os Acertos são divididos pelo número total de itens.

### 2.4.4 Coeficiente de Silhueta

O Coeficiente de Silhueta é uma medida de *clusterização* para avaliar a qualidade da separação entre grupos. Para isso, ele mede o quão bem um objeto se encaixa em seu grupo comparado a grupos vizinhos. Quanto maior o resultado, melhor é a qualidade do agrupamento. O cálculo da Silhueta é definido por:

$$Silhouette = \frac{b - a}{max(a, b)} \tag{2.6}$$

Para:

- $a$ = média da distância entre o objeto e todos do mesmo grupo
- $b$ = média da distância entre o objeto e todos objetos de um grupo vizinho

# Capítulo 3
# TRABALHOS RELACIONADOS

Neste capítulo estão descritos artigos e pesquisas relacionados e utilizados para a realização do tema deste trabalho.

## 3.1 Modelos de Visão e Linguagem Com Dados Espaciais

A capacidade de interpretação de imagens dos modelos do estado da arte, apesar de poderosos para algumas tarefas, ainda tem dificuldades em outras questões principalmente envolvendo raciocínio lógico visual e mapeamento estrutural de elementos presentes nas imagens. Por isso, diversas pesquisas de testes e *benchmarks* são relevantes para avaliar constantemente a evolução desses modelos.

De acordo com os resultados obtidos de (WANG et al., 2024), os modelos multimodais utilizados, tanto LLMs quanto LMMs, mostraram dificuldade em interpretar questões matemáticas envolvendo texto e imagem para a resolução de problemas. Os principais tópicos das questões envolviam Geometria Espacial, Topologia, Teoria dos Grafos, Transformações Geométricas e entre outros. Os autores concluíram que as habilidades de raciocínio dos modelos ainda estão distantes dos seres humanos e, portanto, ainda não são confiáveis para realizarem tais tarefas.

A pesquisa de (GUO et al., 2025) demonstra a dificuldade dos modelos retornar uma saída multimodal (texto e imagem) ao responder questões multidisciplinares, como por exemplo: resolução de labirintos, conexão de pontos e ilustração de formas geométricas. Apesar de apresentar uma evolução considerável, os resultados ainda estão bem abaixo do necessário para serem aplicados com o mínimo grau de confiança.

O modelo desenvolvido no trabalho de (WANG et al., 2025) apresentou resultados positivos em reconhecimento e geração de imagens químicas de estruturas moleculares. A tarefa de entender as estruturas moleculares e gerar um texto, ou gerar uma imagem a partir da representação textual, é muito parecida com os objetivos de treinar um modelo especializado em Teoria dos Grafos. Esta conquista foi uma das inspirações para o projeto proposto neste trabalho.

## 3.2 Modelos de Imagem Com Grafos

Grafos, apesar de possuírem representação textual, muitas vezes são utilizados em sua forma estrutural visual para melhorar sua interpretabilidade em diferentes contextos para humanos. Porém, modelos multimodais possuem limitações para interpretá-los.

(GUO et al., 2023) e (DAS et al., 2024) estudaram a proficiência de LLMs aprenderem a compreender grafos em diversas formas estruturais, semânticas, em texto e imagem. Ambas pesquisas apresentaram as limitações das LLMs em classificações, detecções, abstrações e manipulações de grafos.

Um dos grandes obstáculos dos modelos para aprenderem a entender o que é um grafo é a sua diversidade multidimensional e multimodal. Um único grafo pode ter a mesma lista de arestas, mas a posição espacial dos nós ser diferente (isomorfismo), o grafo pode possuir arestas e/ou nós com pesos, arestas direcionadas, etc.

O artigo (WEI et al., 2024) propôs o uso de aumento de dados alterando apenas um aspecto do grafo de cada vez, enquanto mantém o restante constante, e concluiu que essa é uma das formas de melhorar a eficiência de treinamento de modelos baseados em raciocínio visual de grafos.

Utilizando LMMs, (LI et al., 2024) e (BABAIEE et al., 2025) avaliaram modelos em questões sobre Teoria dos Grafos. O primeiro introduziu um agente (DPR) capaz de aprimorar a performance de LMMs ao facilitar algumas tarefas em que os modelos tiveram maior dificuldade, usando ferramentas de descrição, decomposição e geração de código. Já (BABAIEE et al., 2025) usou 6 *datasets* com três tarefas distintas para avaliar os modelos: detecção de isomorfismo, cálculo de caminho e análise de ciclos. Seus resultados apontaram que, mesmo alterando os *layouts* das estruturas de grafos, os modelos foram incapazes de responder minimamente corretamente as questões propostas, enquanto que humanos atingiram 100% de acurácia em muitas perguntas.

Os autores (HOU et al., 2024) e (ZHU et al., 2025) obtiveram resultados interessantes sobre LVLMs e suas análises são pertinentes quanto às limitações dos modelos. Apesar de não ter usado explicitamente grafos, o primeiro autor utilizou diagramas para testar as LVLMs e constatou um comportamento vicioso dos modelos de dependerem muito da descrição textual e conhecimento previamente treinado para responder algumas questões mais complexas. Além disso, os testes revelam a facilidade dos modelos detectarem e entenderem as entidades relacionadas, mas não as relações em si. O segundo autor sugere um *framework* (MCDGraph) supervisionado de *fine-tunning* para LVLMs a fim de melhorar a performance. O MCDGraph refina o aprendizado dos modelos aplicando 3 questões para reforçar o entendimento dos problemas de estrutura, relações, detecção de isomorfismo, interpretação e sumarização de grafos. A aplicação envolve aumento de dados ao esconder um nó, aprendizado contrastivo de grafos isomórficos e descrição textual da imagem fornecida.

## 3.3 Aprendizado Contrastivo

O modelo CLIP de (RADFORD et al., 2021), criado pela OpenAI, é um duplo *encoder* que associa uma imagem a um texto de linguagem natural e tenta manter o par correto o mais próximo possível e afastar os incorretos. Para isso se utiliza a técnica de Aprendizado Contrastivo que calcula a perda contrastiva InfoNCE cujo resultado retorna as similaridades entre pares de imagem e textos.

O artigo (TERASHITA et al., 2025) foi o principal motivador da escolha do modelo CLIP para este trabalho. Nele, os autores explicam como o CLIP é popularmente usado em grandes VLMs e LMMs como *encoder* de imagem, mas falha ao enxergar arestas presentes em grafos, sendo muito dependente de uma descrição textual ou mapeamento trivial dos nós presentes. Ao treinar o modelo CLIP especificamente para identificar grafos, sua performance supera a do GPT-4o (OpenAI, 2025) e LLaVA-Mistral (LIU et al., 2024) pré-treinados. Isso reforça a ideia de treinar o modelo do zero para se especializar em detecção de arestas ao invés de depender de entradas textuais adicionais que podem atrapalhar em uma análise mais objetiva sobre grafos.

