

MINISTÉRIO DA EDUCAÇÃO
UTFPR - UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA

WILLIAM SDAYLE MARINS SILVA

DISSERTAÇÃO DE MESTRADO EM BIOINFORMÁTICA

**MELHORIAS NA CLASSIFICAÇÃO DE IMAGENS CONTEXTUAIS ATRAVÉS DE
GRAFOS CONVOLUCIONAIS E CAMINHADA ALEATÓRIA.**

CORNÉLIO PROCÓPIO

2023

WILLIAM SDAYLE MARINS SILVA

**MELHORIAS NA CLASSIFICAÇÃO DE IMAGENS CONTEXTUAIS ATRAVÉS DE
GRAFOS CONVOLUCIONAIS E CAMINHADA ALEATÓRIA.**

**CONTEXTUAL IMAGE CLASSIFICATION THROUGH GRAPH CONVOLUTIONAL
NETWORKS AND RANDOM WALKS.**

Dissertação apresentada como requisito para obtenção do
título de Mestre em Bioinformática pela Universidade
Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Prof. Dr. Pedro Henrique Bugatti

Coorientador(a): Profa. Dra. Priscila Tiemi Maeda Saito

CORNÉLIO PROCÓPIO

2023



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio



WILLIAM SDAYLE MARINS SILVA

MELHORIAS NA CLASSIFICAÇÃO DE IMAGENS CONTEXTUAIS ATRAVÉS DE GRAFOS CONVOLUCIONAIS E CAMINHADA ALEATÓRIA

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Bioinformática da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Bioinformática.

Data de aprovação: 15 de Março de 2023

Dr. Pedro Henrique Bugatti, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Andre Yoshiaki Kashiwabara, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Fabricio Martins Lopes, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Marcelo Ponciano Da Silva, Doutorado - Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

Dra. Priscila Tiemi Maeda Saito, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 15/03/2023.

Dedico este trabalho a minha família e a
todos os meus amigos.

AGRADECIMENTOS

Gostaria de agradecer à Deus por me capacitar e me encorajar a realizar tal projeto. Agradeço aos meus pais, irmãos e namorada por todo amor, carinho e ajuda nos momentos difíceis da pesquisa e no decorrer do programa. Meus mais sinceros agradecimentos a meu orientador Prof. Dr. Pedro Henrique Bugatti e minha coorientadora Profa. Dra. Priscila Tiemi Maeda Saito, por todos os ensinamentos, momentos disponibilizados para auxílio, tempo investido e confiança depositada. Gostaria também de agradecer a UTFPR pela estrutura de ensino oferecida em meio ao momento de turbulência causado pela pandemia do novo coronavírus. Meus sinceros agradecimentos a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo financiamento e confiança em minha pesquisa.

Nunca diga nunca, pois os limites são
como os medos: sempre são apenas
ilusão. (Michael Jordan)

RESUMO

SILVA, Sdayle Marins. **MELHORIAS NA CLASSIFICAÇÃO DE IMAGENS CONTEXTUAIS ATRAVÉS DE GRAFOS CONVOLUCIONAIS E CAMINHADA ALEATÓRIA.** 2023. 114 f. Dissertação (Mestrado em Bioinformática) – Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2023.

Sabe-se que pessoas conseguem assimilar o contexto entre informações a partir de dados pouco percebidos, e essa afirmação se aplica ao contexto em imagens. No aprendizado de máquina, o contexto de uma imagem pode ser representado e previsto usando *Graph Convolutional Networks* (GCNs). Nesse contexto, as redes convolucionais em grafos precisam da estrutura do grafo para aplicar o método convolucional entre os nós. Com base nos trabalhos anteriores, grafos convolucionais com políticas de conexão totalmente conectadas e por cortes com base em pesos são políticas que podem ser usadas para criar gráficos de contexto de imagem. No entanto, as políticas de conexão entre objetos geralmente utilizadas na criação de grafos convolucionais podem não ser as melhores em termos de classificação ou número de arestas, o que pode confundir os modelos de classificação. Com o objetivo de tornar os grafos de contexto mais expressivos e melhorar a precisão das classificações de imagens contextuais, este trabalho propõe um método eficiente baseado em passeios aleatórios que podem obter melhores resultados de classificação em comparação com a literatura. Para aumentar o nível de comparação do método proposto, foram utilizados outros dois métodos aleatórios; corte aleatório e arestas aleatórias. Algumas arquiteturas de redes neurais pré-treinadas foram usadas para extrair características de cada objeto das imagens. Diferentes tamanhos de passeios também foram testados para o passeio aleatório, tornando o projeto mais abrangente. Foram utilizados diferentes contextos de rótulos em quatro conjuntos de imagens para avaliar as políticas de conexão de arestas do grafo. A abordagem proposta apresentou melhores resultados em quase todos os conjuntos e conjuntos de arestas menores do que a política totalmente conectada em todos os datasets. O método proposto performou 56,20% de precisão em comparação com 50,37% da política totalmente conectada e 53,85% do método por cortes com base em pesos.

Palavras-chave: Redes Neurais Convolucionais. Grafos Convolucionais. Caminhada Aleatória.

ABSTRACT

SILVA, Sdayle Marins. **Contextual image classification through Graph Convolutional Networks and random walks..** 2023. 114 p. Dissertation (Master's Degree in Bioinformatic) – Federal University of Technology – Paraná. Cornélio Prociópio, 2023.

Humans can manage to assimilate the context between information from little perceived data, and this statement applies to the context in images. In machine learning, an image's context can be represented and predicted using Graph Convolutional Networks (GCNs). In this context, graph convolutional networks need the graph structure to apply the convolutional method between nodes. Based on previous work, fully connected and threshold weights are edge policies that can be used to create image context graphs. However, the connection policies between objects generally used in the creation of convolutional graphs may not be the best in terms of classification or number of edges, which may confuse the classification models. Aiming to make context graphs more expressive and improve the accuracy of contextual image classifications, this paper proposes an efficient method based in random walks that can get better classification results compared to the literature. To increase the level of comparison of the proposed method, two other random methods were used; random cut and random edge. Some pre-trained neural network architectures were used to extract features of each object from the images. Different sizes of walks were also tested for the random walk, making the project more comprehensive. Were used different label contexts in four datasets to evaluate the graph edge connection policies. The proposed approach could perform better results in almost all the datasets and having edge sets smaller than the fully connected policy in all datasets. The proposed method could perform 56.20% of precision compared to 50.37% of the fully connected policie and 53.85% of the threshold weights method.

Keywords: Convolutional Neural Networks. Graph Convolutional Networks. Random Walks.

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo totalmente conectado	56
Algoritmo 2 – Algoritmo de corte aleatório	57
Algoritmo 3 – Algoritmo de arestas aleatórias	59
Algoritmo 4 – Algoritmo de poda para limiar T	60
Algoritmo 5 – Algoritmo de caminhada aleatória	61

LISTA DE ILUSTRAÇÕES

Figura 1 – Como grafos podem representar o contexto de uma imagem, imagem retirada da base Mit67 (Quattoni; Torralba, 2009).	18
Figura 2 – Ilustração do cenário em que conectar todos os objetos da imagem em um grafo totalmente conectado pode não ser vantajoso.	19
Figura 3 – Como imagens são processadas computacionalmente, inspirado em (GONZALEZ, 2009).	22
Figura 4 – Como imagens são interpretadas computacionalmente. a) exemplo de uma imagem de profundidade de 1 bit, nesse contexto, essa imagem poderia assumir valores entre 1 e 0, representando branco e preto respectivamente, b) exemplo de uma imagem em tons de cinza e como suas intensidades podem variar mais dentro do espectro, em relação a uma imagem binária, c) exemplo de uma imagem no espaço de cor RGB e como suas intensidades podem variar dentro de cada plano de 8 bits.	24
Figura 5 – Ilustração do neurônio biológico, extraído de (SILVA DANILO HERNANE SPATTI, 2019) e inspirado em (MACHADO, 2013).	24
Figura 6 – Neurônio artificial proposto por Frank Rosenblatt em (ROSENBLATT, 1958).	26
Figura 7 – Estrutura da rede multicamadas perceptron (<i>Multi Layer Perceptron</i>).	27
Figura 8 – Ilustração do treinamento de uma rede neural. Em setas pretas estão as comunicações realizadas entre os neurônios no <i>feedforward</i> e em setas pontilhadas azul estão as comunicações feitas entre os neurônios no <i>back propagation</i> , baseado em (BENGIO IAN GOODFELLOW, 2015).	29
Figura 9 – Exemplo de funcionamento do <i>Max Pooling</i> , retirado de (KRIZHEVSKY; SUTSKEVER; HINTON, 2017).	32
Figura 10 – Transferência de aprendizado em modelos de <i>machine learning</i> , extraído de (MARI et al., 2020).	33
Figura 11 – Diferentes topologias de grafos, baseado em (NETTO, 2012) e (CHARTRAND; ZHANG, 2019).	35
Figura 12 – Matriz de adjacência e seu grafo. Em (a) está o grafo G e em (b) a matriz de adjacência X	36
Figura 13 – Diferentes abordagens para redes neurais em grafos, extraído de (CASALEGNO, 2021).	38
Figura 14 – Comparação da convolução entre CNN e GCN, em (a) está a convolução para a rede neural convolucional tradicional (CNN) e em (b) está a convolução para a rede neural convolucional em grafos, imagem extraída de (WU et al., 2020).	40
Figura 15 – Caminhada aleatória no eixo x , extraído de (ZHOU; XU; DENG, 2020).	42
Figura 16 – Comunidades destacadas de um grafo G , inspirado em (Li et al., 2015). O grafo G é toda a área em azul, e as comunidades estão destacadas entre C	45
Figura 17 – Exemplo de aplicação da matriz de confusão.	47
Figura 18 – Metodologia proposta para o trabalho	53

Figura 19 – Processo de extração de características de objetos de imagens. Figura da base de imagens UnRel, da classe <i>Bike and Person</i>	55
Figura 20 – Grafo totalmente conectado e sua matriz de adjacência.	57
Figura 21 – Grafo com poda aleatória e sua matriz de adjacência.	58
Figura 22 – Grafo com criação de arestas aleatórias e sua matriz de adjacência.	59
Figura 23 – Grafo com poda a partir do limiar T e sua matriz de adjacência.	61
Figura 24 – Grafo com método de caminhada aleatória e sua matriz de adjacência.	62
Figura 25 – Pipeline de treinamento de rede neural convolucional tradicional.	64
Figura 26 – Pipeline de treinamento de rede neural em grafos convolucionais.	65
Figura 27 – Ilustração do pipeline de processamento das imagens e como os vetores de características serão salvos em cada classe. Na figura, cada caixa simula uma classe da base de imagens UnRel.	67
Figura 28 – Distribuição das imagens e como os sub-grafos G' são controlados em G . Neste exemplo é ilustrado como o grafo G , que contém N vértices e E arestas, é construído através do somatório de G'	68
Figura 29 – Exemplo de <i>K-Fold cross validation</i>	69
Figura 30 – Exemplo de como os objetos presentes constituem o rótulo da imagem. Esta Figura é da base de imagem Mit67, da classe <i>bedroom</i>	72
Figura 31 – Anotação dos objetos no dataset UnRel, a imagem ilustrada é da classe <i>dog wear tie</i>	73
Figura 32 – Exemplo de imagem do dataset Mit67, a classe da imagem é <i>computer room</i>	74
Figura 33 – As 6 variações de células presentes no dataset Malaria. Em verde as células não infectadas e em laranja as células infectadas.	76
Figura 34 – Exemplo de imagem médica de células humanas, de acordo com (LJOSA; SOKOLNICKI; CARPENTER, 2012) a imagem contém células denominadas de gametócitos, o que faria do conjunto de células presentes na imagem infectado.	76
Figura 35 – Exemplo de relacionamento entre os objetos para o dataset VRD, a classe da imagem é <i>near</i>	78
Figura 36 – Correlação entre quantidade de arestas do grafo e sua respectiva precisão para GCN utilizando extrator ResNet50.	81
Figura 37 – Matriz de confusão da topologia totalmente conectada.	83
Figura 38 – Matriz de confusão da topologia de caminhada aleatória (GCN RW).	84
Figura 39 – Classificações de imagens para cada topologia aplicada. Em (A) está a Figura da classe <i>bike-and-person</i> , em (B) da classe <i>dog-wear-hat</i> , em (C) da classe <i>elephant-wear-pants</i> e em (D) da classe <i>person-inside-tree</i>	85
Figura 40 – Correlação entre quantidade de arestas do grafo e sua respectiva precisão para GCN utilizando extrator VGG16.	88
Figura 41 – Matriz de confusão da topologia totalmente conectada.	89
Figura 42 – Matriz de confusão da topologia de caminhada aleatória.	90
Figura 43 – Classificações de imagens para cada topologia aplicada. Em (A) está a Figura da classe <i>airport-inside</i> , em (B) da classe <i>church-inside</i> , em (C) da classe <i>bathroom</i> e em (D) da classe <i>office</i>	91
Figura 44 – Correlação entre quantidade de arestas do grafo e sua respectiva precisão para GCN utilizando extrator InceptionResNetV2.	95
Figura 45 – Matriz de confusão para a topologia totalmente conectada.	96

Figura 46 – Matriz de confusão para a topologia de caminhada aleatória.	97
Figura 47 – Classificações de imagens para cada topologia aplicada.	98
Figura 48 – Correlação entre precisão e número de arestas.	101
Figura 49 – Matriz de confusão da topologia totalmente conectada.	102
Figura 50 – Matriz de confusão da topologia de caminhada aleatória.	103
Figura 51 – Classificações de imagens para cada topologia aplicada. Em (A) está a Figura da classe <i>above</i> , em (B) da classe <i>behind</i> , em (C) da classe <i>has</i> e em (D) da classe <i>next-to</i>	104

LISTA DE TABELAS

Tabela 1 – Diferentes tipos de abstração de relações entre indivíduos através de grafos, extraído de (NETTO, 2012).	34
Tabela 2 – Cara ou coroa descrevendo os pontos em que o caminhador aleatório caminhará, extraído de (ZHOU; XU; DENG, 2020).	42
Tabela 3 – Descrição do deslocamento do caminhador aleatório.	42
Tabela 4 – Comparação entre os trabalhos correlatos e o projeto proposto. . .	49
Tabela 5 – Tamanho dos vetores de características de acordo com as arquiteturas escolhidas.	54
Tabela 6 – Ambientes e a distribuição das imagens para cada ambiente na base Mit67.	74
Tabela 7 – Quantidade de imagens para cada classe na base Malária.	75
Tabela 8 – Quantidade de imagens para cada classe na base VRD.	77
Tabela 9 – Resultados globais obtidos na base de imagens UnRel.	80
Tabela 10 – Tempo para construção dos grafos de cada politica de conexão. . .	81
Tabela 11 – Resultados globais obtidos na base de imagens MIT67.	87
Tabela 12 – Tempo para construção dos grafos de cada politica de conexão. . .	88
Tabela 13 – Resultados globais obtidos na base de imagens Malária.	93
Tabela 14 – Tempo para construção dos grafos de cada politica de conexão. . .	94
Tabela 15 – Resultados globais obtidos na base de imagens VRD.	99
Tabela 16 – Tempo para construção dos grafos de cada politica de conexão. . .	100

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNNs	Redes Neurais Convolucionais (<i>Convolutional Neural Networks</i>)
GCNs	Grafos Convolucionais (<i>Graph Convolutional Networks</i>)
RC	Poda Aleatória (<i>Random Cut</i>)
RE	Arestas Aleatórias (<i>Random Edge</i>)
RW	Caminhada Aleatória (<i>Random Walk</i>)
RWE	Corte por pesos (<i>Random Weighted</i>)

SUMÁRIO

1	INTRODUÇÃO	17
1.1	FORMULAÇÃO DO PROBLEMA	18
1.2	MOTIVAÇÃO	19
1.3	OBJETIVOS	20
1.3.1	Objetivo Geral	20
1.3.2	Objetivos Específicos	20
1.4	ORGANIZAÇÃO	20
2	REVISÃO BIBLIOGRÁFICA	22
2.1	PROCESSAMENTO E VISUALIZAÇÃO DE IMAGENS DIGITAIS	22
2.2	REDES NEURAIS ARTIFICIAIS(<i>ARTIFICIAL NEURAL NETWORKS</i>)	23
2.2.1	Neurônio biológico	23
2.2.2	<i>Perceptron</i>	25
2.2.3	<i>Multi Layer Perceptron</i>	26
2.3	REDES NEURAIS CONVOLUCIONAIS (<i>CONVOLUTIONAL NEURAL NETWORKS</i>)	30
2.3.1	Convolução	30
2.3.2	Pooling	31
2.3.3	Transferência de aprendizado	32
2.4	GRAFOS	33
2.5	GRAFOS CONVOLUCIONAIS (<i>GCN</i>)	36
2.6	CAMINHADA ALEATÓRIA - <i>RANDOM WALK</i>	41
2.7	MÉTRICAS DE AVALIAÇÃO	44
2.8	TRABALHOS RELACIONADOS	46
3	METODOLOGIA	53
3.1	EXTRAÇÃO DE CARACTERÍSTICAS	53
3.2	CRIAÇÃO DOS GRAFOS	55
3.2.1	Grafo totalmente conectado - (FC)	56
3.2.2	Grafo de poda aleatória - (RC)	56
3.2.3	Grafo com criação de arestas aleatórias - (RE)	58
3.2.4	Grafo com poda a partir de limiar T - (RWE)	59
3.2.5	Grafo de caminhada aleatória - (RW)	60
3.3	MODELAGEM DAS REDES NEURAIS	62
4	DESENVOLVIMENTO	66
4.1	PROCESSAMENTO DAS IMAGENS	66
4.2	DISTRIBUIÇÃO DAS IMAGENS E CRIAÇÃO DAS ESTRUTURAS	67
4.3	EXPERIMENTOS	69
4.3.1	Bases de imagens	71
4.3.1.1	UnRel	72
4.3.1.2	Mit67	73
4.3.1.3	Malaria	75
4.3.1.4	VRD	77

4.3.2	Resultados	78
4.3.2.1	UnRel	79
4.3.2.2	Mit67	86
4.3.2.3	Malária	92
4.3.2.4	VRD	98
5	CONCLUSÃO	105
	REFERÊNCIAS	107

1 INTRODUÇÃO

Sabe-se que a palavra contexto está fortemente ligada à semântica ou informações implícitas ou explícitas. Em geral, contexto pode ser definido como o sentido por de trás de um determinado texto, ou semântica de uma oração (RIBEIRO; MARTINS; ANTUNES, 2017). Por exemplo, na frase: “que ótimo!”, não está implícito ou explícito o que de fato seja ótimo. Já na frase: “acabei de ser promovido, que ótimo!”, nota-se o sentimento de euforia ou otimismo, devido ao contexto empregado na frase.

Em imagens, contextos podem ser identificados por meio de texturas, formas e informações temporais semelhantes (KOSTI et al., 2020) (MASTAN; RAMAN, 2020). Para (GONZALEZ, 2009), imagens podem ser computacionalmente processadas através da estrutura de matriz, em que cada célula indicará uma intensidade de cor, podendo variar sua profundidade de acordo com a estrutura de coloração utilizada.

Considera-se que, Redes Neurais Convolucionais (*Convolutional Neural Networks*) (CNNs) são uma boa alternativa para solucionar problemas de visão computacional, tais como, reconhecimento de objetos em imagens (ALOM et al., 2017), (ZANGENEH; RAHMATI; MOHSENZADEH, 2020a), também podendo ser utilizada para classificação de objetos em imagens (ZANGENEH; RAHMATI; MOHSENZADEH, 2020b), (ABBAS; ABDELSAMEA; GABER, 2020). Entretanto, quando as imagens são rotuladas pelo contexto nelas apresentado, CNNs podem não obter grande sucesso, devido a sua estrutura de extração de características (BUGATTI; SAITO; DAVIS, 2019), (LI et al., 2020).

Buscando solucionar o problema de classificação dos rótulos contextuais em imagens, as estruturas de Grafos Convolucionais (*Graph Convolutional Networks*) (GCNs) podem ser utilizadas, visando interligar informações entre os objetos, para melhorar a representação do contexto das imagens (CHEN; LIN; CHO, 2017), (CHEN et al., 2019). O contexto que visa representar o rótulo de uma imagem é apresentado na Figura 1, em que é criada a estrutura hierárquica de grafos para representar a composição semântica de “Jogo de Baseball”.

Entretanto, algumas estruturas empregadas para representação das imagens podem ser computacionalmente custosas devido à política de conexão dos objetos de cada imagem (BUGATTI; SAITO; DAVIS, 2019), (DUVENAUD et al., 2015). Portanto,

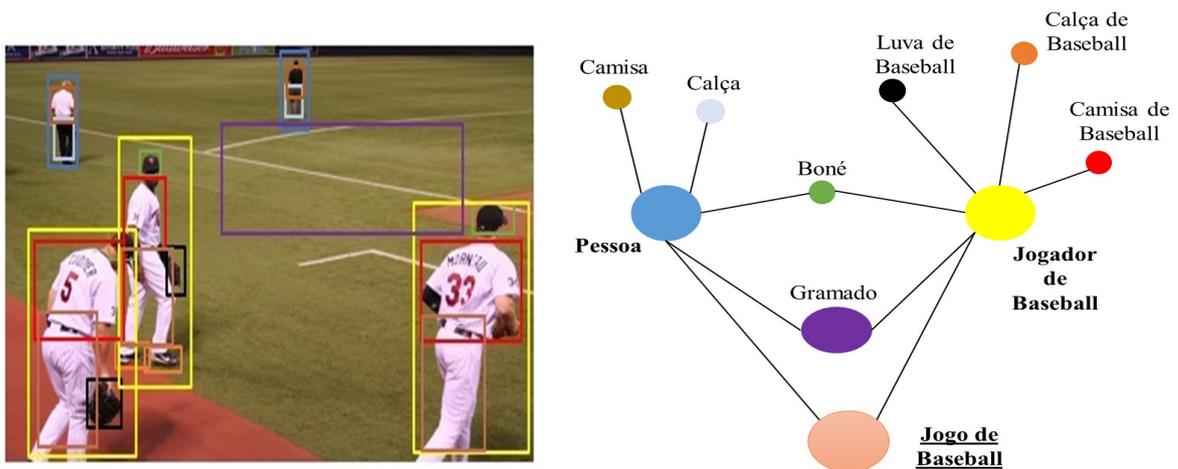


Figura 1 – Como grafos podem representar o contexto de uma imagem, imagem retirada da base Mit67 (Quattoni; Torralba, 2009).

esse projeto visa encontrar conexões adequadas entre os objetos, para que, dessa forma, a representação seja eficaz, computacionalmente vantajosa e obter resultados equiparáveis ou melhores do que o estado da arte.

1.1 FORMULAÇÃO DO PROBLEMA

Seguindo a topologia apresentada em (BUGATTI; SAITO; DAVIS, 2019), avaliada utilizando a estrutura de GCNs exposto em (KIPF; WELLING, 2016), as conexões empregadas poderiam ser mais descritivas para o contexto geral da imagem. A estrutura proposta em (BUGATTI; SAITO; DAVIS, 2019), considera que todos os objetos necessitam estar conectados para que o contexto seja representado. Para ilustrar como a política de conexão apresentada em (BUGATTI; SAITO; DAVIS, 2019) funciona, apresenta-se a Figura 2, que ilustra uma imagem da classe “clínica dentária” da base de imagens Mit67 (Quattoni; Torralba, 2009).

Percebe-se que, alguns objetos presentes na imagem realmente poderiam ser utilizados para definir o contexto de clínica dentária. Porém, alguns objetos fora do contexto da classe podem atrapalhar modelos de redes neurais a fazerem classificações adequadas, conforme apresentado nos resultados de (BUGATTI; SAITO; DAVIS, 2019) e (CAMPOS et al., 2021). As Figuras 2(a) e 2(b), apresentam a imagem e seus respectivos objetos. Em (b) estão destacados os objetos que fazem parte do contexto e os que não fazer parte.

Conforme apresentado na Figura 2, caso seja utilizada a política apresentada

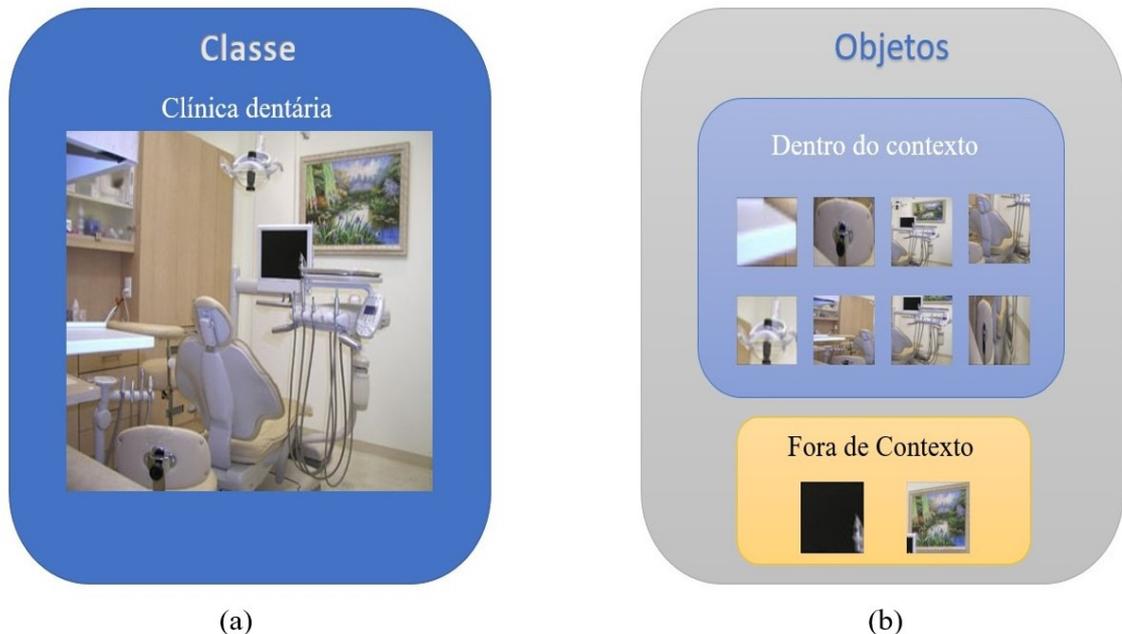


Figura 2 – Ilustração do cenário em que conectar todos os objetos da imagem em um grafo totalmente conectado pode não ser vantajoso.

em (BUGATTI; SAITO; DAVIS, 2019), estaria sendo considerado que quadros e monitores necessitam estar conectados a cadeiras e objetos odontológicos para que o contexto geral seja detectado.

Outro ponto a ser analisado é o custo computacional que grafos com grande quantidade de arestas pode ocasionar em seu processamento, aumentando seu tempo de construção e complexidade de processamento, e também, podendo resultar em classificações pouco vantajosas.

Portanto, há dois pontos a serem analisados como possíveis problemas, sendo eles: a qualidade da classificação das imagens e o custo computacional causado pela alta quantidade de arestas do grafo. O presente trabalho, terá seu foco principal em melhorar a qualidade da classificação das imagens.

1.2 MOTIVAÇÃO

Dentro do campo matemático, grafos podem abstrair desde problemas simples até mais complexos, havendo a possibilidade de ser aplicado em diversas abordagens. Sabe-se que o método de caminhada aleatória pode ser empregado na estrutura de grafo, devido ao seu formato de abstração (vértices e arestas).

Desta maneira, o grafo é uma estrutura ideal para representar conexões entre

objetos, simuladas através de vértices (objetos) e arestas (conexões). Visando melhorar a qualidade das classificações, pode-se assumir que com conexões mais descritivas, adequadas e em menor quantidade, os resultados sejam melhores ou equivalentes.

Levando em consideração o que foi descrito, o presente trabalho utiliza da alta aplicabilidade dos grafos em abstração de informações, dentre elas o contexto de imagens almejando bons resultados apresentados por grafos convolucionais em classificações de imagens contextuais apresentadas em (BUGATTI; SAITO; DAVIS, 2019) e (CAMPOS et al., 2021).

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Utilizar caminhada aleatória para aumentar ou tornar equivalente a classificação de imagens por meio de grafos convolucionais em imagens contextuais.

1.3.2 Objetivos Específicos

- Reduzir a quantidade de arestas dos grafos em relação ao modelo totalmente conectado;
- Comparar os resultados da precisão entre os métodos de grafos convolucionais: totalmente conectado, passeio aleatório, poda aleatória, arestas aleatórias, poda aleatória por pesos (limiar T) e redes neurais convolucionais tradicionais;
- Alcançar melhores resultados em classificações de imagens contextuais.

1.4 ORGANIZAÇÃO

O presente trabalho está organizado nos seguintes capítulos:

- No capítulo 2 serão discutidos conceitos e o estado da arte para o presente trabalho, tais como: processamento de imagens, redes neurais e redes neurais convolucionais (CNNs), grafos, redes neurais convolucionais em grafos (GCNs), caminhada aleatória Caminhada Aleatória (*Random Walk*) (RW) e as métricas

que podem ser utilizadas para avaliar os resultados de cada política. Ao fim do capítulo, serão abordados trabalhos que propuseram soluções na mesma linha de pesquisa que o presente projeto.

- No capítulo 3 será abordado a metodologia implementada no projeto, através dos tópicos: extração de características, criação dos grafos e modelagem das redes neurais.
- No capítulo 4 serão detalhados os procedimentos realizados para desenvolver o projeto. Processamento das imagens, distribuição e criação dos conjuntos de treino e teste e os experimentos executados. Em experimentos estarão todos os resultados alcançados pelo presente trabalho.
- No capítulo 5 é apresentada a conclusão do trabalho e possíveis melhorias futuras.

2 REVISÃO BIBLIOGRÁFICA

Nesta seção, serão apresentados os conteúdos teóricos utilizados como base para construção do método proposto. Havendo informações para os assuntos de processamento de imagens, inteligência artificial, mais especificamente em aprendizado de máquina e aprendizado profundo. Posteriormente, serão abordados assuntos relacionados à estrutura de grafos e será exposto o método de caminhada aleatória com foco na aplicação em grafos, e por fim, alguns trabalhos relacionados ao tema proposto.

2.1 PROCESSAMENTO E VISUALIZAÇÃO DE IMAGENS DIGITAIS

Computacionalmente, uma imagem pode ser processada através da estrutura de matriz, em que cada elemento da matriz é denominado de *pixel* (*Picture Element*). Para (GONZALEZ, 2009), cada *pixel* informa a intensidade de luz que será processada para uma determinada célula da matriz, ou seja, cada *pixel*, informará cores, texturas, profundidades, entre outras informações relacionadas à imagem. Para (DOUGHERTY, 2020), uma imagem pode ser definida pela função $f(m, n)$ em que m é o número de linhas da matriz e n o número de colunas. Pode ser afirmado que $M * N$ informa a resolução da imagem em quantidade de *pixels*, conforme a Figura 3.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix}$$

Figura 3 – Como imagens são processadas computacionalmente, inspirado em (GONZALEZ, 2009).

Para (DOUGHERTY, 2020), em geral, podem haver 3 tipos de imagens: binárias, tons de cinza e coloridas. Essas afirmações estão relacionadas à profundidade da imagem e sua estrutura. Uma imagem binária pode assumir apenas duas intensidades em seus pixels $[0, 1]$, sendo 0 a intensidade mais escura e 1 a intensidade mais clara. Tal imagem pode ser definida como imagem de profundidade de 1 bit (GONZALEZ, 2009). Uma imagem em tons de cinza pode variar um pouco mais no espectro, podendo assumir intensidades entre $[0-255]$, formando uma variedade de 256 possíveis cores para cada *pixel*. Essas imagens, são caracterizadas por possuírem profundidade de 8 bits.

Em (DOUGHERTY, 2020) e (GONZALEZ, 2009), é destacado o espaço de cor RGB para colorização de imagens. Para essa estrutura, a imagem terá em cada pixel uma nova dimensão, denominada de canal de cor. Para (GONZALEZ, 2009), o termo *full color* ou imagem colorida é utilizado para expressar uma imagem de cores *RGB* de 24 bits, em que, o número de cores possíveis nessa estrutura seja de $(2^8)^3 = 16.777.216$. Nesse formato, a imagem pode ser representada pela função $f(m, n, z)$, sendo z a intensidade para uma determinada cor do canal. Essa estrutura funciona combinando intensidades para vermelho, verde e azul até formarem uma intensidade única. A Figura 4 ilustra exemplos dos 3 tipos de imagens, a) binária, b) em tons de cinza e c) colorida (RGB).

2.2 REDES NEURAIAS ARTIFICIAIS(*ARTIFICIAL NEURAL NETWORKS*)

Redes neurais artificiais são algoritmos computacionais que se modelam através de funções matemáticas, tendo como exemplo o próprio neurônio de organismos inteligentes e que adquirem conhecimento através da experiência (SILVA DANILO HERNANE SPATTI, 2019). De acordo com (MEHLIG, 2019), redes neurais podem conter dezenas ou milhares de neurônios artificiais.

2.2.1 Neurônio biológico

Para (MACHADO, 2013), o sistema nervoso é composto por um conjunto complexo de células, denominadas de neurônios, sendo eles os principais articuladores para o funcionamento e comportamento do raciocínio humano. Os neurônios são for-

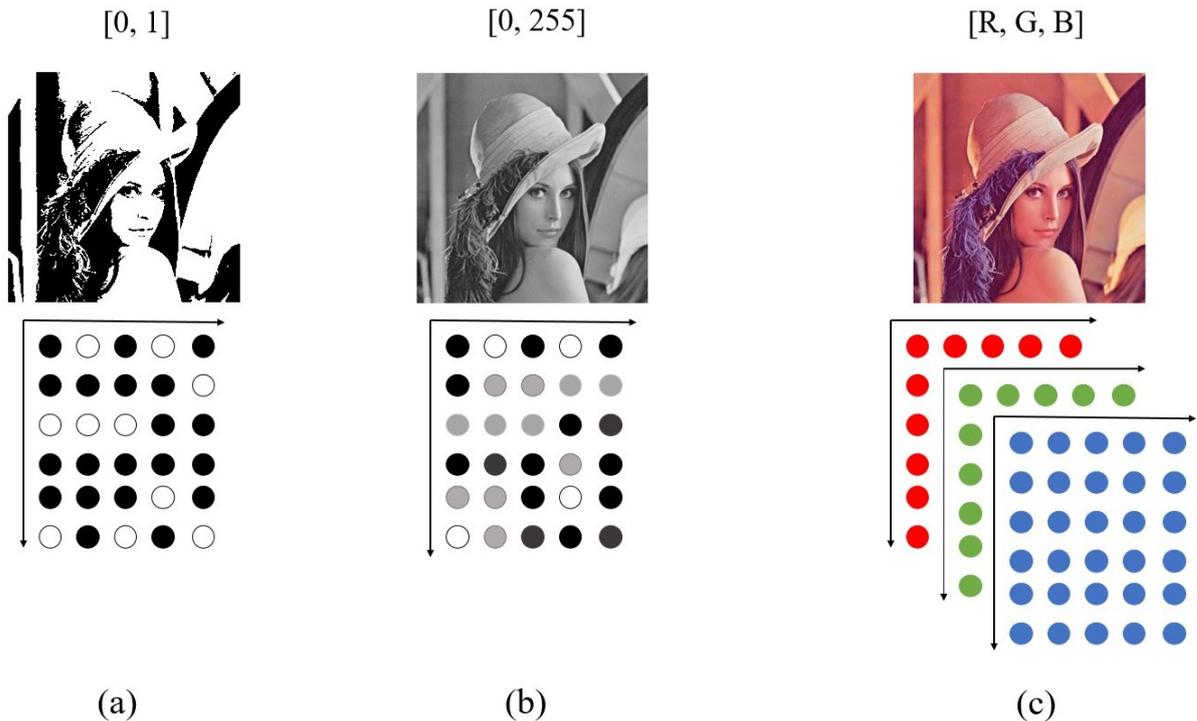


Figura 4 – Como imagens são interpretadas computacionalmente. a) exemplo de uma imagem de profundidade de 1 bit, nesse contexto, essa imagem poderia assumir valores entre 1 e 0, representando branco e preto respectivamente, b) exemplo de uma imagem em tons de cinza e como suas intensidades podem variar mais dentro do espectro, em relação a uma imagem binária, c) exemplo de uma imagem no espaço de cor RGB e como suas intensidades podem variar dentro de cada plano de 8 bits.

mados pelos dendritos, que são um conjunto de terminais de entrada, pelo corpo celular, núcleo, axônio e seus terminais. A Figura 5 ilustra o neurônio biológico, seus componentes e a direção em que o impulso nervoso é transmitido.

