



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

WALACY DA SILVA CAMPOS

**CLASSIFICAÇÃO CONTEXTUAL DE IMAGENS POR MEIO DE REDES NEURAIS
CONVOLUCIONAIS DE GRAFOS E PODA POR SIMILARIDADE**

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2022

WALACY DA SILVA CAMPOS

CLASSIFICAÇÃO CONTEXTUAL DE IMAGENS POR MEIO DE REDES NEURAIAS CONVOLUCIONAIS DE GRAFOS E PODA POR SIMILARIDADE

Contextual image classification through fine-tuned graph neural networks

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. Pedro Henrique Bugatti

Co-orientador: Prof^a. Dra. Priscila Tiemi Maeda Saito

CORNÉLIO PROCÓPIO

2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



**Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio**



WALACY DA SILVA CAMPOS

CLASSIFICAÇÃO CONTEXTUAL DE IMAGENS POR MEIO DE REDES NEURAIS CONVOLUCIONAIS DE GRAFOS E PODA POR SIMILARIDADE

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

Data de aprovação: 24 de Fevereiro de 2022

Prof Pedro Henrique Bugatti, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Daniel Dos Santos Kaster, Doutorado - Universidade Estadual de Londrina (UEL)

Prof Silvio Ricardo Rodrigues Sanches, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 18/03/2022.

Dedico esta pesquisa a minha família e ao marido Matheus, por sempre acreditarem em mim e me apoiarem durante todos os momentos.

AGRADECIMENTOS

A minha família e meu amado marido, Matheus Lambertucci, pelo amor e suporte incondicional que foram substanciais durante toda essa jornada. Aos meus queridos amigos, que tanto torceram por mim. Aos meus orientadores, professores e todos aqueles que contribuíram para a minha formação humana e intelectual. Agradeço a todos que me ajudaram a chegar até aqui.

RESUMO

CAMPOS, Walacy. CLASSIFICAÇÃO CONTEXTUAL DE IMAGENS POR MEIO DE REDES NEURAS CONVOLUCIONAIS DE GRAFOS E PODA POR SIMILARIDADE. 82 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Técnicas de visão computacional se tornaram populares na indústria de tecnologia. Esse sucesso é derivado do avanço em recursos computacionais e na geração de dados complexos. Desta forma, o uso de algoritmos como redes neurais convolucionais para tarefas de classificação foi difundido entre muitas companhias. Apesar de tais redes apresentarem resultados interessantes para classificação de imagens, sua dificuldade em lidar com aspectos de relacionamento entre os dados abriu a possibilidade para o uso das redes neurais convolucionais de grafos. Contudo, essas redes de grafos possuem um gargalo relacionado à escalabilidade de recursos que é pouco explorado pelo estado da arte para o contexto de visão computacional. Desta forma, este trabalho propõe uma abordagem de poda com objetivo de reduzir o consumo de recursos computacionais das redes neurais convolucionais de grafos. Foi realizada uma avaliação experimental extensiva, a qual mostra que a abordagem proposta pode ser promissora, sendo capaz de reduzir o tempo de treinamento em 60,24%, além do consumo de memória de placa gráfica em 23,11%. O trabalho pode contribuir para a redução de recursos computacionais em *data centers*, sendo um dos principais ofensores no que tange a custos e emissão de carbono.

Palavras-chave: Redes Neurais Artificiais, Redes Neurais Convolucionais, Redes Neurais de Grafos, Podas, Otimização

ABSTRACT

CAMPOS, Walacy. Contextual image classification through fine-tuned graph neural networks. 82 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Computer vision techniques have become popular in the technology industry. This success is derived from advances in computational resources and the generation of complex data. In this way, the use of resources such as convolutional neural networks for classification tasks was widespread among companies. Although such networks present interesting results for image classification, their difficulty in dealing with data relationship aspects opened a possibility for the use of graph convolutional neural networks. However, these networks have a bottleneck related to the scalability of resources, which is not well explored by the state of the art for the context of computer vision. In this way, this research proposes an approach with the objective of reducing the consumption of computational resources of the graph convolutional neural networks. In this research, there were several tests executed, which revealed this work can be promissory in many scenarios, capable of reducing the training effort by reductions in the memory consumption of up to 60.24% and memory usage from a graphical card in 23.11%. The work can contribute to the reduction of computing resources in data centers, being one of the main offenders in terms of costs and carbon emissions.

Keywords: Artificial Neural Networks, Convolutional Neural Networks, Graph Neural Networks, Pruning, Optimization

LISTA DE FIGURAS

FIGURA 1	– Exemplo do funcionamento do modelo para a em busca na base RBCC, extraído de Hong et al. (2015)	26
FIGURA 2	– Exemplo do funcionamento do modelo para a em busca na DBPedia, extraído de Hong et al. (2015)	26
FIGURA 3	– Exemplo de <i>lattice</i> invertida, adaptado de Hong et al. (2015)	27
FIGURA 4	– Exemplo de poda anti-monotônica, adaptado de Hong et al. (2015)	28
FIGURA 5	– Poda vertical, adaptado de (Hong et al., 2015)	29
FIGURA 6	– Poda horizontal, adaptado de Hong et al. (2015)	30
FIGURA 7	– Poda <i>DeepLight</i> , adaptado de Deng et al. (2021)	31
FIGURA 8	– Pipeline da abordagem proposta	33
FIGURA 9	– Amostra adaptada de retângulos envolventes em imagem da base de dados MIT67	34
FIGURA 10	– Ilustração da relação entre grafo e vetores de característica	35
FIGURA 11	– Ilustração de grafos totalmente conectados para grafos podados	35
FIGURA 12	– Ilustração de grafos totalmente conectados para grafos com arestas podadas	38
FIGURA 13	– Ilustração de grafos com arestas podadas para grafos com vértices podados	40
FIGURA 14	– Distribuição de imagens por base de imagens	45
FIGURA 15	– Distribuição de retângulos envolventes por base de imagens	45
FIGURA 16	– Imagem extraída da base de imagens MIT-67 da classe airport (aeroporto)	46
FIGURA 17	– Imagem extraída da base de imagens MIT-67 da classe library (livraria)	47
FIGURA 18	– Imagem extraída da base de imagens MIT-67 da classe restaurant (restaurante)	47
FIGURA 19	– Distribuição de imagens por classe para a base de imagens MIT-67	48
FIGURA 20	– Distribuição de retângulos envolventes por classe para a base de imagens MIT-67	49
FIGURA 21	– Imagem extraída da base de imagens Unrel da classe cat on the top of dog (gato sob cachorro)	50
FIGURA 22	– Imagem extraída da base de imagens Unrel da classe dog ride bike (cachorro pilota a bicicleta)	50
FIGURA 23	– Imagem extraída da base de imagens Unrel da classe person under table (pessoa embaixo da mesa)	51
FIGURA 24	– Distribuição de imagens por classe para a base de imagens Unrel	52
FIGURA 25	– Distribuição de retângulos envolventes por classe para a base de	

imagens Unrel	53
FIGURA 26 – Imagem extraída da base de imagens VRD da classe has (possui)	54
FIGURA 27 – Imagem extraída da base de imagens VRD da classe has (possui)	55
FIGURA 28 – Imagem extraída da base de imagens VRD da classe wearing (vestindo)	55
FIGURA 29 – Distribuição de imagens por classe para a base de imagens VRD .	56
FIGURA 30 – Distribuição de retângulos envolventes por classe para a base de imagens VRD	57
FIGURA 31 – Gráfico de radar dos resultados dos experimentos na arquitetura EfficientNetB7 para a base de dados MIT-67	61
FIGURA 32 – Gráfico de radar dos resultados dos experimentos na arquitetura InceptionV3 para a base de dados MIT-67	61
FIGURA 33 – Gráfico de radar dos resultados dos experimentos na arquitetura ResNet50 para a base de dados MIT-67	62
FIGURA 34 – Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados MIT-67	62
FIGURA 35 – Visualização geral dos resultados dos experimentos para a base de dados MIT-67	63
FIGURA 36 – Gráfico de radar dos resultados dos experimentos na arquitetura EfficientNetB7 para a base de dados Unrel	64
FIGURA 37 – Gráfico de radar dos resultados dos experimentos na arquitetura InceptionV3 para a base de dados Unrel	65
FIGURA 38 – Gráfico de radar dos resultados dos experimentos na arquitetura ResNet50 para a base de dados Unrel	65
FIGURA 39 – Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados Unrel	66
FIGURA 40 – Visualização geral dos resultados dos experimentos para a base de dados Unrel	67
FIGURA 41 – Gráfico de radar dos resultados dos experimentos na arquitetura EfficientNetB7 para a base de dados VRD	69
FIGURA 42 – Gráfico de radar dos resultados dos experimentos na arquitetura InceptionV3 para a base de dados VRD	69
FIGURA 43 – Gráfico de radar dos resultados dos experimentos na arquitetura ResNet50 para a base de dados VRD	70
FIGURA 44 – Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados VRD	70
FIGURA 45 – Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados VRD	71

LISTA DE TABELAS

TABELA 1	– Número de classes de imagens para arquiteturas sem poda e com poda para cada uma das base de dados	42
TABELA 2	– Total de combinações por parametrizações dos experimentos.	42
TABELA 3	– Recursos de máquina utilizados	43
TABELA 4	– Bibliotecas e pacotes utilizados	44
TABELA 5	– Métricas utilizadas nos experimentos	59

LISTA DE SIGLAS

(AP)	Apredizado de Máquina
(RNA)	Redes Neurais Artificiais
(IA)	Inteligência Artificial
(AP)	Aprendizado Profundo
(RNA)	Rede Neural Artificial
(MCP)	Modelo McCulloch e Pitts
(RNR)	Redes Neurais Recorrentes
(RNC)	Redes Neurais Convolucionais
(RNG)	Redes Neurais de Grafos
(RNCG)	Redes Neurais Convolucionais de Grafos
(RBCC)	Repositório Bibliográfico de Ciência da Computação
(TA)	Transferência de Aprendizado
(GRAM)	Memória RAM da Placa Gráfica
(VRD)	Visual Relationship Detection

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo	14
1.2	Organização do Trabalho	14
2	CONCEITOS RELACIONADOS	15
2.1	Aprendizado de Máquina	15
2.1.1	Aprendizado Profundo	16
2.1.2	Redes Neurais Artificiais	17
2.1.3	Redes Neurais Convolucionais	18
2.2	Domínio de Grafos	19
2.2.1	Redes Neurais em Grafos	19
2.2.2	Convolução em Grafos	20
2.2.3	Grafos Espectrais e Espaciais	22
2.2.4	Poda em Redes Neurais	23
2.2.5	Trabalhos Relacionados	24
3	ABORDAGEM PROPOSTA	33
3.1	Algoritmo de Poda	36
3.1.1	Poda de Arestas	38
3.1.2	Poda de Vértices	39
3.2	Experimentos	40
3.2.1	Cenários Experimentais	41
3.2.1.1	Seleção e Particionamento dos Dados	41
3.2.1.2	Busca em Grade	42
3.2.1.3	Especificações Técnicas da Máquina	43
3.2.1.4	Recursos e Bibliotecas	43
3.2.2	Descrição das Bases de Imagens	44
3.2.2.1	MIT-67	46
3.2.2.2	Unrel	49
3.2.2.3	Visual Relationship Detection	54
3.2.3	Treinamento dos Modelos em Escala	57
3.2.3.1	Computação em Nuvem	58
3.2.4	Coleta de Métricas	58
3.3	Resultados Obtidos	59
3.3.1	MIT-67	60
3.3.2	Unrel	64
3.3.3	VRD	68

4 CONSIDERAÇÕES FINAIS	72
4.1 Trabalhos Futuros	73
4.1.1 Algoritmos de matriz de fatoração para encontrar arestas e vértices relevantes	73
4.1.2 Poda por grau de diversidade	73
4.1.3 Poda por posição de elementos	73
4.2 Publicações	74
REFERÊNCIAS	75
Apêndice A – DEMAIS CENÁRIOS EXPERIMENTAIS	79
Apêndice B – DECISÕES DO TRABALHO	80
Apêndice C – REPOSITÓRIO DA IMPLEMENTAÇÃO	81

1 INTRODUÇÃO

A visão computacional se tornou popular na indústria de tecnologia moderna. Grande parte desse sucesso é derivado do crescimento dos dados em diversos contextos, avanços em recursos computacionais, e do poder de generalização das técnicas de visão computacional, que possuem uma grande representatividade em aplicações comerciais (BRONSTEIN et al., 2017). Dentre uma ampla gama de técnicas envolvendo a classificação de imagens em visão computacional, são destacadas abordagens envolvendo redes neurais convolucionais, que são instrumentos interessantes para generalização. No entanto, abordagens que consideram relação, como as redes neurais convolucionais de grafos (BRUNA et al., 2014) têm apresentado resultados interessantes em tarefas de classificação complexas.

As redes neurais convolucionais de grafos utilizam da sua estrutura de dados para representar relacionamentos, que por sua vez, são importantes para o processo de aprendizado da rede. No entanto, elas possuem algumas lacunas que são relacionadas à sua escalabilidade, com o alto consumo de memória. Outrossim, as conexões do grafo também são tratadas com importância equânime (KIPF; WELLING, 2019), desconsiderando aspectos contextuais, como similaridade.

No contexto de redes neurais artificiais, técnicas de poda neural são exploradas desde 1980, como uma forma de reduzir o consumo de recursos computacionais da rede (BLALOCK et al., 2020)(CHECHIK et al., 1998). Alguns trabalhos no estado da arte elaboraram políticas de poda para diferentes contextos, (Chung et al., 2019) (LI et al., 2018) (Hong et al., 2015). Outros não exploram, contudo, destacam aspectos de escalabilidade (KIPF; WELLING, 2019). Entretanto, as abordagens focadas em redes neurais de grafos em visão computacional são pouco

exploradas na literatura.

Desta forma, essa pesquisa almeja construir um algoritmo de poda para redes neurais de grafos no contexto de visão computacional capaz de reduzir o consumo de recursos computacionais utilizando a informação contextual presente nas imagens, como aspecto de escalabilidade.

Os resultados obtidos por meio dos experimentos executados mostraram que o método pode ser interessante para sua aplicabilidade em diversos contextos para diferentes bases de imagens. É possível verificar que existe uma redução considerável, no que tange ao consumo de infraestrutura, com pequenos impactos em métricas de assertividade.

1.1 OBJETIVO

A partir da observação da literatura e da descoberta do gargalo identificado no campo de redes neurais de grafos, esta pesquisa tem como objetivo trazer uma abordagem de poda utilizando as informações contextuais obtidas no relacionamento entre as imagens de um grafo para redução do consumo de recursos computacionais.

1.2 ORGANIZAÇÃO DO TRABALHO

O presente trabalho foi estruturado da seguinte forma:

- No capítulo 2, são abordados os principais tópicos de conhecimentos relacionados a esta pesquisa buscados na literatura;
- No capítulo 3, a abordagem proposta para atingir o objetivo deste trabalho, assim como todo o processo da execução dos experimentos realizados e os resultados obtidos;
- Por fim, no capítulo 4, são listados os possíveis trabalhos futuros a partir da pesquisa e publicações realizadas.

2 CONCEITOS RELACIONADOS

2.1 APRENDIZADO DE MÁQUINA

O campo de aprendizado de máquina (AP) teve diversas contribuições ao longo da história. De Aristóteles, com a introdução do Associacismo e a exploração do aprendizado, até as Redes Neurais Artificiais (RNA) mais modernas, presentes em vários contextos diferentes na área de Inteligência Artificial (IA) (WANG; RAJ, 2017).

O tema de aprendizado de máquina também é tratado como uma questão de como elaborar programas de computador que melhoram ao decorrer de sua aquisição de experiências, como descrito por Thomas Mitchell (MITCHELL, 1997).

Arthur Samuel, também define o tópico como um campo de estudo em que habilita um computador a aprender, de forma que ele não precise ter sido programado anteriormente (RUSSELL; NORVIG, 2010). O mesmo tema já fora levantado no artigo de Alan Turing, com o questionamento de que se as máquinas poderiam realmente pensar ou não, sendo este, um trabalho que trouxe um processo de avaliação para identificar um robô de um ser humano, rotulado como "O Jogo da Imitação"(TURING, 1950).