# Capítulo 4
# METODOLOGIA E DESENVOLVIMENTO

Esta seção descreve os dados, ferramentas e técnicas utilizadas no desenvolvimento deste trabalho.

## 4.1 Natureza da Pesquisa

A natureza deste trabalho é de Pesquisa Aplicada Experimental, pois seu propósito é utilizar ferramentas já existentes para realizar uma tarefa e avaliar empiricamente seu desempenho, a fim de contribuir para esta área de visão computacional recém explorada no estado da arte.

## 4.2 Ferramentas

As ferramentas e bibliotecas utilizadas para desenvolvimento e treinamento do modelo CLIP, bem como as análises estatísticas dos experimentos são *open-source* e gratuitas. São estas:

- Software de Edição de Texto: *Visual Studio Code*
- Linguagem de programação: *Python* 3.9
- Framework para execução do projeto: *PyTorch*
- API para processamento dos tensores: CUDA
- Computação numérica: *NumPy*
- Análise e manipulação dos dados: *Pandas*
- Cálculo de métricas: *Scikit Learn*
- Plot de gráficos e imagens: *MatplotLib*

Descrição técnica do dispositivo utilizado, disponibilizado pelo laboratório *Image and Network Analysis Group* (INAG) do Departamento de Computação (DC) da Universidade Federal de São Carlos (UFSCar):

- Processador: Intel i9-12900KF
- Placa gráfica: GeForce RTX 3090 24GB
- Memória RAM: 64 GB

## 4.3 Conjunto de Dados

Foram utilizadas duas bases de dados para este projeto, uma menor e mais simples com apenas 2 tipos de estruturas e 2500 itens e outra maior com 5 variações de um mesmo modelo com 5070 itens. Cada imagem é composta por um fundo preto, círculos preenchidos em vermelho para representar os nós, linhas brancas como arestas conectando os nós e valores numéricos em preto para identificação. Para ambas bases o princípio de geração é o mesmo, com a diferença do tipo de estrutura espacial. Para fins de aleatoriedade e variância, o posicionamento dos nós foi parcialmente fixado e as conexões totalmente aleatorizadas.

O código[1] gerador das bases de dados foi fornecido pelo Dr. Cesar Henrique Comin e modificado pelo autor para atender as necessidades do trabalho.

### 4.3.1 Conjunto Estrela-Grade

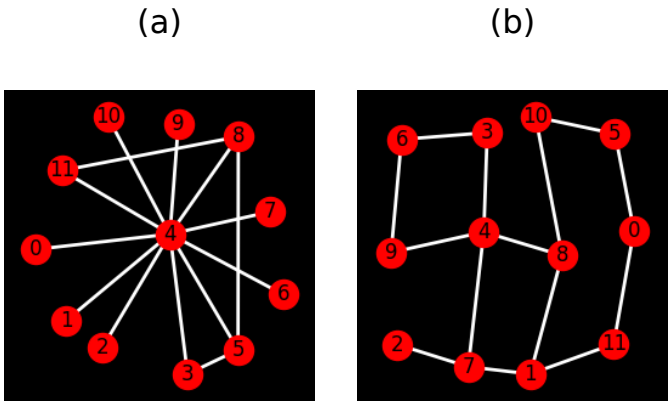
O conjunto Estrela-Grade contém 2500 imagens 2D de grafos artificialmente gerados de dimensão $256 \times 256$ e uma lista de 2500 textos contendo as respectivas listas de arestas. Essa base de dados está divida em 2 grupos de acordo com sua estrutura espacial em Estrela (*hub* central) e Grade (grafo aleatório com os nós alinhados em 3 linhas e 4 colunas). Os valores identificadores dos nós também foram aleatorizados para permitir maior dissimilaridade entre eles. Exemplos de imagens da base são mostrados na Figura 7.

### 4.3.2 Conjunto Watts-Strogatz

Já a base Watts-Strogatz contém 5070 imagens 2D de grafos artificialmente gerados de dimensão $256 \times 256$ *pixels* e uma lista com 5070 textos contendo as listas de arestas das respectivas imagens. Cada imagem contém pequenas diferenças como posições de nós, centralidade e arestas, mas os rótulos dos nós foram previamente fixados. Cada grafo está associado a um número de reconexões (*rewires*) previamente determinados, variando entre 1, 2, 3, 6 e 10 reconexões. Para cada quantidade de reconexão foram gerados 1250 grafos, com exceção do

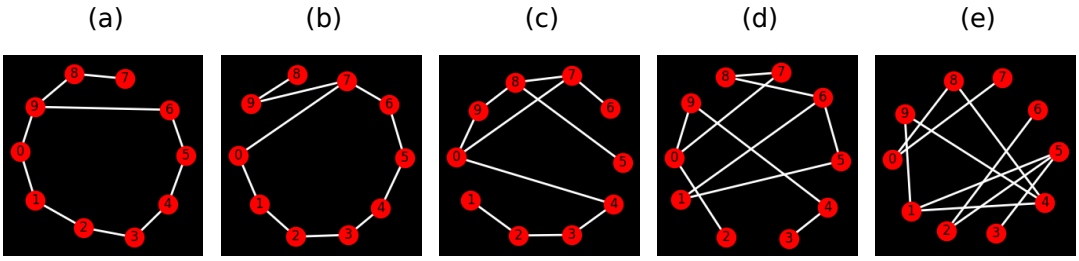
[1] Código fonte da geração de dados: https://github.com/Krayth/TCC-graphs-database

**Figura 7 – Exemplos de imagens do conjunto de dados Estrela-Grade.**




**(a) Grafo tipo Estrela; (b) Grafo tipo Grade. Fonte: Próprio autor**

**Figura 8 – Exemplos de imagens do conjunto de dados Watts-Strogatz.**




**(a)1 *rewire*; (b)2 *rewires*; (c)3 *rewires*; (d)6 *rewires*; (e)10 *rewires*. Fonte: Próprio autor**

valor 1, que contém apenas 70 possibilidades. Exemplos de cada *rewire* são mostrados na Figura 8.

### 4.3.3 Subconjuntos de Treinamento e Validação

Os conjuntos Estrela-Grade e Watts-Strogatz foram divididos em 80% de imagens para trenamento e 20% para validação.

## 4.4 Estratégias e Técnicas

Aqui serão descritas as especificações e particularidades das configurações e treinamento do modelo, assim como quais métricas foram calculadas e selecionadas.

### 4.4.1 Configuração do Modelo

O modelo CLIP utilizado segue a mesma arquitetura do original, com dois *encorders* de texto e imagem, cálculo de Similaridade Cosseno entre cada texto e todas as imagens e a *Loss*

InfoNCE. A CNN utilizada para recuperar informações das imagens é a ResNet-50[2] pré-treinada na base de dados ImageNet e o *transformer* de linguagem foi o DistilBERT[3] (SANH et al., 2019), uma versão menor, mais rápida, mas tão potente quanto sua versão original, BERT.