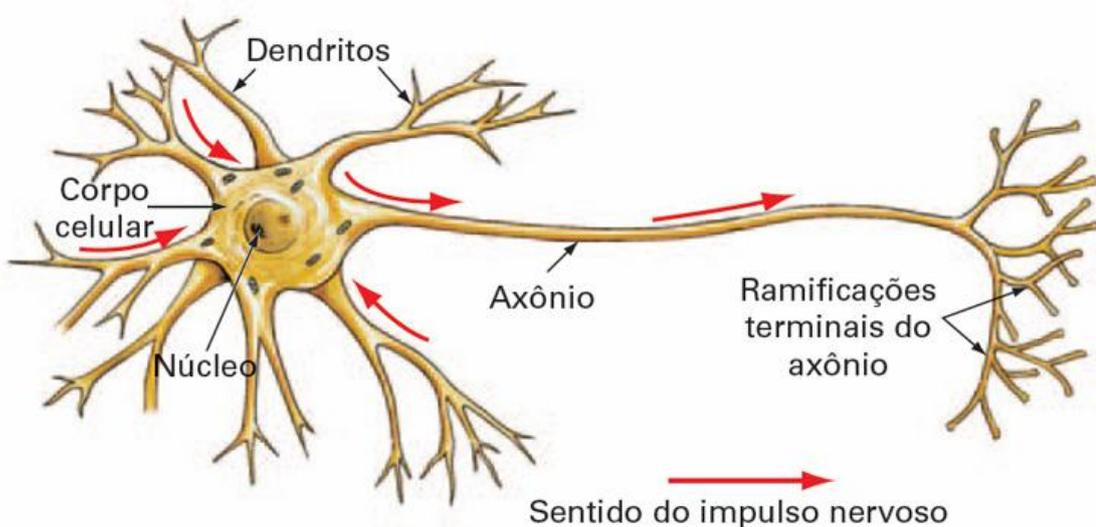


Figura 5 – Ilustração do neurônio biológico, extraído de (SILVA DANILO HERNANE SPATTI, 2019) e inspirado em (MACHADO, 2013).

O neurônio receberá sinais através de seus dendritos, onde serão ponderados

e enviados para o axônio, neste etapa o sinal pode ou não seguir em frente. Tendo em vista que, ao sinal passar pelo neurônio ele pode ser aumentado ou diminuído, variando de acordo com o dendrito de origem, pois cada condutor está ligado a um peso que alterará o sinal, os pesos podem ser denominados de memória (BENGIO IAN GOODFELLOW, 2015).

Visto que o neurônio é o componente principal para o raciocínio humano, pode-se seguir para o que seria o neurônio artificial, que também é o componente base para as redes neurais. O neurônio artificial pode ser denominado de *perceptron*.

2.2.2 *Perceptron*

O modelo perceptron foi desenvolvido pelo pesquisador Frank Rosenblatt em (ROSENBLATT, 1958), levando como base o trabalho de Warren McCulloch e Walter Pitts em (MCCULLOCH; PITTS, 1943). Pela sua facilidade da compreensão, o neurônio perceptron é o mais utilizado para conhecimento do funcionamento de uma rede neural em termos matemáticos. O neurônio perceptron, funciona basicamente tomando entradas $n_1, n_2 \dots n_i$ e gerando uma saída baseada nas entradas. Um neurônio perceptron, pode ser chamado de neurônio base artificial (MACHADO, 2013), a Figura 6, ilustra a estrutura de um neurônio artificial.

A saída resultante do processo realizado pelo neurônio perceptron se dá por funções matemáticas, onde cada entrada é impactada por seu respectivo peso w_i e a função $F(n)$ somará os resultados dos produtos entre as entradas e seus pesos. Um exemplo clássico do funcionamento do *perceptron* é a saída binária baseada em um limiar, como apresentado na Equação 1.

$$y = \begin{cases} 0, & \text{se } \sum_i w_i * n_i \leq \text{limiar} \\ 1, & \text{se } \sum_i w_i * n_i > \text{limiar} \end{cases} \quad (1)$$

Nota-se que o neurônio artificial perceptron isolado não consegue simular adequadamente o funcionamento do raciocínio humano. Na Equação 2, $w * x$ seria o coeficiente angular e b o coeficiente linear. Tendo em vista que a função do neurônio é similar a uma função de primeiro grau, ou equação da reta. Dessa maneira, pode não ser a solução ideal para problemas mais complexos.

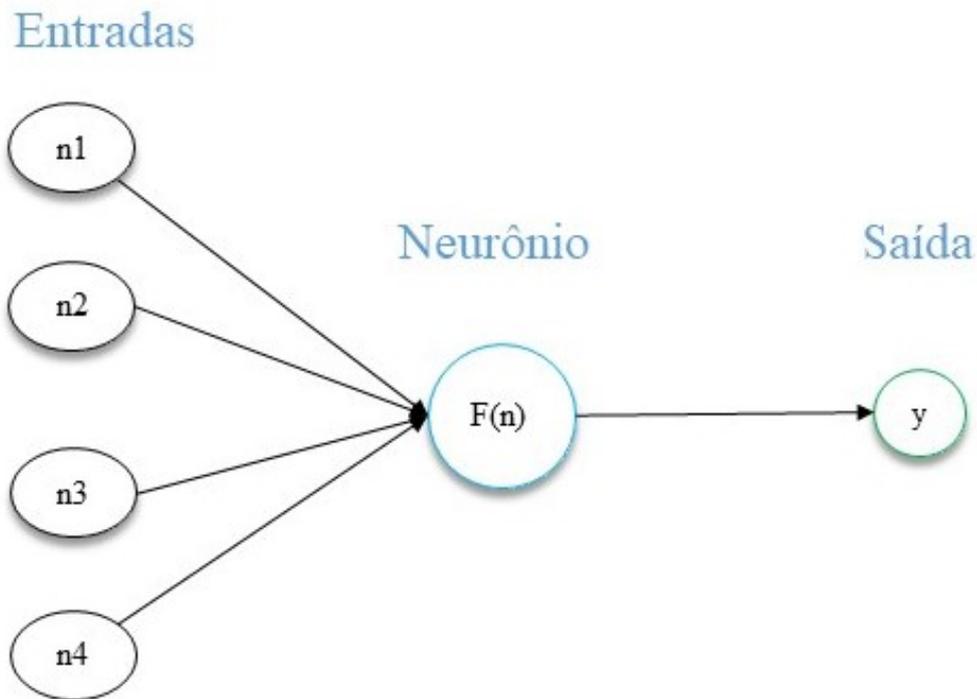


Figura 6 – Neurônio artificial proposto por Frank Rosenblatt em (ROSENBLATT, 1958).

$$y = w * x + b \quad (2)$$

2.2.3 Multi Layer Perceptron

Tendo como base o neurônio perceptron e buscando solucionar problemas mais complexos, (ROSENBLATT, 1958) também foi proposto uma estrutura de rede neural realmente similar ao funcionamento dos neurônios humanos, com trocas de informações, pesos sinápticos sendo perpetuados ou parados, entre outros processos matemáticos.

Uma rede perceptron multicamadas, utilizará de vários neurônios perceptron conectados em camadas transportando informações entre si. As camadas centralizadas na estrutura da rede, é denominada como camada oculta, podendo haver 1 ou k neurônios ligados em 1 ou n camadas. É dado o nome de camada oculta, pois não se pode prever com precisão o resultado obtido desta camada, funcionando quase como uma caixa preta (MITCHELL, 1997). A estrutura de uma rede multicamadas é apresentada na Figura 7.

Diferentemente de um único neurônio perceptron que gera uma saída y , a

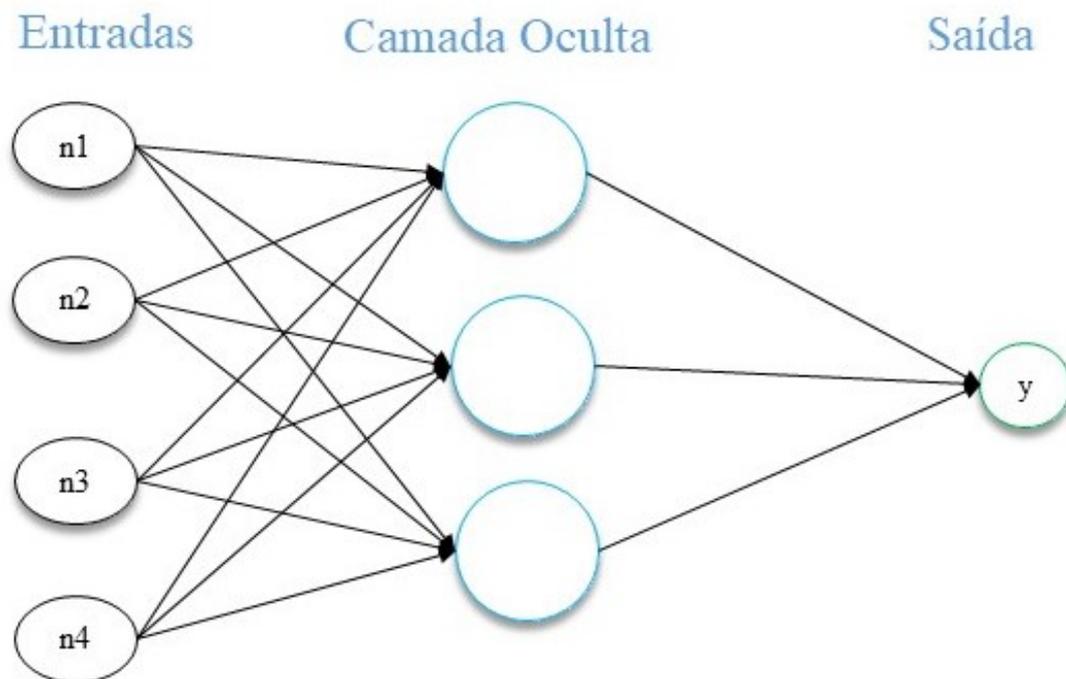


Figura 7 – Estrutura da rede multicamadas perceptron (*Multi Layer Perceptron*).

rede multicamadas pode relacionar vários neurônios para a saída. Para (BENGIO IAN GOODFELLOW, 2015), ativar todos os neurônios pode não ser uma boa opção em custo computacional e também em otimização de parâmetros. Portanto, para o funcionamento adequado da rede e para se aproximar ainda mais do funcionamento dos neurônios biológicos, pode-se adicionar uma função de ativação para o neurônio. As funções de ativação terão como premissa ativar neurônios com resultados adequados e desativar neurônios com resultados abaixo do esperado.

Como o trabalho proposto visa classificar imagens, dessa maneira serão apresentadas algumas funções de ativação voltadas para classificação. Para (BENGIO IAN GOODFELLOW, 2015), as funções de ativação podem variar de acordo com a proposta de classificação. De acordo com (MITCHELL, 1997), para classificações binárias pode ser utilizada a função de ativação binária e para classificações com mais classes podem ser utilizadas as funções *softmax* e a *sigmoid*. As funções citadas, atuariam na última camada da rede neural, buscando ativar os neurônios ideais para obter classificações adequadas. Para ativar os neurônios das camadas ocultas, pode ser utilizada a função *ReLU* (BENGIO IAN GOODFELLOW, 2015). As funções citadas são as mais utilizadas de acordo com (BENGIO IAN GOODFELLOW, 2015) e (MITCHELL, 1997). A função de ativação binária é apresentada na Equação 3, nota-se que

a equação é similar a 1.

$$y = \begin{cases} 0, & \text{se } x < 0 \\ 1, & \text{se } x \geq 0 \end{cases} \quad (3)$$

A Equação 3 mostra que a função binária pode variar entre dois valores apenas $\{0, 1\}$. A função de ativação dos neurônios das camadas ocultas (ReLU), é apresentado na Equação 4.

$$y = \begin{cases} 0, & \text{se } x < 0 \\ x, & \text{se } x \geq 0 \end{cases} = \max(0, x) \quad (4)$$

A Equação 4 mostra que a função ReLU pode variar entre zero e mais infinito $\{0, \infty\}$. A função de ativação dos neurônios da camada de classificação *sigmoid* é apresentada na Equação 5.

$$y = \frac{x}{1 + e^{-x}} \quad (5)$$

A Equação 5 mostra que a função *sigmoid* pode variar entre menos -1 e +1. No entanto, pode-se aplicar uma função modular para que os valores ativados sejam positivados e a função ficar entre 0 e 1 (BENGIO IAN GOODFELLOW, 2015). A função de ativação dos neurônios da camada de classificação *softmax* é apresentada na Equação 6.

$$y = \frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}} \quad (6)$$

A Equação 6 mostra que a função *softmax* pode variar entre menos 0 e 1. Portanto, pode-se dizer que as funções 6, 5 e 3 são para ativar os neurônios da última camada, ou seja, para ativar a camada de classificação. Já a função 4 terá maior efetividade ativando neurônios da camada densa (camada oculta).

Uma vez que as funções de ativação foram apresentadas, pode-se introduzir o método de treinamento das redes neurais multi camadas. O treinamento da rede neural baseia-se em métodos de otimização de parâmetros e aproximações matemáticas. O método mais conhecido para treinamento é o *feedforward*, onde os pesos são inicializados com valores aleatórios positivos. Ao se processar os dados por todas as camadas ocultas da rede, os resultados dos neurônios são passados para a camada

de classificação e o resultado obtido será utilizado para realimentar a rede no método de *back propagation*. O método de realimentação servirá como um calibrador da rede neural, fazendo com que a rede se aproxime do mínimo global, alterando os valores dos pesos de cada neurônio. Esse processo pode ser realizado q vezes, cada processo de *feedforward* e *back propagation* utilizando todo o conjunto de dados é denominado de época. A Figura 8, ilustra como os dois métodos de treinamento realizarão suas operações.

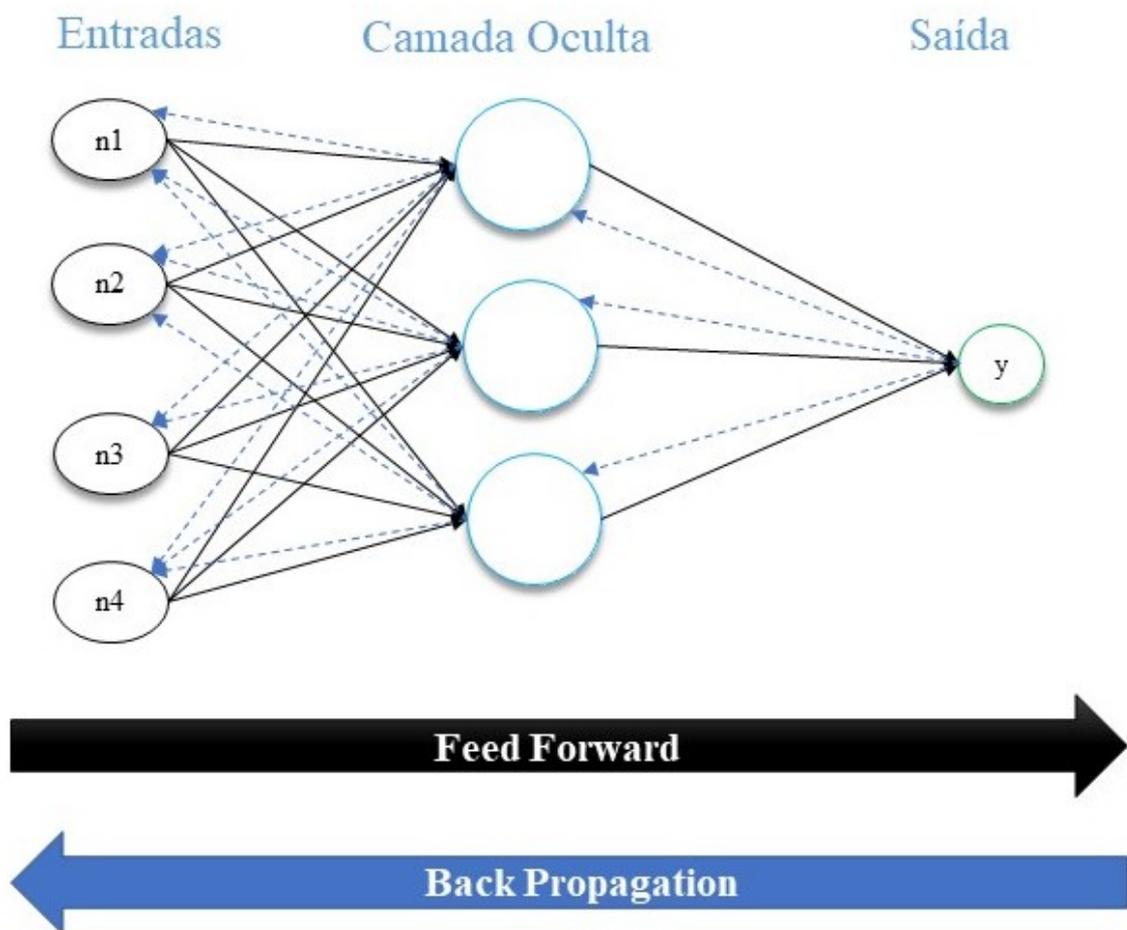


Figura 8 – Ilustração do treinamento de uma rede neural. Em setas pretas estão as comunicações realizadas entre os neurônios no *feedforward* e em setas pontilhadas azul estão as comunicações feitas entre os neurônios no *back propagation*, baseado em (BEN-GIO IAN GOODFELLOW, 2015).

2.3 REDES NEURAS CONVOLUCIONAIS (*CONVOLUTIONAL NEURAL NETWORKS*)

Redes neurais são consideradas uma das principais metodologias de aprendizado profundo (LI et al., 2020), sendo uma das áreas mais prósperas de aprendizado de máquina na linha de pesquisa sobre inteligência artificial, como citado na seção 2.2. Redes neurais convolucionais (CNNs) surgiram entre metade da década de 80 e 90 para solucionar problemas relacionados a reconhecimento de dígitos (LECUN et al., 1998a). O intuito geral era conseguir extrair informações das imagens buscando classificar possíveis dígitos manuscritos. Um classificador linear foi proposto para solucionar esse problema (LECUN et al., 1998b).

Dentro dessa metodologia foi alcançada uma taxa de otimização de erro relativamente alta, em apenas 12% de erro em conjuntos de testes. Ao longo do mesmo período, outras estruturas foram propostas, mas a de melhor resultado foi a proposta em (LECUN et al., 1998a), chegando em 0.95% de erro, utilizando uma única camada de convolução como extrator de características das imagens, buscando solucionar problemas de parametrização (LECUN et al., 1998b).

Dessa forma, foi proposta uma estrutura global para redes neurais convolucionais. Além disso, visando diminuir o número de parâmetros e ainda manter informações relevantes das imagens, foi proposta a camada de convolução como método principal para extração de características.

2.3.1 Convolução

Convolução pode ser definida como a representação matemática de como um sistema linear opera sobre um sinal (HIRSCHMAN, 2012). Matematicamente, convolução é a resposta entre uma entrada e um determinado impulso convoluido a essa entrada. Chama-se convolução em imagens o processo de filtragem efetuado pelo deslocamento de um filtro convolucional, sobre a imagem, pelo avanço de um pixel por vez. Define-se como máscara, uma janela composta por valores ímpares de linhas e colunas (JESUS, 2015).

Em imagens, filtros convolucionais, em geral, podem ser utilizados como extractores de características, buscando otimizar o número de parâmetros das imagens.

Um exemplo de como pode ser calculada a função de convolução é apresentado na Equação 7, onde z é o produto da convolução de g em f .

$$f(x, y) \otimes g(x, y) = z(x, y) \quad (7)$$

Uma imagem pode ser considerada como uma função não linear, sendo aplicada em uma função linear (convolução) (GONZALEZ, 2009). Uma das principais funções da convolução nas redes neurais é a extração de características. O resultado de uma camada convolucional pode ser chamado de mapa de características (JESUS, 2015).

2.3.2 Pooling

O mapa de características gerado pela convolução pode trazer informações variantes espacialmente (DOUGHERTY, 2020), (KRIZHEVSKY; SUTSKEVER; HINTON, 2017). Visando manter a imagem invariante a tais transformações, a camada de *pooling* foi adicionada à estrutura de uma CNNs.

Nesta etapa o objetivo é manter informações valiosas e relevantes (LI et al., 2020), (ZANGENEH; RAHMATI; MOHSENZADEH, 2020a), utilizando diversas políticas de sumarização de características, algumas delas são: *Average Pooling*, *Global Average Pooling*, *Max Pooling* e *Global Max Pooling*.

Uma máscara que pode variar seu tamanho de passada (*stride*), se movendo por todo mapa de características, e passa adiante o quê detectar relevante, conforme sua política empregada. Um exemplo é apresentado na Figura 9, nesse caso, foi utilizado o *Max Pooling*, com janela de 2x2 e um *stride* de tamanho 2 em um mapa 4x4.

O último processo da camada de convolução é o achatamento (*Flatten*), onde as características, que estão em estrutura de matriz (2D), passam a ser um vetor de única dimensão.

Ao final do processo de extração de características, os resultados são passados como entrada para uma rede neural artificial ou *Multi Layer Perceptron*, estrutura apresentada na seção 2.2.3.

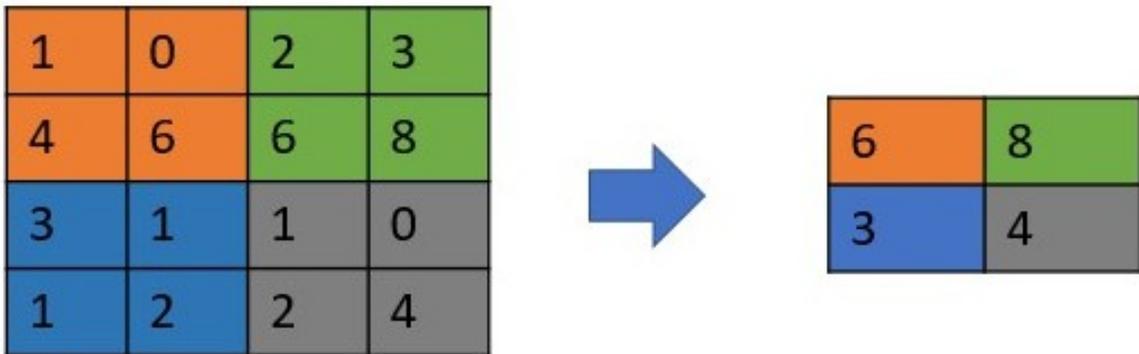


Figura 9 – Exemplo de funcionamento do *Max Pooling*, retirado de (KRIZHEVSKY; SUTSKEVER; HINTON, 2017).

2.3.3 Transferência de aprendizado

Devido ao alto custo computacional para se treinar uma rede neural desde o início (KRIZHEVSKY; SUTSKEVER; HINTON, 2017), pode ser utilizada a técnica de transferência de aprendizado (*Transfer Learning*) que consiste em utilizar modelos pré-treinados e retreiná-los para um novo propósito (ZHUANG et al., 2020).

A transferência de aprendizado é usada para inferir aprendizado a um modelo com domínio semelhante. Essa técnica é muito utilizada em aprendizado de máquina, devido a sua alta aplicabilidade em modelos preditivos (ZHUANG et al., 2020). Em redes neurais convolucionais, a transferência de aprendizado consiste em congelar camadas com filtros já treinados e processar apenas a última camada de uma rede neural, a totalmente conectada, sendo essa a camada onde haverá oportunidade de adaptar a rede para o tipo de problema trabalhado (SHAFABI et al., 2019).

A Figura 10 apresenta a visão geral da metodologia de transferência de aprendizado. Em amarelo, está ilustrada a rede treinada com alto volume de dados D_A , por períodos elevados, que busca atacar um problema genérico T_A . O conceito de transferência de aprendizado está em verde na Figura 10. Fazendo uso dos pesos pré treinados pela rede em amarelo A' , adapta-se os mesmos para o treinamento do novo modelo treinável B , que busca solucionar um problema específico T_B , utilizando D_B como dados de entrada para treinamento.

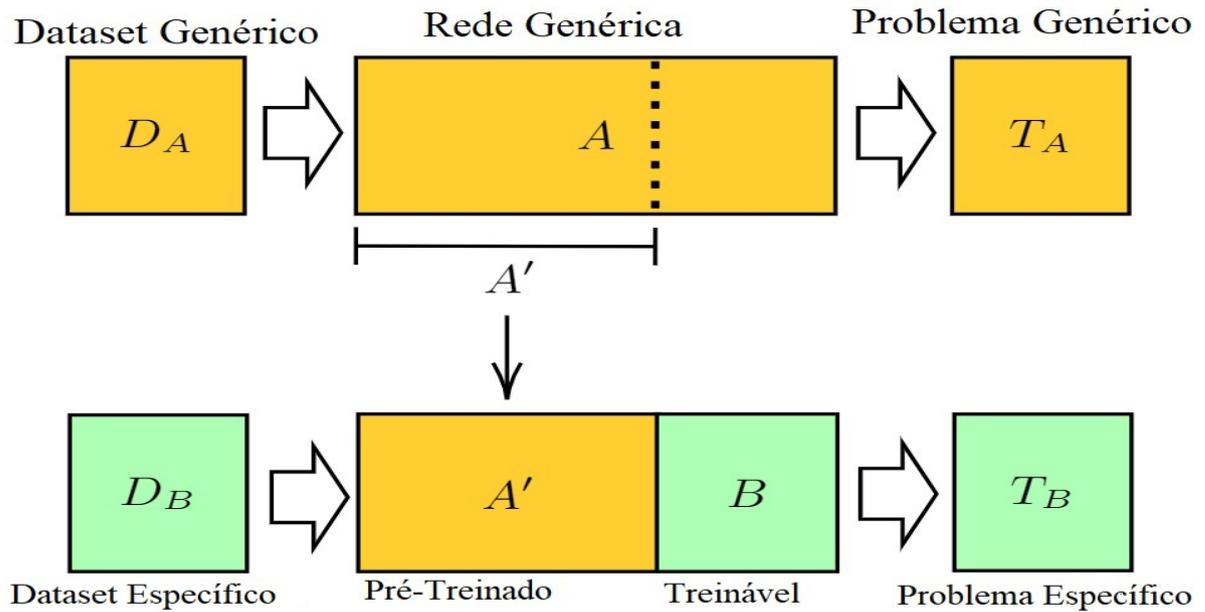


Figura 10 – Transferência de aprendizado em modelos de *machine learning*, extraído de (MARI et al., 2020).

2.4 GRAFOS

Visto que redes neurais convolucionais são ferramentas de grande impacto em extração de características, classificação e detecção e segmentação de objetos (TAN; SONG; BOULARIAS, 2020). Pode-se introduzir a estrutura primária para as redes neurais convolucionais para classificação de imagens contextuais, os grafos.

Buscando abstrair sistemas computacionais, a estrutura de grafo pode ser representada de maneira direcionada ou não direcionada. Um grafo tem uma estrutura similar a de uma rede de computadores, onde os computadores que se conectam são os vértices do grafo e as próprias conexões são as arestas. Portanto, os dois objetos primários de um grafo são as arestas e os vértices (NETTO, 2012).

Um grafo pode ser representado pela Equação 8, onde G é o grafo formado pelo conjunto A de arestas e pelo conjunto V vértices.

$$G = (V, A) \quad (8)$$

Para (CHARTRAND; ZHANG, 2019) e (NETTO, 2012), os grafos podem assumir diversas topologias diferentes para representar algum tipo de informação, no entanto, sua estrutura básica não alterará, se mantendo em conjuntos de vértices e conjunto de arestas. Para descrever o grafo, pode-se utilizar alguns modelos, como é

citado na Tabela 1.

Grafo	Vértice	Arestas
Relações Sociais	Pessoas	Amizades
Transporte	Cidades Aeroportos	Estradas e vias aéreas
Internet	Sites	Links

Tabela 1 – Diferentes tipos de abstração de relações entre indivíduos através de grafos, extraído de (NETTO, 2012).

De acordo com (CHARTRAND; ZHANG, 2019), pode-se definir as estruturas mais conhecidas de grafos, tais como: simples, não dirigido e dirigido, completo, ciclo, bipartido completo, valorado, subgrafo, entre outras possíveis topologias, no entanto, o presente trabalho irá tomar conhecimento das estruturas citadas.

Um grafo simples não possuirá laços nem arestas paralelas, onde uma aresta com vértices terminais u e v pode ser representada por uv . Um grafo dirigido consiste em dois conjuntos finitos, vértices $V(G)$ e arestas dirigidas, em que as arestas são associadas a pares ordenados de vértices (nós terminais). Um grafo não dirigido é similar ao dirigido, no entanto, não haverá direções nas arestas que conectam os vértices (NETTO, 2012).

Um grafo completo é um grafo simples com n vértices, em que o conjunto de arestas contém precisamente uma aresta para cada par de vértice distinto, podendo ser representado por K_n . Para calcular a quantidade de arestas de um grafo completo K_n , aplica-se a Equação 9, sendo $|V|$ o número de vértices contidos no grafo.

$$\frac{|V^2| - |V|}{2} \quad (9)$$

Um grafo em ciclo de n vértices, pode ser apresentado como G_n , em que $n \geq 3$, é um grafo simples com n vértices e a arestas. O nome de ciclo se dá pelo fato de que as conexões do grafo formam uma espécie de laço entre si, onde o primeiro vértice o qual se conecta com o último e ele não é necessariamente completo (NETTO, 2012).

Um grafo bipartido completo com m, n vértices, pode ser denominado de $K_{m,n}$ é um grafo simples com vértices em v e w , onde os vértices estão em paralelo. As propriedades desta topologia são: deve existir uma aresta entre um par de vértices v e w , não pode existir uma aresta entre um par de vértices v e v e não pode haver uma aresta entre um par de vértices w e w (NETTO, 2012).

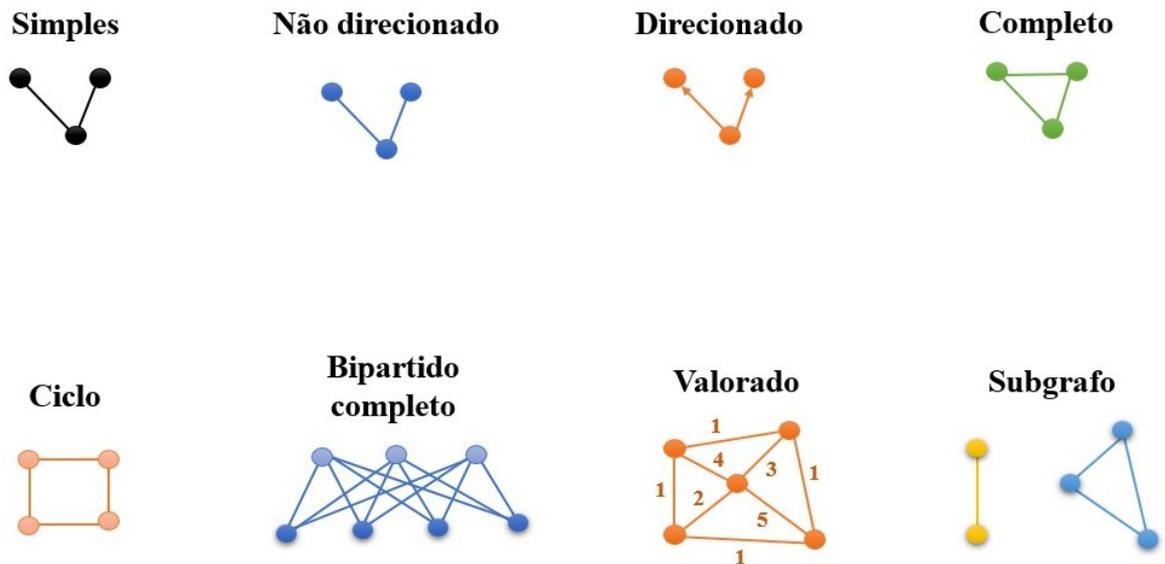


Figura 11 – Diferentes topologias de grafos, baseado em (NETTO, 2012) e (CHARTRAND; ZHANG, 2019).

Em grafos valorados, suas arestas possuem valores associados. Um grafo G consiste em um conjunto de vértices V e um conjunto A de arestas e uma função f de A para P , em que P será o conjunto dos pesos associados às arestas (NETTO, 2012).

Um subgrafo $G' = (V', A')$ é um subgrafo de $G = (V, A)$ se cada vértice de G' estiver contido no conjunto de vértices de G e se cada aresta de G' estiver contida no conjunto de arestas de G (NETTO, 2012).

Dentre as topologias citadas, cabe ainda mostrar que um vértice v de um grafo G pode se conectar com ele mesmo, formando um *loop*, aresta de v para v . Dito como cada topologia pode ser definida, a Figura 11, ilustra cada uma das topologias.

Como apresentado na Figura 11, um grafo pode ser representado neste formato similar a uma rede ou diagrama (NETTO, 2012). No entanto, computacionalmente, tal abstração pode não ser a ideal, portanto, pode-se utilizar de outra estrutura para processar as conexões entre vértices de um grafo, através da matriz de adjacência (CHARTRAND; ZHANG, 2019), (NETTO, 2012).

Uma matriz de adjacência X mostrará as conexões presentes em um grafo, em que suas linhas e colunas são referentes a cada vértice do grafo. Em um par de vértices que exista conexão (aresta), na matriz estará a quantidade de conexões na célula em que a coluna é o vértice v_j e a linha é o vértice v_i .

Portanto, pode-se afirmar que uma matriz de adjacência irá representar o grafo

em um processamento computacional. A matriz terá dimensões $n \times n$, sendo n o número de vértices do grafo. Denotada por $X = [x_{ij}]$ e definida pela Equação 10.

$$x_{ij} = \begin{cases} 1, & \text{se existe uma aresta entre o par de vértices } v_i \text{ e } v_j \\ 0, & \text{se não existe uma aresta entre o par de vértices } v_i \text{ e } v_j \end{cases} \quad (10)$$

Para auxiliar na percepção e diferenciar cada uma das estruturas, seja o grafo ou a matriz, é apresentada a Figura 12, com a matriz ilustrada em (b) e o grafo em (a).

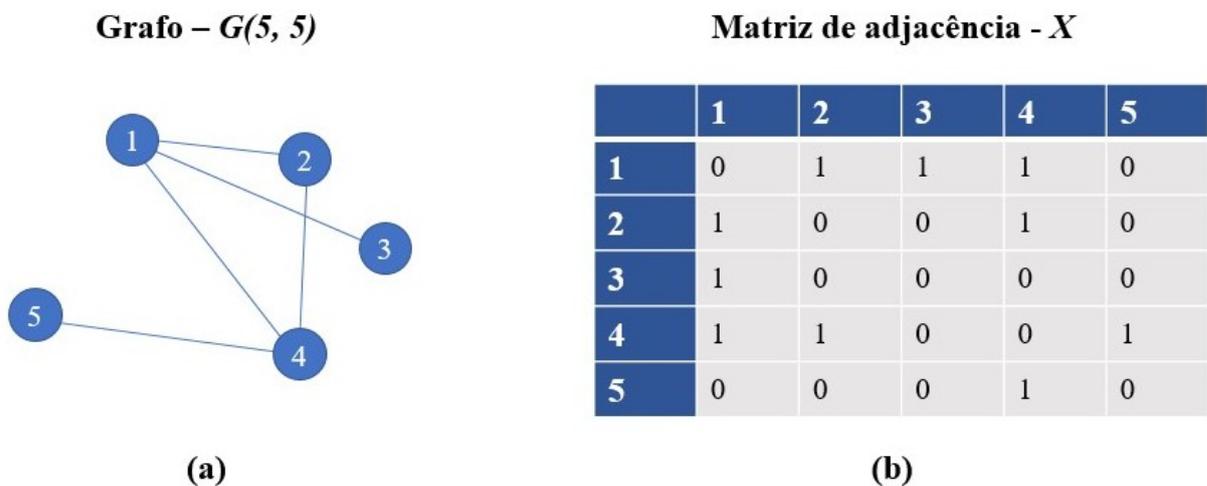


Figura 12 – Matriz de adjacência e seu grafo. Em (a) está o grafo G e em (b) a matriz de adjacência X .