No decorrer da evolução da área e com passar de diversas definições, são inúmeros algoritmos que estão relacionados com o campo de Aprendizado de Máquina. Bonetto e Latzko (2020) apresentaram um grupo de macrocategorias para os definir, sendo eles: supervisionado, que representam modelos que utilizam dados rotulados; não-supervisionado, que representam modelos cujos dados não possuem rótulos definidos; e reforço, que utilizam estratégias de recompensa ou punição para a construção de modelos. Ainda que essas macrocategorias abranjam grande parte dos algoritmos utilizados para problemas cujo contexto são deveras diversos, partindo de

classificação até mesmo clusterização. O autor Sarker (2021) demonstra uma quarta classe de classificações para os algoritmos, sendo ela a classe semi-supervisionado, que define a utilização de dados tanto rotulados quanto não rotulados.

A aplicabilidade do campo, tendo como um dos princípios básicos que caracterizam algoritmos de aprendizado de máquina consistem em otimizar a performance de atividades, dado um conjunto de amostras do passado ou experiências. Exemplos de aplicações destes algoritmos estão presentes em várias áreas, como varejo, bancos, mercado de ações, indústria e até mesmo medicina (ALPAYDIN, 2014). Todas essas áreas geram dados que podem ser utilizados por diversas técnicas e potencializar o uso de métodos ainda mais robustos (SARKER, 2021).

2.1.1 APRENDIZADO PROFUNDO

O aprendizado profundo (AP), é um campo dentro da área de aprendizado de máquina, que também possui uma gama de contribuições ao longo da história, partindo desde a tentativa da compreensão do cérebro humano, passando pela introdução do método de *dropout*, até os dias atuais (WANG; RAJ, 2017).

O termo passou a ser utilizado no campo de Aprendizado de Máquina a partir de 1986, posteriormente sendo adotado para o contexto de Redes Neurais Artificiais (RNA) nos anos 2000, conforme definido em Minar e Naher (2018). Os autores também apontam algumas das técnicas envolvidas no campo, ressaltando que uma das mais populares é a de Redes Neurais Artificiais. Esta técnica, por sua vez, possui uma ampla utilização na área, sendo aplicada em tarefas como, classificação de imagens, detecção de objetos; processamento de fala; reconhecimento e verificação facial; e muitos outros.

Lecun et al. (2015) apontam que o aprendizado profundo consiste na geração de modelos que são compostos por múltiplas camadas de neurônios artificiais, que são capazes de aprender o padrão de dados em vários níveis de abstração. O sucesso do uso de aprendizado profundo veio por meio dos avanços em recursos computacionais, permitindo o uso de uma enorme quantidade de dados no

treinamento de Redes Neurais Artificiais com várias camadas. Com este feito, várias aplicações no campo de Aprendizado Profundo se beneficiaram (BRONSTEIN et al., 2017).

2.1.2 REDES NEURAIAS ARTIFICIAIS

O tema de redes neurais foi inspirado pela observação das redes neurais biológicas no processo de interconexão de neurônios, sendo elas, capazes de prover um método de aprendizado (MITCHELL, 1997). A estrutura de uma rede neural artificial é composta por um número de unidades e conexões em que cada conexão trabalha com um determinado peso e cada unidade possui entradas e saídas de outras unidades, sendo essas saídas, configuradas como um nível de ativação (RUSSELL; NORVIG, 2010).

A estrutura em Redes Neurais Artificiais é similar possui grande similaridade com as primeiras abordagens no tema, como o modelo de Alexander Bain, citado em (WANG; RAJ, 2017). Esse modelo faz uma conexão com o ano de 1873, marcado pela introdução de grupos neurais, sendo um dos modelos em redes neurais artificiais mais antigos já introduzidos, que propeliu e inspirou a teoria Hebbiana, no qual são discutidas questões sobre as sinapses neuronais e o aprendizado (LANGILLE; BROWN, 2018).

A linha do tempo de Wang e Raj (2017), também retrata sobre os eventos decorridos de McCulloch e Pitts, em 1943, que trouxeram o modelo McCulloch e Pitts (MCP), considerado o ancestor das redes neurais artificiais, assim como o de Donald O. Hebb, em 1949, que é considerado o precursor do tema a partir da criação da regra de aprendizado Hebbiana. Vale ressaltar que o primeiro modelo de perceptron artificial foi introduzido logo em seguida, por Frank Rosenblatt, em 1958, sendo este um modelo capaz de aprender através do associacionismo.

As redes neurais artificiais são de grande interesse comum, conforme em Alpaydin (2014), em que é enfatizado que com elas pode ser possível construir melhores sistemas de computador, capazes de operar em diferentes domínios, como em visão, fala e aprendizado. O autor também cita que elas possuem um evidente

impacto econômico quando implementada em máquinas, que também pode ser visualizado nos trabalhos de Bronstein et al. (2017) e Co-operation e Development (2019), além de algumas variações, como as redes neurais Feedforward, redes neurais recorrentes (RNR), redes neurais convolucionais, além de duas variações de domínio, como as redes neurais de grafos (GORI et al., 2005).

2.1.3 REDES NEURAS CONVOLUCIONAIS

As redes neurais convolucionais (RNC) são estruturas que operam no espaço euclidiano (BRONSTEIN et al., 2017) e são derivadas da família de redes neurais artificiais, caracterizadas por suas camadas de convolução (MINAR; NAHER, 2018). A função de uma camada convolucional é realizar processos de convoluções, posteriormente detectando os conjuntos de características da camada anterior.

Além disso, camadas de *pooling*, também denominadas como camadas de subamostragem (WANG; RAJ, 2017) fazem a junção de características que apresentam similaridade. A primeira rede neural convolucional foi proposta em Lecun et al. (2015), nomeada como LeNet. A LeNet é conhecida por ser uma rede classificadora de dígitos manuscritos, ainda que estes apresentassem uma grande variação em suas características, como rotação, escala e espessura (WANG; RAJ, 2017).

As RNCs possuem uma grande conexão com o que se entende do funcionamento do córtex visual humano. O córtex visual humano abrange uma parte do córtex cerebral que é responsável por processar as informações visuais que são obtidas por meio do sistema visual humano. Algumas regiões do córtex são amplamente conhecidas por serem capazes de detectar características específicas, como: bordas, regiões com alto contraste, cores, formatos, bem como tarefas de nível semântico, como reconhecer faces (WANG; RAJ, 2017). As RNCs ainda herdam da família de redes neurais artificiais características de paralelização das operações, aprendizado por amostras e uma ligação com a memória associativa (ARBIB, 2002).

2.2 DOMÍNIO DE GRAFOS

A teoria dos grafos foi iniciada a partir 1735 por Leonhard Euler, conforme em Sachs et al. (1988). Contudo, apenas em aproximadamente 1940, ela foi estabelecida e se tornou popular (XU; BAO, 2002). Um grafo representa uma abstração matemática de contextos nos quais se estabelecem relações.

É possível representar um grafo como $G = G(V, A)$ em que V é um conjunto de vértices e A de arestas. Desta forma, assumindo a representação de um vértice pelo termo v e uma aresta por a , é sabido que $v \in V$ e $a \in A$. A ligação entre um vértice e outro pode ser representada por $a_{i,j} = (v_i, v_j) \in A$, que denota uma aresta entre v_i e v_j . Um grafo também pode ser representado por uma matrix de adjacência Ad de dimensões $N \times M$ sendo $Ad_{ij} = 1$ se $a_{ij} \in A$ e $Ad_{ij} = 0$ se $a_{ij} \notin A$, conforme em Lloyd et al. (1978).

2.2.1 REDES NEURAIAS EM GRAFOS

As redes neurais de grafos (RNG) são métodos baseados em aprendizado profundo dentro do domínio dos grafos (ZHOU et al., 2018). As RNGs são compostas por estruturas de dados que operam no espaço não-euclidiano, ainda que apresentem operações estruturadas como a convolução, estritamente presente nas RNC como definido em Bronstein et al. (2017).

Os primeiros estudos com as RNG foram iniciados com (Sperduti; Starita, 1997), que enfatiza a problemática acerca do uso de RNAs em lidar com estruturas de dados complexas que consideram representação de informações adicionais, como a ligação entre características.

No referido trabalho realizado em (Sperduti; Starita, 1997), é apresentado como algumas arquiteturas de RNAs podem ser generalizadas para lidar com modelos estruturados por meio da aplicação de redes neurais em grafos acíclicos. Esse trabalho foi fundamental aos consecutivos estudos com RNG.

O conceito e apresentação de um modelo de RNG foram propostos em (GORI et al., 2005), sendo considerado pelos autores uma extensão de redes neurais

recorrentes uma vez que as mesmas processam uma gama maior de classes de grafos e podem ser utilizadas para problemas focados no vértice. Esse marco inicial, também teve um aprofundamento com a contribuição de Scarselli et al. (2009) e com Gallicchio e Micheli (2010) com o seu trabalho denominado GraphESN, que se trata de uma generalização da *Echo State Network* no domínio de grafos.

2.2.2 CONVOLUÇÃO EM GRAFOS

No trabalho de Zhou et al. (2018) foi apontado que a primeira motivação das RNGs começa nas redes neurais convolucionais, uma vez que elas possuem a capacidade de extrair características espaciais em múltiplas escalas e construir representações altamente expressivas que levaram ao avanço de quase todas as áreas de aprendizado de máquina (i.e., responsável por um novo marco para o aprendizado profundo). Os pontos primordiais que regem a RNC (i.e., como a conexão local, pesos compartilhados e o uso de múltiplas camadas) são atributos de grande importância no domínio dos grafos.

O uso do processo de convolução nas RNGs se inicia e com a formulação de Bruna et al. (2014), ao tratar das possíveis generalizações que as RNCs podem trazer para o campo de sinais com ênfase no domínio spectral. Posteriormente, em Henaff et al. (2015), Defferrard et al. (2016a), Kipf e Welling (2019) e Levie et al. (2017) também trazem uma abstração sobre o uso de convolução em RNGs. As redes neurais convolucionais de grafos (RNCG) são RNGs que aplicam o processo de convolução em sua arquitetura, conforme formulado em Bruna et al. (2014). Em Li et al. (2018), é possível obter a formalização do processo de convolução para a construção spectral. É supracitado que o modelo, inicialmente, das RNCGs espectrais, conforme a equação 2, era impraticável para grandes grafos. Esse problema foi posteriormente resolvido aproximando o filtro spectral θ (auto valores da matriz laplaciana, definido na equação 4), com os polinômios Chebyshev próximos a ordem de K , conforme a formalização da equação. Os polinômios de Chebyshev, ou Chebyshev de primeira ordem, são denotados pela equação 1 conforme em Karjanto (2020).

$$T_n(\cos\theta) = \cos(n\theta) \quad (1)$$

em que T é um polinômio Chebyshev e n são números inteiros e não negativos.

Um outro ponto relatado pelos autores, Li et al. (2018), denota a aplicação da operação intitulada como K-localized ChebNet, em que a convolução é definida na equação:

$$g_{\theta} \star \approx \sum_{k=0}^K \theta' T_k(L_{sym}) s \quad (2)$$

em que $s \in R^n$ é um sinal de um grafo, sendo R^n um escalar de cada vértice, g_{θ} é um tipo de filtro espectral, a variável \star é o operador de convolução, T_k é propriamente o polinômio de Chebyshev, enquanto $\theta' \in R^K$ se trata de um vetor de coeficientes de Chebyshev.

Ainda em Li et al. (2018), os autores explicam que o modelo de RNCG é uma forma simplificada do então rotulado como Suavização de Laplace, que combina as características de um vértice com os seus vizinhos de maior proximidade. Desta forma, os autores trouxeram a definição de um tipo de grafo laplaciano, definida como L_{sym} , conforme a equação 4, sendo suas versões normalizadas, as equações 4 e 5, dado que D , representa a diagonal da matriz de adjacência Ad e L é um grafo laplaciano.

$$L := D - Ad \quad (3)$$

$$L_{sym} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \quad (4)$$

$$L_{rw} := D^{-1} L \quad (5)$$

Os autores Li et al. (2018) também explicam que a formulação de Kipf e Welling (2019) propõe o valor de $K = 1$ para o modelo exemplificado na equação 2, e aproximando o maior autovalor definido por λ_{max} de L_{sym} por 2, resultando, assim, na simplificação do modelo. Desta forma, a convolução é representada pela equação 6.

$$g_\theta \star s = \theta(I + D^{-\frac{1}{2}}AdD^{-\frac{1}{2}})s \quad (6)$$

em que θ é o coeficiente de convolução Chebyshev restante e I representa uma matriz identidade, conforme em Defferrard et al. (2016b). Os autores em Kipf e Welling (2019), ainda aplicam uma normalização, conforme a equação 7

$$I + D^{-\frac{1}{2}}AdD^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \quad (7)$$

em que $\tilde{A}d = Ad + I$ e $\tilde{D} = \sum_j \tilde{A}d_{ij}$, sendo $Ad = [a_{ij}] \in \mathbb{R}^{n \times n}$, uma matriz adjacente não negativa e $D = \text{diag}(d_1, d_2, \dots, d_n)$, a matriz de graus em que $d_i = \sum_j a_{ij}$, que é o grau do vértice i (Li et al., 2018).

2.2.3 GRAFOS ESPECTRAIS E ESPACIAIS

A colocação da construção espacial e a construção espectral é trazida em Bruna et al. (2014), que enfatiza que a forma mais direta de generalizar RNCs em contextos de grafos é a consideração de várias escalas, hierarquia e campos receptivos locais. Essa afirmação também é relatada em Defferrard et al. (2016c), ao explicitar que as RNCGs podem ser generalizadas com a propagação das características de um dado vértice aos seus respectivos vértices vizinhos.

Neste processo, em Balcilar et al. (2020), é citado que o filtro convolucional é o responsável por definir como as características dos vértices são propagadas aos vizinhos, fato que, por sua vez, assemelha-se ao processo que é derivado das RNCs. No trabalho de Li et al. (2018), é definido que na construção espacial, as RNCs em grafos tratam a convolução como operador de fragmentação para a construção de um vetor para cada vértice, por meio das informações que estão contidas nos vértices vizinhos. Por outro lado, no contexto da construção espectral as RNCs em grafos definem sua convolução pela decomposição de s e R^n , sendo s um sinal de grafo que representa R^n , tratando-se de um escalar para cada um dos vértices do grafo no domínio espectral. Assim, aplicando um filtro espectral g_θ de função de autovalores de L_{sym} , que corresponde a uma versão normalizada de grafos Laplacianos, conforme a equação 4.

A diferença substancial entre os dois, é que no contexto espectral, as arestas de um grafo são decopostas através da matriz Laplaciana, enquanto no espacial, a convolução é através de matrizes utilizando os objetos vizinhos, como descrito em Li et al. (2018).

2.2.4 PODA EM REDES NEURAIS

A definição de técnicas de poda para RNA ser comparada com a poda neural da biologia, em que terminações sinápticas são desativadas quando fracas (CHECHIK et al., 1998). O objetivo da aplicação de tais técnicas no contexto computacional, visam reduzir o consumo de recursos computacionais, através da produção de uma rede neural menor com acurácia similar. Essas técnicas são utilizadas mesmo antes de 1980 que, segundo Blalock et al. (2020), passaram a ganhar muito espaço com a expansão do uso de RNAs. Os autores também trazem uma classificação de metodologia de poda dividida em 4 categorias, conforme:

- **Poda Estrutural**

A Poda Estrutural, é responsável por causar modificações significativas na RNA, sendo capaz de remover por completo neurônios, filtros ou canais.

- **Poda Pontuada**

Essa modalidade de poda considera o valores absolutos, coeficientes ou pontuações relevantes para a RNA, sendo utilizada para podar localmente ou globalmente aqueles que apresentam os menores números quando comparados.

- **Poda Agendada**

A Poda Agendada se trata de uma modalidade de poda divisível em diferentes passos.

- **Poda Refinada**

Envolve o processo de utilização de pesos de treinamentos passados antes da poda.

2.2.5 TRABALHOS RELACIONADOS

O estudo sobre podas em redes neurais convolucionais em grafos é explorado em diversos campos pelo estado da arte com diferentes níveis de abstração. No primeiro trabalho avaliado, em Chung et al. (2019), intitulado como *Predicting Future Participants of Information Propagation Trees* (Prevendo Futuros Participantes de Árvores de Propagação de Informação). Neste trabalho, os autores propuseram um modelo baseado em estruturas de grafos e RNCs, para aplicação em um contexto do entendimento de como a informação pode ser difundida por meio do compartilhamento de postagens em mídias sociais. Isso foi feito de forma que o modelo gerado pudesse prever futuros usuários para a rede de propagações de informações. O referido trabalho se caracteriza por aprender a representação latente de uma árvore de propagação e a combinação desse fator com a representação latente do usuário. Os autores traçaram uma estratégia de poda que parte de uma linha cuja regra de inclusão está formulada da seguinte maneira: um usuário u_i participaria da propagação em uma árvore Tri por meio de um processamento que utiliza as interações entre os usuários para obter um valor entre 0 ou 1 por meio de suas combinações.