### 4.4.2 Parâmetros de Treinamento

Para o treinamento foram usados os seguintes parâmetros:

- Número de Épocas: 100
- Tamanho de *batch*: 8
- Taxa de Aprendizado: $1e^{-5}$
- Decaimento de Pesos: 0.01
- Número de *Workers* do *Dataloader*: 6
- Otimizador de Treinamento: *PyTorch Adam*[4]
- Tokenizador: bert-base-uncased[5]

Além dessas configurações, outras variáveis e características foram necessárias para a obtenção de um resultado mais coerente e interpretável que serão descritas nas subseções seguintes.

### 4.4.3 Treinamento do Modelo e Métricas

O *dataloader* que carrega a base de dados de treinamento foi aleatorizado, mas não o de validação para se ter um resultado mais adequado. Também se fez necessário a desativação dos métodos de *Dropout* da *pipeline* do DistilBERT, pois os textos, que são as listas de arestas dos grafos, são uma sequência de valores numéricos. O modelo possui dificuldade em inferir nós faltantes a partir de nós próximos na lista de arestas, pois estes são arbitrários. A aplicação de *dropouts* dificulta o modelo de aprender durante a fase de treinamento, gerando assim um *underfitting*.

Durante cada época, o modelo entrou em fase de treinamento e, para cada passo, um lote de imagens e respectivas listas de arestas foi carregado, os textos foram *tokenizados*, e uma matriz de *logits* foi gerada entre cada par de imagem e texto. A *loss* InfoNCE foi então calculada, seguida da retropropagação dos pesos, para finalizar com o acúmulo de gradientes e soma do erro dos pares para se obter a perda daquela época.

[2] https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html
[3] https://huggingface.co/distilbert/distilbert-base-cased
[4] https://docs.pytorch.org/docs/stable/generated/torch.optim.Adam.html
[5] https://huggingface.co/google-bert/bert-base-uncased

Ainda na mesma época, depois da fase de treinamento, a fase de validação foi iniciada. O processo foi o mesmo que o da fase de treinamento, com exceção da retropropagação, que não foi realizada.

Após cada época, as médias das *losses* e Acurácia-5 foram calculadas, assim como outras métricas de avaliação de agrupamento, como Coeficiente de Silhueta, Índice de Calinski-Harabasz e Índice de Davies-Bouldin, utilizadas para medir o quão separados estão os grupos e/ou quão próximos estão os objetos de mesmo rótulo.

Contudo, apenas a *loss* e Acurácia-5 se mostraram relevantes o suficiente para serem analisadas, pois as outras medidas não resultaram em valores significativos por conta da natureza dos dados serem muito semelhantes, gerando assim uma dispersão de dados muito aglutinada e sem divisão própria dos grupos.

Após a geração das métricas, outros métodos analíticos foram usados para avaliar a qualidade dos treinamentos do modelo, como a similaridade de todos os pares imagem-texto, a distribuição desses valores entre classes (número de *rewires*) e a distribuição das similaridades de uma imagem com todos os textos, para melhor compreender o motivo das métricas de agrupamento não resultarem em valores relevantes. Também foram aplicados os métodos de projeção UMAP (*Uniform Manifold Approximation and Projection*) e LDA (*Linear Discriminant Analysis*) para visualização dos dados.

O código de treinamento e resultados do modelo podem ser encontrados no repositório público no perfil do autor[6].

[6] https://github.com/Krayth/TCC-graphs-CLIP-test2/tree/1000epochs

# Capítulo 5
# RESULTADOS

Neste capítulo são apresentados os resultados do treinamento e métricas do modelo, assim como alguns gráficos relevantes para serem analisados.

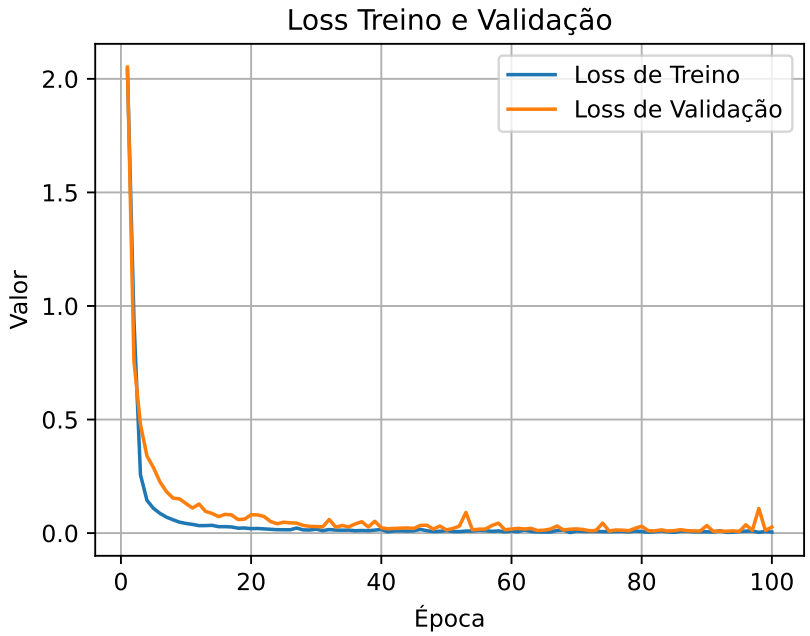
## 5.1 Treinamento e Validação do Modelo

O modelo CLIP utilizado foi baseado na arquitetura original, mas inclui algumas modificações pontuais consideradas necessárias para produzir resultados coerentes e adequados para análise. As configurações do treinamento utilizadas para obtenção dos resultados apesentados neste capítulo seguem abaixo:

1. Divisão do *dataset*: 80% para treino e 20% para validação, com aleatorização do conjunto de treino, mas não da validação;
2. Não utilização de *Data Augmentation* para evitar transformações que causem aleatorização na imagem resultante;
3. Normalização das intensidades dos valores das imagens;
4. Imagens de entrada com resolução original: $256 \times 256$;
5. Taxa de aprendizado igual a $1e^{-5}$;
6. 100 épocas de treinamento;
7. Decaimento de peso igual a 0.01;
8. Tamanho do *batch* igual a 8, por motivos de limitação de *hardware*

Consideradas as características do modelo, seguem os resultados.

**Figura 9 – Cálculo da *Loss* de Treinamento x Validação (Watts-Strogatz)**




**Fonte: Próprio autor**

## 5.2 Principais Métricas Resultantes

Todas as métricas foram calculadas durante as épocas e recuperadas ao final do treinamento para plotagem. Cada treinamento de 100 épocas teve duração média de 1 hora, sendo o principal custo computacional do trabalho.