Além da matriz de adjacência, pode ser utilizada a matriz de graus em grafos não direcionados. A matriz de graus mostrará informações do grau de cada vértice, sendo assim, será computado o número de arestas ligadas a cada vértice (NETTO, 2012). A definição da matriz de graus é apresentada na Equação 11.

$$D_{ij} = \begin{cases} grau(v_i), & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (11)$$

2.5 GRAFOS CONVOLUCIONAIS (GCN)

Visto que a estrutura de dados em Grafo pode representar indivíduos e suas respectivas conexões, como afirmado na Tabela 1, uma outra possível forma de uso dos grafos é na representação de relações entre objetos em imagens (BUGATTI; SAITO; DAVIS, 2019). As redes neurais convolucionais são excelentes ferramentas

para classificação, detecção e segmentação de imagens. No entanto, em alguns casos, onde as imagens são classificadas de acordo com contextos espaciais e estruturais entre objetos presentes na imagem, uma rede neural convolucional pode não alcançar boa performance (BUGATTI; SAITO; DAVIS, 2019), (KIPF; WELLING, 2016), (CHEN; LIN; CHO, 2017).

Buscando obter bons resultados em classificar dados através de contexto (KIPF; WELLING, 2016) propôs um modelo de rede neural estruturado em grafos, onde o grafo G representará as relações entre os dados, sendo uma solução para introduzir o contexto dos dados (BUGATTI; SAITO; DAVIS, 2019), (CHEN et al., 2019).

Os grafos são muito versáteis, graças a sua grande capacidade de expressar relações. Em diversas áreas, modelos de aprendizado de máquina têm sido usados com sucesso para extrair e prever informações sobre dados contidos em grafos, para modelar elementos complexos e suas relações (ZHANG; CUI; ZHU, 2020).

Para (ZHANG; CUI; ZHU, 2020), grafos podem representar diversos problemas do dia-a-dia, sendo uma excelente ferramenta, podendo prever padrões em tráfego de redes rodoviárias (CUI et al., 2018), previsão de interações entre proteínas para desenvolvimento de medicamentos (FOUT et al., 2017), sistemas de recomendação baseados em informações de redes sociais (FAN et al., 2019).

Dentro do campo de aprendizado de máquina, os grafos podem ser utilizados de diferentes maneiras em várias tarefas, tais como: classificação de grafos, classificação de vértices, detecção de comunidades e predição de conexões (Li et al., 2015), (CHARTRAND; ZHANG, 2019), (ZHANG; CUI; ZHU, 2020), (CASALEGNO, 2021).

Para (ZHANG et al., 2019), redes neurais convolucionais ou GCNs, generalizam redes neurais para dados estruturados em grafos. A classificação do grafo será, dado um determinado grafo de entrada o modelo GCN irá prever o conjunto de classes (CASALEGNO, 2021).

Para (KIPF; WELLING, 2016), grafos convolucionais podem ser uma abordagem escalável para aprendizado semi-supervisionado em dados estruturados em grafos, baseado em uma variante eficiente de redes neurais convolucionais que operam diretamente em grafos. Neste caso, a estrutura proposta por (KIPF; WELLING, 2016) seria a de classificação de vértices, onde dado um grafo com rotulagem de nós incompleta, pode-se prever as classes dos nós sem rotulagem (CASALEGNO, 2021).

Para (CARLETTI; FANELLI; LAMBIOTTE, 2021), comunidades em um grafo

são aglomerações de objetos que pertencem a uma mesma classe ou possuem características semelhantes. Para (CASALEGNO, 2021), dado um grafo, comunidades (*clusters*) serão formadas por nós que apresentam similaridades, como características entre objetos, representação de cidades dentro de estados, entre outros.

Para (WANG et al., 2022), grafos convolucionais podem auxiliar na detecção de possíveis conexões entre vértices que possuem alguma similaridade mas sem nenhuma aresta que os conecte. Em (CASALEGNO, 2021), detecção de conexões pode atuar em um grafo com matriz de adjacência incompleta, para prever se cada par de vértices deveria estar conectados. A Figura 13 apresenta as diferentes finalidades em que as redes neurais em grafos podem atacar.

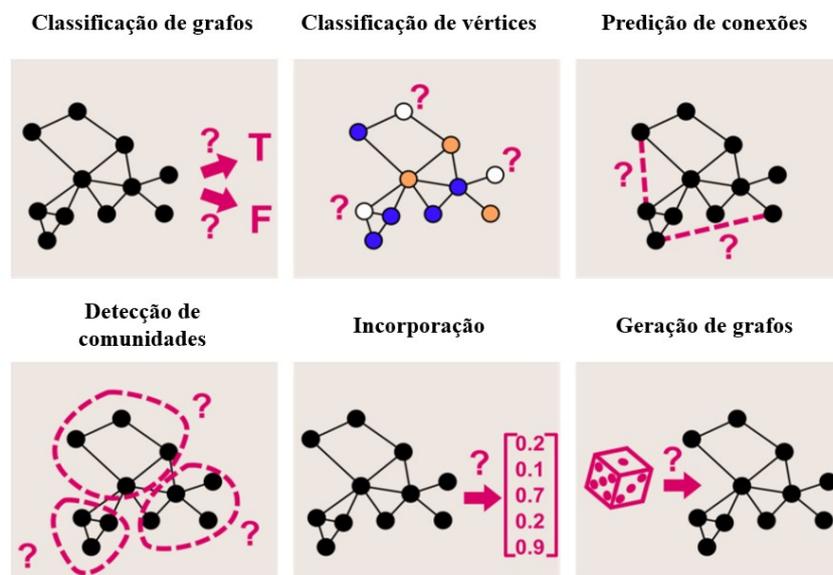


Figura 13 – Diferentes abordagens para redes neurais em grafos, extraído de (CASALEGNO, 2021).

Em (BUGATTI; SAITO; DAVIS, 2019), foi apresentada uma nova funcionalidade de redes neurais convolucionais em grafos, a classificação de imagens contextuais, utilizando datasets como Mit67 (Quattoni; Torralba, 2009), UnRel (PEYRE et al., 2017) e VRD (LU et al., 2016). O intuito principal foi adquirir conhecimento da relação entre os objetos das imagens utilizando o auxílio da estrutura em grafo para simular as relações e utilizando a estrutura de GCN apresentada em (KIPF; WELLING, 2016).

De acordo com (BUGATTI; SAITO; DAVIS, 2019), em (KIPF; WELLING, 2016) os autores propuseram um método de aprendizado escalável para redes neurais convolucionais em grafos chamado *Graph Convolutional Network* (GCN). Para (BUGATTI;

SAITO; DAVIS, 2019), as GCNs utilizariam múltiplas camadas de convolução de grafos para promover agregação de informação de vizinhança, em que após C camadas um determinado vértice irá se fundir com as informações dos seus vizinhos. Ainda é afirmado que GCNs propagariam recursos, buscando agregar cada vez mais informações em cada camada oculta.

Em (BUGATTI; SAITO; DAVIS, 2019), (CHEN; LIN; CHO, 2017) e (CASALEGNO, 2021) um grafo simples não direcionado $G = (V, A)$ em que V é o conjunto de N vértices e $A \in \mathbb{R}^{N \times N}$, produzem uma matriz de adjacência simétrica, onde A_{ij} e A_{ji} são iguais. Em (KIPF; WELLING, 2016) e (BUGATTI; SAITO; DAVIS, 2019), cada vértice v_i é representado por um vetor de características, $x_i \in \mathbb{R}^d$, em que d é a dimensão espacial do vetor, sendo assim, existe uma matriz de características $X \in \mathbb{R}^{N \times d}$ em que estarão todos os vetores de características, como apresentado na Equação 12.

$$X = [x_1, x_2 \dots x_N] \quad (12)$$

Para (BUGATTI; SAITO; DAVIS, 2019), as características em X serão representadas por médias de seus respectivos vizinhos para cada camada de GCN, se identificando com o processo de propagação de características. De acordo com (BUGATTI; SAITO; DAVIS, 2019), em (KIPF; WELLING, 2016), a propagação das características se davam apenas entre vértices de primeira ordem, ou seja, vizinhos diretos.

O método de propagação das características é apresentado na Equação 13 e com base em (BUGATTI; SAITO; DAVIS, 2019) a representação dos vértices iniciais são inicializados com a matriz de características original X , ou seja, ($H^{(0)} = X$).

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \quad (13)$$

De acordo com a Equação 13, l indica o índice da camada convolucional do grafo; σ é a função de ativação das camadas ocultas; W é a matriz de pesos que serão treinados para a camada de grafo convolucional l ; \hat{A} é a normalização de A com loops nos vértices, \hat{A} é definido na Equação 14.

$$\hat{A} = \check{D}^{-\frac{1}{2}} \check{A} \check{D}^{-\frac{1}{2}}, \hat{A} = A + I_N \quad (14)$$

Nota-se que a matriz de graus do grafo é dividida em duas partes $\check{D}^{-\frac{1}{2}}$ para otimização computacional, I_N é a matriz identidade; \check{D} é a matriz de graus de \hat{A} (KIPF;

WELLING, 2016).

Para efeito de comparação, a Figura 14 é apresentada, ilustrando como a convolução é operada para as imagens e para os grafos. Cabe destacar que a rede neural convolucional tradicional utiliza o pixel central do filtro convolucional como valor mais importante, na convolução das redes neurais convolucionais em grafos o processo é similar, no entanto, o vértice central é denominado de centroide da região de convolução do grafo.

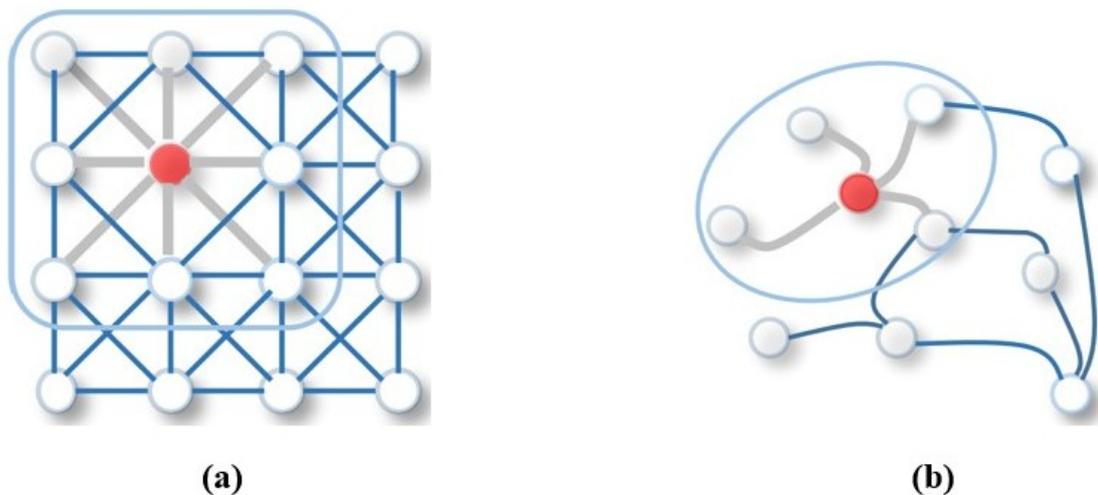


Figura 14 – Comparação da convolução entre CNN e GCN, em (a) está a convolução para a rede neural convolucional tradicional (CNN) e em (b) está a convolução para a rede neural convolucional em grafos, imagem extraída de (WU et al., 2020).

Em (BUGATTI; SAITO; DAVIS, 2019), o método de propagação das informações entre os nós toma conhecimento de atributos visuais e espaciais, denominado de características complementares. Portanto, a propagação das informações nas camadas convolucionais dos grafos em (BUGATTI; SAITO; DAVIS, 2019) é apresentada na Equação 15.

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}S^{(l)}W^{(l)}) \quad (15)$$

Para a última camada da rede, ou denominada de camada de classificação, pode-se utilizar a função de probabilidade *softmax*, em que $H^{(L)}$ apresentará a probabilidade de um conjunto de vértices pertencerem a uma classe c , assumindo que os vértices estão representados em um vetor com dimensão C denominado de *one-hot* vetor $y_i \in \{0, 1\}^C$.

Portanto, de acordo com (KIPF; WELLING, 2016), considerando uma rede neural com duas camadas de grafos convolucionais, a camada de classificação é apresentada na Equação 16.

$$Z = f(X, A) = \text{softmax}(\hat{A}\text{ReLU}(\hat{A}XW^{(0)})W^{(1)}) \quad (16)$$

De acordo com as informações descritas acima sobre a estrutura GCN, nota-se que os grafos exibem um papel muito importante no processo como um todo.

2.6 CAMINHADA ALEATÓRIA - *RANDOM WALK*

Em (PEARSON, 1905), foi abordado o problema de caminhada aleatória, denominado de *The Problem of the Random Walk* ou *Drunkard's Walk*. O problema se iniciava afirmando que, um homem parte de um ponto 0 e caminha l passos em linha reta; ele então vira em qualquer ângulo e anda mais l passos em uma segunda linha reta, repetindo esse processo n vezes, havendo probabilidade de que após n desses trechos ele esteja a uma distância entre r e $r + \delta$ de seu ponto de partida (PEARSON, 1905). Em (ZHOU; XU; DENG, 2020), é apresentado que Lord Rayleigh apresentou uma aplicabilidade do sistema de caminhada aleatória em ondas sonoras, afirmando que caso o valor de n seja relativamente alto, o sistema de caminhada pode ser descrito na Equação 17.

$$\text{random walk} = \frac{2}{nl^2} e^{-r^2/nl^2} r \delta r \quad (17)$$

Para (ZHOU; XU; DENG, 2020), o método descrito como um processo estocástico porque funciona através da aplicação de variáveis aleatórias. A caminhada aleatória essencialmente rastreia etapas incrementais por uma inteligência modelada particular ou uma espécie de ator racional. De acordo com (PONS; LATAPY, 2005), o efeito de cara ou coroa pode auxiliar a compreender o método de caminhada aleatória, conforme apresentado na Tabela 2, em que coroa faz o caminhador ir para frente e cara faz ele caminhar para trás, assumindo que ele está no ponto $r = 0$ e andou p passos, onde p é igual a 4.

De acordo com (ZHOU; XU; DENG, 2020), (WANG; YANG; WANG, 2020) e (Li et al., 2015), o deslocamento do caminhador aleatório pode ser descrito conforme a Tabela 3.

Lançamento	Resultado	Posição
1	Coroa	1
2	Coroa	2
3	Cara	1
4	Coroa	2

Tabela 2 – Cara ou coroa descrevendo os pontos em que o caminhador aleatório caminhará, extraído de (ZHOU; XU; DENG, 2020).

Símbolo	Deslocamento
N	passos sucessivos
l	comprimento da caminhada
p	probabilidade de caminhar para a direita
q	probabilidade de caminhar para esquerda
n_1	passos para a direita
n_2	passos para a esquerda

Tabela 3 – Descrição do deslocamento do caminhador aleatório.

Para reforçar o conhecimento de como o caminhador pode alterar seus passos ao longo do eixo x , é apresentada a Figura 15.

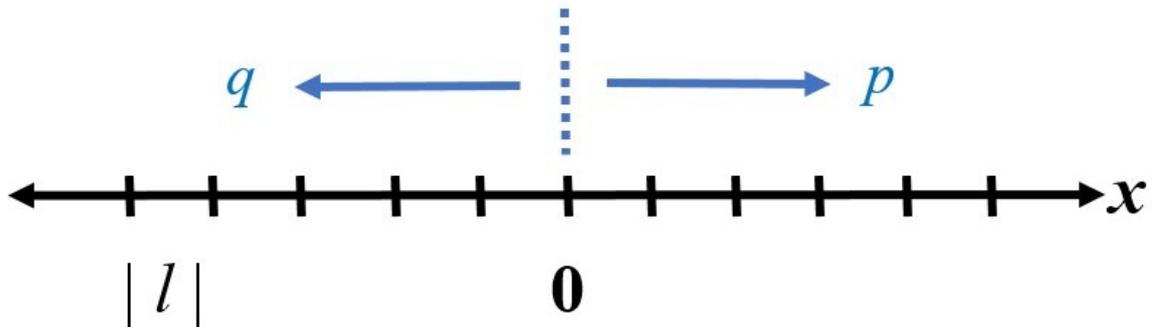


Figura 15 – Caminhada aleatória no eixo x , extraído de (ZHOU; XU; DENG, 2020).

Para que o caminhador saia de um ponto P_i e vá para o ponto P_j é necessário que $p + q \geq 1$, mostrando que um passo foi dado, seja na direção p ou q , na versão descrita em (PEARSON, 1905), o caminhador executará passos sucessivos, não podendo ficar parado.

De acordo com (Li et al., 2015), o deslocamento do caminhador aleatório, pode ser descrito na Equação 18, onde m é o trajeto percorrido, n_1 é o número de passos dado para a direita e n_2 é a quantidade de passos dado para a esquerda, sendo $-N \leq m \leq N$.

$$m = n_1 - n_2 \quad (18)$$

Após o caminhador percorrer N passos, o ponto P em que ele se encontra, pode ser descrito pela Equação 19.

$$x = ml \quad (19)$$

De acordo com (Li et al., 2015), (ZHOU; XU; DENG, 2020) e (PONS; LATAPY, 2005), para descobrir os possíveis trajetos de um caminhador aleatório com N passos, pode-se utilizar a Equação 20.

$$\frac{N!}{n_1!n_2!} \quad (20)$$

Em (WANG; YANG; WANG, 2020), (Li et al., 2015) e (LAWLER; LIMIC, 2010), é citado que existe a possibilidade de calcular a probabilidade de uma determinada sequência de N passos, através da distribuição binomial, afirmando a probabilidade de se haver n_1 passos para a direita e n_2 passos para a esquerda após N passos. A Equação 21 descreve o cálculo.

$$W_N(n_1) = \frac{N!}{n_1!n_2!} p^{n_1} q^{n_2} \quad (21)$$

Verificando a distribuição de probabilidades, afirmando que m seja o deslocamento resultante da Equação 18, e N se da pela soma de n_1 e n_2 , de acordo com (PONS; LATAPY, 2005), (Li et al., 2015) e (ZHOU; XU; DENG, 2020), a probabilidade de se haver um deslocamento m após N passos é apresentado na Equação 22, partindo da primícia que $P_N(m)$ é igual a $W_N(n_1)$.

$$P_N(m) = \frac{N!}{\left(\frac{N+m}{2}\right)! \left(\frac{N-m}{2}\right)!} p^{\frac{N+m}{2}} q^{\frac{N-m}{2}} \quad (22)$$

Visto que há a possibilidade de se calcular a probabilidade de um caminhador aleatório C passar por um determinado ponto P através de caminhadas de tamanho N , (Li et al., 2015) propõe que os mesmos princípios de caminhada aleatória podem ser utilizados em grafos, afirmando que caminhada aleatória é uma técnica fundamental para reconhecer amostras de nós uniformes em um grafo G .

Em (Li et al., 2015) e (WICKET, 2015), o caminhador aleatório pode caminhar pelos vértices de um grafo G , a transição do vértice v_i para v_j só é possível se $\{v_i, v_j\} \in A$, sendo A o conjunto de arestas. A proposta apresentada é que, existe a possibilidade de se prever a probabilidade do caminhador C , durante a caminhada de tamanho N , passar pelo vértice v_i , conforme apresentado na Equação 23 extraído de (WICKET, 2015), sendo \hat{A} a quantidade de arestas do grafo G .

$$w(i) = \frac{d(v_i)}{2\hat{A}} \quad (23)$$

De acordo com (WICKET, 2015) e (Li et al., 2015), caminhada aleatória é uma maneira de dizer para o algoritmo para andar no grafo e escolher transições de maneira aleatória. No entanto, em (ZHOU; XU; DENG, 2020), (PONS; LATAPY, 2005) e (MAZÓN; SOLERA; TOLEDO, 2020), o método de caminhada aleatória pode ser utilizado para reconhecer comunidades em grafos, em que dado um grafo G , o método de caminhada aleatória ficaria preso em subgrafos G' ou comunidades C_i . Para (PONS; LATAPY, 2005), comunidades em grafos são segmentos do grafo em que seus nós compartilham características semelhantes, portanto, as arestas que os conectam possuem alta compatibilidade. Portanto, dado um grafo G , o método de caminhada aleatória pode encontrar C comunidades neste grafo, possibilitando destacar vértices de maior similaridade entre si. Uma ilustração das comunidades do grafo é apresentada na Figura 16.

2.7 MÉTRICAS DE AVALIAÇÃO

Para avaliar a performance dos modelos existem técnicas de seleção em aprendizado de máquina, que buscam identificar os melhores modelos (MITCHELL, 1997). Para avaliar os modelos supervisionados, pode-se utilizar algumas informações: verdadeiros positivos (VP), falsos positivos (FP), falsos negativos (FN) e verdadeiros negativos (VN) (KARIMI, 2021).

Os verdadeiros positivos apontam que as inferências que o modelo previu a classe correta da amostra (e.g. Figura 2 classificação: clinica dentária). Falsos positivos são inferências realizadas pelo modelo e que previram incorretamente a classe da amostra (e.g. Figura 2 classificação: sala). Falsos negativos são inferências realizadas pelo modelo e que previu incorretamente que a amostra não é da classe c (e.g. Figura 2 classificação: não é clinica dentária). Os verdadeiros negativos são inferências em que modelo treinado fez e que previu que a classe da amostra não é c (e.g. Figura 2 classificação: não é sala).

De acordo com (KARIMI, 2021), acurácia, precisão, revocação e f1-score são as métricas mais utilizadas para seleção dos melhores modelos. A acurácia (*accuracy*) quantificará o número de amostras classificadas corretamente, sem levar em

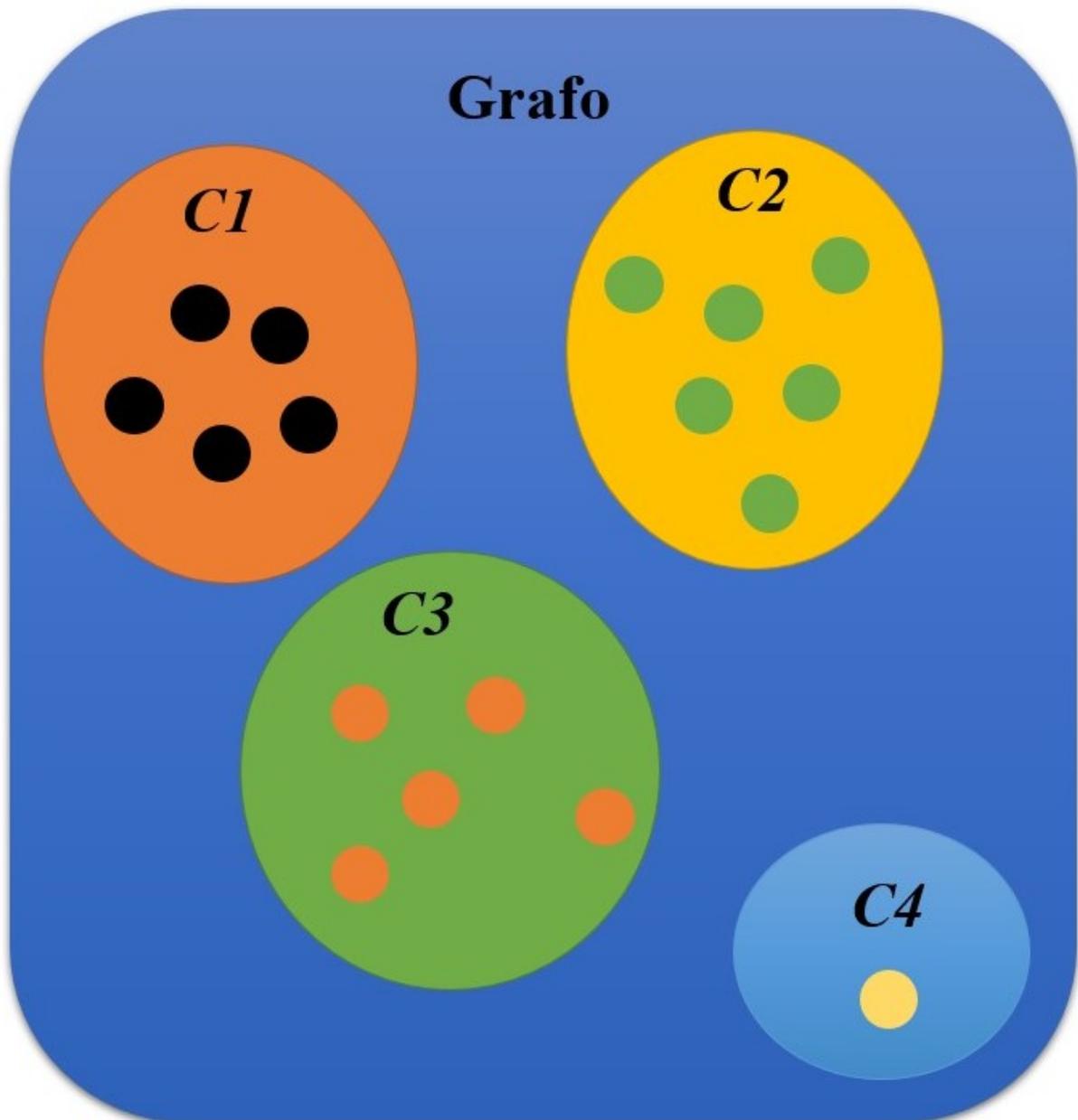


Figura 16 – Comunidades destacadas de um grafo G , inspirado em (Li et al., 2015). O grafo G é toda a área em azul, e as comunidades estão destacadas entre C .

consideração as classes, conforme apresentado na Equação 24.

$$\text{acurácia} = \frac{VP + VN}{VP + FP + FN + VN} \quad (24)$$

Para (KARIMI, 2021), a precisão é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e o total de exemplos classificados como positivos, conforme apresentado na Equação 25.

$$\text{precisão} = \frac{VP}{VP + FP} \quad (25)$$

Em (KARIMI, 2021), a revocação é definida como sensibilidade ou taxa de verdadeiro positivo, dando maior ênfase para os erros por falso negativo, afirmando que a revocação é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e a quantidade de exemplos que são de fato positivos, conforme a Equação 26.

$$\text{revocação} = \frac{VP}{VP + FN} \quad (26)$$

A métrica F1-Score, leva em consideração a precisão e a revocação, sendo justamente a média harmônica entre as duas métricas citadas, conforme apresentado na Equação 27.

$$\text{f1 score} = 2 * \frac{\text{precisão} * \text{revocação}}{\text{precisão} + \text{revocação}} \quad (27)$$

Para (KARIMI, 2021), matriz de confusão é uma ferramenta para análise preditiva de modelos de machine learning. Uma matriz de confusão é uma tabela $C * C$ em que C é a quantidade de classes da base de dados utilizada. Sua estrutura se dá pelas classes verdadeiras nas colunas da matriz e as classes preditas nas linhas, conforme apresentado na Figura 17.

2.8 TRABALHOS RELACIONADOS

Em (BUGATTI; SAITO; DAVIS, 2019), foi proposta uma abordagem capaz de lidar com problemas hierárquicos de objetos em imagens, analisando a possibilidade de um raciocínio baseado em contexto da classe geral do objeto e não de sua classe direta. Neste trabalho foram utilizadas as bases *UnRel* (PEYRE et al., 2017), *Mit67* (Quattoni; Torralba, 2009) e *Visual Relation Detection (VRD)* (LU et al., 2016). A técnica aqui aplicada foi de modelagem dos dados de maneira supervisionada. Foram utilizados modelos pré treinados já conhecidos na literatura, como VGG16, ResNet50, entre outros. Foi utilizada a política de que todos os objetos da imagem conectam entre si para formar o contexto esperado do rótulo. Foram feitas comparações entre o modelo proposto (GCNs) e modelos tradicionais da literatura (CNNs).

Em (KIPF; WELLING, 2016), foi proposta uma abordagem escalável de dados semi-supervisionada estruturados no modelo de grafo, operando através de uma

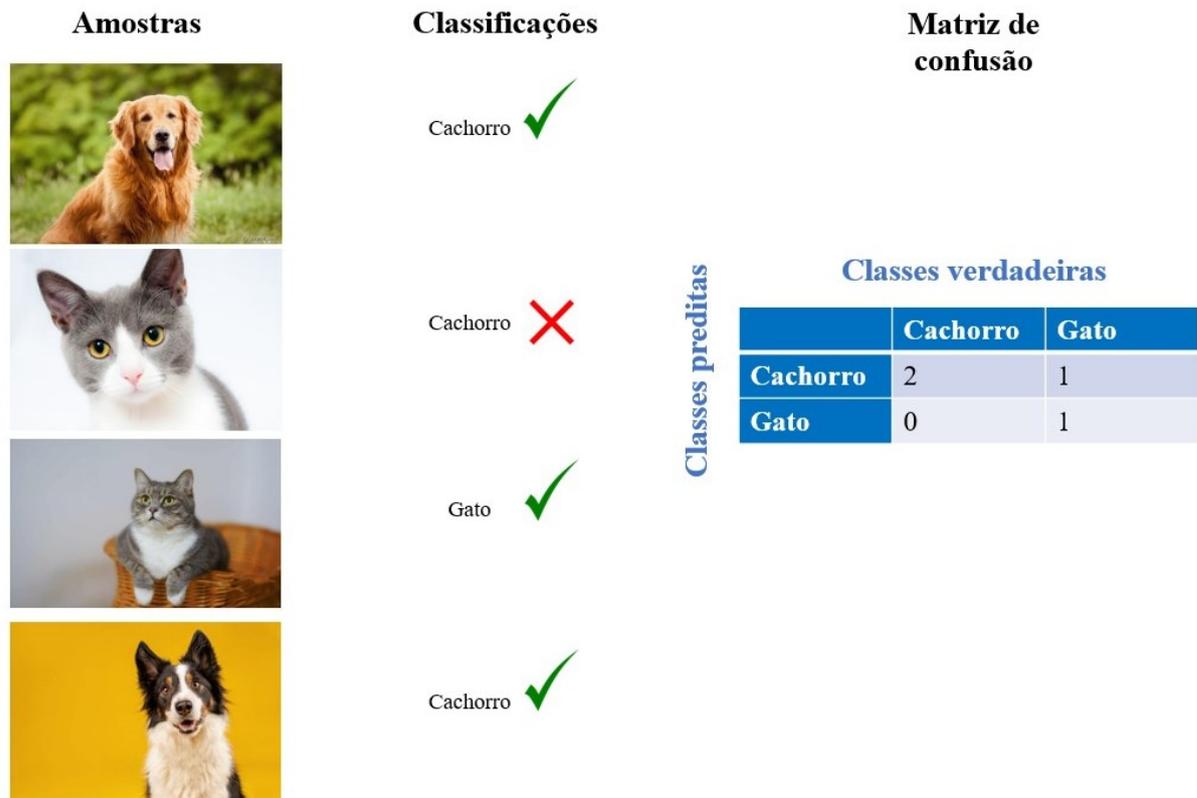


Figura 17 – Exemplo de aplicação da matriz de confusão.

variante eficiente de redes neurais convolucionais. Foi utilizado como conjunto de dados citações e um grafo de conhecimento. Tal trabalho caracterizou-se entre os introdutórios ao conceito de grafos convolucionais. Foram utilizadas bases de dados conhecidas na literatura, tais como, CiteseerX (GILES; BOLLACKER; LAWRENCE, 1998) base com índices de citações automáticas, Cora, Pubmed e NELL.

Em (SAJJAD; DOCHERTY; TYSHETSKIY, 2019), foi proposto a criação de um algoritmo eficiente para melhor representar cada nó de um grafo dinâmico, utilizando métodos de passeio aleatório, variando sua complexidade de acordo com o número de arestas utilizadas. Foram utilizados modelos de aprendizado de máquina e dados de problemas do mundo real para avaliar os grafos propostos. O trabalho se propõe a avaliar o método de caminhada aleatória como algoritmo eficiente para melhor representar os vértices de cada grafo. Foram utilizadas bases de dados como, Cora, Wikipedia, BlogCatalog e CoCit, todas as bases estão relacionadas a tipos de dados interligados entre algum tipo de contexto.

No trabalho de (Zhao et al., 2019), foi proposto um método de previsão da

função de uma determinada proteína através de passeios aleatórios disparados em vários vértices de um grafo. Nesta abordagem, foi utilizado a lei da gravitação universal de Newton para mensurar a distância entre os vértices.

Basicamente, é proposto que sejam encontrados os vértices que melhor representam conjuntos de vértices, ou seja, qual vértice melhor representa seus vizinhos mais próximos. Após o vértice ser encontrado, são então disparadas as caminhadas aleatórias, tendo como ponto inicial o vértice de melhor representação dos seus vizinhos. Essa abordagem tem seu nome de assíncrona, devido a este fato, da dependência da identificação deste super vértice. Algo ainda mais interessante deste trabalho, é a forma com que as estruturas são formadas, podendo ser estruturas inter espécies e intra espécies.

Em (BERTASIUS et al., 2016), foi proposta uma rede neural convolucional baseada em caminhadas aleatórias para segmentar de maneira eficiente partes fragmentadas e com limites mal definidos entre objetos das imagens. Tal proposta busca otimizar os objetivos de afinidade, um pixel por vez, e segmentação semântica. Nessa abordagem, haveria uma nova camada denominada de camada de caminhada aleatória (*random walk layer*), tendo como principal intuito agrupar a otimização entre os objetivos de afinidade e segmentação semântica. O autor ainda cita que foram necessários 131 parâmetros a serem passados para a camada densa de neurônios. O principal objetivo do trabalho foi ajustar o mapa de segmentação semântica de uma imagem através do uso de uma nova camada de caminhada aleatória no modelo de rede neural tradicional (*CNN*).

Em (TAN; SONG; BOULARIAS, 2020), foi criado um sistema capaz de detectar objetos em vídeos de maneira robusta, podendo detectar categorias novas e não conhecidas nos quadros dos vídeos (*self-learning*). Os objetos contidos nos *frames* são segmentados, com o intuito de formar uma rede de conhecimento em formato de grafo. A estrutura de grafo terá um peso para cada aresta de seu conjunto, o peso é calculado através da semelhança entre os objetos. O grafo é usado como amostra de 3 vértices semelhantes e diferentes, tal processo de verificação entre os vértices é feito através do método de caminhada aleatória. Após a verificação dos vértices, o grafo é passado para uma rede siamesa, que projetará os recursos visuais genéricos em uma baixa dimensão.

No trabalho de (NIKOLENTZOS, 2019), é citado que na maioria das redes

neurais em grafos é utilizado o método de propagação de informação entre os nós de um grafo, recebendo como parâmetro uma camada de neurônios totalmente conectada. O autor cita que, o foco do trabalho é tornar a estrutura de GCNs mais intuitiva e transparente. O método basicamente utilizada de caminhadas aleatórias para comparar o grafo de entrada com o conjunto de neurônios da primeira camada de treinamento, buscando produzir representações em formato de grafo.