Primeiramente, são selecionados vértices que possuem alta influência no processo de difusão da informação. As arestas da árvore representam as ações de compartilhamento definidas de duas maneiras, segundo os autores: a compreensão do processo de compartilhamento como difusão da informação de um publicador da postagem para um usuário compartilhador, e a compreensão do processo de compartilhamento como um feedback de interesse do usuário compartilhador.

Neste trabalho, o peso das arestas são definidas pela frequência de compartilhamento dos usuários. Posteriormente, são selecionados os vértices que estiverem próximos de u_i , com a justificativa deles serem os mais influentes no processo difusão da informação. O processo de poda, neste contexto, tem o objetivo de selecionar vértices representativos e eliminar os demais de forma a obter uma melhor representação de uma árvore.

A literatura também contempla alguns trabalhos que denotam um problema

com a escalabilidade dos modelos de convoluções em RNCs em grafos, descrito na segunda pesquisa analisada, Li et al. (2018), no trabalho *Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning* (Aprofundamento de ideias em Redes Neurais Convolucionais de Grafos para Aprendizado Semi-supervisionado). Esse trabalho traz à tona um problema de escalabilidade com a perda de performance das RNCs em grafos ao longo do aumento das camadas de convolução na arquitetura. No trabalho de Li et al. (2018), ocorre uma análise das RNCG em que fica explícita a correlação entre a suavização Laplaciana¹ e o processo de convolução de uma RNCG, fato que os autores consideram requisito chave para o seu funcionamento. Os autores também enfatizam que as convoluções nesse contexto são responsáveis por agrupar vértices adjacentes, de forma que eles convirjam a um ponto unitário no espaço.

O terceiro trabalho analisado nesta pesquisa, dos autores Hong et al. (2015), intitulado como *Subgraph Matching with Set Similarity in a Large Graph Database* (Correspondência de Subgrafos com Semelhança de Conjunto em um Grande Banco de Dados de Grafos), foram propostas abordagens de podas de arestas em um contexto de busca. Os autores propuseram um modelo de busca por meio de um subgrafo baseado em similaridade, de forma que cada vértice estaria associado com um conjunto de elementos de pesos dinâmicos especificados por usuários em diferentes consultas de acordo com os dados envolvidos em cada aplicação alvo. Esse trabalho traz à tona dois exemplos de aplicação. O primeiro, com ênfase em encontrar artigos na base rotulada pelo título, Repositório Bibliográfico de Ciência da Computação (RBCC), que se trata de um repositório bibliográfico de ciência da computação (LEY, 2002). Sendo que nesse contexto, dado um grafo G , cada vértice representa um artigo com um conjunto de palavras-chave com um peso $W(e)$, em que W é uma função de peso e e é o elemento, que é a relevância de uma palavra para o artigo, estabelecidos por arestas que são representadas pelas palavras citadas, conforme a Figura 1.

O segundo exemplo explora o universo da DBPedia, que é um projeto que extrai dados estruturados do site Wikipedia (AUER et al., 2007). Neste cenário, os vértices são um conjunto de palavras que compõem uma entidade, sendo

¹é definido para cada vértice em um reticulado uma nova posição com base em informações locais (i.e., a posição dos vizinhos) e o referido vértice é movido para tal posição

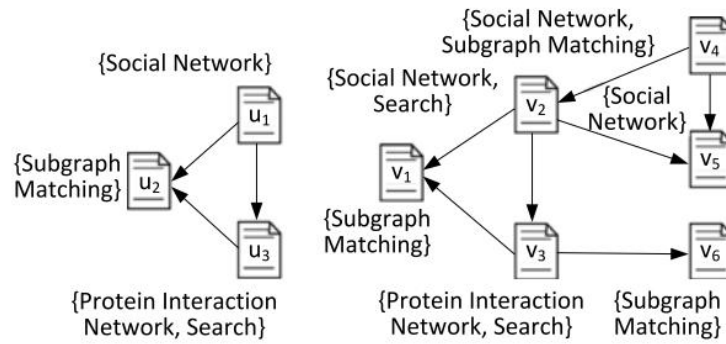


Figura 1: Exemplo do funcionamento do modelo para a em busca na base RBCC, extraído de Hong et al. (2015)

relacionados entre si por arestas que representam fatos entre as entidades, conforme a Figura 2.

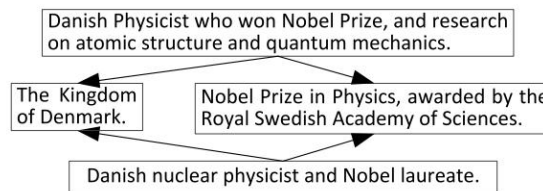


Figura 2: Exemplo do funcionamento do modelo para a em busca na DBPedia, extraído de Hong et al. (2015)

Para ambos os exemplos, o uso do modelo de Hong et al. (2015) retornaria um subgrafo de acordo com a similaridade encontrada entre os grafos após a execução de uma consulta, tratando-se assim de um problema de otimização no cenário de busca em grandes bancos de dados. Os autores relatam a construção da estrutura para lapidação e filtragem, para as abordagens relacionadas, envolvendo processos de podas rotuladas como: anti-monotônica, vertical e horizontal utilizando como base um cálculo que envolve similaridade entre textos, considerando para tanto a medida de similaridade denominada Weighted Jaccard, a qual é definida formalmente na Equação 8.

$$\text{sim}(S(u), S(v)) = \frac{\sum_{a \in S(u) \cap S(v)} W(e)}{\sum_{a \in S(u) \cup S(v)} W(e)} \quad (8)$$

em que $W(e)$ é o peso do elemento e , sendo $W(e)$ não negativo, sim representa a função de similaridade, S representa um conjunto de elementos, u um usuário e v representa um vértice.

A correspondência de um vértice v em um grafo G para uma consulta u acontece se $\text{sim}(S(u), S(v)) > t$, de forma que t é definido pelo usuário. Para os processos de poda, os autores propuseram iniciar pela construção de uma estrutura chamada de reticulado (*lattice*) invertido de padrões, que em suma, trata-se de uma estrutura de vértices.

Para essa construção, primeiro são procurados padrões em conjuntos de todos os vértices em um grafo G . Posteriormente, esses padrões são armazenados na estrutura de reticulado, em que cada vértice intermediário é considerado um padrão frequente, sendo que tais vértices também representam um subconjunto dos nós descendentes. Desta forma, esses padrões são organizados em estruturas adjacentes, conforme a Figura 3. A Figura 3 ilustra os elementos e_1 Correspondência de Subgrafos (*Subgraph Matching*), e_2 Rede de Interação de Proteína (*Protein Interaction Network*), e_3 Rede Social (*Social Network*) e e_4 Busca (*Search*) que são informações dos vértices: $v_1 = \{e_1\}$, $v_2 = \{e_3, e_4\}$, $v_3 = \{e_2, e_4\}$, $v_4 = \{e_1, e_3\}$, $v_5 = \{e_3\}$ e $v_6 = \{e_1\}$.

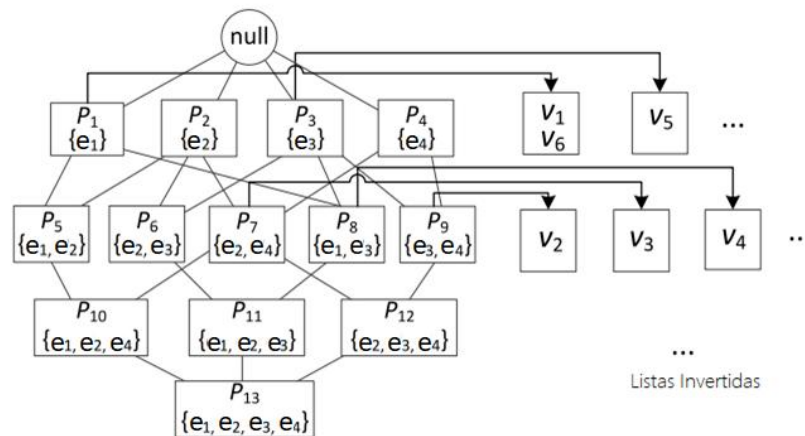


Figura 3: Exemplo de *lattice* invertida, adaptado de Hong et al. (2015)

Uma vez que essa estrutura é construída, a poda anti-monotônica considera que dado um vértice de consulta u , para cada padrão P frequentemente acessado no reticulado invertido, se o resultado de $UB(S(u), P) < t$, sendo UB uma função intitulada como *Upper Bound*, que se trata de uma função para obter um valor superior, todos os vértices nos reticulados $L(P)$ e $L(P')$ poderão ser podados, sendo P' um nó descendente de P no reticulado. O valor de t , conforme relatado, é definido pelo usuário, enquanto a Equação 9 define formalmente a função UB .

$$UB(S(u), S(v)) = \frac{\sum_{e \in S(u)} W(e)}{\sum_{e \in S(u) \cup S(v)} W(e)} \geq sim(S(u), S(v)) \quad (9)$$

Por exemplo, ao considerar $UB(S(u3), P6) = 0.55$ e $t = 0.6$, os nós cinzas, conforme a Figura 4, podem ser podados uma vez que os valores resultantes da Equação da função UB estão abaixo do limiar definido.

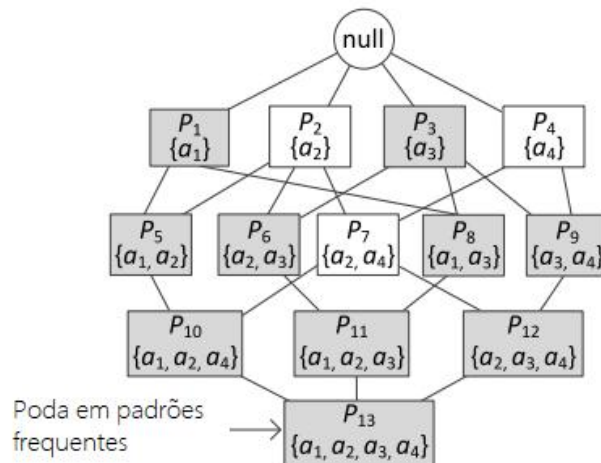


Figura 4: Exemplo de poda anti-monotônica, adaptado de Hong et al. (2015)

Esse processo de poda é o que mais se assemelha ao proposto neste trabalho, uma vez que para a presente abordagem proposta são consideradas arestas ponderadas obtidas por meio da distância euclidiana. Contudo, a abordagem proposta concentra esforços em buscar similaridade entre objetos em imagens. Além disso, a abordagem proposta encontra automaticamente o limiar de poda adequado para cada nó do grafo baseado na vizinhança dos mesmos.

De maneira divergente, o trabalho de Hong et al. (2015) define o limiar com base na relevância que os usuários dão às palavras-chave, desta forma, existe um grande esforço manual para que a política de poda seja traçada.

Passando para a a poda vertical proposta em Hong et al. (2015), dado um conjunto de consultas $S(u)$ e um padrão frequente P no reticulado, se P não for um padrão frequente (ou um descendente) em $S(u)$ ' s p-prefixo, todos os vértices da lista $L(P)$ podem ser podados.

Em linhas gerais, os autores buscam um prefixo que siga o princípio do *prefix*

filter descrito em Chaudhuri et al. (2006). Tal filtro explicita que se dois conjuntos canonizados forem similares, os seus prefixos devem se sobrepor, caso contrário, eles não terão elementos suficientes em comum. Desta forma, elementos que apresentam um dado prefixo podem ser podados. Na Figura 5, os autores elegem o prefixo a_2 .

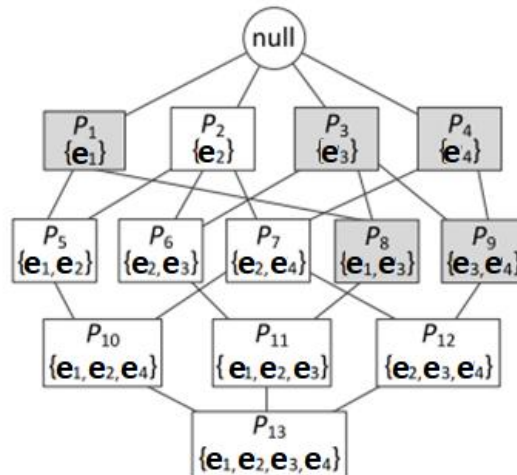


Figura 5: Poda vertical, adaptado de (Hong et al., 2015)

Por fim, ainda sobre Hong et al. (2015), na poda horizontal, a quantidade de itens na lista, definida pelo processo de identificação de padrões, é levada em consideração. Os autores enfatizam que uma consulta $S(u)$ não pode ser similar a um conjunto maior, tampouco a um conjunto menor. Desta forma, esta poda remove os níveis superiores a uma função $LU(u)$, isto é, ao tamanho da lista em consulta. Um exemplo de tal poda é ilustrado na Figura 3, em que $LU(u3) = 2$. Assim, valores maiores que 2 devem ser podado, preservando a integridade de tamanho da lista.

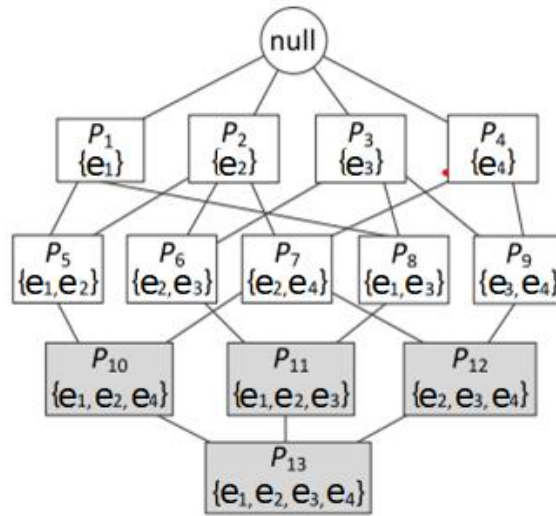


Figura 6: Poda horizontal, adaptado de Hong et al. (2015)

Diferentemente, no presente trabalho, a abordagem proposta não utiliza estratégias de prefixo, bem como não é considerado no processo de poda o número de arestas de uma lista. Nos cenários abordados em Hong et al. (2015), são exploradas técnicas de filtragem (considerando a poda) para melhoria das consultas. Entretanto, não é explorada a similaridade intrínseca a objetos (i.e., por meio de suas características). Os mesmos consideram pesos definidos pelos usuários. Este trabalho Hong et al. (2015), direciona, também, sua poda para os vértices que, por sua vez, são informações. No modelo proposto nesta pesquisa, os vértices não são podados, desta forma, não se descarta nenhuma informação. Outrossim, a implementação do framework proposto é complexa, sendo dividida em vários processos para ser implementada.

Em Deng et al. (2021), o quarto trabalho analisado, intitulado *DeepLight: Deep Lightweight Feature Interactions for Accelerating CTR Predictions in Ad Serving* (DeepLight: interações de características leves e profundas para acelerar as previsões de CTR na servidores de anúncios), os autores exploraram um contexto visando os cenários de servidores de anúncio, que necessitam trabalhar em uma baixa latência, dada a velocidade de resposta que precisam apresentar nos modelos. Os autores aplicaram uma poda estrutural, chamada *DeepLight* em uma arquitetura de *DeepFM*, que utiliza uma combinação de uma RNP e Matrizes de Fatoração (GUO et al., 2017).

A ideia central por trás do trabalho foi reduzir pesos, interações redundantes e alguns elementos esparsos nos vetores de característica, a fim de alcançar três pilares que os autores consideram ótimos para sua proposta: alta qualidade, que consiste em manter uma boa acurácia; baixa latência, sendo esse, um dos aspectos chaves para o trabalho; baixo consumo, reduzindo o uso de memória. A arquitetura final dos autores é ilustrada na Figura 7.