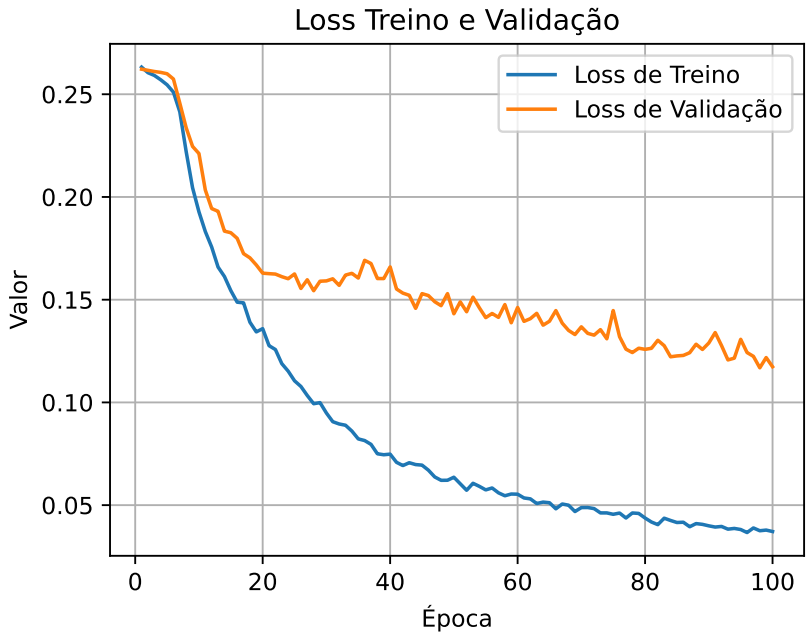
### 5.2.1 Função de Perda (*Loss*)

A primeira métrica analisada foi a Função de Perda por praticidade e simplicidade da recuperação dos valores.

#### 5.2.1.1 Conjunto Watts-Strogatz

Na Figura 9 é notável verificar a verticalidade no início do gráfico por algumas razões. Um dos fatores mais importantes da queda da perda foi a desativação do *dropout* durante o modo de treinamento. Isso foi necessário considerando a natureza dos dados muito parecidos e com poucas diferenças principalmente na versão textual, pois como as listas de arestas não carregam significados relevantes e consideravelmente diferentes, como normalmente é na linguagem natural, a aplicação do *dropout* dificultava mais o modelo de aprender durante a fase de treinamento do que de validação, resultando em um cenário em que a perda de treino era maior que a de validação.

**Figura 10 – Cálculo da *Loss* de Treinamento x Validação (Estrela-Grade)**




**Fonte: Próprio autor**

#### 5.2.1.2 Conjunto Estrela-Grade

Por outro lado, no Conjunto Estrela-Grade, a *loss* se comportou de forma menos acentuada, alcançando valores baixos e promissores para o Treinamento e a Validação alcançou cerca de 3 vezes o valor. Por motivos de custo e tempo, as épocas usadas se limitaram a 100, mas é possível ver na Figura 10 como os dois gráficos apresentam uma leve tendência decrescente ao atingir as últimas épocas, sendo plausível a conclusão de que poderiam atingir valores menores se o modelo fosse treinado por mais tempo.

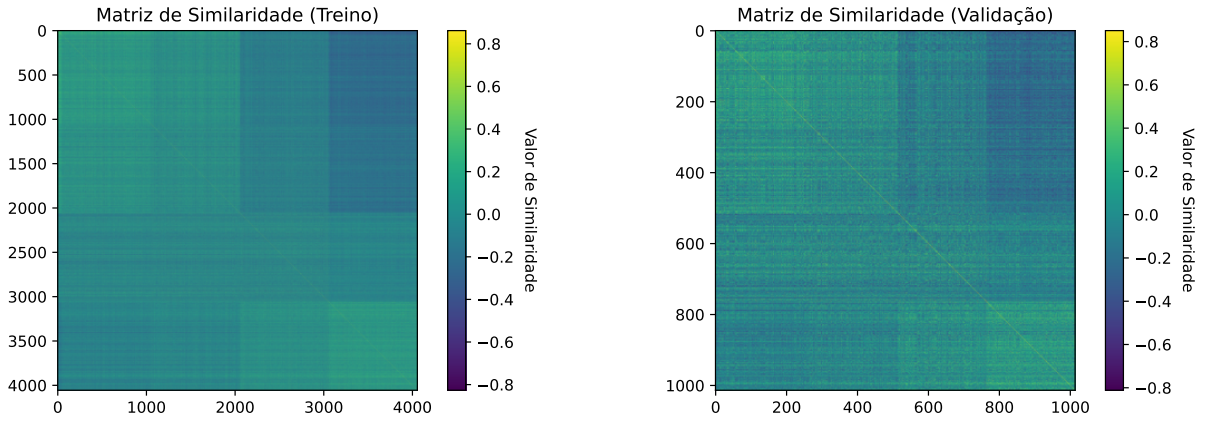
### 5.2.2 Similaridade

A Similaridade foi calculada comparando os *embeddings* de todas as imagens com todos os textos, gerando assim uma matriz. A Matriz de Similaridade, demonstrada por um Mapa de Calor, representa em suas linhas as imagens $i$ e nas colunas as listas de arestas $j$, sendo a diagonal $(i, i)$ o par correto respectivo. As matrizes de similaridade mostram como as diferentes classes dos conjuntos estão parcialmente agrupados em quadrantes e quais grupos mais se destacam.

#### 5.2.2.1 Conjunto Watts-Strogatz

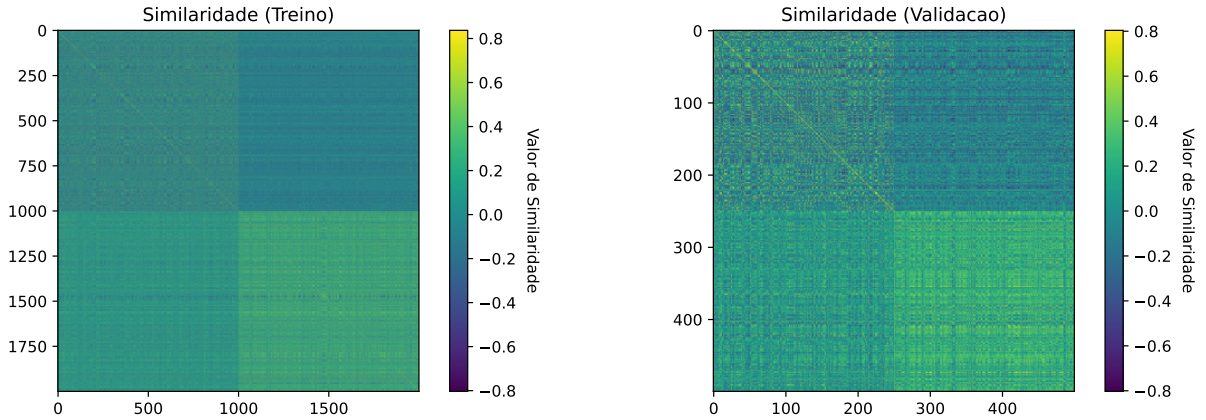
Na Figura 11 são mostradas as matrizes de similaridade do conjunto Watts-Strogatz. É possível ver que as imagens de 10 *rewires* estão mais similares às suas respectivas listas. Já os grupos com 2 e 3 *rewires* parecidos entre si. Enquanto que o grupo com menor distinção é o de 6 *rewires*, possivelmente por ser a transição entre 3 e 10 *rewires* e compartilhando características