Em (ABBAS; ABDELSAMEA; GABER, 2020), foi abordada a técnica de detecção de regiões de interesse em imagens com redes neurais convolucionais. As regiões de interesse das imagens eram partes que poderiam ser referentes a problemas causados pela COVID-19. O autor utilizou da estrutura de rede DeTraC (*Decompose, Transfer and Compose*), tal rede é capaz de lidar com anotações que apontam a parte da imagem onde é necessário a inferência. Os experimentos realizados mostraram que a estrutura foi capaz de detectar casos de COVID-19, utilizando imagens coletadas de vários hospitais do mundo.

Em (CAMPOS et al., 2021), foi apresentada uma técnica de classificação de imagens contextuais utilizando redes neurais em grafos e também podando arestas que contextualizavam as imagens processadas. Tal projeto é similar ao projeto aqui apresentado, no entanto, os métodos de poda são diferentes, tais como: MST e poda por peso. Foram utilizadas algumas das bases processadas no presente trabalho (VRD, UnRel, Mit67).

Relatadas as características dos trabalhos relacionados, na Tabela 4 serão comparados os trabalhos citados com a presente proposta.

Tabela 4 – Comparação entre os trabalhos correlatos e o projeto proposto.

Título	Autores	Resumo	Comparação
<i>Efficient Representation Learning Using Random Walks for Dynamic Graphs</i> - (2019)	Hooman P. Sajjad, Do-cherty A., Yuriy Tyshetskiy	Algoritmos eficientes para aprendizagem de representação de vértice baseados em passeio aleatório	Método proposto pode ser utilizado como parte da abordagem proposta nesse projeto, porém, nesse trabalho, não há relações com grafos convolucionais e nem sua performance em imagens.
Continua na próxima pagina...			

Tabela 4 Comparação entre os trabalhos correlatos e o projeto proposto.

Título	Autores	Resumo	Comparação
<i>HiCoRe: Visual Hierarchical Context-Reasoning</i> - (2019)	Pedro H. Bugatti, Priscila T. M. Saito, Larry S. Davis	Criação de uma estrutura hierárquica baseada no contexto de imagens avaliadas por GCN	Estrutura hierárquica liga todos objetos da imagem entre si, não comparando qual a relevância de seus objetos para o rótulo, para então firmar a relação entre os objetos.
<i>Semi-Supervised Classification with Graph Convolutional Networks</i> - (2016)	Thomas N. Kipf, Max Welling	Abordagem escalável para aprendizado semi supervisionado estruturado em grafos convolucionais	Abordagem se aplica ao método proposto, porém, não foi aplicado sua performance com imagens, apenas ligações entre contextos de artigos científicos.
<i>Cross-Species Protein Function Prediction with Asynchronous-Random Walk</i> - (2019)	Zhao, Yingwen; Wang, Jun; Guo, Maozu; Zhang, Xiangliang; Yu, Guoxian	Método proposto para predição de funções de proteínas de maneira intra espécie e inter espécie, utilizando passeio aleatório assíncrono.	A abordagem proposta pode ser altamente aplicável ao projeto em questão, devido ao seu uso de estruturas em grafos para simular a ontologia dos genes para prever as proteínas. Método também utiliza de adaptação do passeio aleatório que pode ser aplicado ao problema do contexto das imagens.
<i>Convolutional Random Walk Networks for Semantic Image Segmentation</i> - (2017)	Gedas Bertasius and Lorenzo Torresani and Stella X. Yu and Jianbo Shi	Método baseado em caminhadas aleatórias para detectar regiões fragmentadas e mal delimitadas em segmentações, utilizando rede neural convolucional tradicional.	Neste trabalho, foi criado uma camada de rede neural convolucional, que usa de caminhadas aleatórias, para melhor segmentar os objetos das imagens, através da otimização dos objetivos. Comparando-o com o objetivo do presente trabalho, não há nenhum experimento referente a imagens contextuais, nem formação de grafos entre os objetos de interesse. A caminhada aleatória é utilizada apenas para otimizar a quantidade de pixels perdidos no momento da segmentação.

Continua na próxima pagina...

Tabela 4 Comparação entre os trabalhos correlatos e o projeto proposto.

Título	Autores	Resumo	Comparação
<p><i>A Self-supervised Learning System for Object Detection in Videos Using Random Walks on Graphs</i> - (2020)</p>	<p>Juntao Tan and Changkyu Song and Abdeslam Boularias</p>	<p>Sistema que detecta objetos em vídeos utilizando auto aprendizado e caminhada aleatórias para formar os grafos de aprendizado.</p>	<p>Tal sistema utiliza da caminhada aleatória para calcular a similaridade entre os objetos segmentados dos <i>frames</i> do vídeo. Comparando-o com o trabalho proposto, ambos buscando inferir os pesos das ligações de cada conexão, porém, no trabalho citado, a caminhada aleatória não é utilizada para melhorar a representação do contexto através dos grafos. No trabalho citado, os grafos representam o aprendizado feito pelas redes e não o contexto da imagem.</p>
<p><i>Random Walk Graph Neural Networks</i> - (2019)</p>	<p>Giannis Nikolentzos, Michalis Vazirgiannis</p>	<p>A intenção do autor é tornar as GCNs mais intuitivas. O método de caminhada aleatória, é aplicado no grafo de entrada, buscando ser possível a comparação entre os neurônios da primeira camada.</p>	<p>Tal abordagem apresenta traços semelhantes do trabalho proposto. Entretanto, o método aplicado, apenas busca tornar mais intuitiva a compreensão das GCNs. Não se é citado conjuntos de caminhadas aleatórias, nem a construção de alguma hierarquia que represente um determinado contexto em imagem ou outras estruturas.</p>

Continua na próxima página...

Tabela 4 Comparação entre os trabalhos correlatos e o projeto proposto.

Título	Autores	Resumo	Comparação
<i>Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network</i> - (2020)	Asmaa Abbas and Mohamed M. Abdelsamea and Mohamed Medhat Gaber	Método proposto para, detectar regiões de imagens de raio x de tórax que poderiam ter sido afetados pela infecção da COVID-19, através de redes neurais convolucionais com arquiteturas específicas (DeTraC).	Este trabalho utiliza de imagens de raio x de tórax como entrada para detecção de regiões afetadas pela infecção da COVID-19, através do uso de redes neurais convolucionais de arquiteturas diferentes da estrutura proposta no presente trabalho. Comparando-o com o trabalho proposto, pode ser feita a mesma detecção (infectado ou não infectado) nas imagens de raio x de tórax, porém, utilizando as técnicas abordadas no presente trabalho.
<i>Contextual Image Classification Through Fine-Tuned Graph Neural Networks</i> - (2021)	Walacy S. Campos, Luis G. Souza, Priscila T. M. Saito, Pedro H. Bugatti	Método proposto para diminuir o número de arestas de grafos convolucionais e classificar imagens contextuais.	Neste trabalho é proposto um método similar ao nosso, no sentido de diminuir o número de arestas para melhorar a performance do modelo GCN e sua classificação. No entanto, os métodos propostos são MST e poda por pesos, algo que se difere do método proposto por este projeto, que são podas realizadas através do passeio aleatório.

3 METODOLOGIA

Neste capítulo serão apresentados os assuntos relacionados a metodologia empregada ao decorrer do projeto, os modelos pré-treinados para extração de características, quais foram as arquiteturas selecionadas e suas individualidades. Uma subseção sobre a criação dos grafos, apontando pontos positivos e negativos de cada topologia trabalhada e também o algoritmo empregado para criação dos grafos. Por fim, a modelagem das redes neurais, ilustrando como as redes foram criadas e modeladas. A Figura 18, aponta a metodologia proposta para o trabalho.

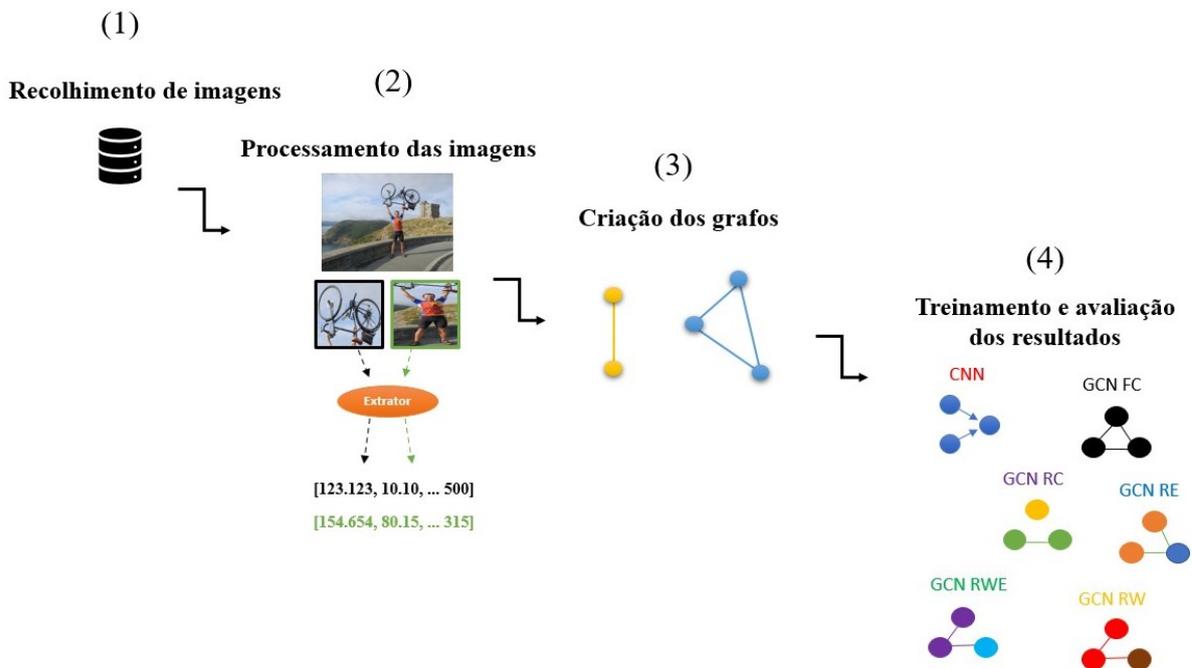


Figura 18 – Metodologia proposta para o trabalho

3.1 EXTRAÇÃO DE CARACTERÍSTICAS

Para utilizar a mesma estrutura de rede neural em grafos convolucionais apresentada em (KIPF; WELLING, 2016), (BUGATTI; SAITO; DAVIS, 2019) e (CAMPOS et al., 2021), pode-se utilizar a proposta de extrair características dos objetos das imagens. Portanto, as características extraídas serão utilizadas na criação da matriz X conforme apresentado na Equação 16 e A será a abordagem de criação de conexões.

Baseado em (BUGATTI; SAITO; DAVIS, 2019), as características foram ex-

traídas apenas dos objetos das imagens, ou seja, apenas das regiões de interesse da imagem (*bounding boxes* anotados do dataset), e não de toda a imagem. Dessa forma, podemos criar as conexões dos objetos que compõem a imagem, buscando através dessa estratégia, representar o contexto da imagem.

Como extrator de características, podem ser utilizadas as camadas de convolução das redes neurais pré-treinadas tradicionais. Como apresentado na subseção 2.3.1, a convolução em redes neurais serão capazes de captar características das imagens buscando diminuir o número de parâmetros treináveis para a camada de neurônios. Como estratégia, foram utilizadas as saídas da camada de convolução das redes pré-treinadas da literatura, visando que, diferentes arquiteturas geram diferentes vetores de características.

Para diversificar as características extraídas, foram utilizadas as arquiteturas: ResNet50 (He et al., 2015), InceptionResNetV2 (SZEGEDY et al., 2016), VGG16 (SIMONYAN; ZISSERMAN, 2014), EfficientNetB7 (TAN; LE, 2020), Xception (CHOLLET, 2016) e InceptionV3 (SZEGEDY et al., 2015). As arquiteturas citadas foram escolhidas devido a sua alta performance em classificação de objetos em imagens e sua alta capacidade de extrair características profundas com grande qualidade. Cada arquitetura escolhida possui um tamanho de vetor de características, conforme apresentado na Tabela 5.

Arquitetura	Tamanho do vetor
InceptionResNetV2	1536
ResNet50	2048
EfficientNetB7	2560
VGG16	512
Xception	2048
InceptionV3	2048

Tabela 5 – Tamanho dos vetores de características de acordo com as arquiteturas escolhidas.

Para exemplificar o processo de extração de características, apresentamos a Figura 19, em que temos uma imagem completa, os objetos contidos na imagem, o extrator de características e seus possíveis vetores X .

Conforme apresentado na Figura 19, o método de destaque dos objetos nas imagens foi o de recorte simples, ou seja, foram utilizadas as mesmas coordenadas disponibilizadas pela base para realizar o recorte do objeto. Nota-se que em ambos os recortes existem partes de outros objetos compondo a Figura (e.g. Figura de bicicleta possui partes da mão do objeto pessoa e o objeto pessoa possui partes do objeto



Figura 19 – Processo de extração de características de objetos de imagens. Figura da base de imagens UnRel, da classe *Bike and Person*.

bicicleta), essa técnica foi utilizada pois no momento de detectar comunidades queremos que os objetos que estão espacialmente conectados façam parte da mesma comunidade e não se agrupem apenas por características físicas e estruturais.

3.2 CRIAÇÃO DOS GRAFOS

Visto que os grafos terão grande atuação no processamento das redes neurais em grafos convolucionais, o presente projeto optou por aplicar 3 métodos novos, o método apresentado em (BUGATTI; SAITO; DAVIS, 2019) totalmente conectado e o método apresentado em (CAMPOS et al., 2021) cortes de arestas por pesos. Os métodos novos são: caminhada aleatória, corte aleatório e geração de arestas aleatória.

Utilizando um grafo G' para representar uma imagem I que contem O objetos, o número de vértices presentes no grafo é igual a $I(O)$ e o número de arestas pode variar de acordo com o método de construção escolhido descrito como γ . A definição do grafo é descrita na Equação 28.

$$G' = (I(O), \gamma) \quad (28)$$

Tendo como base os trabalhos (KIPF; WELLING, 2016) e (BUGATTI; SAITO; DAVIS, 2019), podemos assumir que cada imagem será um subgrafo G' do grafo G . Desta maneira, a construção do grafo G se dará pela somatória de I imagens em todos os subgrafos G' , conforme apresentado na Equação 29.

$$G = \sum_{j=1}^I G' \quad (29)$$

3.2.1 Grafo totalmente conectado - (FC)

O grafo totalmente conectado foi proposto no trabalho (BUGATTI; SAITO; DAVIS, 2019), onde não houve metodologia de podas das arestas, ou seja, as imagens que os grafos devem representar, estariam com todos os objetos conectados com todos. Essa abordagem pode ser precisa em imagens com baixo número de objetos ou datasets com pouca variabilidade. Caso uma imagem tenha vários objetos, essa abordagem pode se tornar computacionalmente custosa e de baixa precisão. A construção da topologia é apresentada em no pseudocódigo 1.

Algoritmo 1 Algoritmo totalmente conectado

```

1: function GRAFO TOTALMENTE CONECTADO(bounding boxes)
2:   v = bounding boxes;
3:   grafo = Grafo();
4:   grafo.adicionar.vertices(v)
5:   i = 0;
6:   j = 0;
7:   while i < v do
8:     while j < v do
9:       grafo.adiciona.aresta((i, j));
10:      j = j + 1;
11:     end while
12:    i = i + 1;
13:  end while
14:  return grafo
15: end function

```

Para auxiliar a compreensão em como a topologia será representada e computada, apresentamos a Figura 20, em que (a) é o grafo totalmente conectado e (b) a matriz de adjacência computada para o grafo G .

3.2.2 Grafo de poda aleatória - (RC)

O grafo de poda aleatória ou corte aleatório, será utilizado como um comparativo ao método de caminhada aleatória. Tendo em vista que, caso o método de caminhada aleatória obtenha resultados superiores ao de corte aleatório, podemos afirmar que a heurística na detecção de comunidades é valiosa. Portanto, o método de corte aleatório funcionará de maneira similar ao de caminhada aleatória, no entanto, esco-

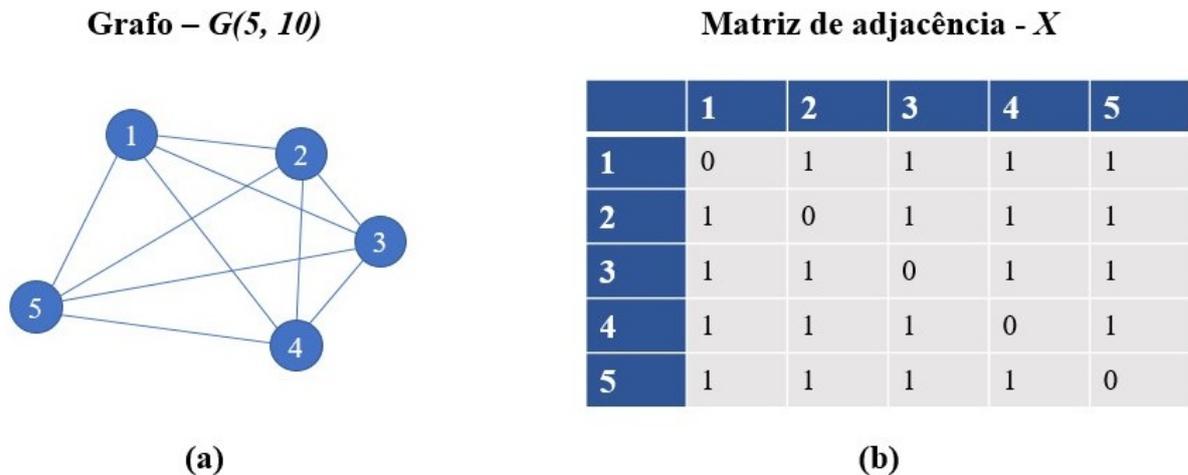


Figura 20 – Grafo totalmente conectado e sua matriz de adjacência.

lhendo arestas para serem cortadas aleatoriamente. Buscando manter a comparação justa, o algoritmo de corte aleatório será limitado a cortar o mesmo número de arestas que o método de caminhada aleatória. O método é apresentado no pseudocódigo 2.

Algoritmo 2 Algoritmo de corte aleatório

```

1: function GRAFO CORTE ALEATÓRIA(grafo totalmente conectado, grafo caminhada aleatória)
2:    $v =$  grafo totalmente conectado vértices;
3:   grafo = grafo totalmente conectado;
4:   arestas caminhada aleatoria = grafo caminhada aleatória arestas;
5:   arestas totalmente conectado = grafo totalmente conectado arestas;
6:   arestas podadas = arestas totalmente conectado - arestas caminhada aleatoria;
7:   if arestas podadas > 0 then
8:     while arestas podadas > 0 do
9:        $i =$  valor aleatório(0,  $v$ );
10:       $j =$  valor aleatório(0,  $v$ );
11:      aresta de poda =  $(i, j)$ ;
12:      grafo remova(aresta de poda);
13:      arestas podadas = arestas podadas - 1;
14:     end while
15:   end if
16:   return grafo
17: end function

```

O Algoritmo 2 mostra que as arestas são criadas através de escolhas aleatórias, no entanto, é tomado como parâmetro o número de arestas que foram podadas através do método de caminhada aleatória, tal parâmetro pode enviesar os cortes do grafo e não gerar a pluralidade desejada entre os métodos. Para ilustrar o formato do grafo e sua matriz de adjacência, a Figura 21 é apresentada em que (a) é o grafo e (b) a matriz de adjacência do grafo G .

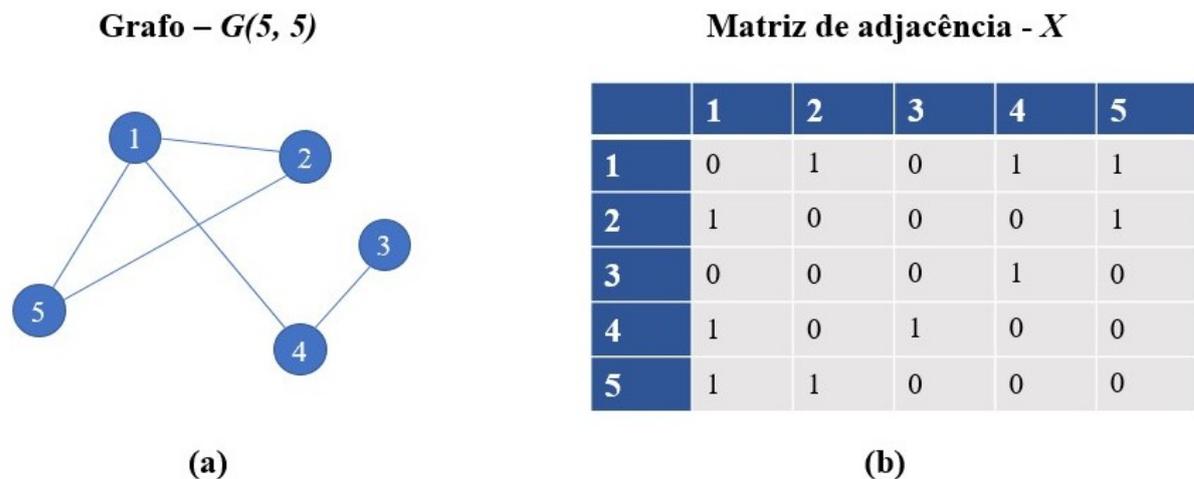


Figura 21 – Grafo com poda aleatória e sua matriz de adjacência.

3.2.3 Grafo com criação de arestas aleatórias - (RE)

Para eliminar um possível viés deixado pelo grafo de caminhada aleatória e tornar totalmente aleatória a criação das arestas para o grafo G , foi criado o algoritmo de criação de arestas aleatórias. Para este método, escolhemos uma semente aleatória que deve ser maior ou igual a zero e menor ou igual ao número de arestas do grafo totalmente conectado, conforme apresentado na Equação 30 em que G_{FC}^A é a quantidade de arestas do grafo totalmente conectado.

$$0 < \text{semente} < G_{FC}^A \quad (30)$$

A semente será a quantidade de arestas a serem criadas, tornando assim totalmente aleatória a escolha de criação. Para criar as novas arestas, foi utilizado a mesma política empregada no corte aleatório, escolhendo os vértices v_i e v_j aleatoriamente. O algoritmo 3 mostra o funcionamento da criação da topologia de arestas aleatórias.

A Figura 22, em(a), ilustra como o grafo de criação de arestas aleatórias ficará após o algoritmo ser executado, e em (b) mostra como ficará a matriz de adjacência do grafo G .

Algoritmo 3 Algoritmo de arestas aleatórias

```

1: function GRAFO DE ARESTAS ALEATÓRIA(grafo totalmente conectado)
2:   v = grafo totalmente conectado vértices;
3:   arestas = grafo totalmente conectado arestas;
4:   semente = valor aleatório(0, arestas);
5:   grafo = Grafo();
6:   while semente > 0 do
7:     i = valor aleatório(0, v);
8:     j = valor aleatório(0, v);
9:     aresta = (i, j);
10:    grafo adicione aresta(aresta);
11:    semente = semente - 1;
12:  end while
13:  return grafo
14: end function

```

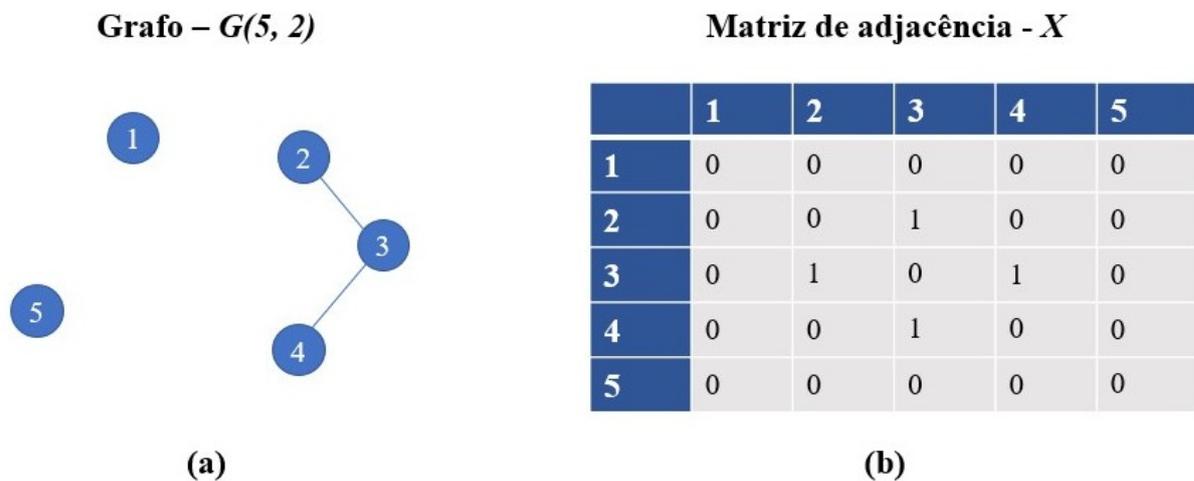


Figura 22 – Grafo com criação de arestas aleatórias e sua matriz de adjacência.

3.2.4 Grafo com poda a partir de limiar T - (RWE)

A topologia com poda por limiar T é uma das abordagens destacadas em (CAMPOS et al., 2021). Para efeito de comparação, tal topologia também foi criada para validar a quantidade de arestas a serem podadas e também para avaliar a qualidade das conexões do método de caminhada aleatória em comparação ao de poda por limiar T .

O método consiste em, dado um par de vértices v_i e v_j de um grafo totalmente conectado G_{FC} , caso a distância (peso da aresta) for superior ao limiar T , a aresta entre o par de vértices é cortada.

Para o limiar T , consideramos a mediana entre todas as distâncias das arestas do grafo totalmente conectado G_{FC} , como apresentado na Equação 31, λ é a distância entre os vértices v_i e v_j do grafo G_{FC} e μ é mediana das distâncias.

$$T = \mu(\lambda(G_{FC})) \quad (31)$$

Para exemplificar o funcionamento do algoritmo, apresentamos o pseudocódigo 4.

Algoritmo 4 Algoritmo de poda para limiar T

```

1: function GRAFO DE PODA POR PESOS(grafo totalmente conectado)
2:    $v =$  grafo totalmente conectado vértices;
3:   grafo = grafo totalmente conectado;
4:   arestas totalmente conectado = grafo totalmente conectado;
5:    $T =$  mediana das distancias(arestas totalmente conectado);
6:    $i = 0$ ;
7:    $j = 0$ ;
8:   while  $i < v$  do
9:     while  $j < v$  do
10:      if distancia entre  $(i, j) > T$  then grafo remove aresta( $i, j$ )
11:      end if
12:    end while
13:     $i = i + 1$ ;
14:  end while
15:  return grafo
16: end function

```

Como apresentado no pseudocódigo 4, o método de corte por limiar T pode ser interessante para manter apenas curtas distâncias. No entanto, caso uma imagem possua apenas objetos similares e nenhum que se destaque como diferente, o algoritmo vai perceber ainda assim que há necessidade de poda entre os elementos do grafo, algo que perante a heurística do método de caminhada aleatória não acontecerá, tendo em vista que o método busca detectar comunidades do grafo. Para visualizar como o grafo com poda em limiar T ficará, apresentamos a Figura 23, em que (a) é o grafo resultante do método e (b) é a sua respectiva matriz de adjacência.

3.2.5 Grafo de caminhada aleatória - (RW)

A abordagem principal para este trabalho é a otimização do grafo totalmente conectado através do algoritmo de caminhada aleatória. Tendo visto que o foco principal do presente trabalho é a melhora na classificação das imagens contextuais, poderiam ser utilizados outros métodos para otimização dos grafos, tal como a entropia. No entanto, caso tal método fosse utilizado, poderia aumentar a complexidade do algoritmo de criação dos grafos como é descrito em (POPOVIC, 2017).

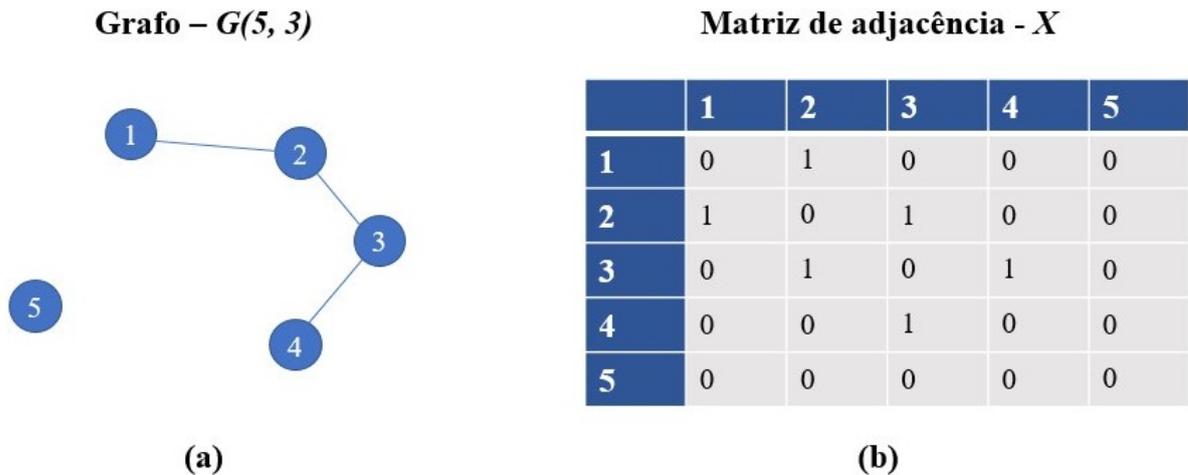


Figura 23 – Grafo com poda a partir do limiar T e sua matriz de adjacência.

Tomando como base os trabalhos (Li et al., 2015), (CARLETTI; FANELLI; LAMBIOTTE, 2021), (PONS; LATAPY, 2005) e (ZHOU; XU; DENG, 2020), foi construído um algoritmo capaz de identificar comunidades em grafos e representá-los de maneiras mais adequada, o pseudocódigo 5 apresenta o esboço do funcionamento do método.

Algoritmo 5 Algoritmo de caminhada aleatória

```

1: function GRAFO CAMINHADA ALEATÓRIA(grafo totalmente conectado, passo, pesos)
2:   grafo = grafo totalmente conectado;
3:   if vertices  $\geq$  3 then
4:     comunidades = caminhada-aleatória(grafo, passo, pesos);
5:     novo-grafo = Grafo();
6:     novo-grafo.adiciona.arestas(comunidades)
7:     grafo = novo-grafo;
8:   end if
9:   return grafo
10: end function

```

O método se baseia na solução proposta por (PONS; LATAPY, 2005) e (Li et al., 2015), em que o caminhador aleatório sempre partirá do ponto zero, ou seja, do vértice zero do grafo, dessa maneira podemos assumir que comparações entre os métodos são justas, já que o ponto inicial de todos é o mesmo. Nota-se que o método de caminhada aleatória só é aplicado em grafos com número igual ou maior a 3, pois para que possa haver uma otimização ou detecção de comunidades o grafo necessita ter no mínimo 3 vértices, ou seja, o grafo necessita apresentar similaridades a estrutura de ciclo.

Para computar as comunidades o método computa o tamanho de cada aresta e calcula a probabilidade do caminhador aleatório de passo P passar pelo vértice

v_i saindo do vértice v_j . Caso a probabilidade do caminhador aleatório de passo P passar por v_j seja alta, o algoritmo identifica que o par de vértices v_i e v_j são da mesma comunidade. As distâncias entre os vértices são representadas por pesos, ou seja, utilizamos a técnica de grafo valorado para computar o método de caminhada aleatória. Os pesos são distâncias euclídeas computadas entre cada objeto O_i e O_j da imagem I .

Após as probabilidades dos vértices pertencerem a comunidades serem computadas, o método devolve as comunidades encontradas no grafo G e podemos visualizar as probabilidades e distâncias utilizando o gráfico hierárquico dendrograma. Para visualizar como o grafo pode ficar após o método de caminhada aleatória ser computado, a Figura 24 é apresentada, em que (a) é o grafo e (b) a matriz de adjacência do grafo G .

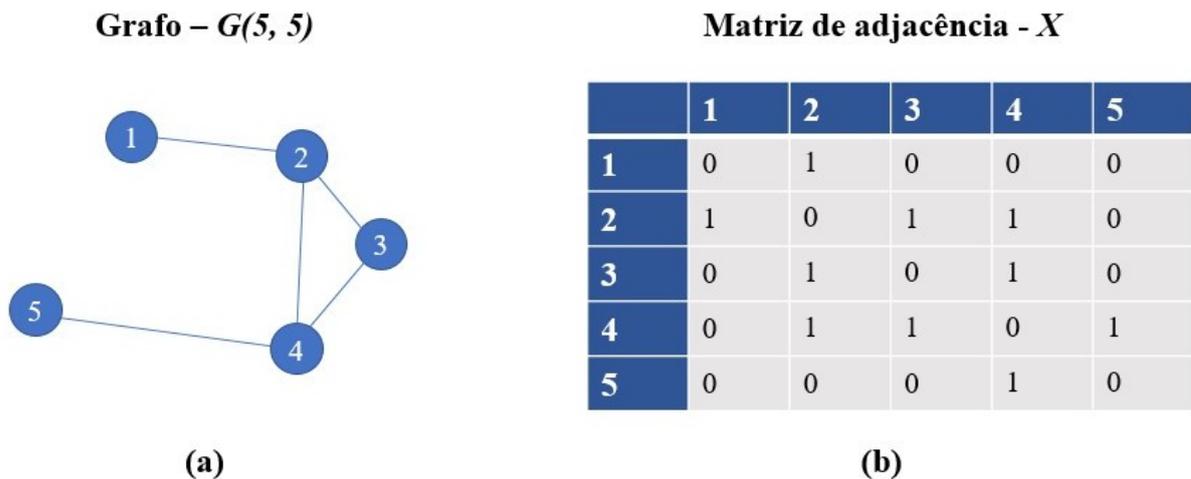


Figura 24 – Grafo com método de caminhada aleatória e sua matriz de adjacência.

3.3 MODELAGEM DAS REDES NEURAIIS

Para o treinamento com as estratégias apresentadas, é necessário a criação dos modelos de rede neural, sejam CNNs ou GCNs. Nesta seção serão apresentadas as informações referentes a modelagem das redes neurais para treinamento com as bases de imagens adotadas.

Para efeito de comparação, utilizamos as CNNs tradicionais da literatura (LI et al., 2020) e (ZANGENEH; RAHMATI; MOHSENZADEH, 2020a). Para que a comparação entre as estruturas seja justa, toda a modelagem é igual, ou seja, parâmetros

e quantidade de camadas são iguais para CNNs e GCNs. Sendo assim, o fator que diferenciará um resultado do outro é apenas a sua estratégia.

Para as redes neurais convolucionais, podem ser utilizados os pesos das redes pré-treinadas citadas em 3.1, com os pesos carregados podemos treinar uma camada oculta da rede e a camada de classificação utilizando o conjunto de treino escolhido, essa técnica é denominada de transferência de aprendizado (*transfer learning*) (SHAFABI et al., 2019). Portanto, o número de camadas de extração de características pode variar de acordo com o modelo pré-treinado utilizado.

Podemos dividir a modelagem das redes em pré-processamento e treinamento, como descrito na Figura 25. Nesta figura, nota-se que a imagem original é dividida em 4 objetos, denominados de *bounding boxes*, portanto, a rede será alimentada com os objetos presentes na imagem e não com a imagem completa. Após os dados serem carregados, passa-se para a construção do modelo de classificação, em que serão carregados os pesos pré-treinados na base de imagens ImageNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2017) e os *bounding boxes* utilizados passarão por essa camada para extração de suas características. Visando aumentar a produção de bons resultados, foram adicionadas duas camadas densas de 256 neurônios, região denominada de camada oculta. Por fim, a camada de classificação é operada com a função de probabilidade *softmax* para gerar os resultados da rede de acordo com os dados de entrada. A Figura 25 ilustra as camadas da rede neural convolucional tradicional (CNNs).