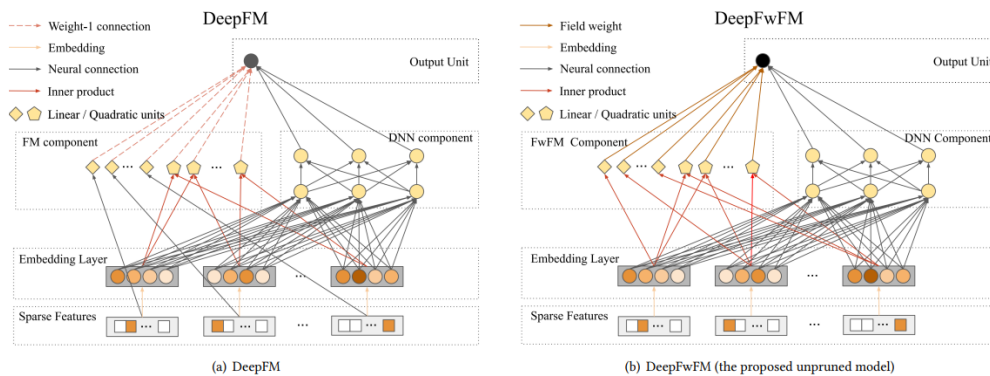


Figura 7: Poda *DeepLight*, adaptado de Deng et al. (2021)

Essa poda envolve o que os autores rotularam como *Deep-Field-weighted Factorization Machine* ou DeepFwFM, que se trata de um modelo de otimização, conforme a equação:

$$\phi_{DeepFwFM}(w, v, vec, R) = \phi_{Deep}(w, vec) + \phi_{FwFM}(v, vec, R) \quad (10)$$

em que w é o parâmetro da RNP que possui pesos e *biases*, $v \in \mathbb{R}^{n \times k}$ e e são conjuntos de vetores de característica, $R \in \mathbb{R}^{n \times k}$ é uma matriz do modelo de fatoração, responsável pela interação de relações relevantes e ϕ_{FwFM} é um modelo de eficiência.

No trabalho de Deng et al. (2021) é importante ressaltar que as análises ocorreram com ênfase em predição, uma vez que o foco era no serviço do modelo. Outrossim, as bases de dados exploradas são do contexto de textos e anúncios, sendo um dado diferente de imagens, propostos neste trabalho.

Por fim, no último trabalho analisado, dos autores Kipf e Welling (2019), em *Semi-supervised classification with graph convolutional networks* (Classificação Semi-supervisionada com Redes Neurais Convolucionais de Grafos), no qual aborda-se a semi-supervisão no contexto de RNCs em grafos, são realizadas análises envolvendo a complexidade de espaço (memória principal) requerida para alocação de grafos no contexto de RNCs. Além disso, em tal trabalho são considerados grafos com areastas não ponderadas. Assim, a questão da poda em grafos para as RNCs em grafos também não foi explorada. Desta forma, o presente trabalho tem potencial de contribuir para que tais redes sejam aprimoradas.

3 ABORDAGEM PROPOSTA

Para validar a hipótese levantada neste trabalho e atingir seu objetivo, como abordagem proposta, foi empregada uma política de poda de arestas e vértices em grafos totalmente conectados utilizando, de forma automática, a agregação de informação contextual ao processo de classificação de imagens por meio de redes neurais convolucionais em grafos. A Figura 8 representa as etapas do modelo proposto.

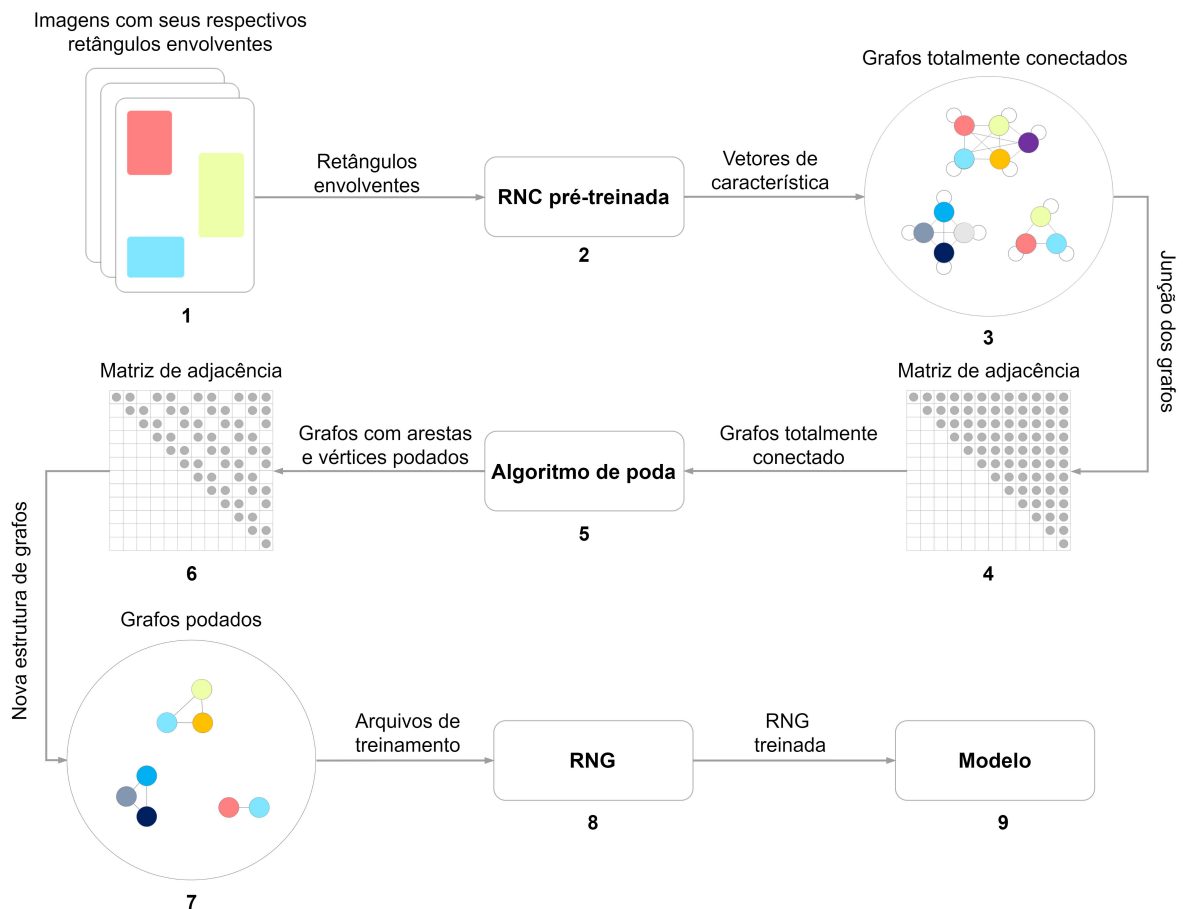


Figura 8: Pipeline da abordagem proposta

Iniciando no passo 1, são obtidos os objetos pertencentes a uma imagem por meio de seus retângulos envolventes (*bounding boxes*). Esse conjunto de informações representando por tais imagens se apresentam como as informações contextuais definidas neste trabalho, pois caracterizam o contexto de uma imagem. Cada retângulo envolvente representa uma subimagem de uma imagem maior, que são interconectadas na estrutura em grafo, conforme exemplificado na Figura 9.

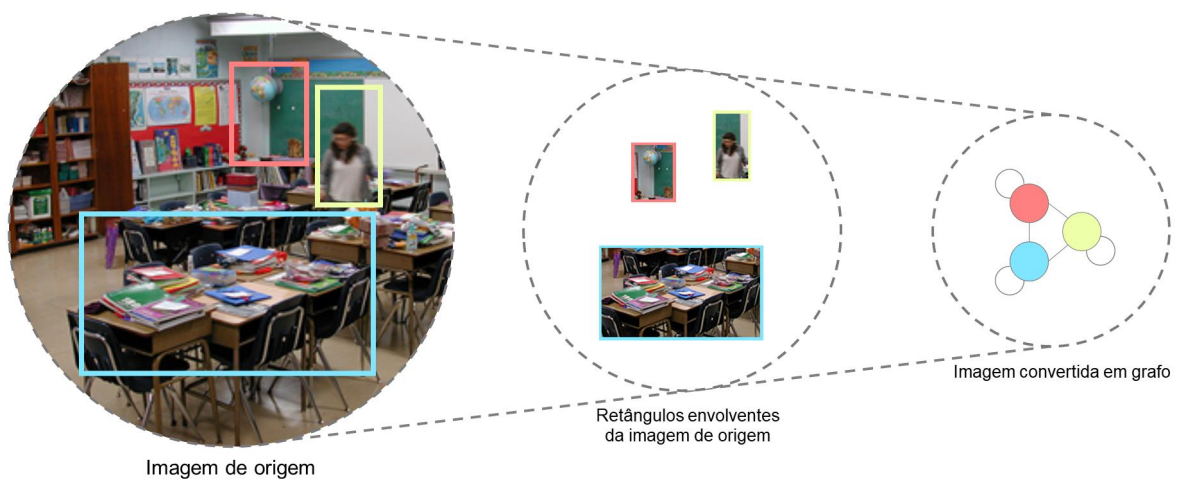


Figura 9: Amostra adaptada de retângulos envolventes em imagem da base de dados MIT67

Seguido pelo passo 2, tais objetos são descritos por meio de redes neurais convolucionais pré-treinadas com pesos carregados da base de imagens *imagenet* (DENG et al., 2009), através do processo de transferência de aprendizado (TA). Nesse processo, cada subimagem é submetida a rede, sendo processada pelas diversas camadas de subamostragem até uma representação final na camada de achatamento (*flatten*), na qual deve originar um vetor de características derivado de uma arquitetura de RNC.

Uma vez que tal descrição ocorreu, no passo 3, tais subimagens podem ser vistas como pontos em um espaço n -dimensional, sendo que o conjunto de objetos pertencentes a uma mesma imagem, se tornaram grafos totalmente conectados. Cada subimagem deverá estar ligada a vizinhança representada em uma mesma imagem. Neste ponto, cada vértice representa um conjunto de informações acerca da sua subimagem e sua própria descrição através de um vetor de características. Além disso, cada aresta representa a sua interconexão em uma mesma imagem, originando,

assim, um grafo totalmente conectado. A característica desse tipo de grafo é marcada por sua topologia, em que todas os vértices possuem relacionamentos entre os seus vizinhos, assim como um auto relacionamento, conforme ilustrado na Figura 10.

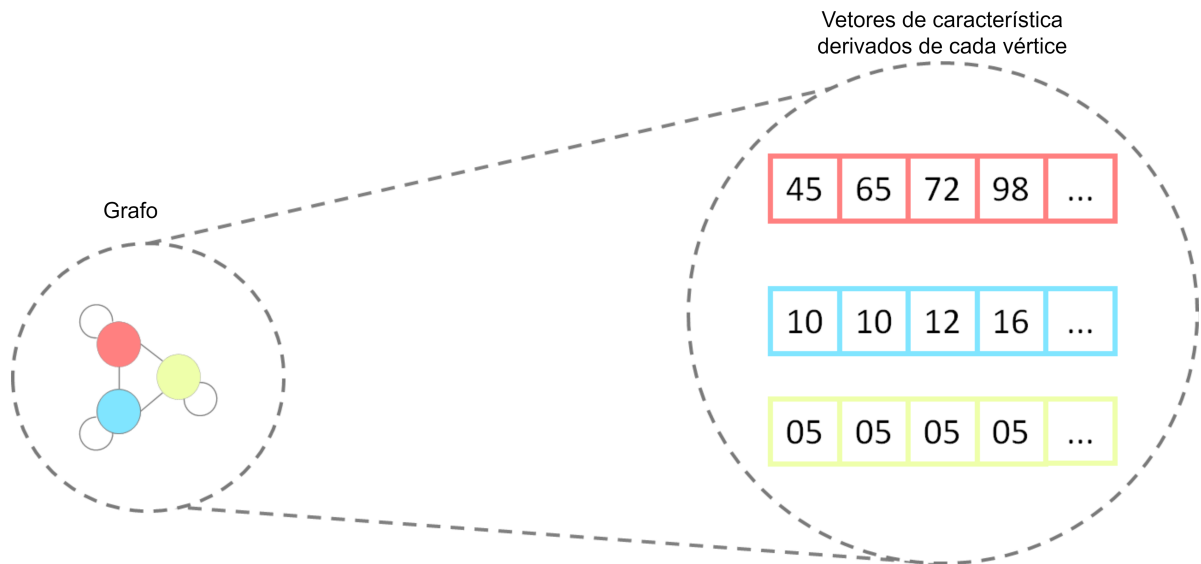


Figura 10: Ilustração da relação entre grafo e vetores de característica

No passo 4, por sua vez, uma matriz de adjacência é gerada para a representação de tais grafos. Essa estrutura gerada, nesse ponto, representa um conjunto de grafos totalmente conectados (ou subgrafos), que serão submetidos ao Algoritmo de poda proposto neste trabalho.

O passo 5 é caracterizado pelo Algoritmo proposto neste trabalho, que será descrito na Seção 3.1. Uma vez que a poda é aplicada, uma abstração de grafos é gerada, alterando sua topologia. A Figura 11 exemplifica o processo decorrido.

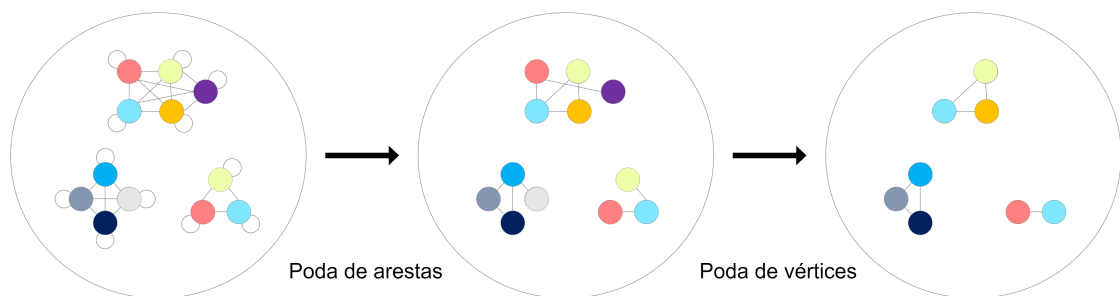


Figura 11: Ilustração de grafos totalmente conectados para grafos podados

No passo 6, a matriz de adjacência pós-poda é gerada a partir do resultado obtido na poda. Então, no passo 7, os grafos são inseridos como entrada para o processo de treinamento de redes neurais convolucionais de grafos. Do passo 1 ao passo 7, todo o fluxo percorrido ocorreu de maneira *offline*, isto é, ele é executado apenas antes da etapa de treinamento.

No passo 8, o treinamento utiliza do grafo podado para estabelecer os processos característicos da arquitetura de rede, sendo um dos grandes responsáveis pelo consumo de recurso computacional ao longo da jornada. Importante ressaltar que neste trabalho o auto-laço não é considerado no treinamento, ainda que seja considerado para política de poda. Por fim, no passo 9, o modelo do treinamento é gerado, pronto para ser utilizado para realizar as predições.

3.1 ALGORITMO DE PODA

O Algoritmo de poda proposto tem como base as informações contextuais presentes em imagens de bases de dados pré rotuladas.

Conforme ilustrado nas Figuras 10 e 11, para os retângulos envolventes (*bounding boxes*) derivados de uma dada imagem i são extraídos vetores de característica por meio de um processo de transferência de aprendizado. Tais vetores são carregados para cada uma das arquiteturas de RNC propostas, descritas na Seção 3.2, com os pesos da base Imagenet (DENG et al., 2009).

Cada retângulo envolvente b_i de uma imagem Im_i é representado posteriormente por vetor de características extraídos através de uma RNC, conforme citado na seção anterior. Esse vetor representa um vértice v_i do grafo G_i . As conexões entre os vetores são estabelecidas por meio de arestas a_i . Cada conjunto de vértices v_i possui um relacionamento entre seus vizinhos, construindo, assim, um grafo totalmente conectado, como apresentado na literatura em Kipf e Welling (2019).

A partir desses relacionamentos, o processo de poda é iniciado pelo mapeamento de similaridade entre os vértices v_i e v_j , tendo como métrica de similaridade principal um valor $sim(v_i, v_j)$ para cada par através de uma função de distância Euclideana de dois ou mais pontos, conforme a Equação 11 (LIBERTI et al.,

2012).

$$dt_{ij} = \|x_i - x_j\|_2 \quad (11)$$

em que dt_{ij} equivale ao valor de distância entre pontos x_i e x_j .

Esse mapeamento de similaridade pela vizinhança é o componente de análise contextual presente no Algoritmo de poda proposto nesta pesquisa. Através dele, é possível identificar objetos que possuem grande semelhança contextual, como um conjunto de lâmpadas idênticas que caracterizam um ambiente fechado de classe “quarto” ou diversos livros que podem indicar que a classe principal é “livraria”. Quanto menor a distância Euclideana entre diferentes subimagens, mais parecidas elas se tornam. A ideia principal é que itens similares não sejam considerados no processo de treinamento das RNCG, pois assim, seriam consideradas amostras redundantes.

A partir da construção do grafo e do cálculo de similaridade para cada vértice, o processo de poda é dividido em duas etapas referentes à poda de arestas e à poda de vértices. As Seções 3.1.1 e 3.1.2 detalham sobre os etapas. O Algoritmo 1 descreve o fluxo geral da abordagem proposta.