**Figura 11 – Matrizes de Similaridade (Watts-Strogatz)**




**Fonte: Próprio autor**

**Figura 12 – Matrizes de Similaridade (Estrela-Grade)**



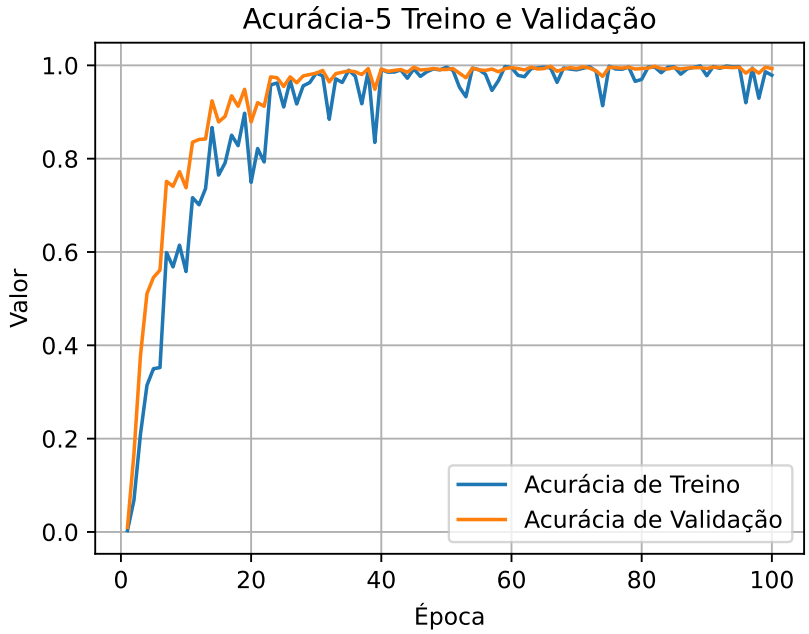

**Fonte: Próprio autor**

com ambos. Outra característica importante das matrizes é verificar os valores da diagonal principal que se destacam melhor no conjunto de validação, por ter menos objetos e ficar mais visível. Isso implica que o modelo está conseguindo parear corretamente os grafos com suas respectivas listas.

#### 5.2.2.2 Conjunto Estrela-Grade

Por possuir apenas duas classes e elas serem mais distintas, na Matriz de Estrela-Grade, representado pela Figura 12, fica mais visível a capacidade do modelo de distinguir os dois grupos comparando suas representações visuais e textuais. O quarto quadrante representa as imagens e listas dos grafos de estrutura em Grade, ganhando destaque pelos seus altos valores em média. A classe Estrela, apesar de estar menos homogênea, ainda consegue atingir bons resultados na diagonal principal, ou seja, os pares corretos estão com a maior Similaridade.

**Figura 13 – Acurácia-5 de Treinamento x Validação (Watts-Strogatz)**




**Fonte: Próprio autor**

### 5.2.3 Acurácia-5

Para o cálculo da acurácia foi aplicada a métrica *Top 5*, que mede se a classe correta está entre as 5 previsões mais prováveis, sendo a classe, nesse caso, a imagem associada a lista de arestas correta. Como se utilizou a similaridade para verificar quais imagens são mais próximas a quais listas de arestas, foi possível verificar se a imagem correspondente a sua respectiva lista está entre as 5 imagens mais similares entre imagem-texto.

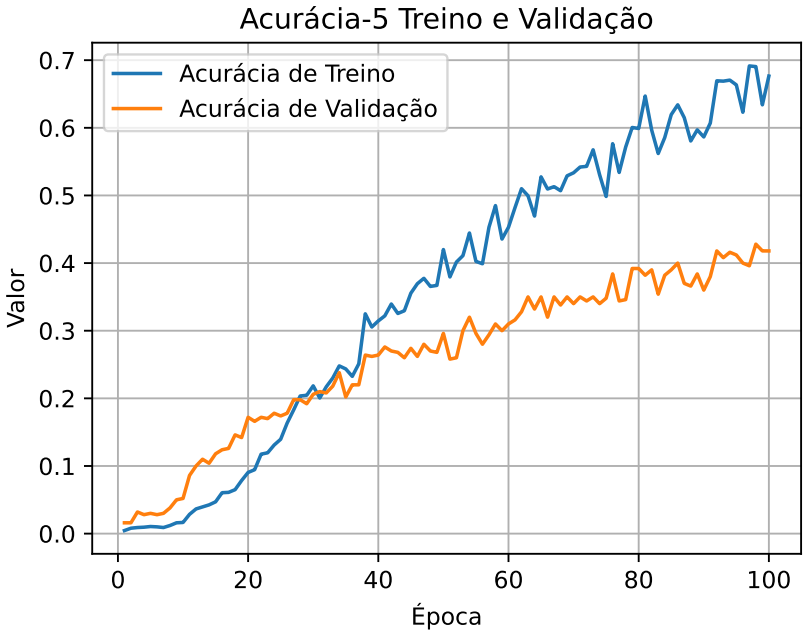
#### 5.2.3.1 Conjunto Watts-Strogatz

A Figura 13 reflete o resultado do cálculo de perda do conjunto Watts-Strogatz ao possuir a mesma verticalidade na Acurácia-5. Isso significa que o modelo está acertando os pares corretos quase com perfeição, chegando a atingir valores próximos a $0.99$ e $1.0$ em algumas épocas.

#### 5.2.3.2 Conjunto Estrela-Grade

A Acurácia-5 do conjunto Estrela-Grade resultou em valores menores do que no conjunto Watts-Storgatz, como pode ser visto na Figura 14. Isso pode ser analisado em conjunto de sua Matriz de Similaridade, da Figura 12, ao verificar o primeiro quadrante da matriz de validação. A matriz revela altos valores de similaridade fora da diagonal principal, significando que o modelo pareou a mesma imagem com mais de uma lista de arestas ou vice-versa. Por isso, sua Acurácia-5 foi prejudicada. Porém, no conjunto de Treinamento atingiu bons resultados, perto de $0.7$. Vale

**Figura 14 – Acurácia-5 de Treinamento x Validação (Estrela-Grade)**




**Fonte: Próprio autor**

ressaltar que, assim como na *loss*, a limitação de épocas de treinamento pode alterar o cenário, visto que ambos os gráficos apresentam uma tendência crescente.