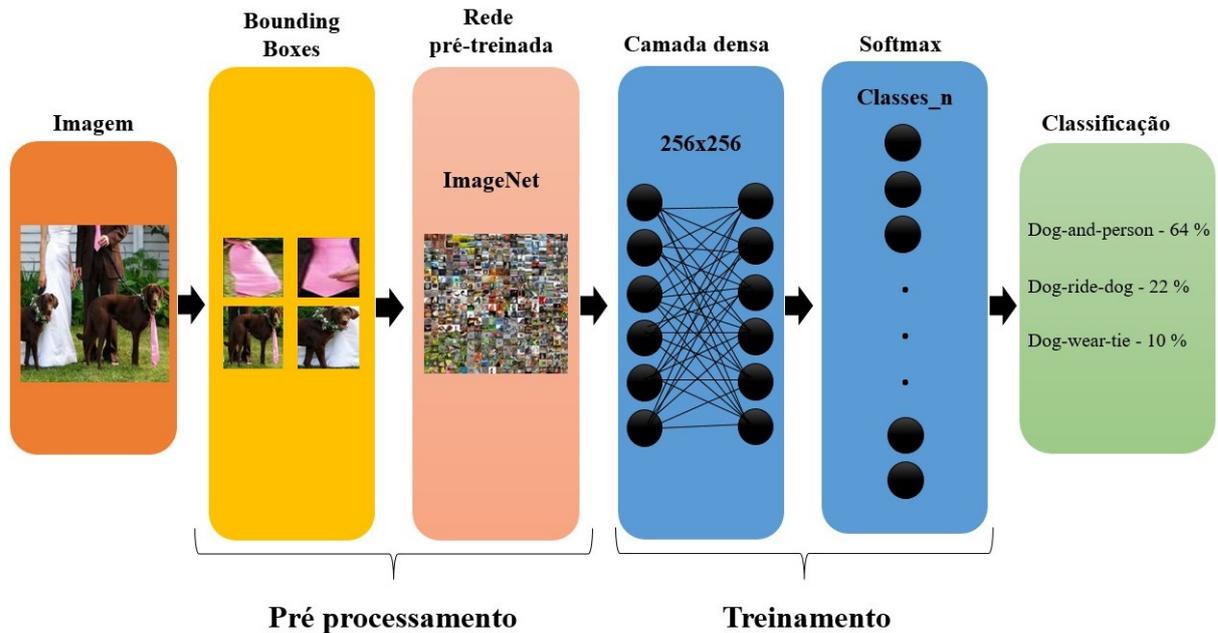


Figura 25 – Pipeline de treinamento de rede neural convolucional tradicional.

Para a construção dos modelos de GCNs, existem alguns passos anteriores. Para seguir o procedimento de treinamento proposto em (KIPF; WELLING, 2016) e (BUGATTI; SAITO; DAVIS, 2019), as imagens do dataset foram processadas em arquivos que possam ser abertos pelo algoritmo de treinamento da rede neural em grafos, tais como: matriz de características X ; matriz com os rótulos de cada amostra em X , denominado de Y , o grafo G para representar as conexões entre os objetos e onde estarão as estratégias propostas neste trabalho; os nós de teste para avaliar o modelo a cada época. Levando em consideração que os dados de entrada estarão nessa estrutura, o modelo de GCN pode ser construído.

Uma imagem I que contem O objetos, formará um grafo de O vértices para a GCN processar, sendo que o número de arestas pode variar de acordo com a topologia escolhida para treinamento. Como proposto em (KIPF; WELLING, 2016), (BUGATTI; SAITO; DAVIS, 2019) e (CAMPOS et al., 2021), a GCN terá duas camadas de convolução para aglomeração de características entre os seus vizinhos, utilizando informações espaciais como descrito na Equação 15, também foram processados os vizinhos de primeira ordem, como proposto em (KIPF; WELLING, 2016). Para a camada oculta, utilizamos a mesma quantidade proposta para as CNNs, duas camadas densas com 256 neurônios, formando a camada oculta. Para camada de classificação, utilizamos a função de probabilidade *softmax*. A Figura 26 mostra o modelo de GCN e seu processo de classificação.

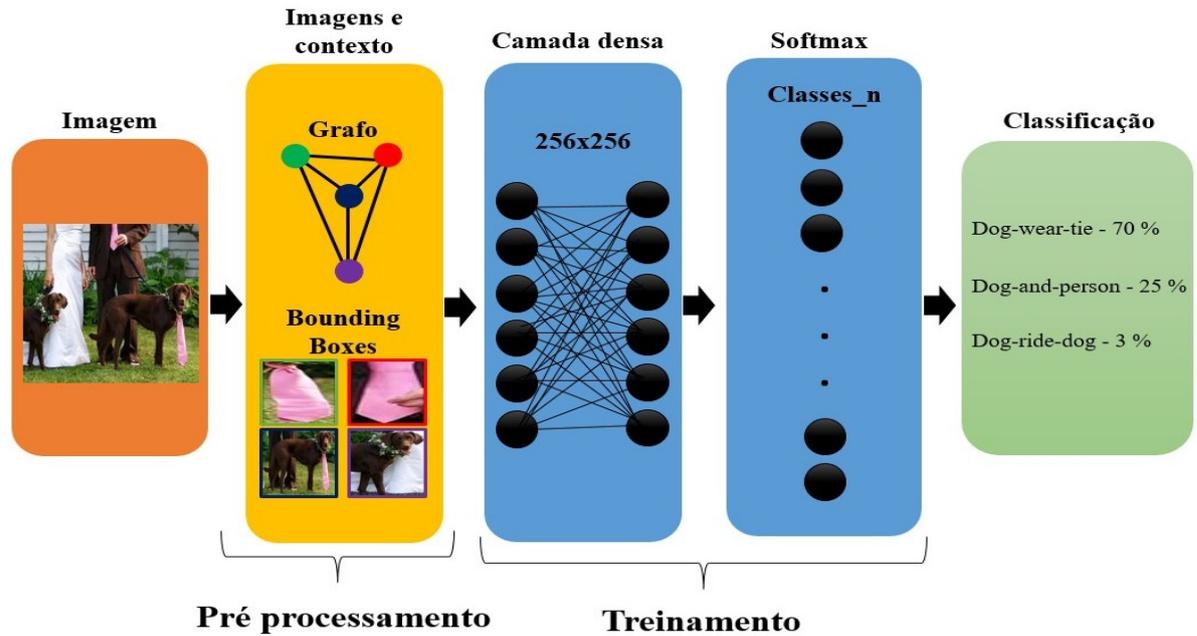


Figura 26 – Pipeline de treinamento de rede neural em grafos convolucionais.

Como foi descrito, ambas as redes são propostas diferentes para solucionar um mesmo problema, classificar imagens contextuais. Nota-se que ambos os modelos se limitaram aos mesmos parâmetros para que exista uma comparação justa entre ambas as propostas.

4 DESENVOLVIMENTO

Neste capítulo será destacado o desenvolvimento do projeto. Assuntos relacionados ao processamento das imagens nas bases, criação das estruturas e da distribuição das imagens nos treinamentos das GCNs e CNNs, a construção dos modelos utilizados e também será descrito como os treinamentos foram realizados, quais bibliotecas foram necessárias, como os resultados foram armazenados, como os modelos foram salvos para posteriormente haver a classificação das imagens e também os parâmetros utilizados para o treinamento dos modelos relacionados a cada experimento realizado.

4.1 PROCESSAMENTO DAS IMAGENS

Como destacado na revisão da literatura, as imagens são computacionalmente processadas através de matrizes e para destacar áreas das imagens é necessário apenas indicar as posições dos pontos mais acima à esquerda e mais abaixo à direita, sendo $P_E(x_1, y_1)$ o ponto mais alto a esquerda e $P_D(x_2, y_2)$ sendo o ponto mais baixo a direita.

As bases de imagens UnRel, Mit67, Malaria e VRD, foram todas processadas utilizando tal técnica de recorte por pontos na imagem. Os objetos das imagens foram recortados e salvos de acordo com a classe da imagem e não de acordo com as classes dos objetos. Todas os objetos recortados foram salvos em figuras com dimensões (224, 224), utilizamos este tamanho para ganhar tempo de processamento e para unificar o processo de carregamento dos dados, evitando erros nas bibliotecas. Em algumas imagens foi necessário tomar cuidado devido a presença do canal Alfa, presentes em imagens da base VRD, algo que poderia ocasionar problemas na extração de características, no entanto, simplesmente ignoramos o canal Alfa e utilizamos apenas os valores de RGB.

Após o recorte dos objetos, cada figura recortada foi processada pelos extractores de características. Cada figura gerou um vetor de características e os valores foram salvos em arquivos simples (e.g. .txt) para serem carregados no processo de distribuição das imagens.

Como apresentado na seção de extração de características, cada arquitetura utilizada gera um vetor composto por diferentes valores e de diferentes tamanhos, diversificando os impactos nos métodos que utilizarão os vetores. Um exemplo de processamento das imagens é apresentado na Figura 27.

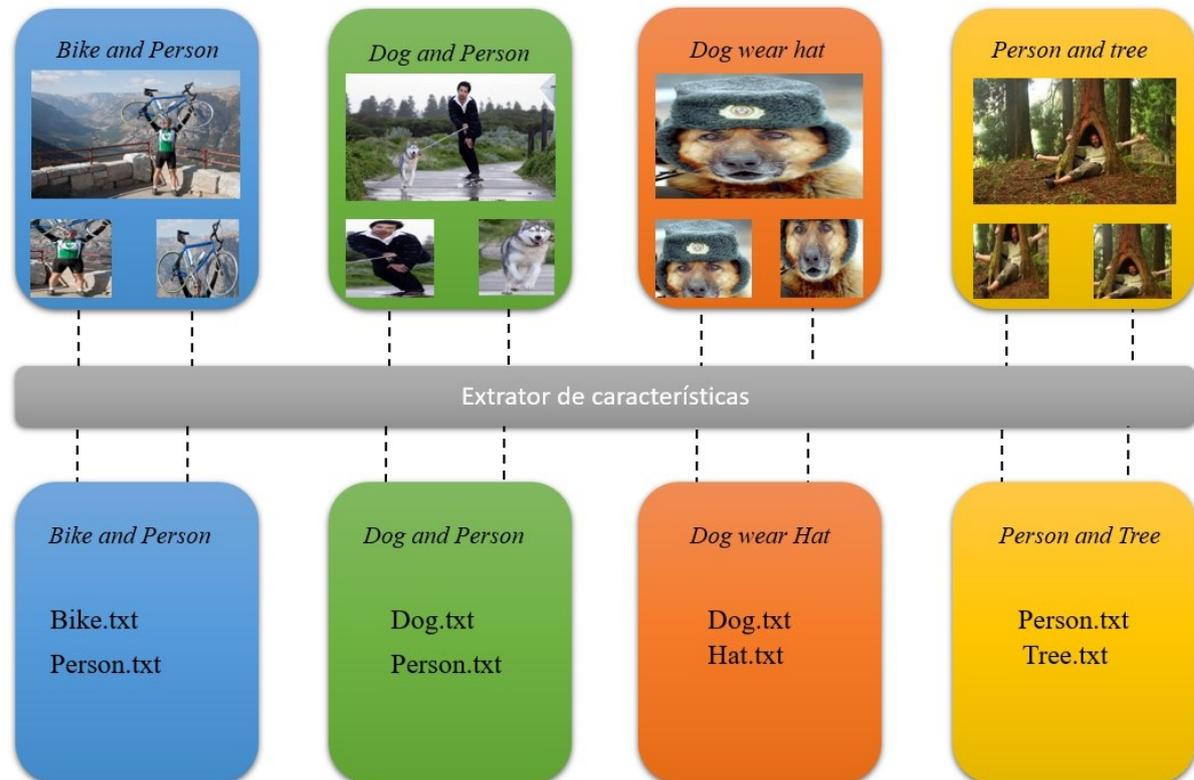


Figura 27 – Ilustração do pipeline de processamento das imagens e como os vetores de características serão salvos em cada classe. Na figura, cada caixa simula uma classe da base de imagens UnRel.

Portanto, nota-se que os objetos foram destacados através de técnicas simples de processamento de imagens digitais e cada objeto recortado passou pelos extratores de características, gerando um arquivo simples para armazenar os valores visando o uso das características no treinamento.

4.2 DISTRIBUIÇÃO DAS IMAGENS E CRIAÇÃO DAS ESTRUTURAS

Visto que cada imagem gera um grafo G' e o emaranhado de imagens da base pode ser descrito como o somatório de todos os grafos, formando G , ou seja, partimos da premissa que cada imagem forma um subgrafo G' do grafo G . Dessa maneira, os métodos de otimização de arestas não sofrerão impacto de imagens vizinhas. Para facilitar o entendimento desta distribuição, apresentamos a Figura 28, em que é ilustrado

como os grafos G são formados por G' .

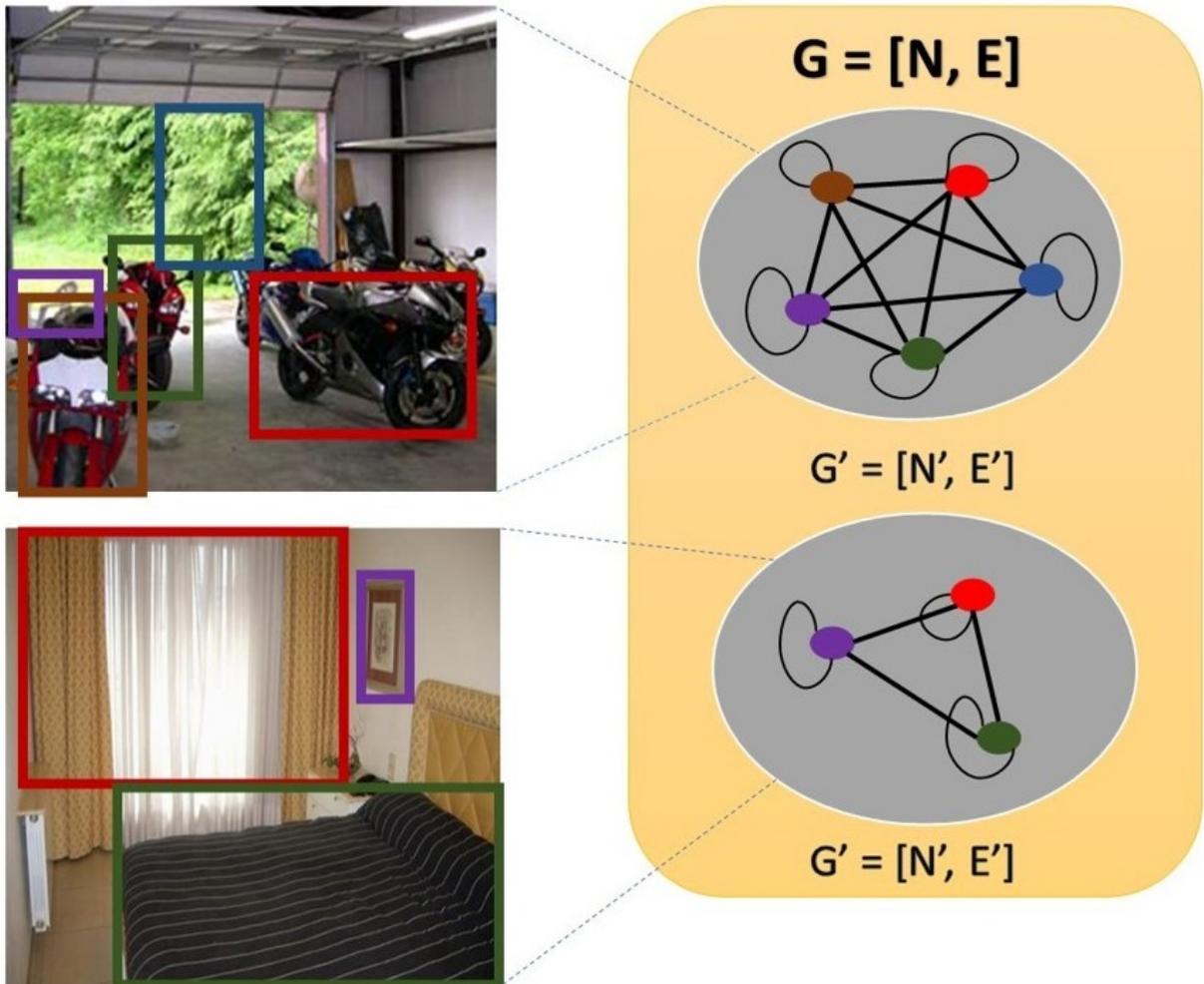


Figura 28 – Distribuição das imagens e como os sub-grafos G' são controlados em G . Neste exemplo é ilustrado como o grafo G , que contém N vértices e E arestas, é construído através do somatório de G' .

Para a distribuição dos dados, utilizamos 80% das imagens para treinamento e 20% para teste, optamos por não utilizar um conjunto de validação já que faremos distribuição em conjuntos distintos. O procedimento escolhido foi o *K-Fold cross validation*, método que nos ajuda a não ser impactado pelo conjunto de amostras presentes no teste e no treino, ou seja, queremos diversificar os conjuntos para que amostras de grande poder descritivo não tenham grande impacto no resultado final, buscando contornar possível *overfitting*. Sendo assim, foram construídos 3 conjuntos de dados distintos, ou seja, em nenhum dos 3 conjuntos foram utilizadas as mesmas amostras para treinamento e teste. Para exemplificar este procedimento, apresentamos a Figura29, que ilustra como as amostras são distribuídas para cada conjunto.

Para que fosse justa uma comparação entre as arquiteturas CNN e GCN, to-

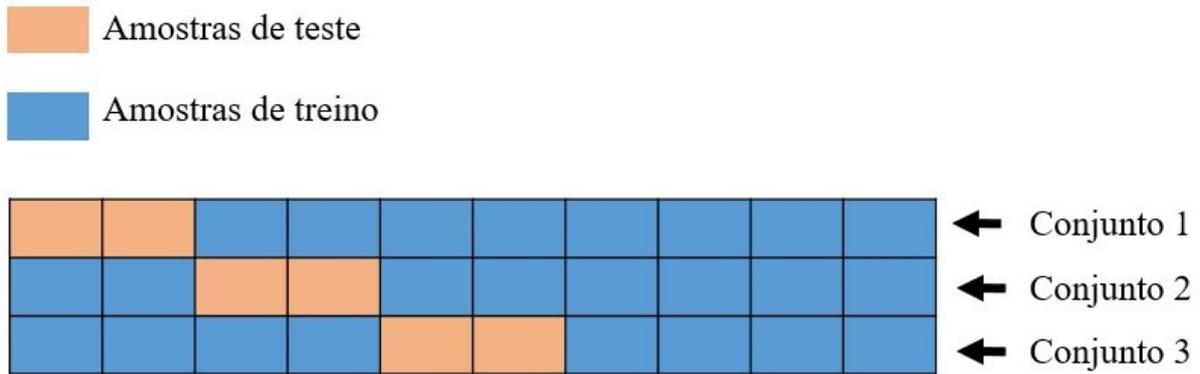


Figura 29 – Exemplo de *K-Fold cross validation*.

dos os treinamentos e testes foram feitos utilizando os mesmos conjuntos de treino e teste, ou seja, os 3 conjuntos de treino construídos foram utilizados para treinar tanto as GCNs quanto as CNNs, o mesmo para os conjuntos de teste.

Para treinamento e teste das CNNs, não houve grandes processamentos necessários, apenas foram criados os conjuntos e utilizamos os carregamentos disponíveis na biblioteca Tensorflow. Já para construção dos dados para treinamento das GCNs, foi necessário um cuidado maior, principalmente para seguir as mesmas estruturas apresentadas em (KIPF; WELLING, 2016), (BUGATTI; SAITO; DAVIS, 2019) e (CAMPOS et al., 2021). Criamos arquivos para matriz X que comportará tanto as amostras de treino quanto as amostras de teste. Para distinguir o que é treino e teste, foi criado um arquivo *test.index* que indicará quais são os vértices de teste no grafo G . E por fim, para representar as conexões, foram criados os arquivos *graph*, neles estarão todas os relacionamentos entre os objetos nas imagens. Cada topologia processada terá um arquivo *graph* diferente.

4.3 EXPERIMENTOS

No que se diz respeito aos experimentos, tanto os realizados em modelos CNNs quanto em GCNs utilizaram os mesmos parâmetros, com os mesmos conjuntos de treino e teste. Dessa maneira, todos os modelos podem ser comparados de maneira justa, onde a diferença entre os modelos é somente sua estrutura e a metodologia adotada para topologia dos grafos.

Para conseguir explorar ao máximo os métodos de otimização aleatória, optamos por utilizar 10 caminhadas para cada algoritmo de otimização abordado, ou seja, 10 caminhadas para RW, Poda Aleatória (*Random Cut*) (RC), Arestas Aleatórias (*Ran-*

dom Edge) (RE) e Corte por pesos (*Random Weighted*) (RWE). O valor 10 foi definido após ser analisado que os modelos não melhoravam seus resultados em caminhadas acima de 10, mantendo o resultado de 10 ou obtendo resultados baixo ao esperado.

Para cada experimento, optamos por combinar vários parâmetros, para aumentar a diversidade dos experimentos e também descobrir em quais configurações os modelos se comportariam melhor. Sendo assim, utilizamos a técnica de *grid search*, onde o modelo era treinado e testado com diferentes parâmetros, em que é definido se uma configuração é melhor que a outra de acordo com o resultado obtido pela métrica de avaliação preferida.

Todos os modelos foram construídos com 2 camadas densas, contendo 256 neurônios em cada uma. O número de épocas para treinamento foi definido em 2000. No entanto, para evitar que o modelo caia em *overfitting* e saia do mínimo global, foi utilizada a técnica de *early stop*, onde a cada época é medido se uma determinada métrica está ou não fazendo progresso, caso o modelo não esteja mais convergindo durante um período, o treinamento é finalizado. Para *learning rate* foram utilizados os seguintes valores: 0.01, 0.001, 0.05, 0.005 e para *dropout*: 0.3, 0.5, 0.9.

Como foi citado na seção 4.2, foram criados 3 conjuntos distintos de treino e teste. Dessa forma, ao final do experimento, obtínhamos 3 resultados oriundos dos 3 conjuntos de teste criados. Para manter um único valor para cada experimento, calculamos a média entre os valores obtidos, ou seja, se os resultados de acurácia para um experimento performaram em 50% no primeiro conjunto, 57% no segundo e 60% no terceiro, o resultado final de acurácia para tal experimento será de 55.66% com desvio padrão de 4.18% entre os conjuntos. Esse procedimento é o mesmo adotado para precisão, revocação e f1-score.

Para avaliar a performance dos modelos, utilizamos sempre os conjuntos de teste, dessa forma, teremos valores mais coniventes com a realidade de um modelo de classificação.

Tendo em vista que o presente projeto visa melhorar a classificação das imagens contextuais, cada experimento necessita retornar os resultados referentes a cada classe da base de imagens utilizada. Para isso, optamos por apresentar a acurácia, precisão, revocação e f1-score de cada classe da base, seguindo a mesma estrutura de média e desvio padrão para obter os resultados. As métricas são calculadas a partir da matriz de confusão, geradas pela avaliação do modelo perante ao conjunto de

teste usado.

Foram utilizadas as bibliotecas *PyTorch* para criação dos modelos GCN e *TensorFlow* para os modelos CNN. Para cada avaliação realizada, a melhor versão do modelo era salva para ser utilizada posteriormente na classificação de novas imagens. Buscando avaliações qualitativas dos modelos, realizamos também classificações utilizando imagens que não foram utilizadas nem para treino e nem para teste, dessa maneira, podemos ver o poder de generalização dos modelos testados de acordo com sua estrutura e topologia.

Ao todo foram somados 72 experimentos realizados com a estrutura de CNNs para cada base de imagens, levando em consideração cada treinamento realizado por conjunto, foram executados 216 treinamentos com as CNNs em cada base de imagens. Em relação às GCNs a quantidade de experimento foi maior por ser o foco principal do trabalho. Foram executados cerca de 2952 experimentos com a estrutura GCN em cada base de imagens, totalizando 8856 treinamentos.

Durante os treinamentos, foram utilizadas 3 máquinas para acelerar e aumentar a produtividade das execuções. Todas as máquinas foram disponibilizadas pela UTFPR (Universidade Tecnológica Federal do Paraná). As 3 máquinas utilizavam o sistema operacional Linux Ubuntu 20.04, a máquina 1 possuía placa de vídeo dedicada NVidia 1080TI com 12 gigabytes de memória, a máquina 2 possuía placa de vídeo dedicada NVidia 2080TI também com 12 gigabytes de memória e a máquina 3 possuía placa de vídeo dedicada NVidia Titan também com 12 gigabytes de memória.

4.3.1 Bases de imagens

Nesta seção serão detalhados os assuntos relacionados às bases de imagens utilizadas para treinamento, teste e classificação ao longo do projeto. Para a finalidade do projeto, foram utilizadas bases em que o conjunto de objetos presentes na figura indicam o rótulo, tornando a imagem classificada pelo contexto apresentado. Por exemplo, na Figura 30 é ilustrado como o conjunto de objetos presentes constituem o contexto geral da imagem (quarto).

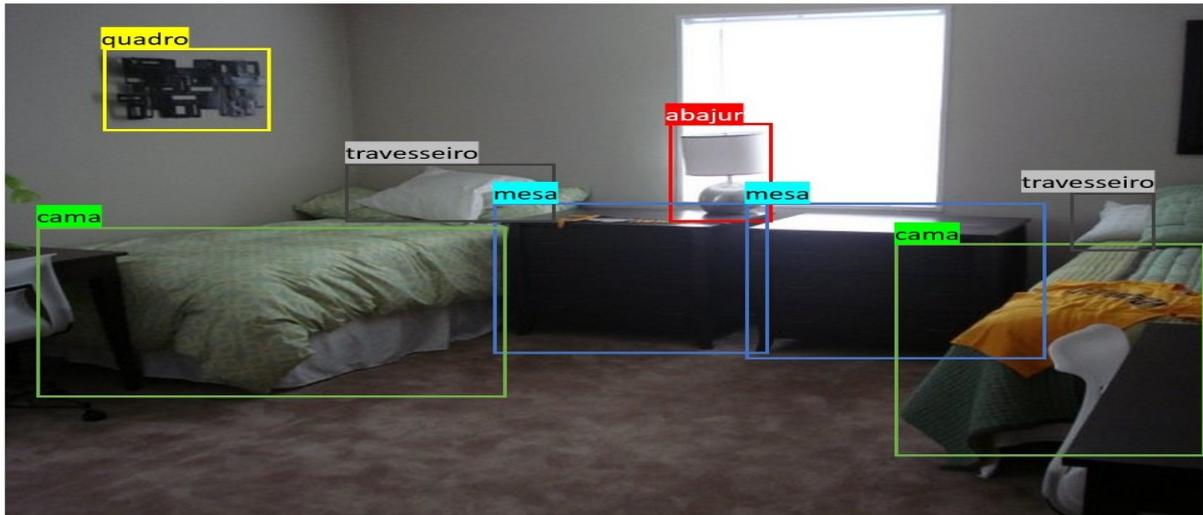


Figura 30 – Exemplo de como os objetos presentes constituem o rótulo da imagem. Esta Figura é da base de imagem Mit67, da classe *bedroom*.

4.3.1.1 UnRel

Em (PEYRE et al., 2017), foi proposta uma abordagem para modelar relações visuais entre pares de objetos, denominando de relação trinca contendo: sujeito, predicado e objeto. Onde o predicado é uma preposição ('acima', 'a frente') ou um verbo ('usa', 'anda') buscando ligar os objetos ou sujeitos. Neste trabalho, a proposta foi identificar as relações entre objetos das imagens através da aproximação entre os mesmos. Aprender tais relações pode ser um desafio pelo fato de que os objetos possuem diferentes configurações e aparências variando de acordo com a relação em que ela ocorre.

O intuito do uso desta base de imagens se dá pelo fato de que as relações entre os objetos estão anotadas e a classe de cada uma dessas relações está rotulada, facilitando o uso e pré-processamento. A Figura31 ilustra como os objetos estão anotados nas imagens e a classe das relações entre esses objetos.

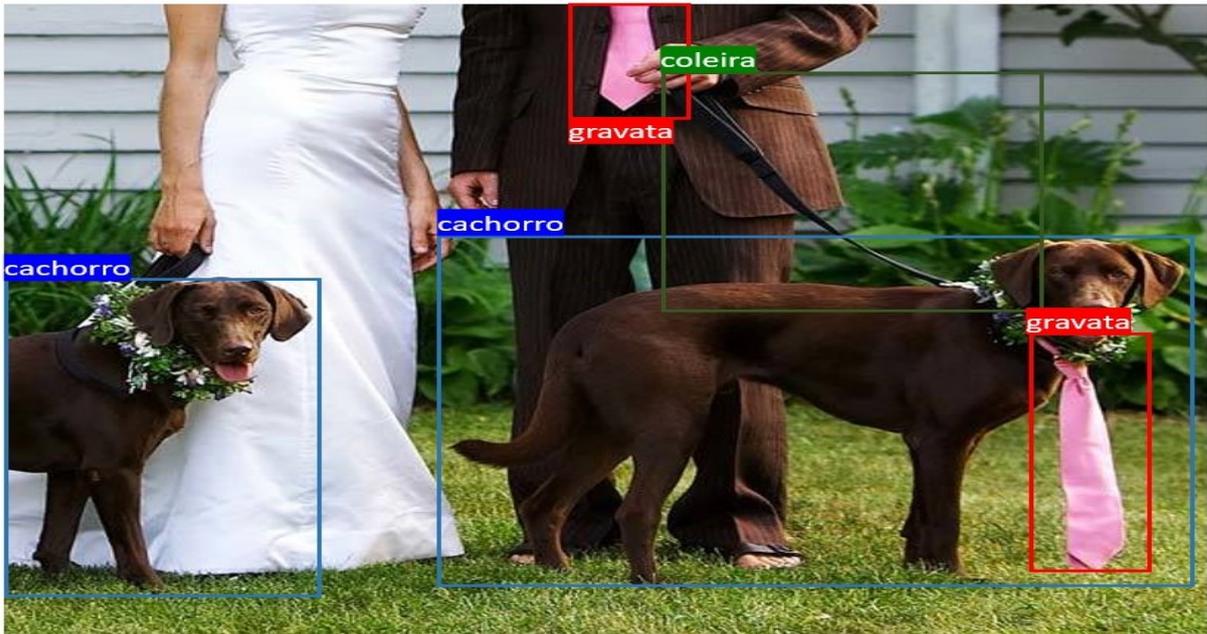


Figura 31 – Anotação dos objetos no dataset UnRel, a imagem ilustrada é da classe *dog wear tie*.

Portanto, podemos analisar que tal dataset disponibiliza imagens com o intuito de classificar as relações entre seus objetos. O dataset possui 1071 imagens para 57 classes de relações distintas, totalizando 2156 objetos nas imagens. Originalmente a quantidade de imagens é maior, no entanto, algumas imagens possuíam apenas um objeto anotado, fazendo com que a necessidade de aplicar tal imagem em grafos fosse nula.

4.3.1.2 Mit67

O dataset Mit67 foi criado originalmente no trabalho (Quattoni; Torralba, 2009), com o intuito de reconhecer cenas internas e reafirmar que reconhecimento de cenas internas é um problema aberto desafiador na visão de alto nível. Como está no nome, a base de imagens utiliza 67 classes de imagens em ambientes internos, que se dividem em 5 classes de ambientes: loja, casa, locais públicos, local de trabalho e locais de lazer. De acordo com (Quattoni; Torralba, 2009), o dataset possui 15620 imagens, divididas em 67 classes para 5 ambientes internos diferentes, é afirmado que todas as imagens possuem resolução mínima de 200 pixels.

Seguindo a mesma estrutura de dados apresentada em (BUGATTI; SAITO; DAVIS, 2019), algumas classes foram extraídas por erros nas anotações, sendo as seguintes classes: “auditorium”, “bowling”, “elevator”, “jewellery shop”, “locker room”,

“hospital room”, “restaurant kitchen”, “subway”, “laboratory wet”, “movie theater”, “museum”, “nursery”, “operating room”, “waiting room”. Portanto, foram utilizadas 53 classes das 67 disponibilizadas pelo dataset.

Para analisar a distribuição das imagens para cada ambiente, é apresentada a Tabela 6, em que estão os 5 ambientes apresentados no dataset, a quantidade de classes em cada ambiente, número de imagens por cada ambiente e número de objetos totais (BUGATTI; SAITO; DAVIS, 2019).

Ambientes	Classes	Imagens	Objetos
Locais de lazer	9	228	4,995
Locais de trabalho	11	477	9,902
Casas	13	1,390	25,074
Lojas	12	284	6,945
Locais públicos	8	232	3,952

Tabela 6 – Ambientes e a distribuição das imagens para cada ambiente na base Mit67.

A Figura 32, mostra um exemplo de como as imagens do dataset são formadas em ambientes internos.

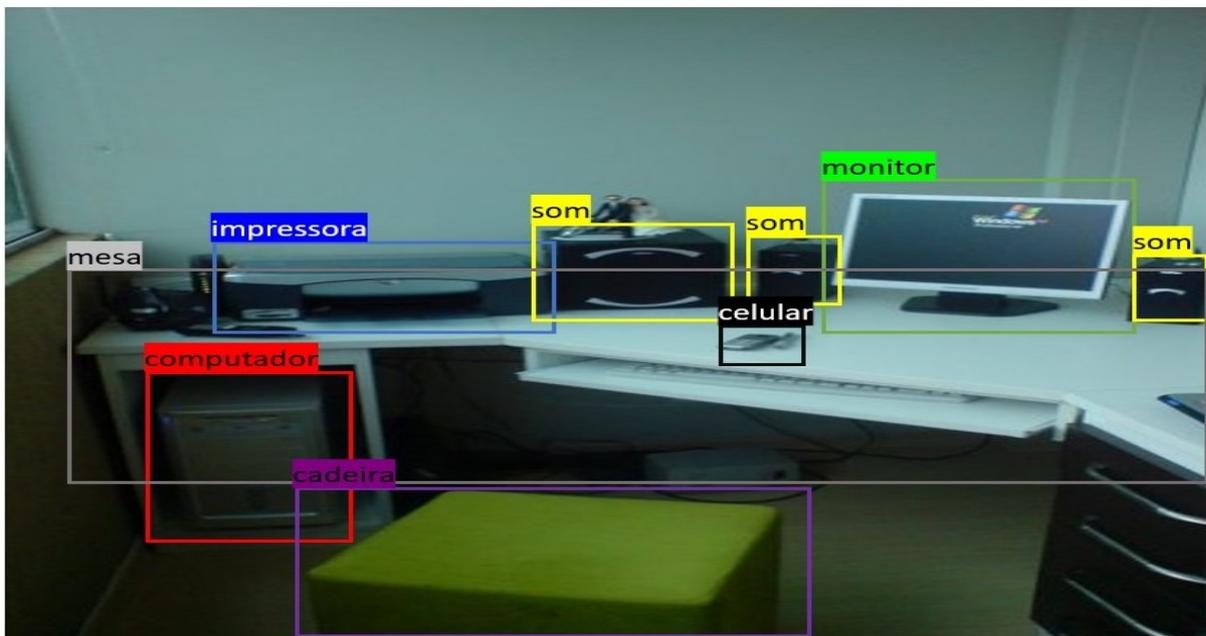


Figura 32 – Exemplo de imagem do dataset Mit67, a classe da imagem é *computer room*.

A principal utilidade deste dataset é conseguir prever as relações dos objetos de acordo com suas classes e não seu ambientes gerais. Portanto, as classificações serão feitas para cada classe do dataset. Em comparação ao dataset UnRel, o Mit67 terá mais imagens para processar, sendo uma boa alternativa para validar o comportamento das estratégias de corte dos grafos de representação.