Algoritmo 1: Algoritmo de treinamento da abordagem proposta

Entrada : conjunto de imagens Im , uma RNC C

Saída : modelo M

Auxiliares: conjuntos de vetores de características extraídos Vec_i , conjunto de treino Vec' , conjunto de teste Vec'' , um grafo não direcionado totalmente conectado G^i , um grafo podado G^Δ

para cada $Im_j, j = 1, 2, \dots, |Im|$ **faça**

$B \leftarrow \text{retangulosEnvolventes}(Im_k)$;

para cada $B_i, i = 1, 2, \dots, |B|$ **faça**

$Vec \leftarrow Vec \cup \text{vetorCaracteristicas}(B_i, C)$;

Dividir Vec ;

$Vec' \leftarrow$ amostras de treinamento aleatórias de Z ;

$Vec'' \leftarrow$ amostras de teste aleatórias de $Vec \setminus Vec'$;

$G^i \leftarrow \text{grafoTotalmenteConectado}(Vec', Vec'')$;

$G^\Delta \leftarrow \text{grafoPodado}(G_i, \text{grafo})$;

Na implementação do algoritmo proposto, há a representação de uma complexidade $O(N^2)$, devido ao número de iterações presentes, conforme em

Calamoneri et al. (2006) Importante ressaltar que se trata de um processo *offline*, isto é, é executado fora do processo de treinamento e também apenas uma vez.

3.1.1 PODA DE ARESTAS

Em um grafo, a poda de arestas está diretamente relacionada à remoção de conexões. Isso impacta diretamente no comportamento da RNG, uma vez que a cada processo de agregação, durante o momento de execução do treinamento de uma RNG, as informações dos vizinhos são relevantes para performance do algoritmo. Desta forma, para explorar o potencial de redução de arestas, essa etapa é o primeiro passo do Algoritmo.

Dado um grafo não direcionado G_i e um conjunto de vértices V_i , sendo $V_i \in G_i$, o valor de τ é definido a partir da Equação 12:

$$\tau = \frac{\sum_{k=0}^{a-1} sim(v'_k, v''_k)}{a} \quad (12)$$

Sendo k o iterador; a o número total de arestas de um grafo G_i , $dist$ a função responsável por calcular a distância entre vértices v'_k, v''_k . O processo também é ilustrado na Figura 12.

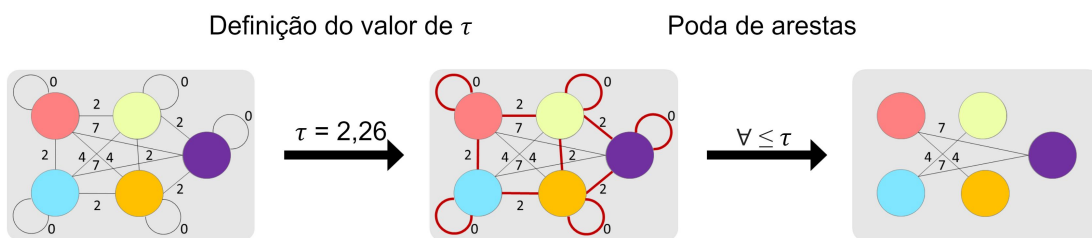


Figura 12: Ilustração de grafos totalmente conectados para grafos com arestas podadas

Arestas que possuem valores iguais ou abaixo do limiar τ são selecionadas para o processo de poda, permitindo que apenas arestas de maior peso permaneçam no grafo. Desta forma, são eliminados objetos muito similares ou até mesmo idênticos, garantindo que o grafo terá maior diversidade de elementos. O Algoritmo 2 ilustra tal

procedimento.

Algoritmo 2: Algoritmo de poda de arestas

Entrada: um grafo não direcionado G^i

Saída: um grafo podado não direcionado G^Ω

Auxiliares: V : conjunto de vértices do grafo de entrada, A : lista de arestas de cada vértice v , τ : limiar de poda para arestas irrelevantes

para cada $v \in V$ **faça**

$A \leftarrow \text{selecionaArestasIncidentes}(v)$

$\tau \leftarrow \text{calculaLimiarArestas}(A)$

para cada $a \in A$ **faça**

se $\text{relevanciaAresta}(a) \leq \tau$ **então**

$\text{removeAresta}(a)$

3.1.2 PODA DE VÉRTICES

Essa modalidade de poda está relacionada com a remoção de vértices da topologia de um grafo. Para a RNG, as informações utilizadas pela rede para o processo de treinamento são representadas pelos vértices, o que pode ser um fator potencializador para redução do consumo de recursos.

Essa etapa utiliza uma medida de centralidade de grafos rotulada como Centralidade de Autovetor, que tem como característica identificar a relevância de um vértice dados os seus vizinhos. O Algoritmo 3 ilustra o fluxo da poda de vértices utilizada. A medida de Centralidade de Autovetor pode ser representada através da definição encontrada em Freitas (2010), conforme Equação 13. Nesta equação, também há o índice p , representado por $p(G) = \text{ind}(G)$, também exposto por Freitas (2010), que explica que grafos conexos possuem suas matrizes de adjacência não divisíveis.

$$c_{eig}(v_k) = \frac{1}{p} \sum_{j=1}^n Ad_{kj} \text{autovec}_j \quad (13)$$

em que v_k representa um vértice de um grafo G , p o índice de G , Ad uma matriz de adjacência, autovec um autovetor não negativo associado a G , j é o iterador e k é uma coordenada do vetor.

O uso desse algoritmo só foi possível dada à Poda de Arestas, a qual é

responsável por formar uma topologia diferente do modelo totalmente conectado. Sem isso, todas as arestas teriam a mesma medida de centralidade.

$$\phi = \frac{\sum_{k=0}^{v-1} \text{AutoValores}(v_k)}{V} \quad (14)$$

Sendo k o iterador; V o número total de vértices de um grafo G_i , e a função de auto valores, descrita na Equação 13, responsável por calcular o auto valor de cada vértice v_k . O processo também é ilustrado na Figura 13.

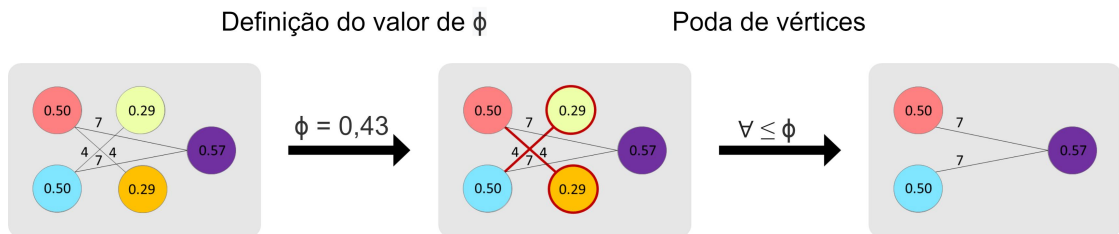


Figura 13: Ilustração de grafos com arestas podadas para grafos com vértices podados

Algoritmo 3: Algoritmo de poda de vértices

Entrada: um grafo com arestas podadas não direcionado G^Δ

Saída: um grafo podado não direcionado G^Ω

auxiliares: V : conjunto de vértices do grafo de entrada, ϕ : limiar de poda para vértices irrelevantes

$\phi \leftarrow \text{calculaLimiarVertices}(V)$

para cada $v \in V$ **faça**

se $\text{autoVetor}(v) \leq \phi$ **então**

$\text{removeVertice}(v)$

3.2 EXPERIMENTOS

Para validar o Algoritmo proposto neste trabalho, foram definidos diversos critérios de execução, como um conjunto de hiper parâmetros, estratégia de validação estatística e configuração de máquina. Todos esses pontos, juntos, somam o total de 8640 experimentos executados em 3 bases de imagens de diferentes dimensões.

Nas Subseções 3.2.1 e 3.2.1.2 são aprofundadas as combinações de hiper parâmetros e estratégia de validação estatística.

Na Subseção 3.2.1.3 é detalhada a configuração de máquina em que os experimentos foram executados. As Subseções 3.2.1.4 e 3.2.2, descrevem os artefatos utilizados, como bibliotecas e bases de imagens. Na Subseção 3.2.3 é especificado como foi o processo de escalabilidade para execução do número de experimentos para a pesquisa. Por fim, na Subseção 3.2.4 são detalhadas as métricas coletadas para o trabalho.

3.2.1 CENÁRIOS EXPERIMENTAIS

A execução dos experimentos foi um processo bastante longo ao decorrer da pesquisa. Toda a jornada envolveu técnicas de Engenharia de Machine Learning (BURKOV, 2020), que visam gerar experiências escaláveis de Aprendizado de Máquina. Com isso, os cenários experimentais se dividiram entre Seleção e Particionamento dos Dados e busca por melhores hiper parâmetros através da Busca em Grade.

3.2.1.1 SELEÇÃO E PARTICIONAMENTO DOS DADOS

Para particionar a base de dados, foram executadas algumas alterações em camadas para cada uma das técnicas de poda. Para ambas as podas foram removidas as imagens cujas classes não possuísem pelo menos 10 imagens para cada uma das bases. Desta forma, é possível garantir que conjuntos de treinamento e de teste possuam pelo menos uma imagem para a validação cruzada com 5 subdivisões.

Na Poda por vértice, foi percebido que algumas classes poderiam deixar de existir a medida que seus vértices e, conseqüentemente seus rótulos, passavam ser podados. Desta forma, não foi possível otimizar um balanceamento de classes.

A Tabela 1 apresenta a relação entre quantidade de imagens e poda para cada uma das bases de dados, assim como as Figuras 14 e 15 apresentam o balanceamento de imagens.

Tabela 1: Número de classes de imagens para arquiteturas sem poda e com poda para cada uma das base de dados

Base de Dados	Topologias de Grafo					
	Totalmente conectado		Poda de Arestas		Poda de Vértices	
	N. Classes	N. Imagens	N. Classes	N. Imagens	N. Classes	N. Imagens
MIT-67	53	58068	53	58068	53	23656
Unrel	33	1809	33	1809	22	207
VRD	23	87696	23	87696	23	42919

3.2.1.2 BUSCA EM GRADE

A técnica de Busca em Grade é caracterizada pela procura da composição de um número de combinações de uma ordem de parametrizações (ANDONIE, 2019). Com isso, são avaliadas todas as possibilidades do grupo, sendo essa a estratégia utilizada na pesquisa, uma vez que fora fechado o espaço amostral de hiperparametrizações. A Tabela 2 apresenta o número de variáveis e combinações para execução dos experimentos.

Hiper parâmetro	Variáveis	Número de parâmetros
Arquiteturas	[EfficientNetB7, InceptionV3, ResNet50, VGG19]	4
Auto Laço	Falso	1
Bases de Imagens	[MIT-67, Unrel, VRD]	3
Número de Camadas	2	1
Número de Épocas	2000	1
Número de Neurônios	[16, 64, 256]	3
Taxa de Aprendizado	[0.001, 0.005, 0.01, 0.05]	4
Taxa de Decadência de Peso	0	1
Taxa de Desligamento	[0.3, 0.5, 0.8, 0.9]	4
Topologias de Grafo	[Totalmente Conectado, Poda de Arestas, Poda de Vértices]	3

Tabela 2: Total de combinações por parametrizações dos experimentos.

O total de combinações possíveis no espaço amostral dos experimentos pode ser descrito pela multiplicação do número de parâmetros da Tabela 2, sendo o valor de 1728 experimentos. Além das combinações já citadas, esse valor foi multiplicado por 5, equivalente ao número de validações cruzadas, para encontrar relevância estatística e remover quaisquer vieses com relação aos dados, realizadas em cada uma das 3 bases de dados. Portanto, o número de experimentos finais é 8640 (1728x5).

3.2.1.3 ESPECIFICAÇÕES TÉCNICAS DA MÁQUINA

A máquina selecionada para execução dos experimentos é um dos pontos mais importantes para garantir a falta de quaisquer vieses técnicos e assegurar que o processo seja totalmente equiparável entre a literatura e o método proposto. Com isso, a Tabela 3 apresenta a configuração de máquina para execução do total de experimentos.

Recurso	Especificação Técnica
Ambiente	Nuvem do Google Cloud Platform
Memória (RAM)	32 GB
Memória em Disco	500 GB
Memória Gráfica (GRAM)	16GB
Placa de Vídeo	Tesla T4, 4000 núcleos CUDA
Processador	Intel Xeon E3000, 6 núcleos e 12 Threads a 3.0 Ghz
Sistema Operacional	Ubuntu 18.4 LTS

Tabela 3: Recursos de máquina utilizados

Essa máquina foi hospedada na plataforma em nuvem do Google (Google Cloud Platform), sendo este um ambiente totalmente auto gerenciável e escalável para alocação de recursos de máquina.

3.2.1.4 RECURSOS E BIBLIOTECAS

Para execução dos experimentos, foram selecionados diversos recursos ao longo da pesquisa para etapas de carregamento, pré-processamento, treinamento e métricas. A etapa de carregamento, pode ser compreendida como a etapa de leitura das bases de dados; pré-processamento representa a fase de tratamento dos dados, assim como preparação dos arquivos de treinamento; treinamento, por sua vez, trata-se do processo de execução da rede neural até sua consolidação; por fim, coleta de métricas, envolve o trabalho de visualização de dados e resultados.

A linguagem de programação Python, em sua versão 3.7.7 foi o recurso principal para o desenvolvimento. Como característico da arquitetura da linguagem, foram instalados os pacotes da Tabela 4.

Pacote	Descrição	Versão	Etapa
dgl-cu102	Pacote Deep Graph Library	0.4.3	Treinamento
EfficientNet	Modelo da EfficientNet	1.1.0	Pré-processamento
matplotlib	Pacote de plotagem	3.1.3	Coléta de métricas
networkx	Pacote de manipulação de estrutura de dados em grafos	2.5.0	Pré-processamento
numpy	Pacote de cálculos e computação	1.19.2	Pré-processamento
pandas	Pacote de manipulação de estrutura de dados em tabela	1.0.3	Todas as etapas
pickleshare	Pacote para manipulação de arquivos Pickle	0.7.5	Pré-processamento
psutil	Pacote para monitoramento de procesoss e sistemas	5.7.0	Treinamento
scipy	Pacote de matemática	1.6.1	Pré-processamento
py3nvm1	Pacote para monitoramento dos recursos da NVIDIA	0.2.5	Treinamento
torch	Pacote Pytorch, para Redes Neurais e uso de GPU	1.7.1	Treinamento
scikit-learn	Pacote de manipulação de dados	0.22.1	Pré-processamento
tensorflow	Pacote Tensorflow, para Redes Neurais e uso de GPU	2.1.0	Pré-processamento
tqdm	Pacote de mensuração de progresso	4.46.0	Pré-processamento

Tabela 4: Bibliotecas e pacotes utilizados

3.2.2 DESCRIÇÃO DAS BASES DE IMAGENS

O trabalho foi um projeto do campo de Visão Computacional, portanto, foram utilizados dados de imagens para sua execução. Foram exploradas bases de imagens com características diferentes, tanto em aspectos visuais quanto quantitativos.

As bases MIT-67, Unrel e VRD são bases da literatura que oferecem não só as imagens, mas também a localização das subimagens através de um mapeamento geométrico dos retângulos envolventes, oferecendo a possibilidade de utilizá-los no treinamento.

Todas as bases passaram por um processo de limpeza, cuja política envolve a remoção de elementos cujas classes não possuam pelo menos 10 imagens presentes. As Figuras 14 e 15 apresentam a distribuição de imagens e subimagens (retângulos envolventes), respectivamente. A média de retângulos envolventes por imagem é de 19 para a MIT-67, 3 para Unrel e 19 para VRD.