## 5.3 Outros Resultados

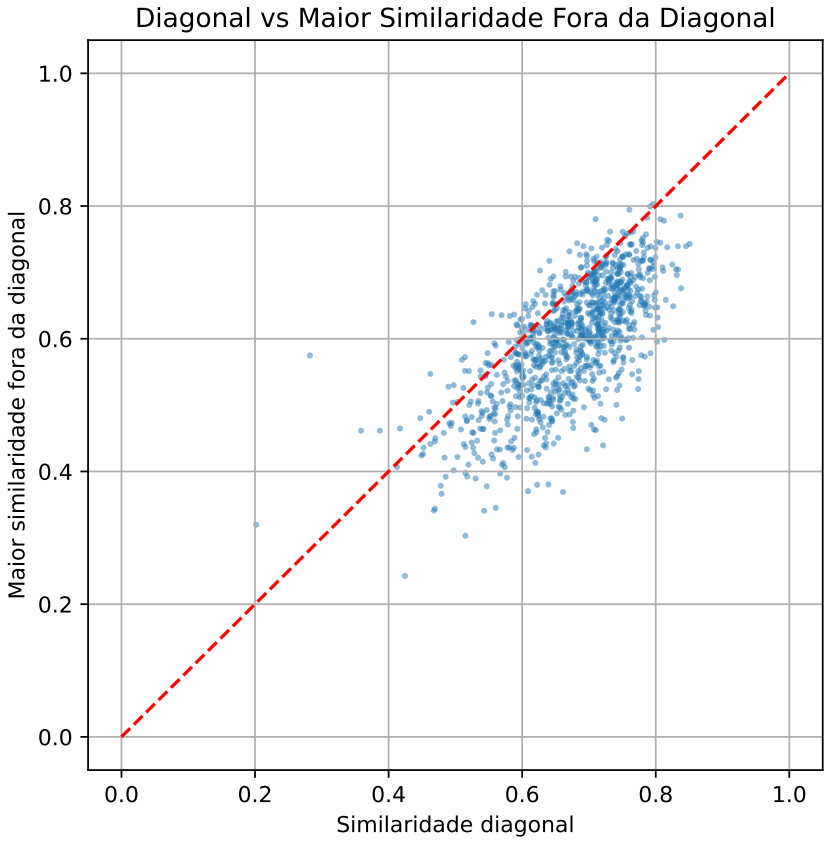
Os resultados apresentados a seguir consideram apenas o conjunto de dados Watts-Strogatz. O conjunto é mais interessante do que o Estrela-Grade porque existe uma medida referência de similaridade entre os grafos, dada pelo número de reconexões realizadas. Por exemplo, espera-se que grafos com uma reconexão sejam mais similares a grafos com duas reconexões do que a grafos com dez reconexões.

### 5.3.1 Comparação entre Similaridades Corretas e Incorretas

Outra forma de analisar as similaridades, além da matriz, é observar a distribuição de pontos que representam a similaridade dos pares corretos (imagem-texto) e o maior valor incorreto. No gráfico da Figura 15 são mostrados no eixo x os valores de similaridade entre os pares de imagens e textos corretos (diagonal da matriz de similaridade) e no eixo y o maior valor de similaridade entre cada imagem e todas as listas de arestas. A linha vermelha mostra a função $y = x$. Pontos abaixo da linha vermelha indicam pares para os quais o modelo é capaz de identificar a imagem e aresta correspondente.

Pode-se notar que há uma concentração maior abaixo do limitante, apesar de não muito

**Figura 15 – Similaridade Diagonal Principal VS Máxima Similaridade Externa**




**Fonte: Próprio autor**

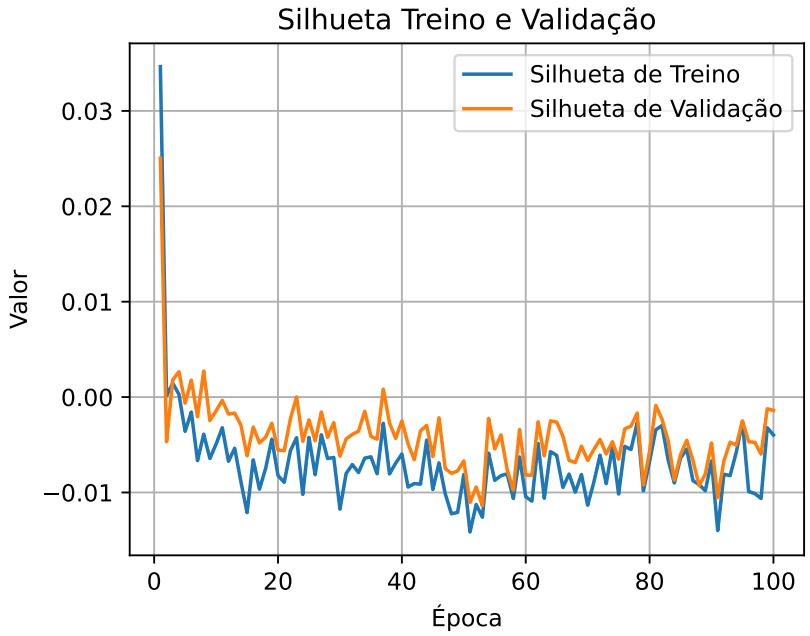
distante. Isso mostra como todos os pares possuem uma semelhança, mas é o suficiente para que possam ser distinguidos e associados corretamente.

### 5.3.2 Coeficiente de Silhueta

O Coeficiente de Silhueta tem a função de medir o quão separados estão os grupos ao mesmo tempo que mede o quão próximos estão os objetos do mesmo grupo. Essa medida é fundamental para verificar se o modelo foi capaz de dividir o conjunto em diferentes números de *rewires* de forma consistente.

A Figura 16, porém, evidencia que o modelo não foi capaz de ajustar divisões claras para se obter uma *clusterização* confiável, provavelmente causado pelo cálculo do Coefficiente de Silhueta usar a distância euclideana entre objetos. O modelo CLIP considera a similaridade de cosseno, que considera apenas os ângulos entre vetores, e não necessariamente as distâncias entre os pontos. O resultado pouco abaixo do valor 0, contudo, não significa um agrupamento ruim, pois a ausência completa de agrupamente ocorre apenas quando o Coeficiente de Silhueta é -1. Portanto, isso abre espaço para que possam ser feitos ajustes finos a fim de se obter um resultado mais positivo.