4.3.1.3 Malaria

O dataset Malaria foi proposto para realizar classificações de imagens médicas proposto por (MISHRA, 2021) e também utilizado em (SCHULER et al., 2022), com o dataset criado por (LJOSA; SOKOLNICKI; CARPENTER, 2012), ambas abordagens buscam classificar imagens de células humanas em infectado e não infectado. Como apresentado na Figura34, as células da imagem podem ser destacadas através de anotações disponibilizadas pelo dataset.

De acordo com a Fundação Oswaldo Cruz, malária é uma doença infecciosa, febril, aguda e potencialmente grave. Ela é causada pelo parasita do gênero *Plasmodium*, transmitido ao homem, na maioria das vezes pela picada de mosquitos do gênero *Anopheles* infectados, também conhecido como mosquito-prego (BRASIL, 2009).

A base de imagens de células com malária é uma proposta para analisar o comportamento dos modelos com conjuntos de dados desbalanceados, visto que neste dataset a quantidade de imagens com células não-infectadas é significativamente maior que o número de imagens de células infectadas, conforme apresentado na Tabela 7.

Classe	Número de imagens	Número de objetos
Infectado	209	9,581
Não infectado	1119	76,454

Tabela 7 – Quantidade de imagens para cada classe na base Malária.

De acordo com (LJOSA; SOKOLNICKI; CARPENTER, 2012), as possíveis infecções detectáveis no dataset se dão pelo reconhecimento das células gametócitos, anéis, trofozoítos e esquizontes. As células não infectadas são as células vermelhas ou hemácias e leucócito. A Figura33 ilustra a coloração e formato das células infectadas e as não infectadas.

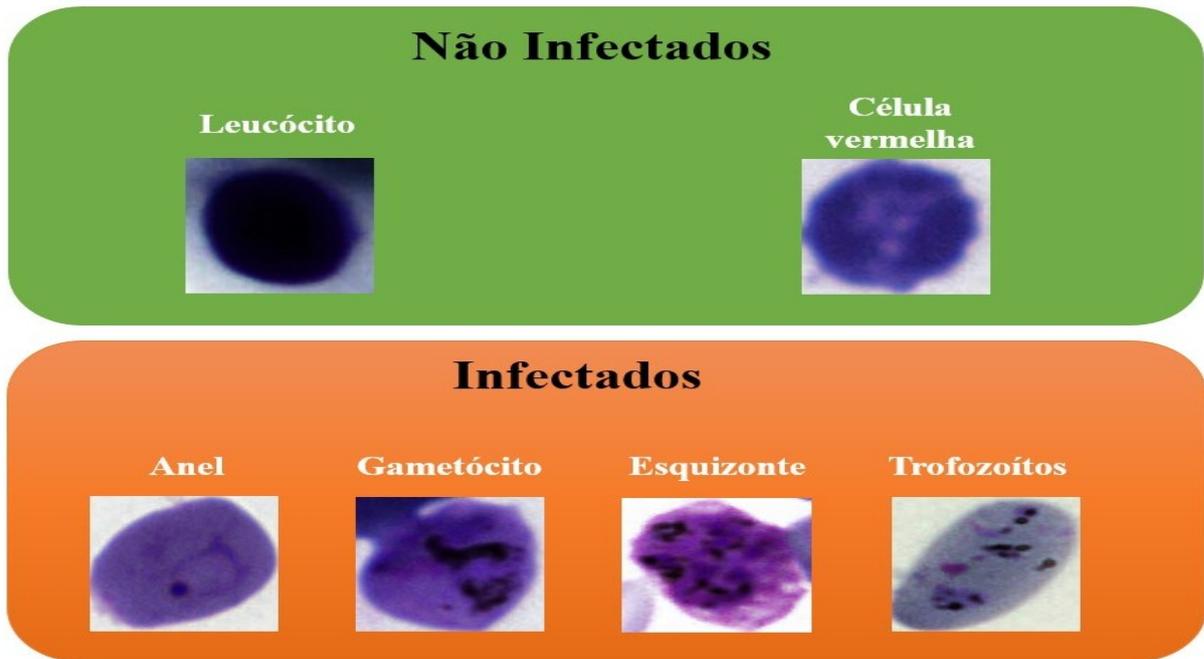


Figura 33 – As 6 variações de células presentes no dataset Malaria. Em verde as células não infectadas e em laranja as células infectadas.

Para ilustrar como as imagens médicas são compostas no dataset Malaria, a Figura 34 apresenta todas as células presentes na imagem e a anotação de cada célula para imagem.

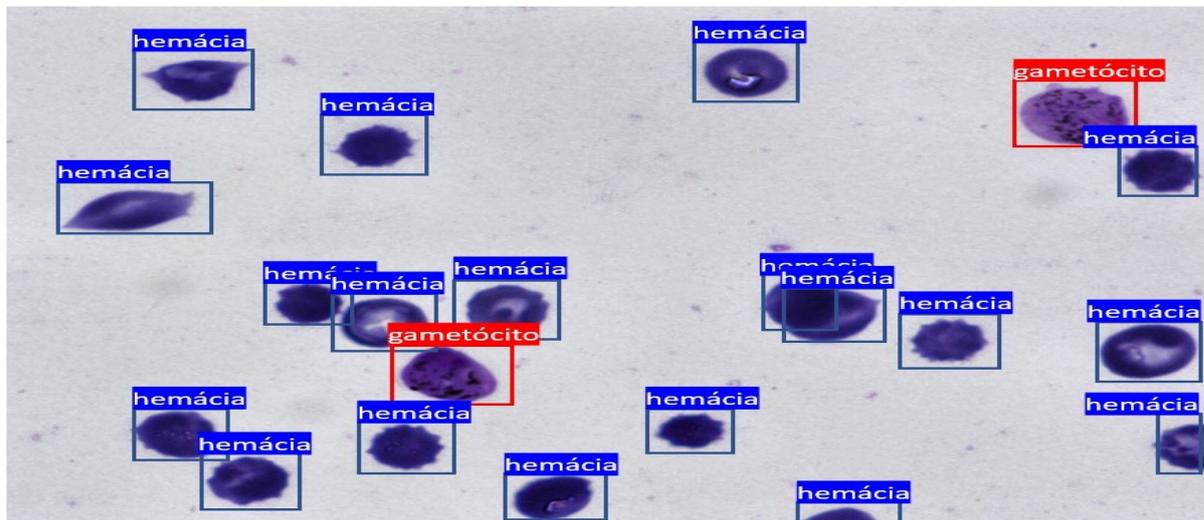


Figura 34 – Exemplo de imagem médica de células humanas, de acordo com (LJOSA; SOKOL-NICKI; CARPENTER, 2012) a imagem contém células denominadas de gametócitos, o que faria do conjunto de células presentes na imagem infectado.

4.3.1.4 VRD

O dataset VRD foi criado para sanar um problema de visão computacional com o objetivo de reconhecer relações ou interações entre objetos em uma imagem, se enquadra como uma tarefa de aprendizado adicional após o reconhecimento de objetos e é essencial para a compreensão completa das imagens, até mesmo do mundo visual (LU et al., 2016).

Como citado, o dataset VRD é muito útil para reconhecer os relacionamentos entre os objetos nas imagens. Sendo um dataset com várias imagens e com todas as imagens anotadas como necessário. O dataset é composto por 23 classes, em que cada classe comporta imagens contextuais e suas classes são descritas pelos relacionamentos das imagens (e.g. na frente de, perto, acima, abaixo). A Tabela 8 mostra cada uma das classes do dataset, a quantidade de imagens por classe e o número de objetos para cada classe.

Classe	Número de imagens	Número de objetos
above	204	3,463
against	16	298
attached-to	21	404
behind	324	5,951
below	73	1, 152
beside	63	1,195
by	44	756
carrying	17	308
has	932	17,161
holding	98	2,001
in	305	5,925
in-front-of	214	3,801
inside	16	254
left-of	37	737
near	117	1,932
next-to	230	4,213
on	1,217	23,607
on-top-of	36	678
over	49	837
right-of	21	422
under	468	9,425
wearing	29	550
with	151	2,626

Tabela 8 – Quantidade de imagens para cada classe na base VRD.

Visto que o dataset não é balanceado, ou seja, as classes não possuem a mesma quantidade de amostras, podemos analisar o comportamento das topologias propostas para esta base e verificar se há bons resultados.

Para ilustrar como as imagens são compostas no dataset VRD, a Figura35 é apresentada.

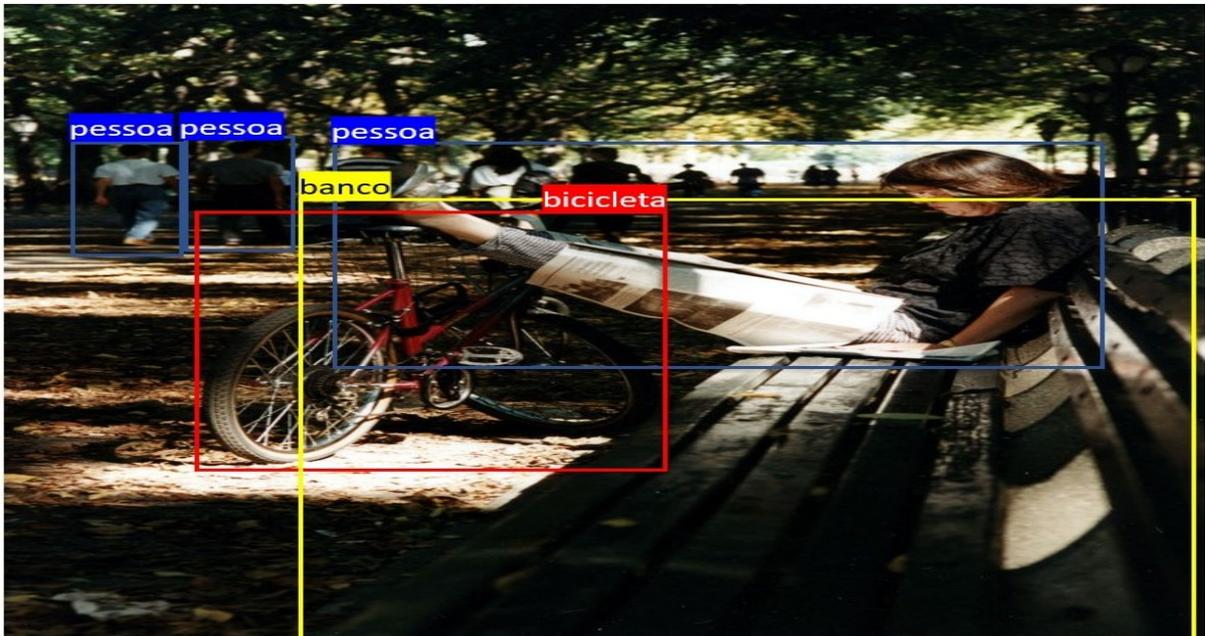


Figura 35 – Exemplo de relacionamento entre os objetos para o dataset VRD, a classe da imagem é *near*.

4.3.2 Resultados

Para apresentar os resultados obtidos ao longo dos experimentos realizados, separamos uma seção para discutir os resultados de cada política, arquitetura e estrutura de rede abordada ao longo do projeto. Os resultados serão introduzidos de acordo com a base de imagens utilizadas.

Para avaliar a performance de cada modelo treinado, utilizamos os resultados apenas dos conjuntos de teste, portanto, todos os resultados apresentados a seguir são oriundos dos conjuntos de teste.

As métricas de avaliação escolhidas para avaliarem a performance dos modelos foram: acurácia, precisão, revocação e média harmônica (F1-Score). Levando em consideração que foram criados 3 conjuntos de teste com imagens distintas, os resultados apresentados são médias percentuais calculadas a partir dos resultados de cada conjunto de teste criado, ou seja, foi analisado o resultado obtido com o conjunto de teste 1, posteriormente com o conjunto 2 e conjunto 3, por fim foi feita a média dos 3 conjuntos e esses serão os valores apresentados nesta seção. Para apresentar o

quanto a média pode variar, ao lado do valor está o desvio padrão, ambos os valores estão em porcentagem.

Os resultados são extraídos da última camada das redes neurais a *Softmax*. Em geral, queremos utilizar o conjunto de objetos e suas relações para classificar o *label* da imagem como um todo, ou seja, caso um objeto *table* esteja tanto em *living room* quanto em *computer room*, espera-se que os modelos classifiquem a imagem completa e não o objeto. Dentro desta perspectiva, a topologia das relações entre os objetos será avaliada, podendo ser capaz de extrair a classe da imagem tendo os objetos como entrada.

Para avaliar a qualidade dos modelos teremos como principal métrica a precisão, buscando medir a quantidade de verdadeiros positivos em relação aos falsos positivos. Dessa maneira a convicção de que determinada topologia obteve melhor performance será maior do que se utilizássemos apenas a acurácia. Portanto, nesta subseção, serão apresentados os melhores resultados obtidos por cada rede neural trabalhada (CNNs, GCNs).

Para apresentar os resultados por classes, foram ranqueadas as 3 classes que obtiveram os maiores valores de precisão. Complementando as métricas, também foram calculadas acurácia, revocação e média harmônica. Para analisar todo o conjunto de teste processado, serão apresentadas as matrizes de confusão dos experimentos que obtiveram os maiores valores de precisão. A matriz de confusão apresentará a média de acertos entre os 3 conjuntos de teste processados.

Os primeiros resultados apresentados são originados dos experimentos que obtiveram a maior precisão dentro da topologia utilizada para criação do grafo de treinamento da rede neural, portanto, os resultados serão apresentados de acordo com cada topologia e os respectivos modelos pré treinados extratores de características.

4.3.2.1 UnRel

ResNet50	Acurácia	Precisão	Revocação	F1-Score
CNN	12.81 \pm 0.00243	0.0059 \pm 0.00368	0.0191 \pm 0.00143	0.0057 \pm 0.00132
GCN FC	64.90 \pm 0.01738	50.37 \pm 0.01919	49.25 \pm 0.00720	47.46 \pm 0.01579
GCN RC	64.75 \pm 0.00416	54.00 \pm 0.05592	51.17 \pm 0.02839	50.09 \pm 0.03852
GCN RE	64.67 \pm 0.02438	55.49 \pm 0.02586	51.39 \pm 0.02418	50.49 \pm 0.02644
GCN RWE	58.75 \pm 0.01993	53.85 \pm 0.01948	49.20 \pm 0.02416	49.47 \pm 0.02544
GCN RW	63.90 \pm 0.00851	56.20 \pm 0.03955	51.52 \pm 0.01648	51.18 \pm 0.02775

VGG16	Acurácia	Precisão	Revocação	F1-Score
CNN	29.90 ±0.01822	20.75 ±0.053924	18.059 ±0.02527	16.50 ±0.03110
GCN FC	58.13 ±0.01500	46.16 ±0.00646	44.81 ±0.01687	43.02 ±0.00964
GCN RC	60.48 ±0.01181	51.32 ±0.02987	48.60 ±0.02930	47.23 ±0.02297
GCN RE	61.29 ±0.01172	51.26 ±0.00932	48.93 ±0.00899	47.93 ±0.00672
GCN RWE	58.35 ±0.00992	51.24 ±0.02962	47.97 ±0.02796	47.73 ±0.02633
GCN RW	61.88 ±0.02352	50.36 ±0.03424	49.76 ±0.02237	47.76 ±0.02197
InceptionResNetV2	Acurácia	Precisão	Revocação	F1-Score
CNN	46.00 ±0.02500	47.04 ±0.01180	37.55 ±0.01181	38.72 ±0.01412
GCN FC	61.07 ±0.01805	47.49 ±0.01751	47.69 ±0.01207	45.12 ±0.01312
GCN RC	61.44 ±0.01985	53.62 ±0.01436	49.71 ±0.00743	48.72 ±0.01148
GCN RE	62.03 ±0.04696	51.91 ±0.06877	48.08 ±0.05287	46.73 ±0.06266
GCN RWE	55.40 ±0.01651	53.10 ±0.01811	47.26 ±0.01323	47.94 ±0.01703
GCN RW	63.64 ±0.01849	55.30 ±0.02942	49.83 ±0.02100	49.73 ±0.02569
InceptionV3	Acurácia	Precisão	Revocação	F1-Score
CNN	44.56 ±0.00677	46.49 ±0.05692	34.51 ±0.00480	35.42 ±0.01450
GCN FC	56.95 ±0.01125	44.29 ±0.01873	42.64 ±0.01121	40.96 ±0.01644
GCN RC	57.54 ±0.00453	48.17 ±0.01375	44.07 ±0.00529	43.72 ±0.01034
GCN RE	55.77 ±0.02112	45.91 ±0.01182	42.60 ±0.00732	41.77 ±0.01008
GCN RWE	48.19 ±0.01303	43.47 ±0.04492	41.18 ±0.03318	40.77 ±0.03251
GCN RW	58.49 ±0.01407	49.18 ±0.03646	44.14 ±0.02076	44.41 ±0.02695
Xception	Acurácia	Precisão	Revocação	F1-Score
CNN	49.77 ±0.01095	50.90 ±0.04322	42.13 ±0.03013	43.06 ±0.01140
GCN FC	61.66 ±0.01303	46.08 ±0.01991	46.47 ±0.00492	43.58 ±0.01436
GCN RC	61.95 ±0.01985	50.91 ±0.00987	47.98 ±0.00806	46.67 ±0.00814
GCN RE	64.16 ±0.01574	50.84 ±0.02532	49.40 ±0.01234	48.04 ±0.01330
GCN RWE	55.11 ±0.01985	50.34 ±0.01249	46.80 ±0.02471	46.78 ±0.01932
GCN RW	64.16 ±0.03246	51.59 ±0.05471	50.34 ±0.02260	48.68 ±0.03574
EfficientNet	Acurácia	Precisão	Revocação	F1-Score
CNN	50.91 ±0.02379	49.57 ±0.01846	43.30 ±0.02285	43.87 ±0.01891
GCN FC	23.62 ±0.02256	16.37 ±0.01347	15.43 ±0.00792	14.29 ±0.01198
GCN RC	30.46 ±0.01430	22.26 ±0.02748	21.13 ±0.02341	20.43 ±0.02517
GCN RE	30.16 ±0.01172	21.38 ±0.02393	20.71 ±0.01568	19.61 ±0.01839
GCN RWE	39.22 ±0.03175	34.80 ±0.04066	30.56 ±0.04534	30.43 ±0.04350
GCN RW	27.74 ±0.00632	21.45 ±0.00805	18.96 ±0.00573	18.28 ±0.00320

Tabela 9 – Resultados globais obtidos na base de imagens UnRel.

De acordo com a Tabela 9, a maior precisão, em âmbito geral, da base UnRel, foi alcançada pela topologia GCN RW (*Random Walk*) com extrator ResNet50 e passo de tamanho 1, performando 56.20%. Porém, analisando o desvio padrão dos resultados, podemos notar um empate técnico entre as topologias de corte aleatório, dentro dos resultados do mesmo extrator, já que a precisão de GCN RW pode variar em 0.0395%. No entanto, o resultado obtido pela topologia GCN FC (*Fully Connected*) foi menor que a performance alcançada pela rede GCN RW mesmo considerando o

desvio padrão de ambas as redes, sendo que a rede GCN FC pode variar em 0.0119%. Dessa forma, pode-se dizer que, para a base UnRel a rede GCN RW obteve melhores resultados perante a rede GCN FC e utilizou menos arestas para construir o grafo de conexões, tendo em vista que a rede GCN FC utilizou 7054 arestas e a rede GCN RW contou com 5132 arestas, havendo uma queda de 27.24% no número de arestas. Vale ressaltar também a eficácia dos métodos em relação ao totalmente conectado na construção dos grafos, apresentado na tabela 10.

Método	Tempo (s)
FC	0.31
RC	0.27
RE	0.28
RWE	0.27
RW	0.27

Tabela 10 – Tempo para construção dos grafos de cada política de conexão.

Nota-se que os métodos de otimização foram mais eficazes que o grafo totalmente conectado, construindo o grafo global em menos tempo. A Figura 36 mostra a correlação entre a quantidade de arestas e a precisão performada pelo modelos treinados com o extrator ResNet50, que de acordo com a Tabela 9, obteve os maiores resultados para estrutura GCN.

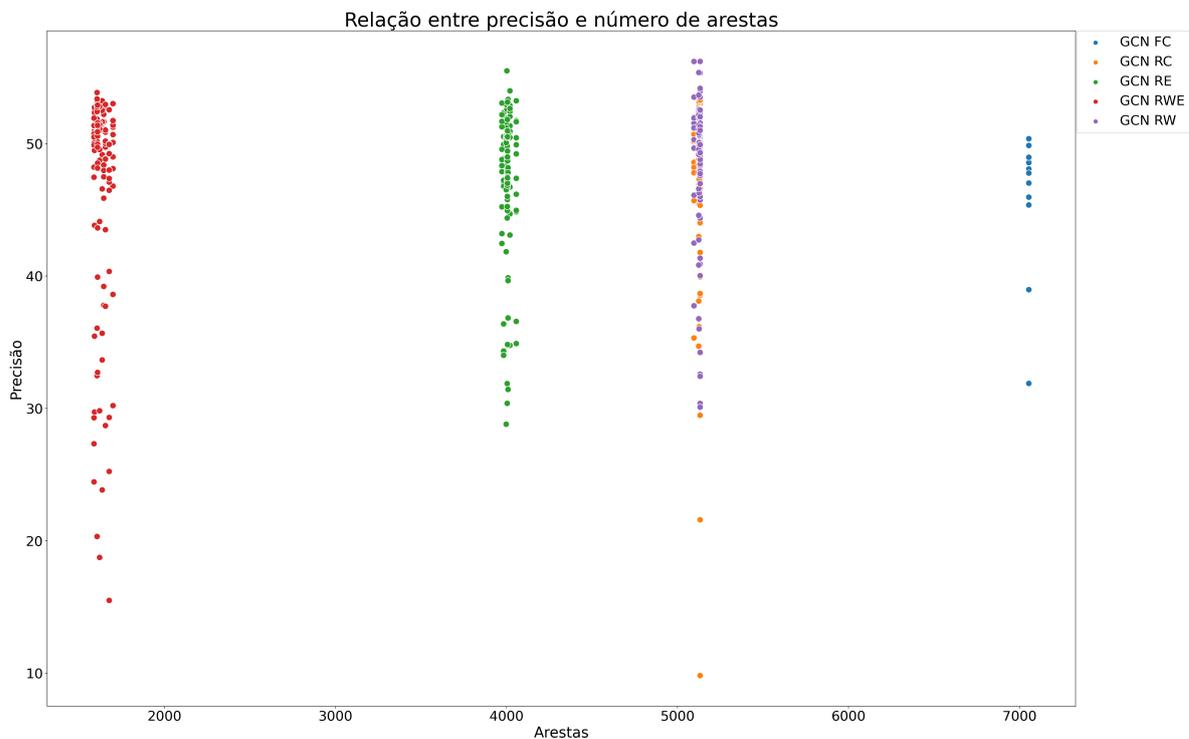


Figura 36 – Correlação entre quantidade de arestas do grafo e sua respectiva precisão para GCN utilizando extrator ResNet50.

De acordo com Figura 36, nota-se que a topologia totalmente conectada possui maior quantidade de arestas que qualquer método aleatório. No entanto, o método RWE foi a topologia que obteve o menor número de arestas e com resultados expressivos perante as outras topologias processadas. Cabe aqui dizer que os resultados obtidos com os métodos aleatórios, em sua maioria, obtiveram um empate técnico, já que podem variar dentro da mesma escala. Porém, analisando os dados apresentados na Figura 36, pode-se perceber que os resultados da topologia totalmente conectada variam menos que os algoritmos baseados em caminhada aleatória.

Para analisar os resultados entre as classes, foi utilizado a métrica matriz de confusão para apresentar os acertos e erros das redes treinadas. Para efeito de comparação, em vista que o projeto visa superar o estado da arte dado no trabalho (BUGATTI; SAITO; DAVIS, 2019), serão apresentadas as matrizes de confusão das topologias totalmente conectada e caminhada aleatória (que obteve maior precisão). A Figura 37 apresenta a matriz de confusão obtida nos conjuntos de teste da base UnRel utilizando a topologia totalmente conectada.

A matriz de confusão nos auxiliará a avaliar os acertos da rede em relação às classes da base de imagens, ou seja, identificar os verdadeiros positivos. Na matriz apresentada na Figura 37, ao lado da matriz, há uma paleta de cores que auxiliará em identificar a quantidade de vezes que um rótulo foi classificado pela rede. Quanto mais escura for a cor dentro da matriz, mais vezes o rótulo foi classificado pela rede neural. Cabe analisar que, quanto mais escura for as cores na diagonal principal da matriz, maior a quantidade de acertos da rede neural. A classe de maior acerto da topologia totalmente conectada foi "*person in cart*", performando 93.88% de precisão nesta classe, sua revocação foi de 81.81% e o F1-score foi de 86.38%.

A matriz de confusão performada com a topologia de caminhada aleatória está apresentada na Figura 38, os resultados foram obtidos utilizando caminhada de tamanho 1.

A classe de maior precisão utilizando a topologia de caminhada aleatória foi "*dog in tree*", performando 95.83% de precisão, 75% de revocação e 83.37% de F1-Score.

Visto que o presente trabalho busca aumentar a precisão na classificação de imagens contextuais, podemos apresentar como os modelos performados se comportam perante a imagens que não estão nos conjuntos de treino e teste da base UnRel.



(A)



(B)



(C)



(D)

Figura 39 – Classificações de imagens para cada topologia aplicada. Em (A) está a Figura da classe *bike-and-person*, em (B) da classe *dog-wear-hat*, em (C) da classe *elephant-wear-pants* e em (D) da classe *person-inside-tree*.

cada imagem presente na Figura 39, há uma caixa de classificação no canto superior direito. Dentro da caixa estão os modelos com os métodos utilizados, as classes preditas e as probabilidades para cada topologia e estrutura testada, nesta ordem.

De acordo com a Figura 39, percebemos que a topologia de caminhada aleatória obteve boas classificações para todas as imagens. No entanto, a imagem (B) é a única em que nenhuma das topologias classificou corretamente. Para efeitos de comparação entre a topologia totalmente conectada e caminhada aleatória, somente na imagem (A) a topologia GCN FC obteve resultado superior ao da GCN RW, no entanto, ambas as topologias classificaram a imagem corretamente. Cabe ressaltar

que a topologia GCN RWE obteve a maior probabilidade de todas as topologias e classificou a imagem corretamente.

A imagem (C), somente a topologia GCN RW a classificou corretamente e na imagem (D) as topologias GCN RW, GCN RE e GCN RWE classificaram a imagem corretamente, no entanto, a topologia GCN RW conseguiu a maior probabilidade entre as demais (53.88%).

Portanto, pode-se dizer que as topologias de otimização obtiveram um impacto muito positivo perante aos resultados performados pela topologia totalmente conectada, principalmente a topologia proposta no presente trabalho, a caminhada aleatória.

4.3.2.2 Mit67

Seguindo a estrutura de apresentação dos resultados para UnRel foram recolhidos os experimentos que obtiveram os maiores valores para precisão de acordo com a topologia utilizada. Dessa forma, os resultados apresentados na Tabela 11 são oriundos dos experimentos com maior precisão para a base de imagens Mit67.

ResNet50	Acurácia	Precisão	Revocação	F1-Score
CNN	13.59 ±0.02474	0.0312 ±0.00032	0.01898 ±0.00016	0.00491 ±0.00015
GCN FC	15.33 ±0.00052	2.50 ±0.00296	2.90 ±0.00038	0.99 ±0.00076
GCN RC	25.74 ±0.01668	31.90 ±0.05904	15.24 ±0.02886	17.28 ±0.03071
GCN RE	20.06 ±0.00443	21.89 ±0.01288	7.40 ±0.00360	8.02 ±0.00422
GCN RWE	27.72 ±0.03255	32.59 ±0.04215	19.57 ±0.01462	21.23 ±0.02424
GCN RW	36.63 ±0.01691	42.78 ±0.05634	25.66 ±0.03286	28.24 ±0.03647
VGG16	Acurácia	Precisão	Revocação	F1-Score
CNN	15.31 ±0.01933	0.01287 ±0.00408	0.02730 ±0.00239	0.01338 ±0.00223
GCN FC	91.71 ±0.00072	89.68 ±0.00116	86.65 ±0.00412	87.33 ±0.00177
GCN RC	65.05 ±0.00262	66.40 ±0.00278	59.00 ±0.01282	61.86 ±0.00838
GCN RE	85.00 ±0.00278	83.59 ±0.00790	79.12 ±0.00393	80.90 ±0.00618
GCN RWE	54.06 ±0.00145	56.35 ±0.00720	47.31 ±0.00205	50.61 ±0.00424
GCN RW	88.95 ±0.00293	86.58 ±0.00599	82.77 ±0.00773	84.12 ±0.00797
InceptionV3	Acurácia	Precisão	Revocação	F1-Score
CNN	15.35 ±0.00063	0.0520 ±0.00088	0.01932±0.00010	0.00617 ±0.00023
GCN FC	15.45 ±0.00094	2.37 ±0.00152	2.03 ±0.00018	0.8 ±0.00022
GCN RC	23.19 ±0.00321	27.35 ±0.02945	11.09 ±0.00664	12.52 ±0.00814
GCN RE	18.55 ±0.00144	17.60 ±0.02090	5.14 ±0.00431	5.28 ±0.00452
GCN RWE	25.45 ±0.00153	29.65 ±0.00854	13.95 ±0.00449	16.01 ±0.00491
GCN RW	30.94 ±0.00495	33.41 ±0.01845	18.75 ±0.00135	21.36 ±0.00279
InceptionResNetV2	Acurácia	Precisão	Revocação	F1-Score
CNN	14.50 ±0.01181	0.0352 ±0.00089	0.01883 ±0.000050	0.00587 ±0.00121

GCN FC	15.29 ±0.00052	3.50 ±0.00307	2.08 ±0.00007	0.97 ±0.00021
GCN RC	23.54 ±0.01939	33.27 ±0.02238	12.52 ±0.01795	14.01 ±0.01872
GCN RE	19.89 ±0.00375	22.16 ±0.01391	7.22 ±0.00393	8.12 ±0.00501
GCN RWE	28.61 ±0.00067	34.90 ±0.00263	17.88 ±0.00497	21.33 ±0.00507
GCN RW	34.91 ±0.00391	42.78 ±0.02849	24.72 ±0.01268	28.80 ±0.00806
Xception	Acurácia	Precisão	Revocação	F1-Score
CNN	14.64 ±0.00649	0.0609 ±0.00173	0.02034 ±0.00118	0.0074 ±0.00014
GCN FC	15.00 ±0.00246	2.23 ±0.00229	2.11 ±0.00081	1.09 ±0.00170
GCN RC	23.59 ±0.00402	26.52 ±0.01224	12.24 ±0.01479	14.10 ±0.01628
GCN RE	18.35 ±0.00331	19.69 ±0.01673	6.77 ±0.00091	7.45 ±0.00044
GCN RWE	25.75 ±0.00694	27.34 ±0.02722	17.62 ±0.00234	19.51 ±0.00585
GCN RW	26.95 ±0.04463	38.71 ±0.04749	17.87 ±0.05852	19.07 ±0.06235
EfficientNet	Acurácia	Precisão	Revocação	F1-Score
CNN	15.34 ±0.00	0.0028 ±0.00	0.01886 ±0.00	0.0050 ±0.00
GCN FC	9.61 ±0.00688	2.06 ±0.00231	2.01 ±0.00048	1.72 ±0.00105
GCN RC	17.75 ±0.00446	18.59 ±0.01488	5.84 ±0.00954	6.71 ±0.01172
GCN RE	12.24 ±0.00185	8.37 ±0.02621	3.43 ±0.00129	3.45 ±0.00229
GCN RWE	21.96 ±0.01647	25.81 ±0.03541	10.11 ±0.02564	12.21 ±0.02667
GCN RW	19.75 ±0.01754	20.53 ±0.03263	6.51 ±0.02711	7.13 ±0.03122

Tabela 11 – Resultados globais obtidos na base de imagens MIT67.

De acordo com a Tabela 11, a maior precisão performada com as redes neurais para a base Mit67 foi alcançada através de um GCN com topologia totalmente conectada (GCN FC) utilizando os extratores da VGG16, com precisão de 89.68%. A rede com topologia de caminhada aleatória (GCN RW) performou 86.58% de precisão e passo de tamanho 10, sendo superada pela rede GCN FC até mesmo em seu desvio padrão.

Para a base Mit67, nota-se que o extrator VGG16 obteve os maiores valores de precisão, um possível motivo é pelo fato da base comportar muitos objetos nas imagens e o vetor de características da VGG16 ser consideravelmente menor que o dos outros extratores, conforme apresentado na Tabela 5. No entanto, nota-se que a rede GCN FC obteve um resultado superior à rede GCN RW em 0.0310%, podendo variar de acordo com o desvio padrão. Mas na verdade, o que chama a atenção é que a GCN RW utilizou em sua topologia, apenas 145130 arestas, enquanto a topologia totalmente conectada utilizou 1395296 arestas. Portanto, a rede GCN RW conseguiu otimizar em 89.60% o número de arestas da topologia totalmente conectada e obteve um resultado de precisão inferior em apenas 0.0310%. Apresenta-se também a eficácia dos métodos em relação ao totalmente conectado na construção dos grafos,

apresentado na tabela 12.

Método	Tempo (s)
FC	65.88
RC	28.50
RE	28.83
RWE	22.61
RW	28.51

Tabela 12 – Tempo para construção dos grafos de cada política de conexão.

Nota-se que os métodos de otimização foram mais eficazes que o totalmente conectado, sendo a política RWE a mais rápida em relação as outras. A Figura 40, ilustra a relação entre a quantidade de arestas e a precisão performada pelos experimentos na base MIT67 através do extrator VGG16.

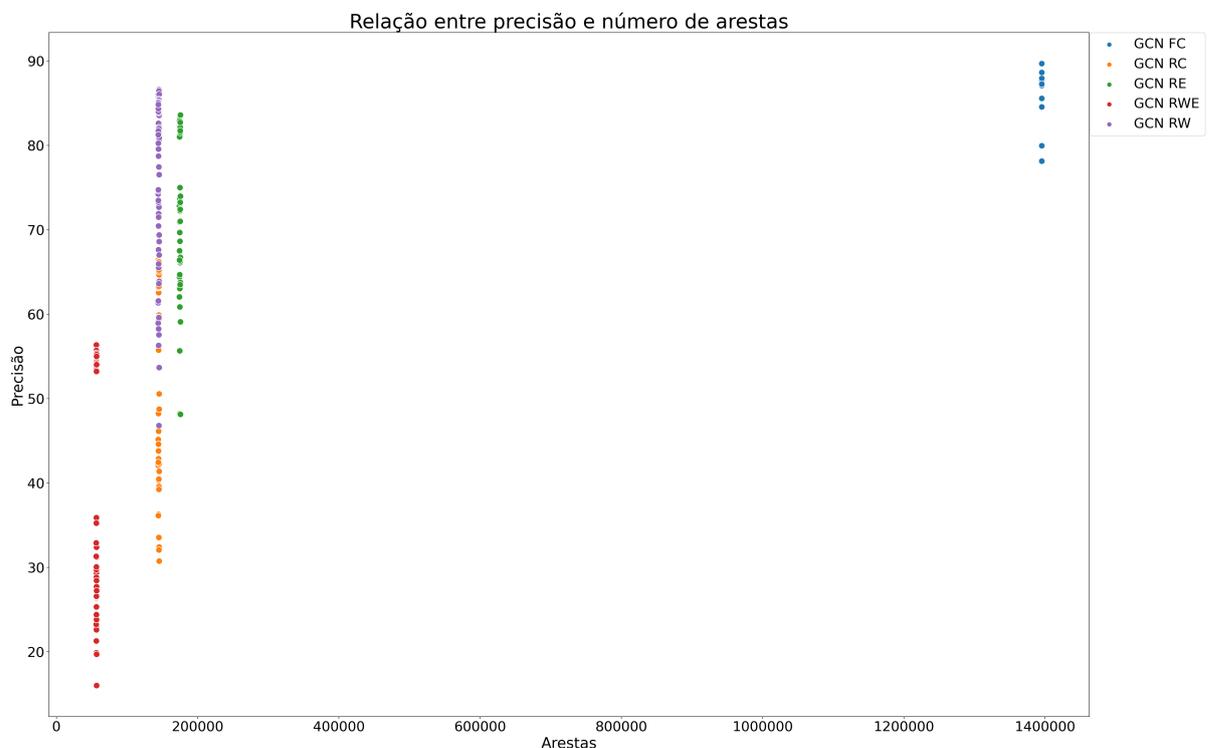


Figura 40 – Correlação entre quantidade de arestas do grafo e sua respectiva precisão para GCN utilizando extrator VGG16.