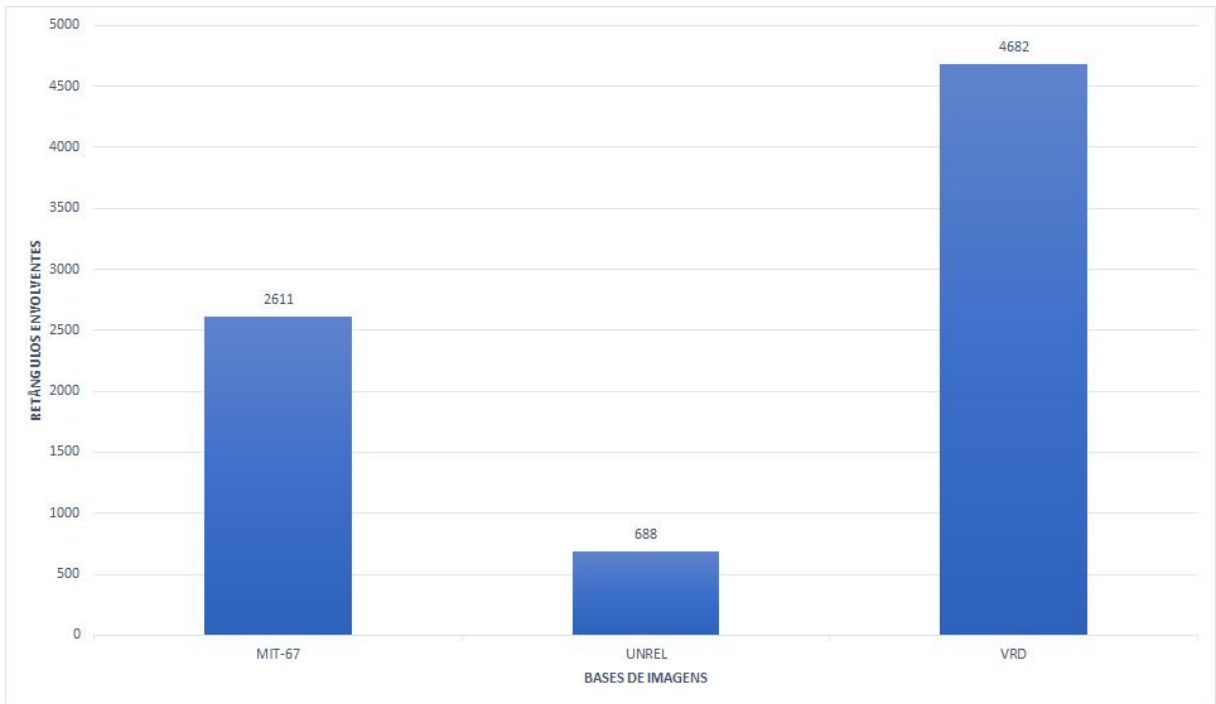


Figura 14: Distribuição de imagens por base de imagens

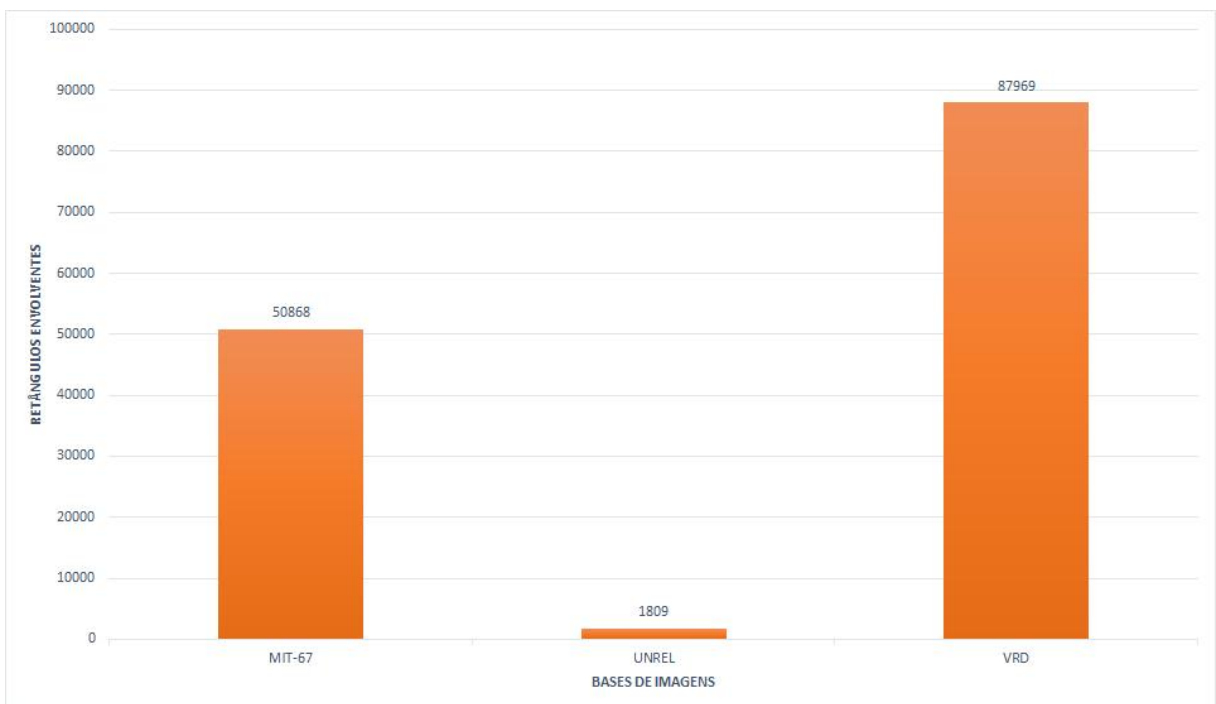


Figura 15: Distribuição de retângulos envolventes por base de imagens

3.2.2.1 MIT-67

A base de imagens MIT-67 contém 67 classes e 15620 imagens, sendo uma base coletada por meio de mecanismos de busca na internet. A base apresenta imagens de cenas internas de diversos ambientes, como aeroportos, cozinhas e livrarias.

Essa base foi utilizada como parte do estudo para classificação de cenas de locais internos, considerado como um grande desafio de Visão Computacional e Aprendizado de Máquina, devido às suas características.

A resolução mínima das imagens presentes na base são a partir de 200 pixels e seu formato está em JPG (QUATTONI; TORRALBA, 2009). Na pesquisa, foram utilizadas apenas 53 classes e 2611 imagens, conforme representadas nas Figuras 19 e 20.

São geradas as subimagens por meio das anotações presentes, representando o total de 50868 imagens. As Figuras de 16 a 18 representam amostras da base de dados.



Figura 16: Imagem extraída da base de imagens MIT-67 da classe airport (aeroporto)



Figura 17: Imagem extraída da base de imagens MIT-67 da classe library (livraria)



Figura 18: Imagem extraída da base de imagens MIT-67 da classe restaurant (restaurante)

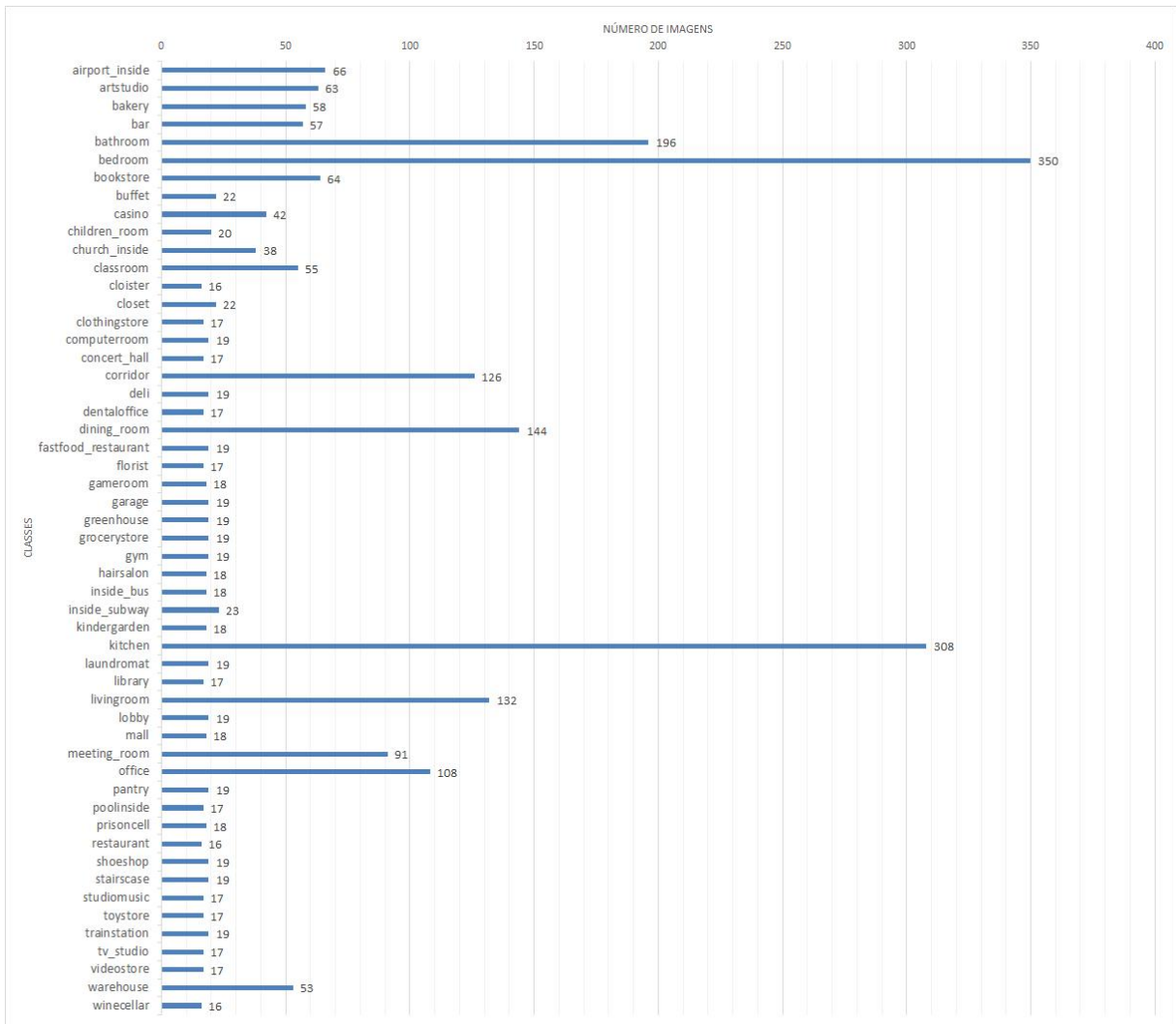


Figura 19: Distribuição de imagens por classe para a base de imagens MIT-67

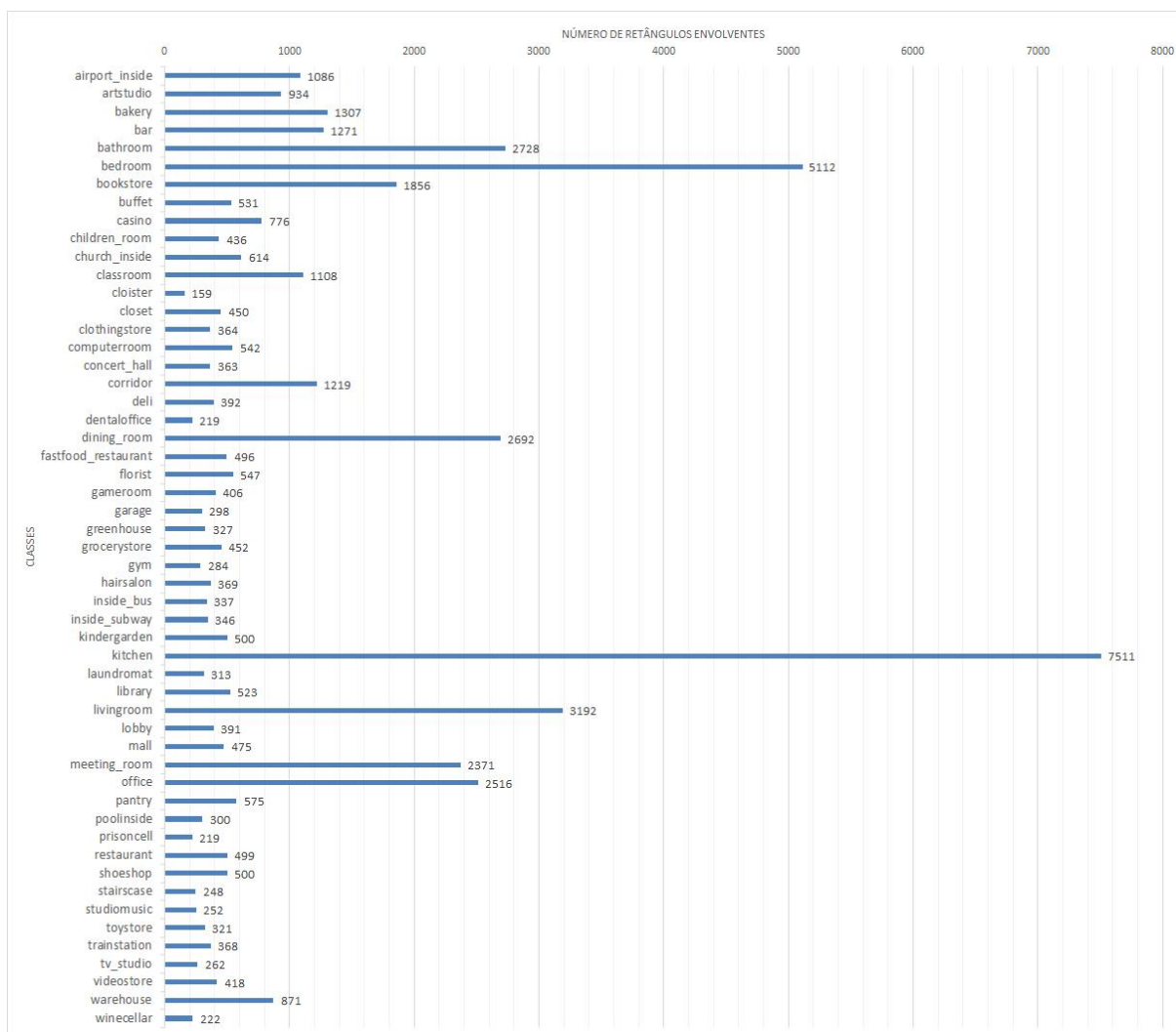


Figura 20: Distribuição de retângulos envolventes por classe para a base de imagens MIT-67

3.2.2.2 UNREL

A base Unrel possui 76 classes e 1196 imagens. Sua característica principal é relacionar objetos cujo contexto pode parecer atípico para tarefas de classificação comuns, como um gato utilizando gravata, por exemplo. Sua resolução é variada e o formato carregado foi o JPG (PEYRE et al., 2017). Para o trabalho, foram utilizadas apenas 57 classes e 688 imagens, possuindo 1809 retângulos envolventes, conforme a distribuição nas Figuras 24 e 25.

As Figuras de 21 a 23, também representam amostras da base de dados.