**Figura 16 – Coeficiente de Silhueta ao Longo das Épocas**

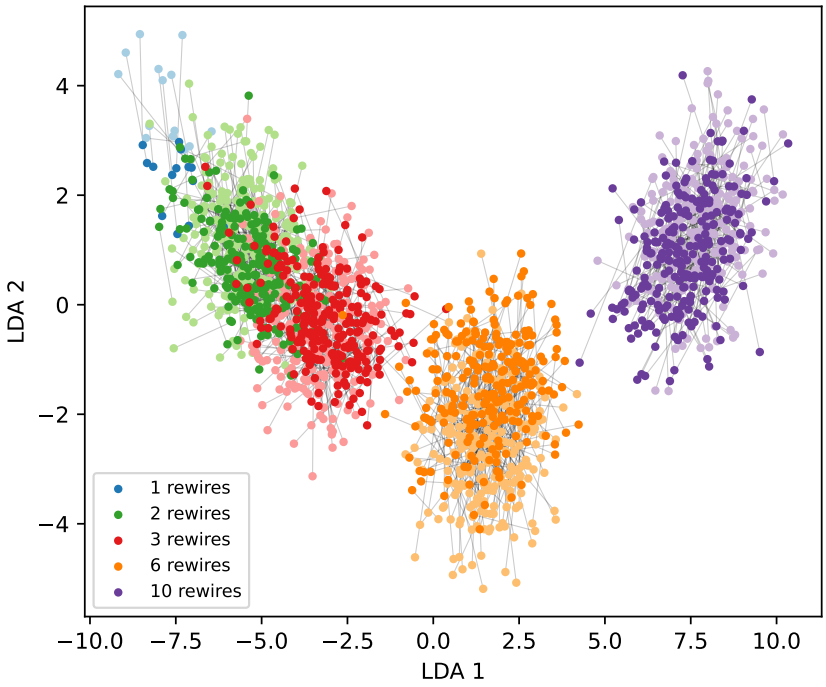



**Fonte: Próprio autor**

### 5.3.3 Análise Discriminante Linear

A Análise Discriminante Linear (LDA) (FISHER, 1936), é uma projeção linear de dados que busca maximizar a distância entre classes e minimizar a variabilidade dentro das classes ao mesmo tempo que reduz a dimensionalidade e seleciona as características que melhor dividem os grupos, assim como a técnica *Principal Component Analysis* (PCA), mas para modelos supervisionados. Para isso se utilizou os *embeddings* de imagens e textos e a rotularização dos objetos de acordo com o número de *rewires*. Na Figura 17 é notável perceber como as classes de *rewires* estão bem agrupadas, tanto as imagens (cores pasteis) como os textos (cores escuras), com seus respectivos pares conectados por linhas pretas, dando destaque ao grupo de 10 *rewires* que está mais distante de todos, os *rewires* 1, 2 e 3 mais próximos entre si e o de 6 sendo intermediário. Em particular, nesta ocasião os grupos estão bem definidos pelo eixo x, sendo possível gerar uma projeção unidimensional.

**Figura 17 – LDA**




**Tons pasteis: *embeddings* das imagens; Tons escuros: *embeddings* dos textos. Fonte: Próprio autor**

# Capítulo 6
## CONCLUSÃO

A aplicação de modelos de visão e linguagem do estado da arte para detectar e entender grafos ainda não atingiu a performance necessária para se obter um resultado satisfatório. A exploração de outros métodos é fundamental para que se possa contornar esse obstáculo, além de reduzir os custos ao diminuir o escopo do problema e buscar soluções mais factíveis de serem implementadas em futuras pesquisas, para continuar o progresso.

Considerando isso, o trabalho escrito apresenta uma análise do potencial de modelos contrastivos para classificação e agrupamento de estruturas de grafos com base em suas listas de arestas. Para que isso fosse possível, uma sequência de experimentos e estudos sobre modelos aplicados a imagens de grafos foram realizados, a fim de encontrar uma alternativa capaz de atingir um desempenho relevante para o avanço do estado da arte. Por consequência, descobriu-se a capacidade notória do modelo CLIP para tarefas que envolvem grafos e seu potencial de aplicação em diferentes contextos.

O modelo CLIP se mostrou eficiente em agrupar imagens com diferentes estruturas e parâmetros de grafos. O modelo aprendeu com sucesso um espaço compartilhado entre imagens de grafos e respectivas listas de arestas, possibilitando o cálculo da similaridade entre os dois conjuntos de dados.

Uma das maiores dificuldades que se apresentaram na pesquisa foi compreender o comportamento dos resultados e quais parâmetros de treinamento, dados e modelos tinham maior relevância no desempenho final do experimento.

A revisão bibliográfica foi fundamental para que se pudesse chegar na conclusão do uso do modelo aplicado assim como na forma em que seria utilizado. Portanto, houve um intenso foco analítico e interpretativo sobre o potencial do modelo CLIP, além da necessidade de estudar mais a fundo sobre métricas e representações visuais capazes de traduzir de forma mais legível os resultados obtidos.

## 6.1 Trabalhos futuros

A partir dos resultados obtidos neste trabalho, esta seção apresenta potenciais direções que podem ser exploradas em pesquisas subsequentes.

1. Treinar o modelo por mais épocas até atingir a convergência das métricas;
2. Aplicação do modelo em outras estruturas de grafos;
3. Exploração de outras arquiteturas de modelo, como outras versões de *Transformers* e/ou CNNs;
4. Desenvolvimento de modelos generativos capazes de gerar listas de arestas a partir de imagens;
5. Identificação da lista de arestas de um grafo mapeado a partir de vasos sanguíneos.

# REFERÊNCIAS

BABAIEE, Z.; KIASARI, P. M.; RUS, D.; GROSU, R. Visual graph arena: Evaluating visual conceptualization of vision and multimodal large language models. *arXiv preprint arXiv:2506.06242*, 2025. Citado na página 24.

CAUCHY, A. et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, v. 25, n. 1847, p. 536–538, 1847. Citado na página 17.

COMIN, C. H.; GALVÃO, W. N. Vessshape: Few-shot 2d blood vessel segmentation by leveraging shape priors from synthetic images. *arXiv preprint arXiv:2510.27646*, 2025. Citado na página 11.

DAS, D.; GUPTA, I.; SRIVASTAVA, J.; KANG, D. Which modality should i use-text, motif, or image?: Understanding graphs with large language models. In: *Findings of the Association for Computational Linguistics: NAACL 2024*. [S.l.: s.n.], 2024. p. 503–519. Citado na página 24.

DAVID, S. A.; MAHESH, C.; KUMAR, V. D.; POLAT, K.; ALHUDHAIF, A.; NOUR, M. Retinal blood vessels and optic disc segmentation using u-net. *Mathematical Problems in Engineering*, Wiley Online Library, v. 2022, n. 1, p. 8030954, 2022. Citado na página 11.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *kdd*. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231. Citado na página 17.

FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936. Citado na página 38.

FREITAS-ANDRADE, M.; COMIN, C. H.; SILVA, M. V. da; COSTA, L. d. F.; LACOSTE, B. Unbiased analysis of mouse brain endothelial networks from two-or three-dimensional fluorescence images. *Neurophotonics*, Society of Photo-Optical Instrumentation Engineers, v. 9, n. 3, p. 031916–031916, 2022. Citado 2 vezes nas páginas 11 e 12.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, Springer, v. 36, n. 4, p. 193–202, 1980. Citado na página 17.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. *Deep learning*. [S.l.]: MIT press Cambridge, 2016. v. 1. Citado na página 15.

GOODFELLOW, I. J.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. *Advances in neural information processing systems*, v. 27, 2014. Citado na página 19.

Google. *Gemini API (versão 2.5 Pro)*. 2025. <ai.google.dev>. Acesso em: 10 dez. 2025. Citado na página 18.

GUO, J.; DU, L.; LIU, H.; ZHOU, M.; HE, X.; HAN, S. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023. Citado na página 24.

GUO, M.-H.; CHU, X.; YANG, Q.; MO, Z.-H.; SHEN, Y.; LI, P.-l.; LIN, X.; ZHANG, J.; CHEN, X.-S.; ZHANG, Y. et al. Rbench-v: A primary assessment for visual reasoning models with multi-modal outputs. *arXiv preprint arXiv:2505.16770*, 2025. Citado na página 23.

HO, J.; JAIN, A.; ABBEEL, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, v. 33, p. 6840–6851, 2020. Citado na página 19.

HOU, Y.; GILEDERELI, B.; TU, Y.; SACHAN, M. Do vision-language models really understand visual language? *arXiv preprint arXiv:2410.00193*, 2024. Citado na página 24.

JR, J. H. W. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, Taylor & Francis, v. 58, n. 301, p. 236–244, 1963. Citado na página 17.

KAUFMAN, L.; ROUSSEEUW, P. J. *Finding groups in data: an introduction to cluster analysis*. [S.l.]: John Wiley & Sons, 2009. Citado na página 17.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 2002. Citado na página 17.

LI, Y.; HU, B.; SHI, H.; WANG, W.; WANG, L.; ZHANG, M. Visiongraph: Leveraging large multimodal models for graph theory problems in visual context. *arXiv preprint arXiv:2405.04950*, 2024. Citado na página 24.

LIU, H.; LI, C.; LI, Y.; LI, B.; ZHANG, Y.; SHEN, S.; LEE, Y. J. *Llavanext: Improved reasoning, ocr, and world knowledge*. 2024. Citado na página 25.

MACQUEEN, J. Multivariate observations. In: *Proceedings ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability*. [S.l.: s.n.], 1967. v. 1, p. 281–297. Citado na página 17.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 15.

Meta AI. *Llama 3: Open Foundation Models*. 2024. <ai.meta.com>. Acesso em: 10 dez. 2025. Citado na página 18.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Citado na página 19.

NIELSEN, M. A. *Neural networks and deep learning*. [S.l.]: Determination press San Francisco, CA, USA, 2015. v. 25. Citado na página 16.

OORD, A. v. d.; LI, Y.; VINYALS, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. Citado 2 vezes nas páginas 19 e 21.

OpenAI. *ChatGPT*. 2025. Versão GPT-4o, Modelo de linguagem IA. Acesso em: 26 nov. 2025. Disponível em: <https://openai.com/>. Citado 2 vezes nas páginas 18 e 25.

RADFORD, A.; KIM, J. W.; HALLACY, C.; RAMESH, A.; GOH, G.; AGARWAL, S.; SASTRY, G.; ASKELL, A.; MISHKIN, P.; CLARK, J. et al. Learning transferable visual models from natural language supervision. In: PMLR. *International conference on machine learning*. [S.l.], 2021. p. 8748–8763. Citado 2 vezes nas páginas 19 e 25.

RGUIBI, Z.; HAJAMI, A.; ZITOUNI, D.; ELQARAOUI, A.; BEDRAOUI, A. Cxai: Explaining convolutional neural networks for medical imaging diagnostic. *Electronics*, MDPI, v. 11, n. 11, p. 1775, 2022. Citado na página 18.

ROSENBLATT, F. et al. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. [S.l.]: Spartan books Washington, DC, 1962. v. 55. Citado na página 16.

ROUT, S.; DWIVEDI, V.; SRINIVASAN, B. Numerical approximation in cfd problems using physics informed machine learning. *arXiv preprint arXiv:2111.02987*, 2021. Citado na página 15.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986. Citado na página 17.

SANH, V.; DEBUT, L.; CHAUMOND, J.; WOLF, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. Citado na página 29.

SOHL-DICKSTEIN, J.; WEISS, E.; MAHESWARANATHAN, N.; GANGULI, S. Deep unsupervised learning using nonequilibrium thermodynamics. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 2256–2265. Citado na página 19.

TERASHITA, N.; TOZAKI, Y.; OMOTE, H.; NGUYEN, C.; NAKAMOTO, R.; KOREEDA, Y.; OZAKI, H. Can visual encoder learn to see arrows? *arXiv preprint arXiv:2505.19944*, 2025. Citado 2 vezes nas páginas 13 e 25.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. Citado na página 18.

VINYALS, O.; TOSHEV, A.; BENGIO, S.; ERHAN, D. Show and tell: A neural image caption generator. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3156–3164. Citado na página 19.

WANG, J.; YANG, H.; WU, J.; HE, Y.; WEI, X.; WANG, Y.; LIU, C.; GE, L.; WU, L.; WANG, B. et al. Gtr-cot: Graph traversal as visual chain of thought for molecular structure recognition. *arXiv preprint arXiv:2506.07553*, 2025. Citado na página 23.

WANG, K.; PAN, J.; SHI, W.; LU, Z.; REN, H.; ZHOU, A.; ZHAN, M.; LI, H. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, v. 37, p. 95095–95169, 2024. Citado na página 23.

WEI, Y.; FU, S.; JIANG, W.; ZHANG, Z.; ZENG, Z.; WU, Q.; KWOK, J.; ZHANG, Y. Gita: Graph to visual and textual integration for vision-language graph reasoning. *Advances in Neural Information Processing Systems*, v. 37, p. 44–72, 2024. Citado na página 24.

ZHU, Y.; BAI, X.; CHEN, K.; XIANG, Y.; YU, J.; ZHANG, M. Benchmarking and improving large vision-language models for fundamental visual graph understanding and reasoning. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [S.l.: s.n.], 2025. p. 30678–30701. Citado 2 vezes nas páginas 11 e 24.