Conforme apresentado na Figura 40, os experimentos realizados pelas redes GCN FC obtiveram baixa variação, notando que a maioria de seus experimentos estão localizados em regiões próximas. Já os métodos com topologias otimizadas, a variação dos resultados é maior, porém, nota-se que os resultados da GCN RW não variam tanto quanto os demais.

Para analisar e comparar os resultados de classificações das redes realizadas, podemos analisar as matrizes de confusão geradas no momento do teste das

algumas classificações erradas. As 3 classes que obtiveram os maiores resultados para a base Mit67 utilizando a topologia de caminhada aleatória foram: *bedroom* - (95.13%), *gameroom* - (95.52%) e *church-inside* - (87.76%).

Para que o poder de classificação das redes performadas seja testado, realizamos alguns testes de classificação com imagens extraídas da internet, imagens condizentes com as classes: *airport-inside*, *church-inside*, *bathroom* e *office*. Utilizamos os modelos que performaram os maiores resultados na base Mit67 com todas as topologias testadas. A Figura 43, apresenta as classificações realizadas.

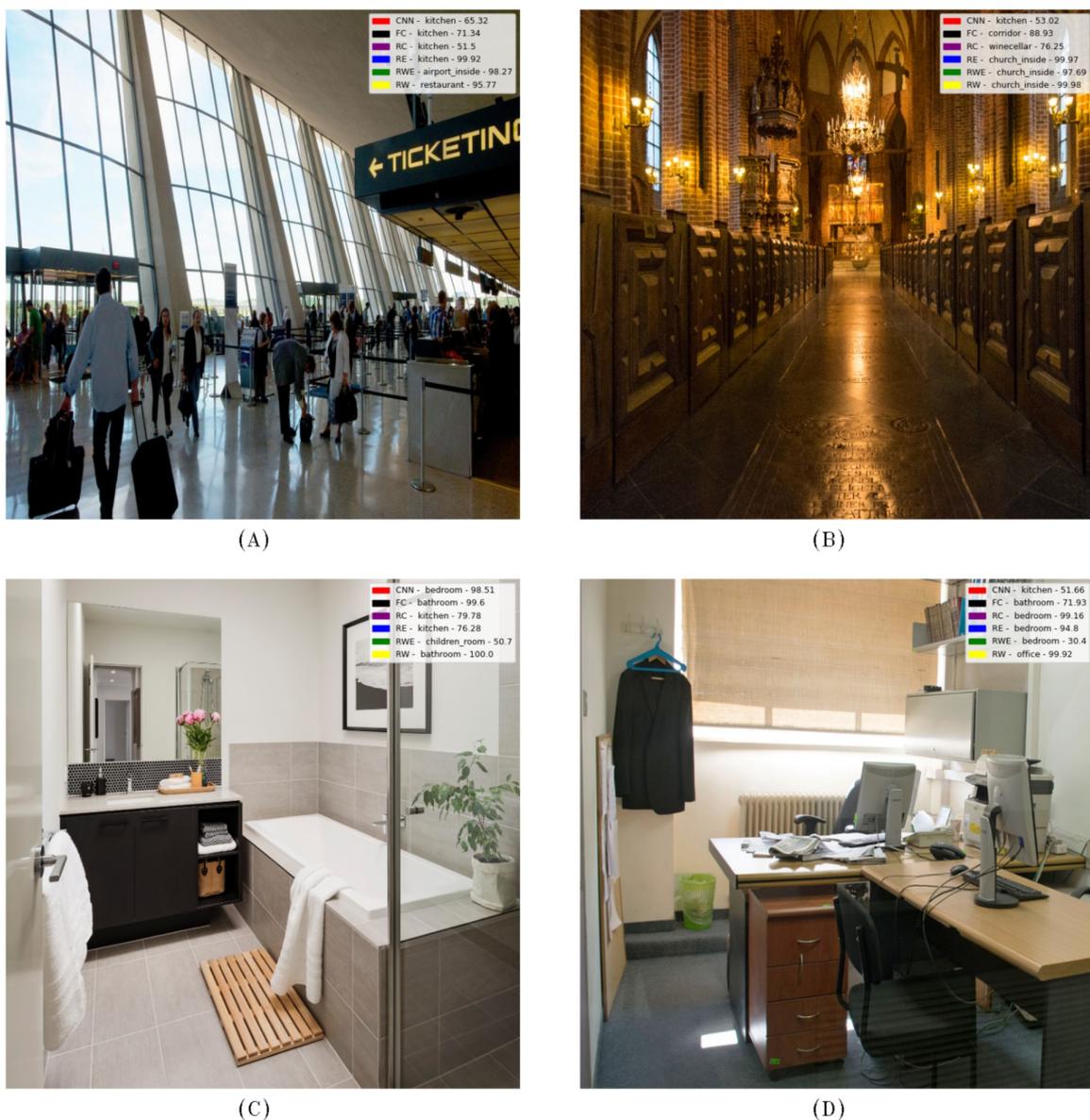


Figura 43 – Classificações de imagens para cada topologia aplicada. Em (A) está a Figura da classe *airport-inside*, em (B) da classe *church-inside*, em (C) da classe *bathroom* e em (D) da classe *office*.

Analisando as imagens classificadas, percebe-se que a topologia de cami-

nhada aleatória classificou corretamente 3 das 4 imagens testadas e foi a única topologia que acertou a classe *office* da imagem (D). Nota-se também que, na imagem (C) a classificação feita pela topologia RW obteve a maior probabilidade das classificações realizadas (100%), obtendo resultado superior ao da topologia totalmente conectada e CNN que também acertaram a classe da imagem. Em (A), somente a topologia RWE classificou corretamente a imagem, enquanto todas as outras topologias confundiram a Figura com ambientes culinários. A imagem (B) cabe uma análise, nota-se que a Figura é similar às imagens de corredores (*corridor*), no entanto, com características diferentes dos corredores, tendo luzes baixas, símbolos e uma arquitetura própria. Dessa forma, nota-se que mesmo obtendo precisão maior que as demais topologias, a GCN FC pode se confundir quando as imagens possuem formas parecidas, levando em consideração que em (B) a topologia GCN FC classificou *church-inside* como *corridor*.

4.3.2.3 Malária

Seguindo a estrutura de apresentação dos resultados, foram recolhidos os experimentos que obtiveram os maiores valores para precisão de acordo com a topologia utilizada. Dessa forma, os resultados apresentados na Tabela 13 são os experimentos com maior precisão para a base de imagens Malária.

ResNet50	Acurácia	Precisão	Revocação	F1-Score
CNN	89.91 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN FC	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN RC	89.45 ±0.00157	81.10 ±0.06114	54.07 ±0.02042	54.64 ±0.03637
GCN RE	89.12 ±0.00030	83.26 ±0.04201	50.79 ±0.00350	48.73 ±0.00727
GCN RWE	89.62 ±0.00429	83.50 ±0.07787	55.60 ±0.03940	56.63 ±0.06686
GCN RW	90.33 ±0.00907	84.89 ±0.06818	60.79 ±0.07571	62.64 ±0.10744
VGG16	Acurácia	Precisão	Revocação	F1-Score
CNN	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN FC	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN RC	89.76 ±0.00058	79.05 ±0.00522	56.13 ±0.00649	58.18 ±0.00985
GCN RE	88.99 ±0.00050	81.35 ±0.1386	50.18 ±0.00140	47.51 ±0.00352
GCN RWE	89.89 ±0.00029	81.11 ±0.01443	56.35 ±0.00610	58.53 ±0.00903
GCN RW	89.89 ±0.00724	83.30 ±0.03569	56.17 ±0.05217	57.25 ±0.08048
InceptionResNetV2	Acurácia	Precisão	Revocação	F1-Score
CNN	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN FC	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN RC	89.29 ±0.00169	80.60 ±0.05915	52.79 ±0.01832	52.45 ±0.03487
GCN RE	89.07 ±0.00011	80.41 ±0.04662	50.45 ±0.00041	48.06 ±0.00100

GCN RWE	89.47 ±0.00149	80.85 ±0.03196	53.66 ±0.01572	54.03 ±0.02835
GCN RW	89.43 ±0.00302	87.32 ±0.05033	52.68 ±0.01995	52.21 ±0.03750
InceptionV3	Acurácia	Precisão	Revocação	F1-Score
CNN	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.000000	47.09 ±0.00000
GCN FC	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN RC	89.38 ±0.00037	79.01 ±0.01444	53.03 ±0.00549	53.04 ±0.01010
GCN RE	89.02 ±0.00067	74.82 ±0.06313	50.62 ±0.00214	48.45 ±0.00509
GCN RWE	89.49 ±0.00054	79.29 ±0.04185	54.57 ±0.01794	55.52 ±0.02933
GCN RW	89.69 ±0.00201	82.79 ±0.02159	54.30 ±0.00906	55.24 ±0.01554
Xception	Acurácia	Precisão	Revocação	F1-Score
CNN	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.000000	47.09 ±0.00000
GCN FC	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.00000	47.09 ±0.00000
GCN RC	89.28 ±0.00115	78.71 ±0.04561	52.96 ±0.01624	52.78 ±0.03054
GCN RE	89.06 ±0.00008	82.23 ±0.01351	50.38 ±0.00072	47.89 ±0.00154
GCN RWE	86.46 ±0.00315	83.83 ±0.07635	53.88 ±0.02756	54.16 ±0.04998
GCN RW	89.64 ±0.00462	86.25 ±0.05946	54.38 ±0.03375	54.89 ±0.05821
EfficientNet	Acurácia	Precisão	Revocação	F1-Score
CNN	89.01 ±0.00000	44.50 ±0.00000	50.00 ±0.000000	47.09 ±0.00000
GCN FC	88.92 ±0.00117	46.07 ±0.02222	49.91 ±0.00011	47.15 ±0.00087
GCN RC	89.05 ±0.00223	74.14 ±0.04801	52.70 ±0.01309	52.36 ±0.02320
GCN RE	89.01 ±0.00033	71.89 ±0.02983	50.84 ±0.00468	48.91 ±0.00996
GCN RWE	89.50 ±0.00127	77.81 ±0.02636	54.71 ±0.01234	55.85 ±0.02051
GCN RW	89.14 ±0.00194	77.72 ±0.06126	53.28 ±0.02158	53.17 ±0.03563

Tabela 13 – Resultados globais obtidos na base de imagens Malária.

De acordo com a Tabela 13, a maior precisão performada com as redes neurais para a base Malária foi alcançada através de um GCN com topologia de caminhada aleatória (GCN RW) utilizando os extratores da InceptionResNetV2 com passo de tamanho 7 e precisão de 87.32%. A rede com topologia totalmente conectada performou seu maior resultado com o extrator EfficientNet, alcançando 46.07% de precisão.

Para a base Malária, nota-se que os resultados das topologias totalmente conectadas não conseguiram atingir bons resultados, e as topologias otimizadas performaram resultados maiores com menos arestas. A topologia totalmente conectada utilizou 7.085.737 arestas no grafo para simular as conexões. Enquanto o método de caminhada aleatória utilizou apenas 248.667 arestas, havendo uma queda de 96.25% no número de arestas de uma topologia para a outra. Visto que o método de corte por limiar T empatou tecnicamente com o método de caminhada aleatória, podemos apresentar seus resultados e arestas podadas. A topologia alcançou a precisão de 80.35%

e de acordo com seu desvio padrão, está tecnicamente empatada com a topologia de caminhada aleatória. O modelo GCN RWE também usou menos arestas para simular as relações entre os objetos nas imagens, utilizando apenas 90.803 arestas, enquanto o método de caminhada aleatória usou 248.667 arestas, uma queda de 63.49% das arestas em relação ao modelo GCN RW. Apresenta-se também a eficácia dos métodos em relação ao totalmente conectado na construção dos grafos, apresentado na tabela 14.

Método	Tempo (s)
FC	184.71
RC	29.08
RE	28.78
RWE	23.07
RW	24.69

Tabela 14 – Tempo para construção dos grafos de cada política de conexão.

Nota-se que os métodos de otimização foram mais eficazes que o totalmente conectado, ficando abaixo de 30 segundos na construção do grafo global, sendo a política RWE a mais rápida em relação as outras. A Figura 44 ilustra uma correlação entre a quantidade de arestas e os resultados de precisão performados pelos modelos. Nota-se que a maior precisão da GCN FC foi obtida com o extrator EfficientNet e não com a InceptionResNetV2, porém, o número de arestas para a topologia totalmente conectada não varia de acordo com o extrator escolhido.

Percebe-se que, os resultados obtidos pelos modelos de topologias otimizadas alcançaram resultados melhores utilizando menos arestas que o totalmente conectado. De acordo com a Figura 44, a topologia que mais otimizou as arestas foi a GCN RWE, no entanto, o maior resultado foi obtido através da topologia de caminhada aleatória GCN RW.

Sendo assim, podemos afirmar que, para a base de classificação de imagens de Malária, o método de caminhada aleatória foi consideravelmente superior ao modelo totalmente conectado, seja em número de arestas reduzidas ou em precisão performada. No entanto, o modelo de corte aleatórios por pesos (GCN RWE) otimizou mais arestas e empatou tecnicamente com o modelo GCN RW. Dessa forma, para a base Malária, os modelos GCN RW e GCN RW empataram em precisão e o modelo GCN RW foi superado em otimização de arestas pelo GCN RWE.

Para compararmos as classificações entre os modelos GCN FC e GCN RW, apresentamos a matriz de confusão obtida pelos modelos treinados com a melhor

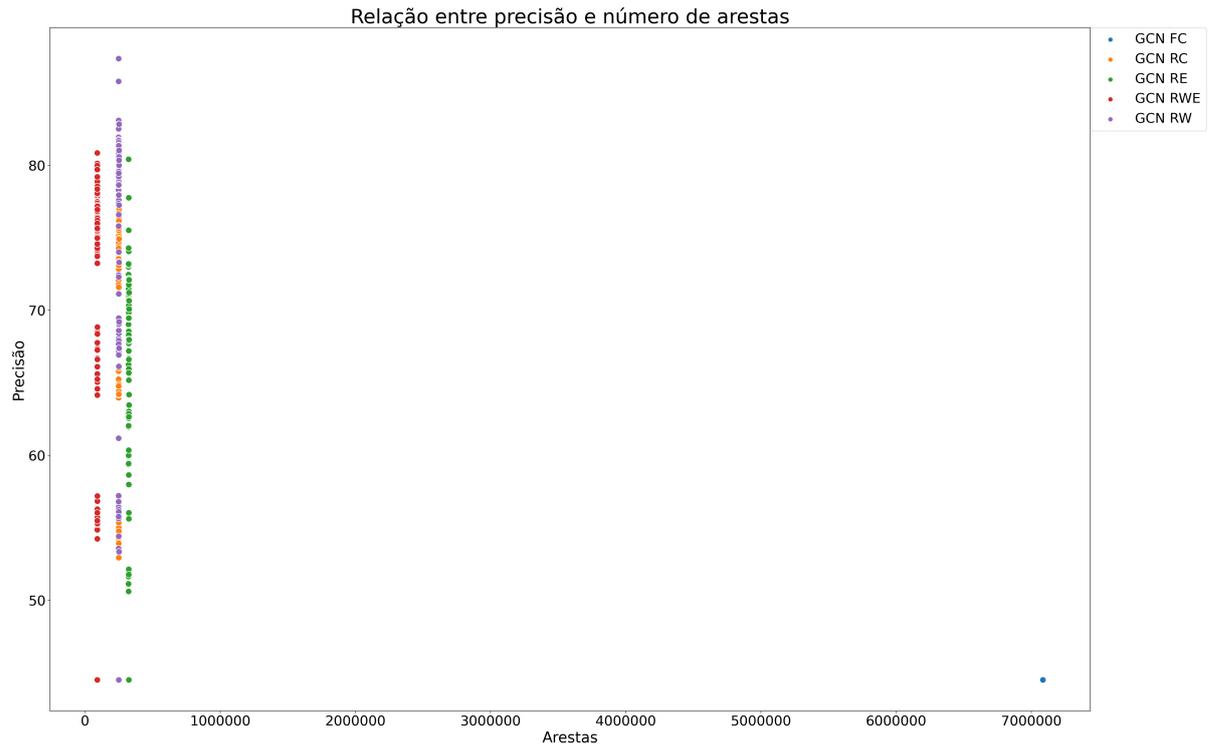


Figura 44 – Correlação entre quantidade de arestas do grafo e sua respectiva precisão para GCN utilizando extrator InceptionResNetV2.

combinação de parâmetros para base Malária. A Figura 45, apresenta a matriz de confusão para o modelo GCN FC.

De acordo com a Figura 45, apenas uma amostra da base de imagens Malária foi classificada corretamente utilizando a topologia totalmente conectada, conforme ilustrado na matriz de confusão. Essa informação pode ser interpretada como um ponto negativo para o uso de uma topologia totalmente conectada para uma base de imagens ampla como a Malária, com vários objetos presentes nas imagens.

Este exemplo de resultado é exatamente o que este presente trabalho buscou atacar, a busca pelo conhecimento de que há a necessidade de uma heurística para melhor representar os relacionamentos entre os objetos, e para afirmar que, uma topologia baseada no método de caminhada aleatória pode ser muito vantajosa para classificações mais precisas.

Para efeito de comparação, apresentamos a Figura 46, com a matriz de confusão com o experimento de maior precisão para a topologia de caminhada aleatória (GCN RW).

Analisando as Figuras 45 e 46, nota-se que a topologia de caminhada aleatória classificou mais amostras corretamente do que a topologia totalmente conectada.

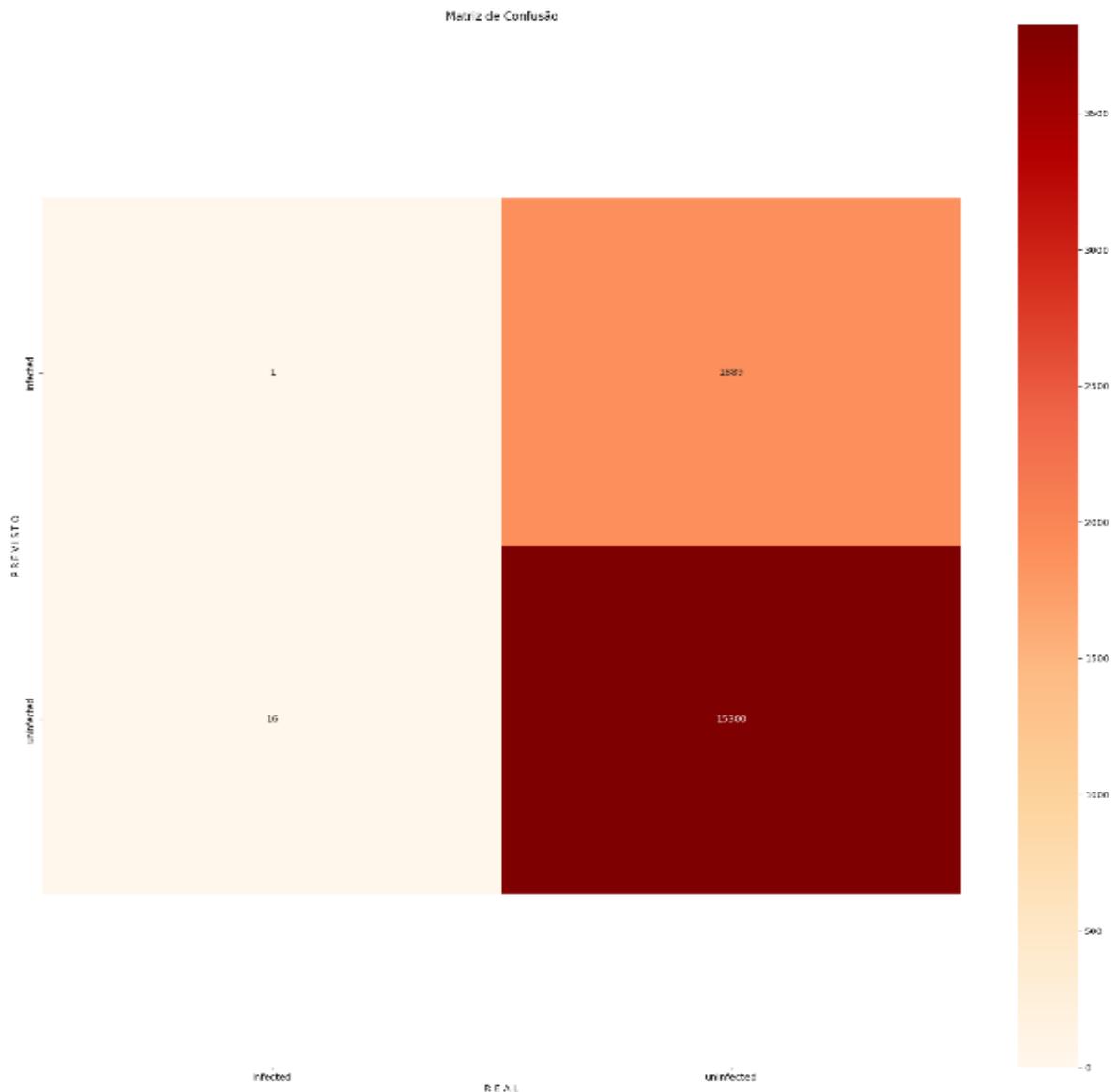


Figura 45 – Matriz de confusão para a topologia totalmente conectada.

tada. Porém, ainda assim, nota-se que há uma quantidade alta de erros entre as duas abordagens, havendo uma quantidade alta de erros considerável para classificações que eram não infectada sendo classificadas como infectada. O resultado da métrica revocação aponta exatamente essa informação, a existência de falsos negativos em excesso. Contudo, a topologia de caminhada aleatória ainda assim foi superior a totalmente conectada, tanto em nível de otimização de arestas quanto em precisão, revocação e f1-score.

Para demonstrar classificações que poderiam ser realizadas no dia-a-dia, apresentamos a Figura 47, que aponta imagens a serem classificadas pelo modelos treinados. Em (A) e (B) são imagens de células infectadas e em (C) e (D) são imagens

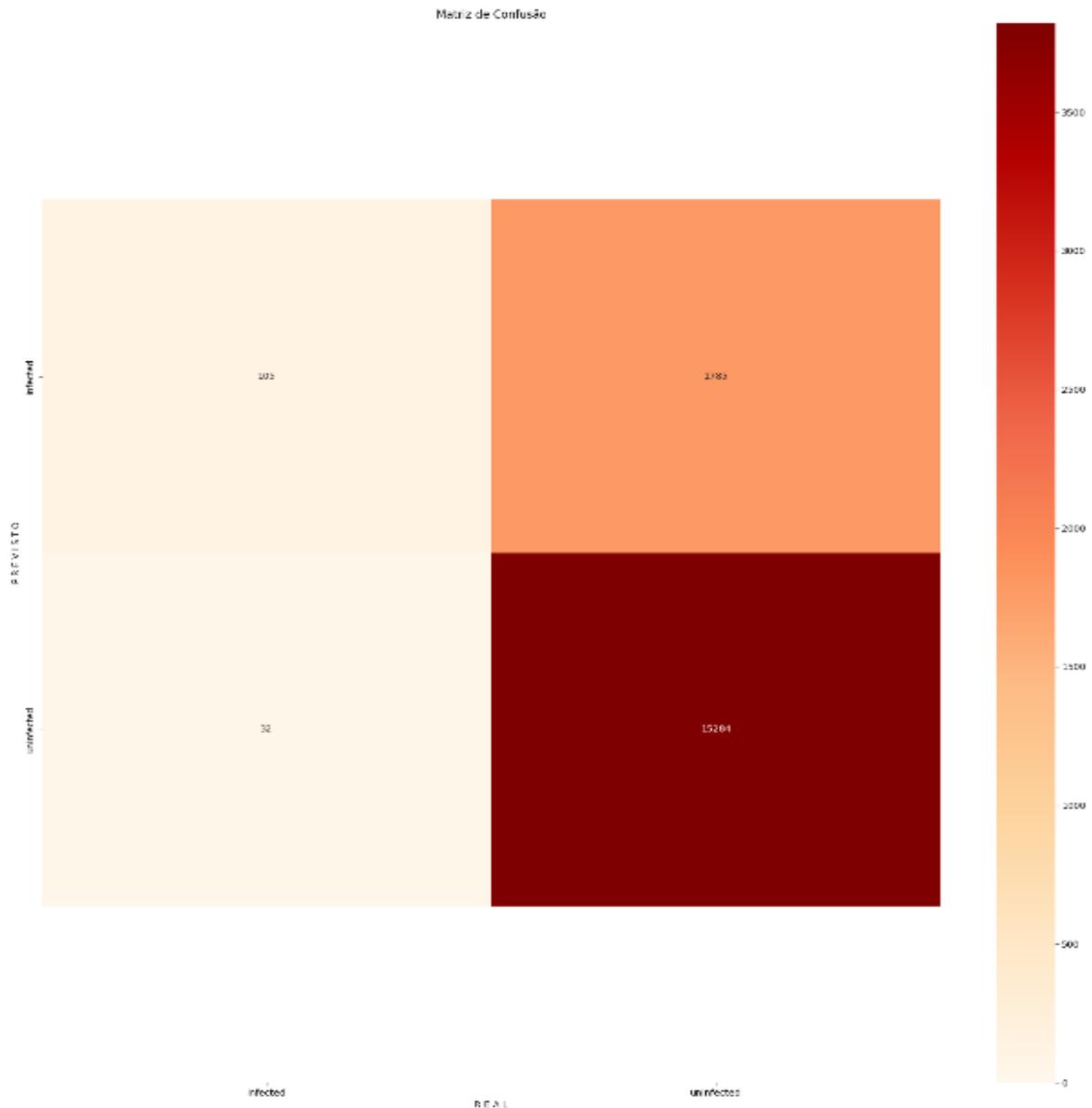


Figura 46 – Matriz de confusão para a topologia de caminhada aleatória.

de células não infectadas.

De acordo com a Figura 47, o modelo de caminhada aleatória classificou corretamente 3 das 4 imagens apresentadas. Nota-se que a topologia totalmente conectada classificou todas as imagens incorretamente, enquanto as topologias GCN RWE e GCN RE classificaram 2 das 4 imagens corretamente, porém, com resultados idênticos, levantando suspeita para que os modelos apenas identifiquem imagens infectadas. O modelo de caminhada aleatória erra apenas a imagem (D), enquanto as topologias GCN RC, GCN FC e a rede CNN classificaram corretamente.

Dessa forma, podemos afirmar que a topologia totalmente conectada foi superada pela topologia de caminhada aleatória e para as outras topologias apresentadas.

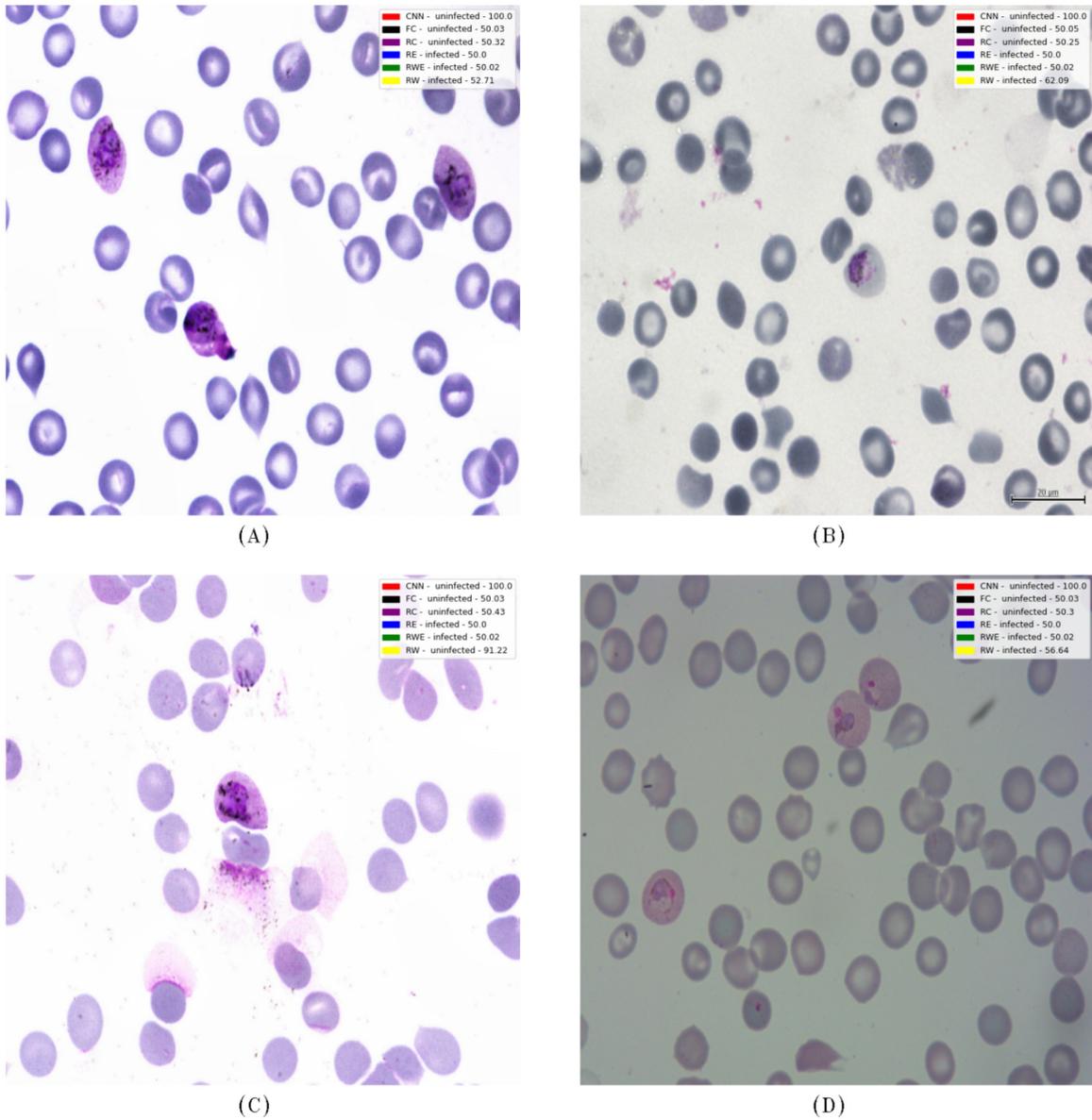


Figura 47 – Classificações de imagens para cada topologia aplicada.

No entanto, a GCN RW obteve a maior precisão geral para a base de imagens, empatando tecnicamente com o modelo do estado da arte GCN RWE.

4.3.2.4 VRD

Seguindo a estrutura de apresentação dos resultados, foram recolhidos os experimentos que obtiveram os maiores valores para precisão de acordo com a topologia utilizada. Dessa forma, os resultados apresentados na Tabela 15 são os experimentos com maior precisão para a base de imagens VRD.

ResNet50	Acurácia	Precisão	Revocação	F1-Score
----------	----------	----------	-----------	----------

CNN	27.23 ±0.00563	0.0237 ±0.00733	0.0436 ±0.00027	0.02143 ±0.00365
GCN FC	27.63 ±0.00000	1.20 ±0.00000	4.34 ±0.00000	1.88 ±0.00000
GCN RC	27.72 ±0.00049	8.36 ±0.00563	5.01 ±0.00031	3.57 ±0.00061
GCN RE	27.55 ±0.00060	4.62 ±0.01462	4.51 ±0.00012	2.36 ±0.00047
GCN RWE	28.28 ±0.00079	8.36 ±0.01101	5.34 ±0.00033	3.98 ±0.00055
GCN RW	27.94 ±0.00271	7.77 ±0.00031	5.73 ±0.00162	4.17 ±0.00076
VGG16	Acurácia	Precisão	Revocação	F1-Score
CNN	27.56 ±0.00044	0.0265 ±0.00482	0.04377 ±0.00021	0.0204 ±0.00107
GCN FC	26.81 ±0.00060	0.29 ±0.00103	0.43 ±0.00015	0.25 ±0.00041
GCN RC	25.94 ±0.00346	11.34 ±0.02380	0.51 ±0.00036	0.41 ±0.00081
GCN RE	26.52 ±0.00050	0.78 ±0.01196	0.41 ±0.00010	0.31 ±0.00030
GCN RWE	26.45 ±0.00209	14.45 ±0.03033	0.61 ±0.00110	0.59 ±0.00138
GCN RW	27.78 ±0.00193	15.00 ±0.00894	0.63 ±0.00142	0.55 ±0.00343
InceptionResNetV2	Acurácia	Precisão	Revocação	F1-Score
CNN	27.55 ±0.00118	0.01436 ±0.00332	0.0434 ±0.00000	0.01956 ±0.00105
GCN FC	27.63 ±0.00000	0.01201 ±0.00000	0.04347 ±0.00000	0.01882 ±0.00000
GCN RC	27.87 ±0.00036	0.04859 ±0.02050	0.04745 ±0.00041	0.02851 ±0.00138
GCN RE	27.71 ±0.00000	0.02592 ±0.00098	0.04430 ±0.00018	0.02087 ±0.00049
GCN RWE	28.17 ±0.00079	0.07076 ±0.01170	0.05134 ±0.00039	0.03634 ±0.00076
GCN RW	28.30 ±0.00068	0.03520 ±0.00022	0.05138 ±0.00043	0.03415 ±0.00106
InceptionV3	Acurácia	Precisão	Revocação	F1-Score
CNN	24.87 ±0.03905	0.02295 ±0.00918	0.04349 ±0.00000	0.0173 ±0.00221
GCN FC	27.63 ±0.00000	1.20 ±0.00000	4.34 ±0.00000	1.88 ±0.00000
GCN RC	27.76 ±0.00028	3.55 ±0.00618	4.71 ±0.00005	2.65 ±0.00045
GCN RE	27.66 ±0.00010	2.45 ±0.00100	4.39 ±0.00016	2.01 ±0.00047
GCN RWE	27.45 ±0.00444	5.48 ±0.01870	5.19 ±0.00034	3.72 ±0.00089
GCN RW	28.11 ±0.00069	3.88 ±0.00953	4.99 ±0.00337	2.83 ±0.00398
Xception	Acurácia	Precisão	Revocação	F1-Score
CNN	22.22 ±0.03825	0.0180 ±0.00451	0.0435 ±0.00000	0.01675 ±0.00197
GCN FC	27.63 ±0.00000	1.20 ±0.00000	4.34 ±0.00000	1.88 ±0.00000
GCN RC	27.30 ±0.00255	8.00 ±0.01570	5.14 ±0.00071	3.72 ±0.00051
GCN RE	27.69 ±0.00035	3.15 ±0.00706	4.49 ±0.00025	2.28 ±0.00082
GCN RWE	27.91 ±0.00038	6.91 ±0.02248	5.16 ±0.00020	3.77 ±0.00030
GCN RW	28.42 ±0.00104	4.18 ±0.00996	5.35 ±0.00110	3.81 ±0.00063
EfficientNet	Acurácia	Precisão	Revocação	F1-Score
CNN	27.63 ±0.00000	0.0120 ±0.00000	0.0434 ±0.000000	0.0188 ±0.00000
GCN FC	19.11 ±0.00488	4.39 ±0.00273	4.35 ±0.00094	4.08 ±0.00090
GCN RC	22.21 ±0.00278	8.42 ±0.04190	5.17 ±0.00178	4.88 ±0.00299
GCN RE	26.38 ±0.00269	5.56 ±0.02029	4.44 ±0.00037	2.80 ±0.00053
GCN RWE	21.51 ±0.00729	10.27 ±0.01940	7.19 ±0.00302	7.48 ±0.00353
GCN RW	22.20 ±0.00233	7.85 ±0.03001	5.30 ±0.00046	4.94 ±0.00066

Tabela 15 – Resultados globais obtidos na base de imagens VRD.