Figura 21: Imagem extraída da base de imagens Unrel da classe cat on the top of dog (gato sob cachorro)



Figura 22: Imagem extraída da base de imagens Unrel da classe dog ride bike (cachorro pilota a bicicleta)



Figura 23: Imagem extraída da base de imagens Unrel da classe person under table (pessoa embaixo da mesa)

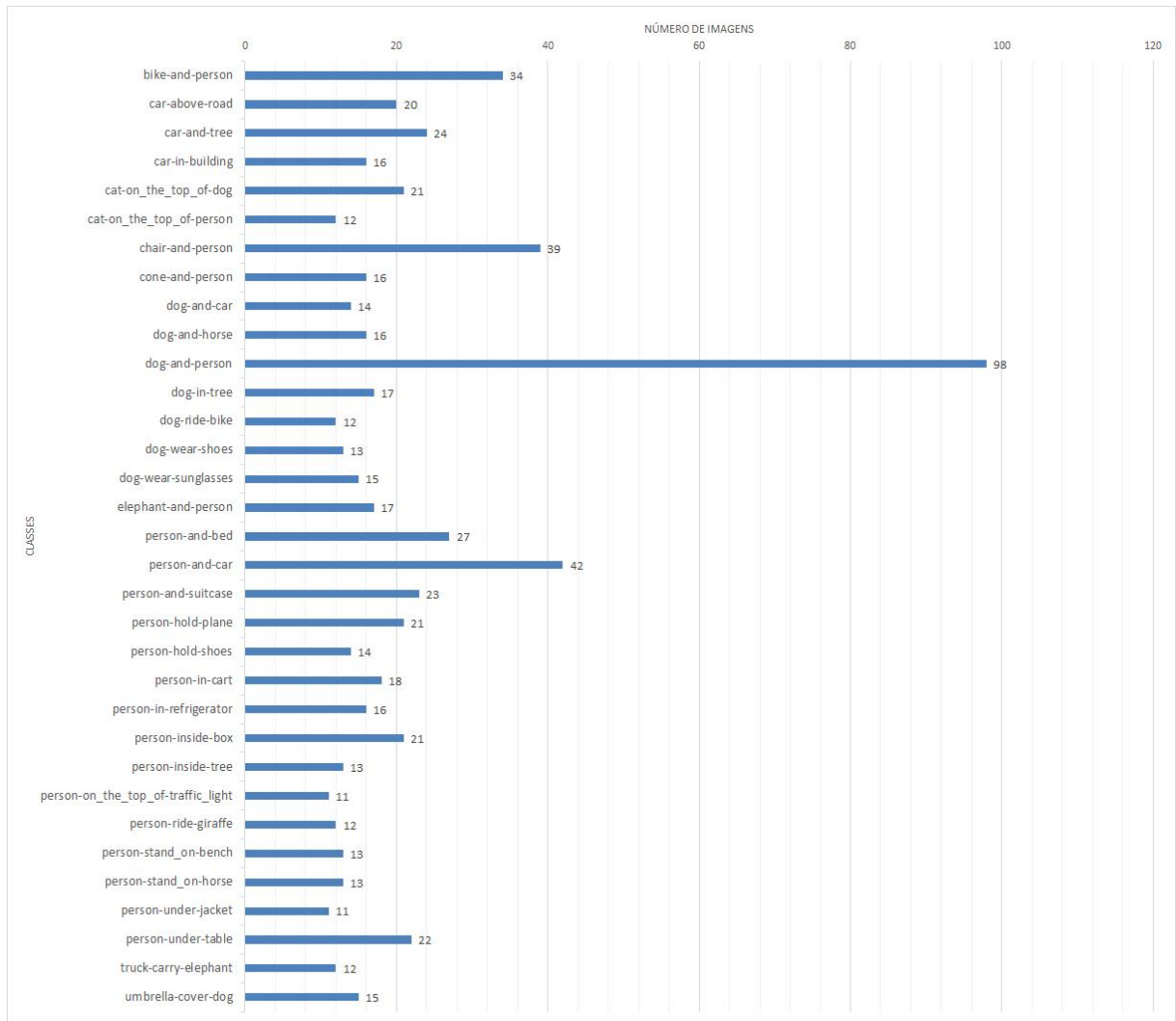


Figura 24: Distribuição de imagens por classe para a base de imagens Unrel

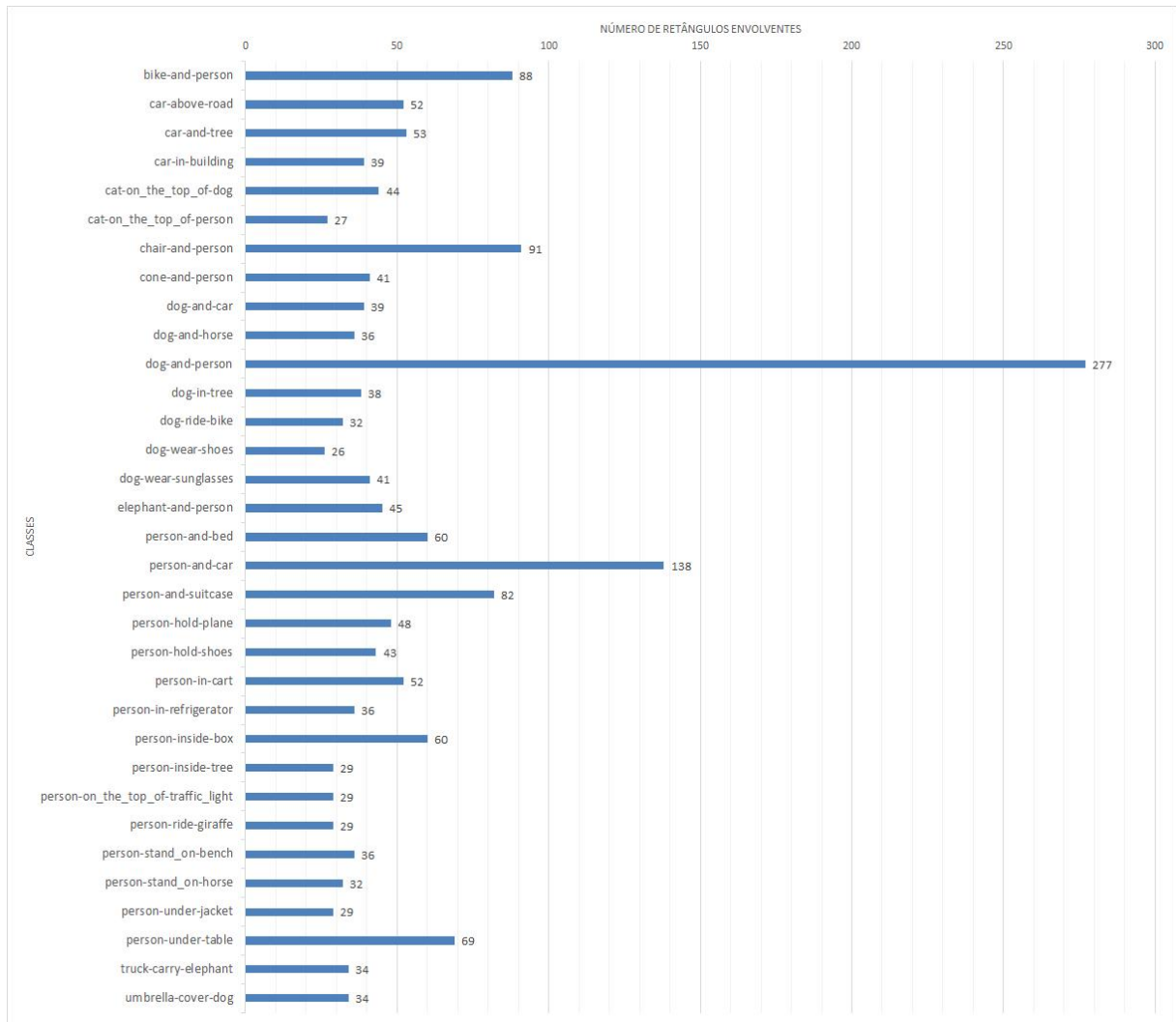


Figura 25: Distribuição de retângulos envolventes por classe para a base de imagens Unrel

3.2.2.3 VISUAL RELATIONSHIP DETECTION

A base de imagens Visual Relationship Detection (VRD), também considerada a maior base para os experimentos. Contém 100 objetos e 70 predicados, dos quais são utilizados nesta pesquisa como classes e 5000 imagens.

São relacionamentos cujos contextos envolvem um objeto seguido de um predicado e outro objeto, conforme explicitado pelos autores. A base explora cenários em que, os relacionamentos de interação com esses objetos, possam ser interessantes para a classificação, como "pessoa empurrando uma bicicleta", por exemplo. Os tamanhos das imagens são variados e a base foi carregada no formato JPG (LU et al., 2016). Apenas 23 classes e 4682 imagens foram utilizadas na pesquisa, conforme as Figuras 29 e 30.

Por fim, as Figuras de 26 a 28, também representam amostras da base de dados.



Figura 26: Imagem extraída da base de imagens VRD da classe has (possui)



Figura 27: Imagem extraída da base de imagens VRD da classe has (possui)



Figura 28: Imagem extraída da base de imagens VRD da classe wearing (vestindo)

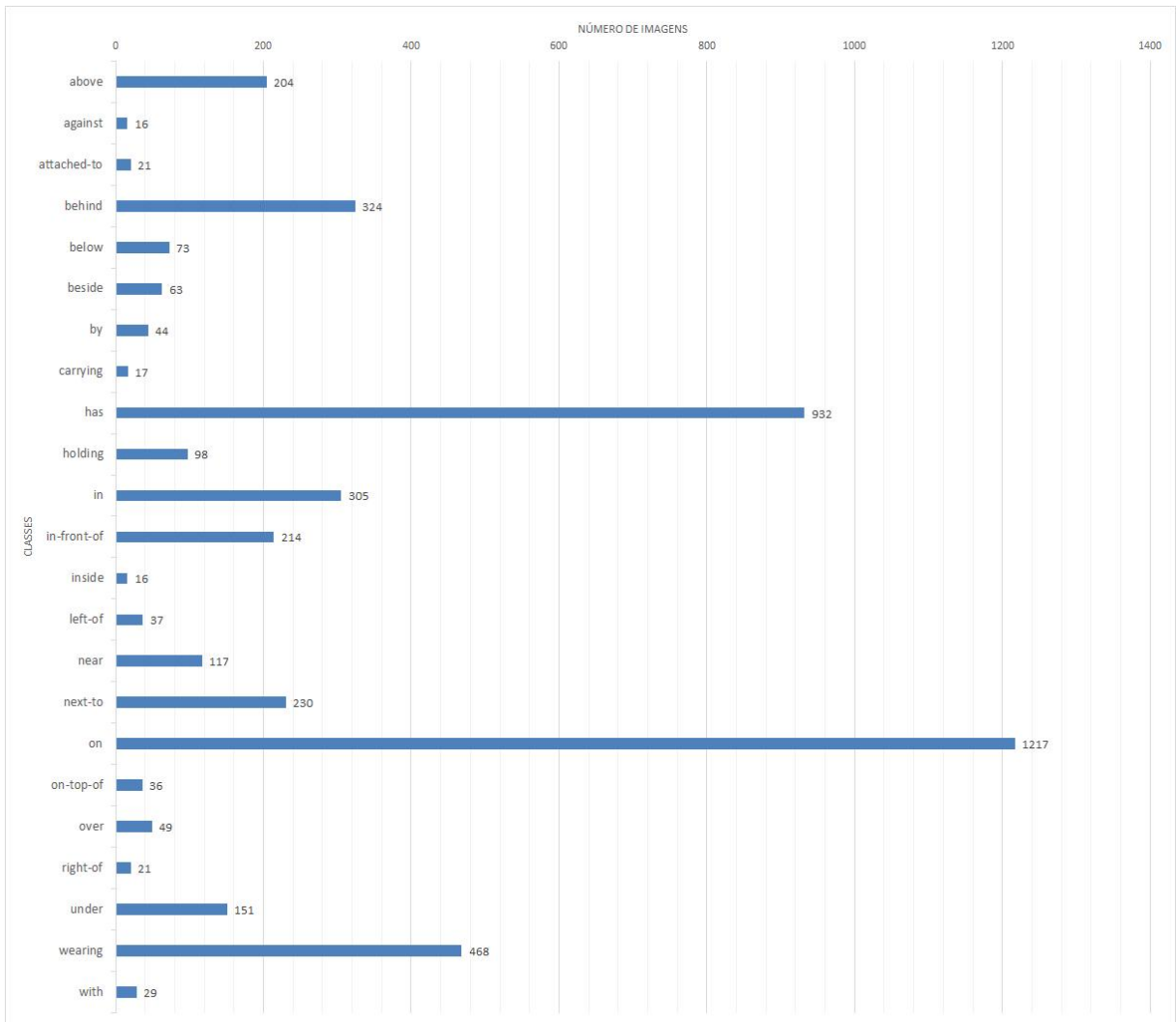


Figura 29: Distribuição de imagens por classe para a base de imagens VRD

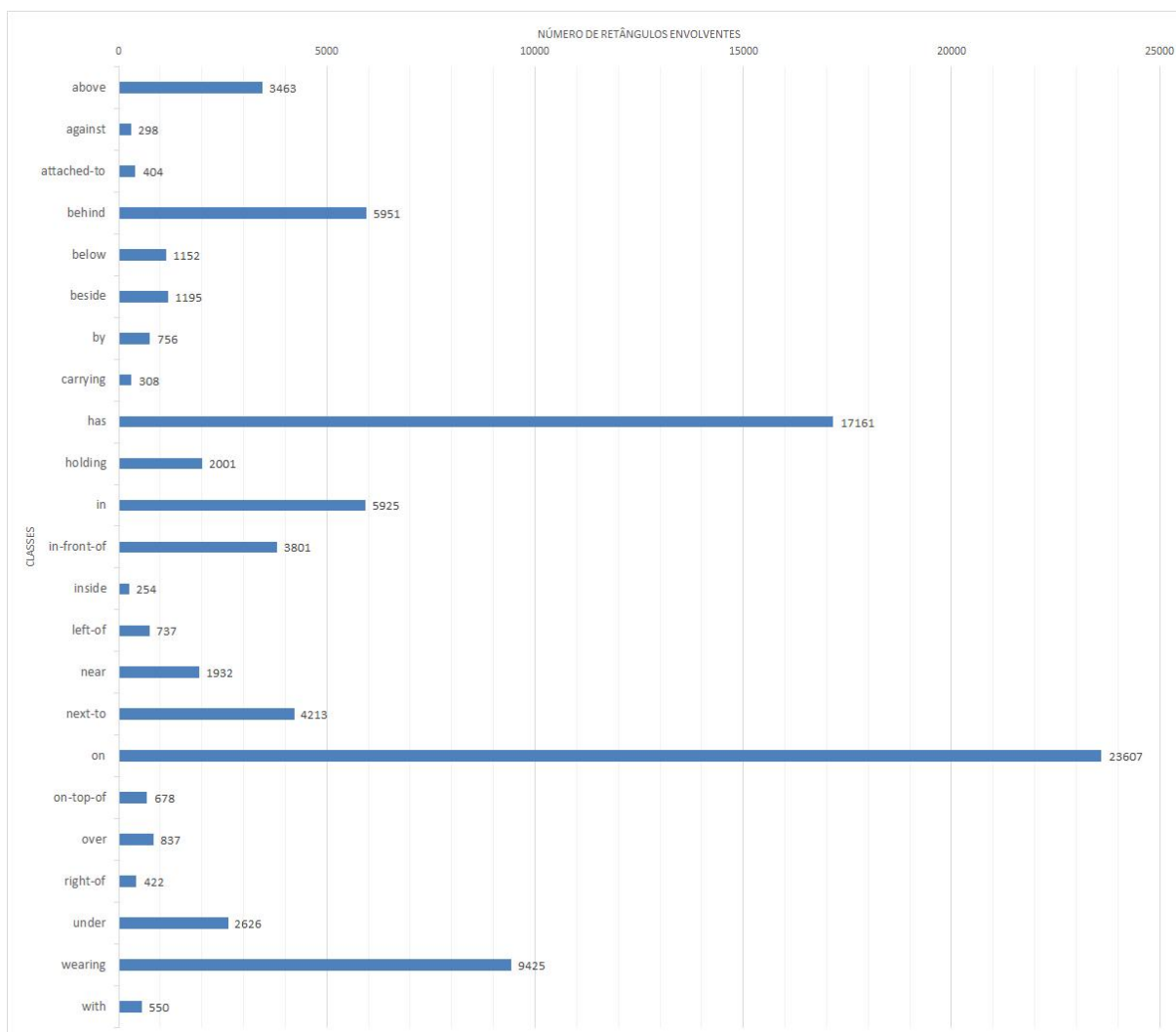


Figura 30: Distribuição de retângulos envolventes por classe para a base de imagens VRD

3.2.3 TREINAMENTO DOS MODELOS EM ESCALA

Os experimentos realizados na pesquisa exigiram uma estrutura de escalabilidade capaz de suportar o treinamento contínuo dos modelos ao longo de uma série de parametrizações. Para a viabilização da escala, foi desenvolvido um mecanismo de geração de hiperparametrizações, com objetivo de criar uma planilha com as combinações da Busca em Grade. Cada linha representando uma combinação e, conseqüentemente, um experimento. Por fim, essa planilha seguiu a estrutura de dados em fila e passou por um processo de escalabilidade de forma sequencial. O processo mencionado consiste em utilizar máquinas apresentando um mesmo padrão, para assegurar que os experimentos sejam executados e comparados de maneira

equivalente.

3.2.3.1 COMPUTAÇÃO EM NUVEM

Os experimentos foram executados utilizando uma infraestrutura virtual, rotulada como Computação em Nuvem, que é caracterizada por oferecer recursos de máquina remotamente.(KHAN; ALI, 2015)(GOOGLE. . . , 2012). Toda essa estrutura capacitou a alocação de recursos computacionais de forma modular. Outro ponto acerca do tema, é que todo o processamento ocorreu em ambiente totalmente controlado, isto é, sem que agentes externos executassem outros programas na mesma máquina alocada

3.2.4 COLETA DE MÉTRICAS

O uso de métricas na pesquisa foi dividido em três vertentes principais: assertividade, performance e topologia. As métricas de assertividade foram focadas em analisar acurácia, precisão, revocação e F1-score ao longo da execução dos experimentos. Em performance, métricas como tempo de execução e consumo da placa gráfica foram os pilares para avaliação do processo de poda. Por fim, as métricas acerca da topologia, que visaram mensurar a redução de arestas e vértices durante o processo de poda.