De acordo com a Tabela 15, a maior precisão performada com as redes neurais para a base VRD foi alcançada através de um GCN com topologia caminhada aleatória (GCN RW) utilizando os extratores da VGG16, com precisão de 15%. A rede

com topologia totalmente conectada (FC) foi superada por todas as topologias per-
formadas de acordo com os resultados para precisão. Os resultados em geral são
baixos, algo que foi citado em (BUGATTI; SAITO; DAVIS, 2019). As imagens da base
VRD necessitam de um grande poder de generalização para que os resultados sejam
adequados. No entanto, no geral, podemos analisar que os resultados obtidos pelas
topologia GCN RW e GCN RWE são os de maior precisão.

A topologia de caminhada aleatória utilizou 248.572 arestas no grafo que si-
mula as relações entre os objetos em todas as imagens, com passo de tamanho
3, enquanto o totalmente conectado utilizou 1.792.334 arestas. De acordo com os
dados apresentados, a topologia de caminhada aleatória otimizou em 86.14% o nú-
mero de arestas do grafo e mesmo assim obteve uma precisão maior que a GCN FC.
Apresenta-se também a eficácia dos métodos em relação ao totalmente conectado na
construção dos grafos, apresentado na tabela 16.

Método	Tempo (s)
FC	192.86
RC	99.66
RE	99.30
RWE	78.89
RW	87.31

Tabela 16 – Tempo para construção dos grafos de cada política de conexão.

Nota-se que o método mais eficaz foi o da topologia RWE construindo o grafo
global em 78.89 segundos. Para ilustrar a relação entre precisão e arestas, apresenta-
mos a Figura 48, os dados apresentados na mesma foram obtidos por meio do extrator
VGG16 que obteve os maiores resultados para a base VRD.

Conforme apresentado na Figura 48, nota-se que o maior valor geral para
precisão foi obtido por meio do método de caminhada aleatória. Porém, o método de
corte por limitar T foi o que mais otimizou as arestas do grafo.

Para analisarmos a capacidade do modelo GCN FC, apresentamos a matriz
de confusão na Figura 49, a mesma é retirada do experimento que obteve a maior
precisão geral para o dataset VRD. Com a matriz podemos analisar em quais classes
o modelo obteve maior precisão e verificar quais são os erros comuns de classificação
para este modelo.

Analisando a Figura 49, é explícito que o modelo não obteve capacidade para
generalizar todas as classes da base VRD, gerando confusão entre as classes *has* e

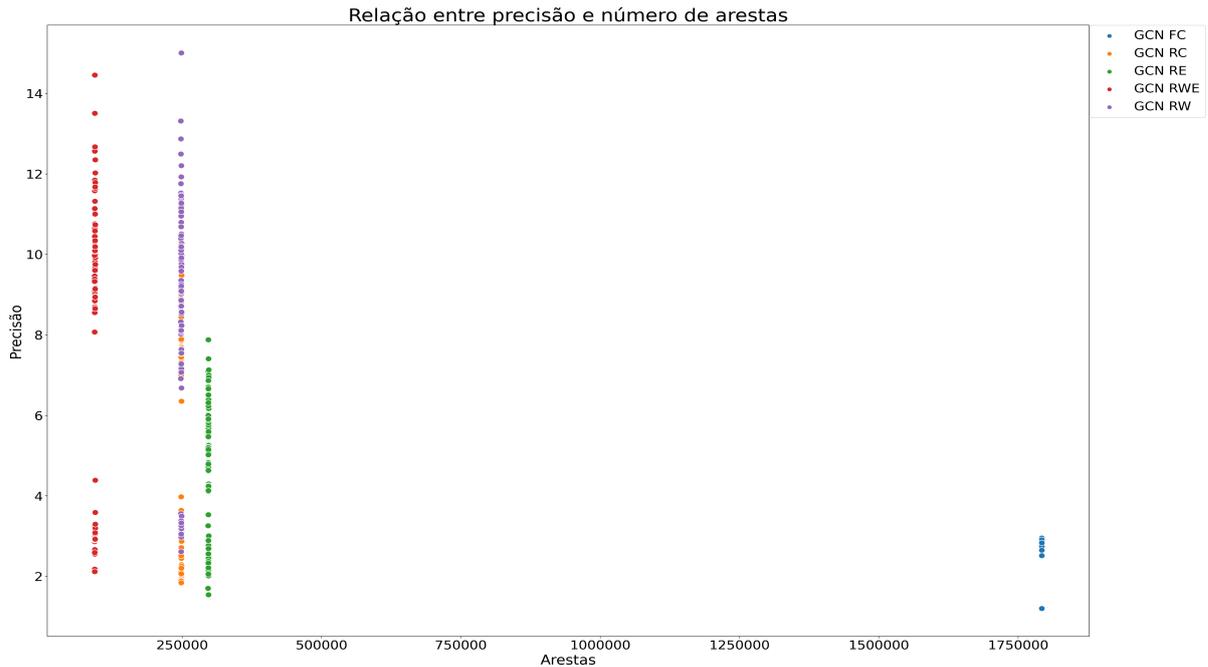


Figura 48 – Correlação entre precisão e número de arestas.

on. Sendo assim, notamos que o modelo não obteve bons resultados entre as classes da base VRD, já que a diagonal principal da matriz de confusão não está preenchida.

Para checar a performance do modelo GCN RW em classificar corretamente as classes, apresentamos a matriz de confusão na Figura 50, a mesma é retirada do experimento que obteve a maior precisão geral para o dataset VRD.

Analisando a Figura 50, também podemos analisar que o modelo não obteve poder de generalização suficiente para performar bons resultados para a base VRD. Porém, podemos analisar que o modelo GCN RW conseguiu classificar corretamente mais classes que o modelo GCN FC. No entanto, encontra-se com o mesmo problema de generalização e confusão na classificação das demais classes. O modelo GCN RW obteve seus maiores resultados para as classes: *on*, *has* e *wearing*, nesta ordem. O modelo GCN FC obteve seus maiores resultados para as classes: *on*, *has* e *in*, nesta ordem.

Para realizar as classificações, foram extraídas imagens do mundo real presentes em paginas web e estão disponíveis para uso. Utilizamos imagens que con- dizem com as classes: *above*, *behind*, *has* e *next-to*. As imagens foram classificadas pelos melhores modelos para cada topologia e arquitetura utilizada. Adicionamos uma imagem para a classe *has* buscando perceber se os modelos conseguem a classificar corretamente e outras classes também. As classificações realizadas pelos modelos

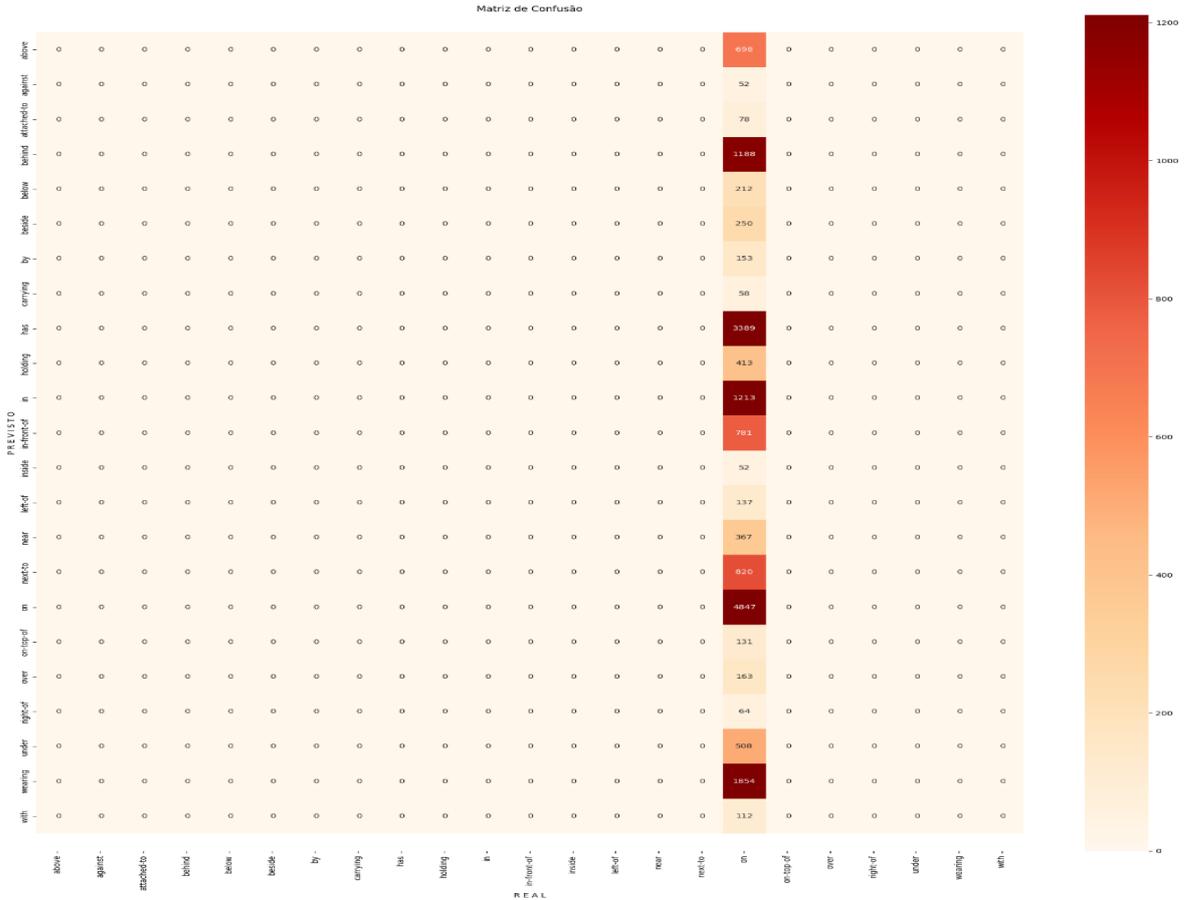


Figura 49 – Matriz de confusão da topologia totalmente conectada.

estão na Figura 51, em cada imagem presente na Figura há uma caixa de classificação no canto superior direito, dentro da caixa estão as classes preditas e as probabilidades para cada topologia e estrutura testada.

De acordo com a Figura 51, percebe-se que nenhuma das topologias testadas conseguiu classificar corretamente a imagem (A) da classe *above* e imagem (D) da classe *next-to*. No entanto, todas as topologias, exceto a estrutura CNN, classificaram corretamente a imagem (C), com destaque para o modelo GCN RW que obteve a maior probabilidade entre os modelos performados para a imagem. O maior destaque para os modelos GCN RW vai para a imagem (B) em que somente este modelo foi capaz de classificar a imagem corretamente (*behind*).

Portanto, para a base de imagens VRD, podemos afirmar que os resultados não foram satisfatórios. No entanto, cabe dizer que os modelos GCN RW e GCN RWE obtiveram resultados superiores aos modelos GCN FC, mesmo utilizando menos arestas para representar os grafos de relação.

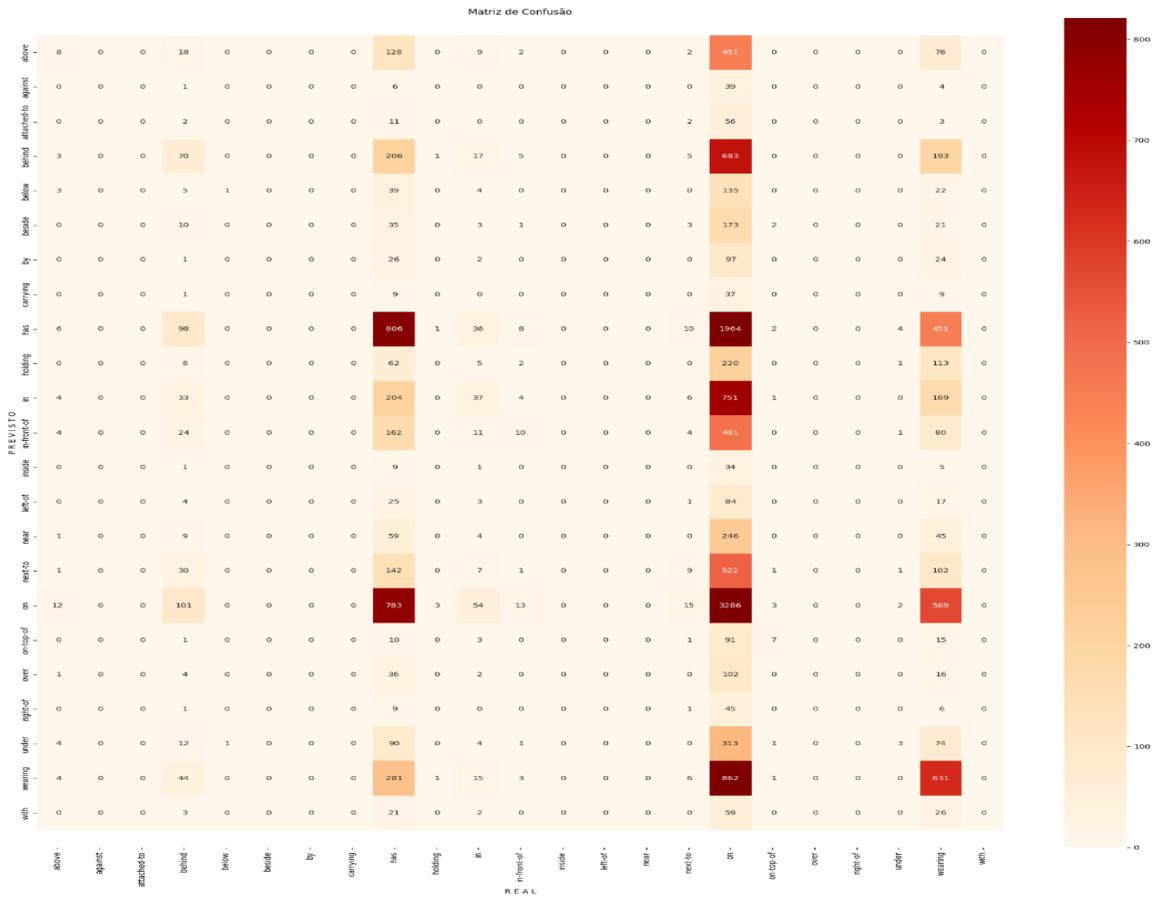
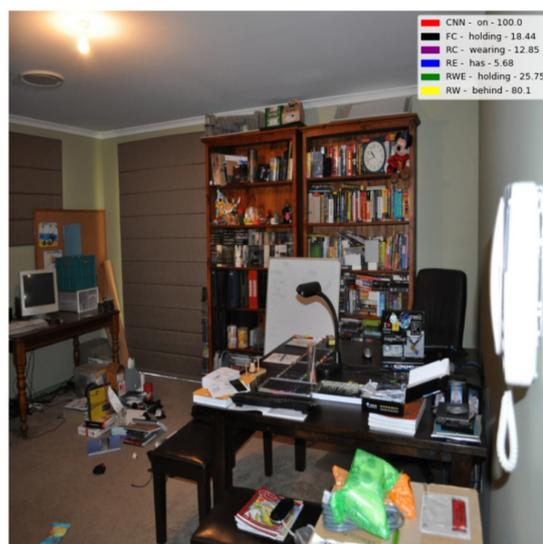


Figura 50 – Matriz de confusão da topologia de caminhada aleatória.



(A)



(B)



(C)



(D)

Figura 51 – Classificações de imagens para cada topologia aplicada. Em (A) está a Figura da classe *above*, em (B) da classe *behind*, em (C) da classe *has* e em (D) da classe *next-to*.

5 CONCLUSÃO

O presente trabalho propôs uma nova abordagem de classificação de imagens utilizando redes neurais em grafos convolucionais que foi baseada em (BUGATTI; SAITO; DAVIS, 2019), (KIPF; WELLING, 2016) e (CAMPOS et al., 2021). A proposta foi aumentar a precisão na classificação de imagens contextuais e diminuir o número de arestas dos grafos através de caminhada aleatória, ou seja, foi proposto que os novos modelos fossem mais assertivos com menos relações nas topologias.

Foram utilizadas 4 bases de imagens com características semelhantes, as bases possuem classes que são definidas pelo contexto apresentado nas imagens. As bases escolhidas foram: Mit67 (Quattoni; Torralba, 2009), UnRel (PEYRE et al., 2017), VRD (LU et al., 2016) e Malária (MISHRA, 2021). Utilizamos processamento de imagens para destacar os objetos de cada imagem, sendo assim possível simular o relacionamento dos objetos através de grafos. O método utilizado foi de corte simples entre pontos mais a esquerda e mais a direita.

Para efeitos de comparação, foram criadas 2 topologias com escolha de arestas aleatórias: Cortes aleatórios (RC) e Arestas aleatórias (RE). Também foram utilizadas duas topologias propostas na literatura, cortes por limiar T (RWE) proposto em (CAMPOS et al., 2021) e a topologia totalmente conectada (FC) proposta em (BUGATTI; SAITO; DAVIS, 2019). Para aumentar a credibilidade dos métodos propostos, também testamos as redes neurais convolucionais tradicionais (CNNs).

Para treinamento dos modelos utilizamos as bibliotecas Tensorflow para redes neurais convolucionais tradicionais e PyTorch para treinamento e teste das redes neurais em grafos convolucionais. Utilizamos a técnica de *grid search* para encontrar parâmetros adequados nas redes. Para avaliar as redes, utilizamos como métrica principal a precisão, buscando aumentar a credibilidade dos resultados performados pelo método proposto.

Os resultados obtidos pelo método proposto foram dentro do esperado. Tendo como principal comparação a topologia totalmente conectada, o método de caminhada aleatória foi capaz de diminuir o número de arestas na base UnRel em 27.24% e ainda assim obter uma precisão superior, performando 56.20%, enquanto a topologia totalmente conectada alcançou 50.37%.

Para a base Mit67, o método de caminhada aleatória não obteve resultados superiores ou tecnicamente iguais aos da topologia totalmente conectada. No entanto, o método foi capaz de otimizar em 89.60% o número de arestas do grafo totalmente conectado, e ainda assim alcançar uma precisão de 86.58%, porém sendo superado pelo totalmente conectado que performou 89.68% de precisão.

Na base Malária, o método de caminhada aleatória obteve a maior precisão entre todos os métodos propostos. O método foi capaz de diminuir em 96.25% das arestas em relação a topologia totalmente conectada e alcançou uma precisão de 87.32%, enquanto a topologia totalmente conectada performou 46.07%. Cabe ressaltar que o método proposto em (CAMPOS et al., 2021) obteve bons resultados para a base de imagens analisada.

A base VRD foi a base em que os resultados não alcançaram resultados interessantes em nenhuma abordagem. No entanto, analisando os valores de precisão performados, o método de caminhada aleatória é superior a todos os métodos testados. Em comparação entre os métodos de caminhada aleatória e totalmente conectado, a topologia RW diminuiu em 86.14% o número de arestas da topologia FC e alcançou 15% de precisão, enquanto a topologia FC obteve 4.39%.

Portanto, tendo como objetivo central obter precisão tecnicamente igual ou superior na classificação de imagens contextuais utilizando menos arestas em grafos convolucionais através do método de caminhada aleatória, o presente trabalho alcançou tal objetivo nas bases de imagens VRD, UnRel e Malária, não alcançando um valor tecnicamente igual ao totalmente conectado somente na base Mit67. No entanto, cabe ressaltar que nenhuma das abordagens testadas chegou mais perto da precisão performada pela topologia totalmente conectada do que a de caminhada aleatória. Porém, mesmo não alcançando resultados tecnicamente iguais ou superiores, o método de caminhada aleatória ainda assim conseguiu diminuir o número de arestas de topologia totalmente conectada em 89.60%.

Foram comparados os resultados performados por todas as topologias escolhidas no presente trabalho, utilizando métricas e conjuntos de dados justos para comparação. O método de caminhada aleatória também obteve melhores classificações nas imagens contextuais em relação ao totalmente conectado, seja em relação a prever o rótulo corretamente ou em obter maior probabilidade. Dessa forma, podemos afirmar que o presente trabalho alcançou os objetivos elencados ao longo do projeto.

REFERÊNCIAS

ABBAS, Asmaa; ABDELSAMEA, Mohammed M.; GABER, Mohamed Medhat. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. **medRxiv**, Cold Spring Harbor Laboratory Press, p. 2–3, 2020. Citado 2 vezes nas páginas 17 e 49.

ALOM, Md. Zahangir et al. Improved inception-residual convolutional neural network for object recognition. **CoRR**, abs/1712.09888, p. 3–5, 2017. Citado na página 17.

BENGIO IAN GOODFELLOW, Aaron Courville Yoshua. **Deep Learning**. [S.l.]: Nature, 2015. 331–372 p. Citado 5 vezes nas páginas 10, 25, 27, 28 e 29.

BERTASIUS, Gedas et al. Convolutional random walk networks for semantic image segmentation. **CoRR**, abs/1605.07681, p. 1–4, 2016. Citado na página 48.

BRASIL, Clarisse Bressan e Patrícia. **Malária**. 2009. Citado na página 75.

BUGATTI, Pedro Henrique; SAITO, Priscila T. M.; DAVIS, Larry S. Hicore: Visual hierarchical context-reasoning. **CoRR**, abs/1909.00848, p. 2–9, 2019. Disponível em: <<http://arxiv.org/abs/1909.00848>>. Citado 20 vezes nas páginas 17, 18, 19, 20, 36, 37, 38, 39, 40, 46, 53, 55, 56, 64, 69, 73, 74, 82, 100 e 105.

CAMPOS, Walacy S. et al. Contextual image classification through fine-tuned graph neural networks. In: RUTKOWSKI, Leszek et al. (Ed.). **Artificial Intelligence and Soft Computing**. Cham: Springer International Publishing, 2021. p. 15–24. ISBN 978-3-030-87897-9. Citado 10 vezes nas páginas 18, 20, 49, 53, 55, 59, 64, 69, 105 e 106.

CARLETTI, Timoteo; FANELLI, Duccio; LAMBIOTTE, Renaud. Random walks and community detection in hypergraphs. **Journal of Physics: Complexity**, IOP Publishing, v. 2, n. 1, p. 2–7, apr 2021. Disponível em: <<https://doi.org/10.1088/2632-072x/abe27e>>. Citado 2 vezes nas páginas 37 e 61.

CASALEGNO, Francesco. **Graph Convolutional Networks - Deep Learning in Graphs**. 2021. Disponível em: <<https://towardsdatascience.com/>

graph-convolutional-networks-deep-99d7fee5706f>. Citado 4 vezes nas páginas 10, 37, 38 e 39.

CHARTRAND, Gary; ZHANG, Ping. **Chromatic Graph Theory**. [S.l.]: CRC Press, 2019. v. 2. 304–400 p. Citado 5 vezes nas páginas 10, 33, 34, 35 e 37.

CHEN, Meihao; LIN, Zhuoru; CHO, Kyunghyun. Graph convolutional networks for classification with a structured label space. **CoRR**, abs/1710.04908, p. 3–7, 2017. Citado 3 vezes nas páginas 17, 37 e 39.

CHEN, Zhao-Min et al. Multi-label image recognition with graph convolutional networks. **CoRR**, abs/1904.03582, p. 2–5, 2019. Citado 2 vezes nas páginas 17 e 37.

CHOLLET, François. Xception: Deep learning with depthwise separable convolutions. **CoRR**, abs/1610.02357, 2016. Disponível em: <<http://arxiv.org/abs/1610.02357>>. Citado na página 54.

CUI, Zhiyong et al. High-order graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. **CoRR**, abs/1802.07007, 2018. Disponível em: <<http://arxiv.org/abs/1802.07007>>. Citado na página 37.

DOUGHERTY, Edward R. **Digital Image Processing Methods**. [S.l.]: CRC Press, 2020. v. 1. 200–320 p. Citado 3 vezes nas páginas 22, 23 e 31.

DUVENAUD, David et al. Convolutional networks on graphs for learning molecular fingerprints. **CoRR**, abs/1509.09292, p. 2–5, 2015. Disponível em: <<http://arxiv.org/abs/1509.09292>>. Citado na página 17.

FAN, Wenqi et al. Graph neural networks for social recommendation. **CoRR**, abs/1902.07243, 2019. Disponível em: <<http://arxiv.org/abs/1902.07243>>. Citado na página 37.

FOUT, Alex et al. Protein interface prediction using graph convolutional networks. In: . Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 6533–6542. ISBN 9781510860964. Citado na página 37.

GILES, C. Lee; BOLLACKER, Kurt D.; LAWRENCE, Steve. Citeseer: an automatic citation indexing system. In: **INTERNATIONAL CONFERENCE ON DIGITAL LIBRARIES**. [S.l.]: ACM Press, 1998. p. 89–98. Citado na página 47.

GONZALEZ, Richard E. Woods Rafael C. Book. **Processamento de Imagens Digitais**. 3. ed. [S.l.]: Pearson Universidades, 2009. 50–420 p. ISBN 8576054019. Citado 5 vezes nas páginas 10, 17, 22, 23 e 31.

He, Kaiming et al. Deep Residual Learning for Image Recognition. **arXiv e-prints**, p. arXiv:1512.03385, dez. 2015. Citado na página 54.

HIRSCHMAN, David V. Widder Isidore Isaac. **Convolution Transform**. [S.l.]: Blucher, 2012. v. 1. Citado na página 30.

JESUS, Roberto Costa Jr. Edison O. A utilização de filtros gaussianos na análise de imagens digitais. In: . [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 30 e 31.

KARIMI, Zohreh. **Confusion Matrix**. 2021. Citado 3 vezes nas páginas 44, 45 e 46.

KIPF, Thomas N.; WELLING, Max. Semi-supervised classification with graph convolutional networks. **CoRR**, abs/1609.02907, p. 1–5, 2016. Citado 12 vezes nas páginas 18, 37, 38, 39, 40, 41, 46, 53, 55, 64, 69 e 105.

KOSTI, Ronak et al. Context based emotion recognition using emotic dataset. **CoRR**, abs/2003.13401, p. 2755–2766, 2020. Citado na página 17.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 60, n. 6, p. 84–90, maio 2017. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/3065386>>. Citado 4 vezes nas páginas 10, 31, 32 e 63.

LAWLER, Gregory F.; LIMIC, Vlada. **Random Walk: A Modern Introduction**. [S.l.]: Cambridge University Press, 2010. 30–52 p. (Cambridge Studies in Advanced Mathematics). Citado na página 43.

LECUN, Yann et al. Handwritten digit recognition with a back-propagation network. In: TOURETZKY, D. (Ed.). **Advances in Neural Information Processing Systems**. Morgan-Kaufmann, 1998. v. 2, p. 396–404. Disponível em: <<https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf>>. Citado na página 30.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 30.

Li, R. et al. On random walk based graph sampling. In: **2015 IEEE 31st International Conference on Data Engineering**. [S.l.: s.n.], 2015. Citado 8 vezes nas páginas 10, 37, 41, 42, 43, 44, 45 e 61.

LI, Zewen et al. A survey of convolutional neural networks: Analysis, applications, and prospects. **CoRR**, abs/2004.02806, p. 1–12, 2020. Disponível em: <<https://arxiv.org/abs/2004.02806>>. Citado 4 vezes nas páginas 17, 30, 31 e 62.

LJOSA, Vebjorn; SOKOLNICKI, Katherine; CARPENTER, Anne. Annotated high-throughput microscopy image sets for validation. **Nature methods**, v. 9, p. 637, 06 2012. Citado 3 vezes nas páginas 11, 75 e 76.

LU, Cewu et al. Visual relationship detection with language priors. **CoRR**, abs/1608.00187, 2016. Disponível em: <<http://arxiv.org/abs/1608.00187>>. Citado 4 vezes nas páginas 38, 46, 77 e 105.

MACHADO, Lucia Machado Haertel Ângelo. **Neuroanatomia funcional**. [S.l.]: Atheneu, 2013. 12–112 p. Citado 4 vezes nas páginas 10, 23, 24 e 25.

MARI, Andrea et al. Transfer learning in hybrid classical-quantum neural networks. **Quantum**, Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, v. 4, p. 340, Oct 2020. ISSN 2521-327X. Disponível em: <<http://dx.doi.org/10.22331/q-2020-10-09-340>>. Citado 2 vezes nas páginas 10 e 33.

MASTAN, Indra Deep; RAMAN, Shanmuganathan. Deepcfl: Deep contextual features learning from a single image. **CoRR**, abs/2011.03712, p. 1–8, 2020. Citado na página 17.

MAZÓN, José M.; SOLERA, Marcos; TOLEDO, Julián. The total variation flow in metric random walk spaces. **Calculus of Variations and Partial Differential Equations**, Springer Science and Business Media LLC, v. 59, n. 1, p. 1–10, Jan 2020. ISSN 1432-0835. Disponível em: <<http://dx.doi.org/10.1007/s00526-019-1684-z>>. Citado na página 44.

MCCULLOCH, Warren; PITTS, Walter. A logical calculus of ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 127–147, 1943. Citado na página 25.

MEHLIG, B. Artificial neural networks. **CoRR**, Springer US, abs/1901.05639, p. 81–100, 2019. Citado na página 23.

MISHRA, Saurav. Malaria parasite detection using efficient neural ensembles. **Journal of Electronics, Electromedical Engineering, and Medical Informatics**, v. 3, n. 3, p. 119–133, Oct. 2021. Disponível em: <<http://jeeemi.org/index.php/jeeemi/article/view/147>>. Citado 2 vezes nas páginas 75 e 105.

MITCHELL, Tom M. **Machine Learning**. [S.l.: s.n.], 1997. Citado 3 vezes nas páginas 26, 27 e 44.

NETTO, Paulo Oswaldo Boaventura. **Grafos: Teoria, Modelos, Algoritmos**. [S.l.]: Blucher, 2012. v. 5. 11–30 p. Citado 6 vezes nas páginas 10, 13, 33, 34, 35 e 36.

NIKOLENTZOS, Michalis Vazirgiannis Giannis. Random walk graph neural networks. Curran Associates, Inc., p. 4–12, 2019. Citado na página 48.

PEARSON, KARL. The Problem of the Random Walk. **Nature**, v. 72, n. 1865, p. 294–294, 1905. Citado 2 vezes nas páginas 41 e 42.

PEYRE, Julia et al. **Weakly-supervised learning of visual relations**. 2017. Citado 4 vezes nas páginas 38, 46, 72 e 105.

PONS, Pascal; LATAPY, Matthieu. Computing communities in large networks using random walks. In: YOLUM, plnar et al. (Ed.). **Computer and Information Sciences - ISICIS 2005**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 9–14. Citado 4 vezes nas páginas 41, 43, 44 e 61.

POPOVIC, Marko. **Researchers in an Entropy Wonderland: A Review of the Entropy Concept**. arXiv, 2017. Disponível em: <<https://arxiv.org/abs/1711.07326>>. Citado na página 60.

Quattoni, A.; Torralba, A. Recognizing indoor scenes. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 413–420. Citado 6 vezes nas páginas 10, 18, 38, 46, 73 e 105.

RIBEIRO, Ana; MARTINS, Bruno; ANTUNES, Elton. Linguagem, sentido e contexto: considerações sobre comunicação e história. **Revista FAMECOS**, v. 24, p. 27047, 08 2017. Citado na página 17.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psych. Rev.**, v. 65, p. 12–26, 1958. Citado 3 vezes nas páginas 10, 25 e 26.

SAJJAD, Hooman Peiro; DOCHERTY, Andrew; TYSHETSKIY, Yuriy. **Efficient Representation Learning Using Random Walks for Dynamic Graphs**. 2019. Citado na página 47.

SCHULER, Joao Paulo Schwarz et al. An enhanced scheme for reducing the complexity of pointwise convolutions in cnns for image classification based on interleaved grouped filters without divisibility constraints. **Entropy**, v. 24, p. 1264, 09 2022. Citado na página 75.

SHAFABI, Ali et al. Adversarially robust transfer learning. **CoRR**, abs/1905.08232, p. 5–8, 2019. Disponível em: <<http://arxiv.org/abs/1905.08232>>. Citado 2 vezes nas páginas 32 e 63.

SILVA DANILO HERNANE SPATTI, Rogério Andrade Flauzino Ivan Nunes da. **Redes Neurais Artificiais**. [S.l.]: Artliber, 2019. Citado 3 vezes nas páginas 10, 23 e 24.

SIMONYAN, Karen; ZISSERMAN, Andrew. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. 2014. Citado na página 54.

SZEGEDY, Christian et al. **Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning**. 2016. Citado na página 54.

SZEGEDY, C. et al. **Rethinking the Inception Architecture for Computer Vision**. 2015. Citado na página 54.

TAN, Juntao; SONG, Changkyu; BOULARIAS, Abdeslam. **A Self-supervised Learning System for Object Detection in Videos Using Random Walks on Graphs**. 2020. Citado 2 vezes nas páginas 33 e 48.

TAN, Mingxing; LE, Quoc V. **EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**. 2020. Citado na página 54.

WANG, Liang et al. **SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models**. arXiv, 2022. Disponível em: <<https://arxiv.org/abs/2203.02167>>. Citado na página 38.

WANG, Tianyu; YANG, Lin F.; WANG, Zizhuo. **Random Walk Bandits**. 2020. Citado 2 vezes nas páginas 41 e 43.

WICKET, Marcos. **Random Walks (Caminhadas aleatórias em Grafos)**. 2015. Citado 2 vezes nas páginas 43 e 44.

WU, Zonghan et al. A comprehensive survey on graph neural networks. **IEEE Transactions on Neural Networks and Learning Systems**, Institute of Electrical and Electronics Engineers (IEEE), p. 1–21, 2020. ISSN 2162-2388. Disponível em: <<http://dx.doi.org/10.1109/TNNLS.2020.2978386>>. Citado 2 vezes nas páginas 10 e 40.

ZANGENEH, E.; RAHMATI, M.; MOHSENZADEH, Y. Low resolution face recognition using a two-branch deep convolutional neural network architecture. **Expert Systems with Applications**, v. 139, 2020. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417419305561>>. Citado 3 vezes nas páginas 17, 31 e 62.

ZANGENEH, Erfan; RAHMATI, Mohammad; MOHSENZADEH, Yalda. Low resolution face recognition using a two-branch deep convolutional neural network architecture. **Expert Systems with Applications**, v. 139, p. 112854, 2020. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417419305561>>. Citado na página 17.

ZHANG, Zhen et al. **Hierarchical Graph Pooling with Structure Learning**. arXiv, 2019. Disponível em: <<https://arxiv.org/abs/1911.05954>>. Citado na página 37.

ZHANG, Ziwei; CUI, Peng; ZHU, Wenwu. **Deep Learning on Graphs: A Survey**. 2020. Citado na página 37.

Zhao, Y. et al. Cross-species protein function prediction with asynchronous-random walk. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, p. 1–5, 2019. Citado na página 47.

ZHOU, Tian; XU, Pengbo; DENG, Weihua. **Lévy walk dynamics in mixed potentials from the perspective of random walk theory**. 2020. Citado 7 vezes nas páginas 10, 13, 41, 42, 43, 44 e 61.

ZHUANG, Fuzhen et al. **A Comprehensive Survey on Transfer Learning**. 2020. Citado na página 32.