A Tabela 5 apresenta as principais definições sobre as métricas coletadas neste trabalho, assim como sua interpretação, conforme em Dalianis (2018). As Equações 15-19 representam o cálculo das métricas utilizadas.

$$ACURACIA = \frac{VP + VN}{VP + VN + FP + FN} \quad (15)$$

em que VP representa o valor de verdadeiro positivo, VN o valor de verdadeiro negativo, FP o valor de falso positivo e FN o valor de falso negativo.

$$PRECISAO = \frac{VP}{VP + FP} \quad (16)$$

em que VP representa o valor de verdadeiro positivo e FP o valor de falso positivo.

$$REVOCAÇÃO = \frac{VP}{VP + FN} \quad (17)$$

em que VP representa o valor de verdadeiro positivo e FN o valor de falso negativo.

$$F1SCORE = 2 * \frac{P * R}{P + R} \quad (18)$$

em que P representa o valor de precisão e R representa o valor de Revocação.

$$ARESTAS = ((v(v - 1))/2) + v \quad (19)$$

em que v representa o número de vértices de um grafo G , sendo G um grafo bidirecional.

	Métrica	Descrição	Equação
Assertividade	Acurácia	Proporção de acertos dentre todas as classificações	(15)
	F1-Score	Média harmônica entre Precisão e Revocação	(18)
	Precisão	Acertos entre verdadeiros positivos e falsos positivos	(16)
	Revocação	Proporção de acertos entre verdadeiros positivos e falsos negativos	(17)
Performance	Memória GRAM	Consumo em GB de memória RAM da placa gráfica	-
	Tempo de treinamento	Tempo em milissegundos do treinamento do modelo	-
Topologia de Grafo	Arestas	Número de conexões entre vértices em um grafo	(19)
	Vértices	Número de vértices em um grafo	-

Tabela 5: Métricas utilizadas nos experimentos

3.3 RESULTADOS OBTIDOS

Os resultados obtidos foram agrupados por cada uma das bases de dados conforme os experimentos descritos na Seção 3.2. Cada um dos agrupamentos conta com um gráfico de radar, contendo as principais métricas de assertividade, performance e topologia de grafo, assim como um gráfico de barras com linhas, sendo as barras as métricas de assertividade e as linhas as métricas de performance.

Os gráficos de radar podem ser interpretados com uma escala de 0 a 100, em percentual, sendo a maior área da plotagem considerada a de melhor resultado. Os gráficos de barra com linhas possuem escalas diversas para 3 dimensões diferentes, sendo o eixo de assertividade de 0 a 100 para as barras e 0 a 4000 para o uso de memória GRAM, representados pelas linhas.

3.3.1 MIT-67

Através dos experimentos executados neste trabalho, foi identificado que a poda de arestas apresentou um desempenho capaz de reduzir até 9,70% do consumo de memória GRAM, enquanto a poda de vértices reduziu em 40,57% comparado ao grafo totalmente conectado. Também foi possível observar que houve uma redução no tempo de treinamento, sendo em até 7,76% para poda de arestas e 57,68% para vértices. O decaimento de acurácia foi de 3,90% para redução de arestas e 12,60% para vértices.

As Figuras 31 a 34 apresentam as diferentes arquiteturas RNCs citadas nos cenários experimentais. De maneira geral, o mesmo comportamento pode ser observado para as diferentes arquiteturas, nas quais a poda por arestas gera um melhor custo benefício, ao passo que a poda por vértice gera um decaimento da acurácia.

A Figura 35 ilustra uma visualização geral em gráfico de barras e valores com respectivos desvios para todas as métricas consideradas, bem como todas as arquiteturas RNCs e políticas de poda. A partir das mesmas, é possível observar que a poda de arestas gera ganho computacional, bem como as diversas métricas obtidas (acurácia, F1-score, etc) apresentam empate estatístico, quando comparadas entre grafo totalmente conectado e o grafo com poda por arestas.

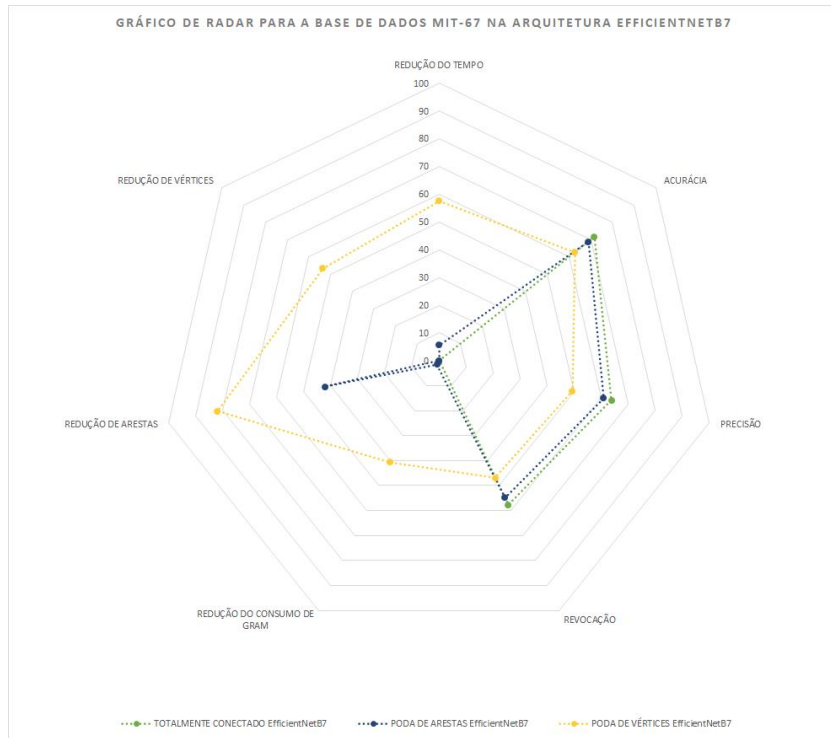


Figura 31: Gráfico de radar dos resultados dos experimentos na arquitetura EfficientNetB7 para a base de dados MIT-67

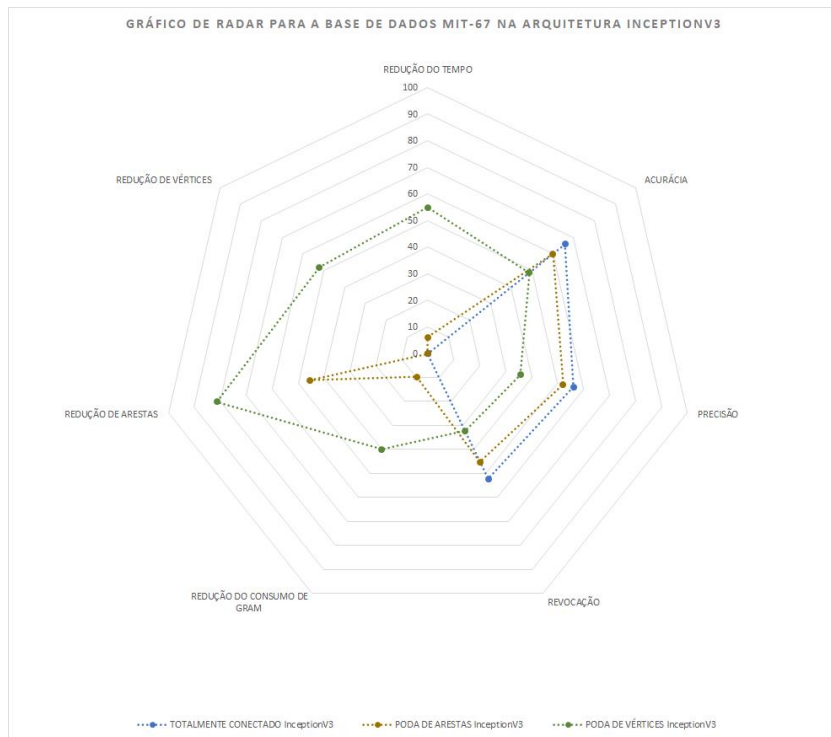


Figura 32: Gráfico de radar dos resultados dos experimentos na arquitetura InceptionV3 para a base de dados MIT-67

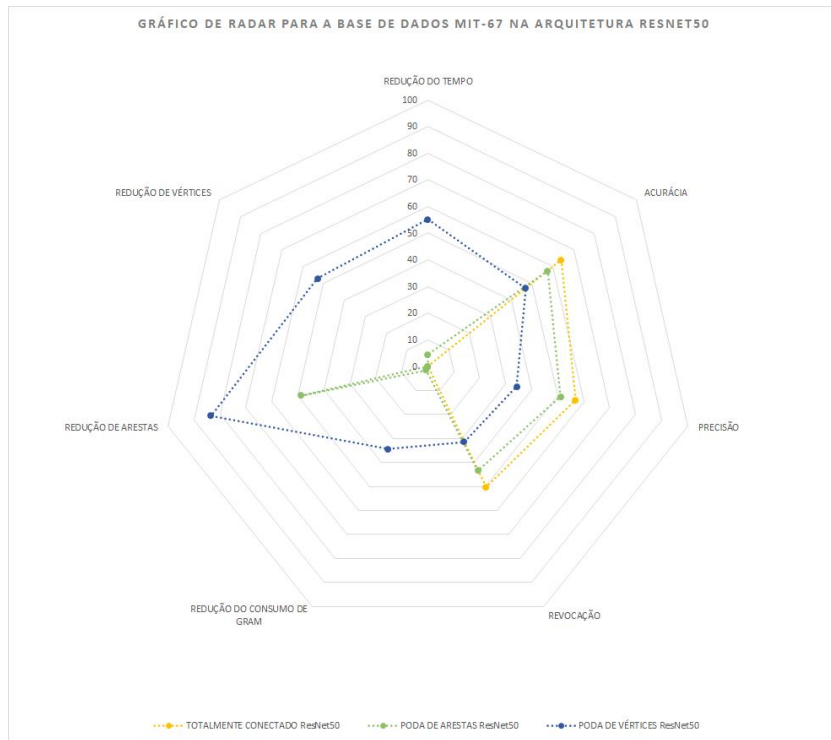


Figura 33: Gráfico de radar dos resultados dos experimentos na arquitetura ResNet50 para a base de dados MIT-67

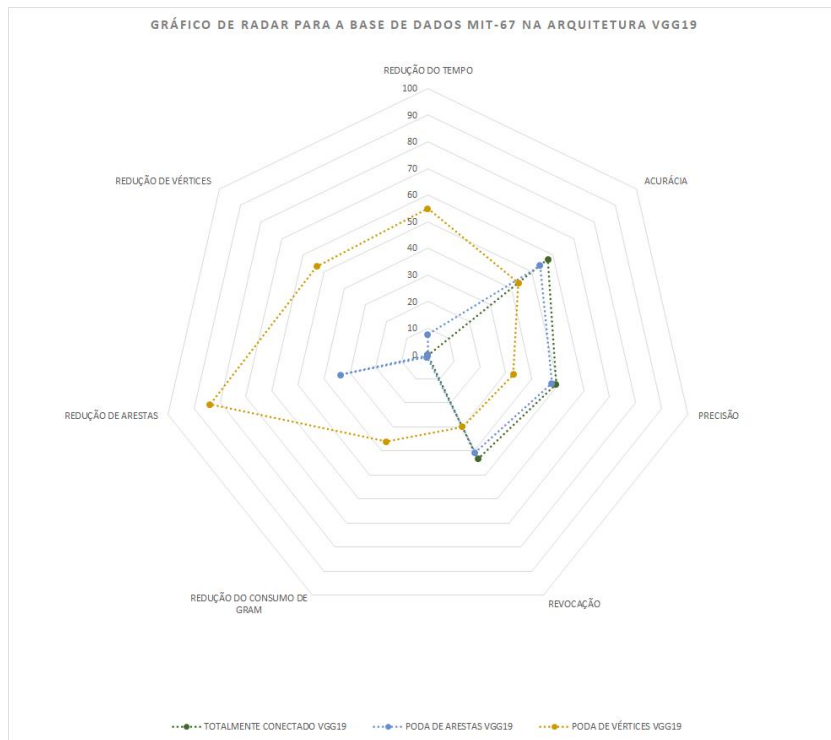


Figura 34: Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados MIT-67

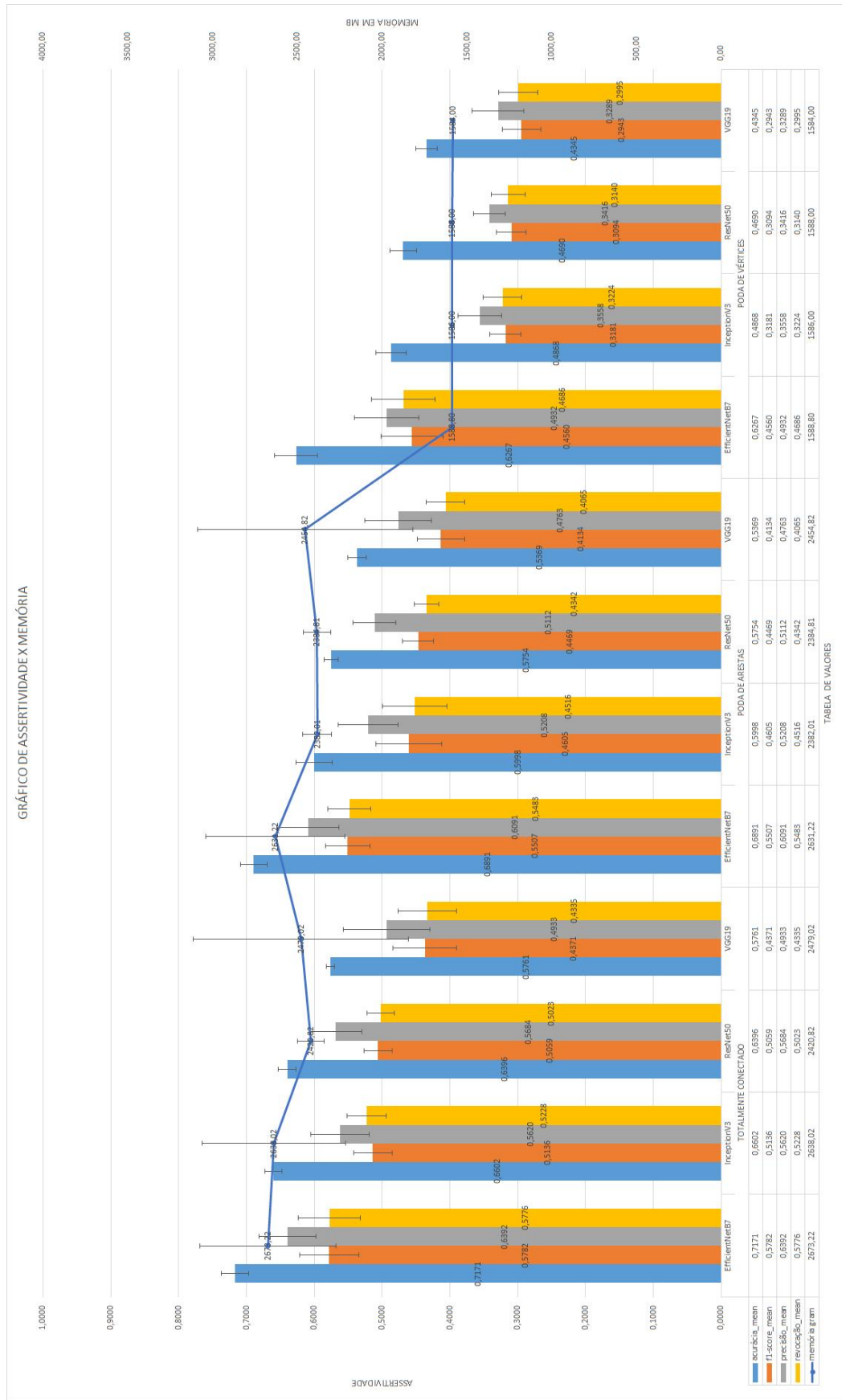


Figura 35: Visualização geral dos resultados dos experimentos para a base de dados MIT-67

3.3.2 UNREL

De maneira análoga à base MIT67, as Figuras 36 a 39 representam as diferentes arquiteturas RNCs citadas nos cenários experimentais. A poda de arestas resultou em até 0,02% de redução do consumo de gram, 0,93% do tempo de treinamento e 20,56% da acurácia. A poda por vértices, por sua vez, foi capaz de reduzir em até 6,77% o consumo de memória gram, 20,59% do tempo de treinamento e 36,53% de acurácia. Pode-se observar os mesmos comportamentos para a base Unrel, na qual as diferentes arquiteturas com poda por arestas apresentam um melhor custo benefício, em relação às outras.

A Figura 40 ilustra uma visualização geral em gráfico de barras e valores com respectivos desvios para todas as métricas consideradas, bem como todas as arquiteturas RNCs e políticas de poda. Novamente, pode-se observar que a poda por vértices gera uma redução considerável em relação a custos computacionais, mas ao preço de redução da acurácia e de outras métricas.

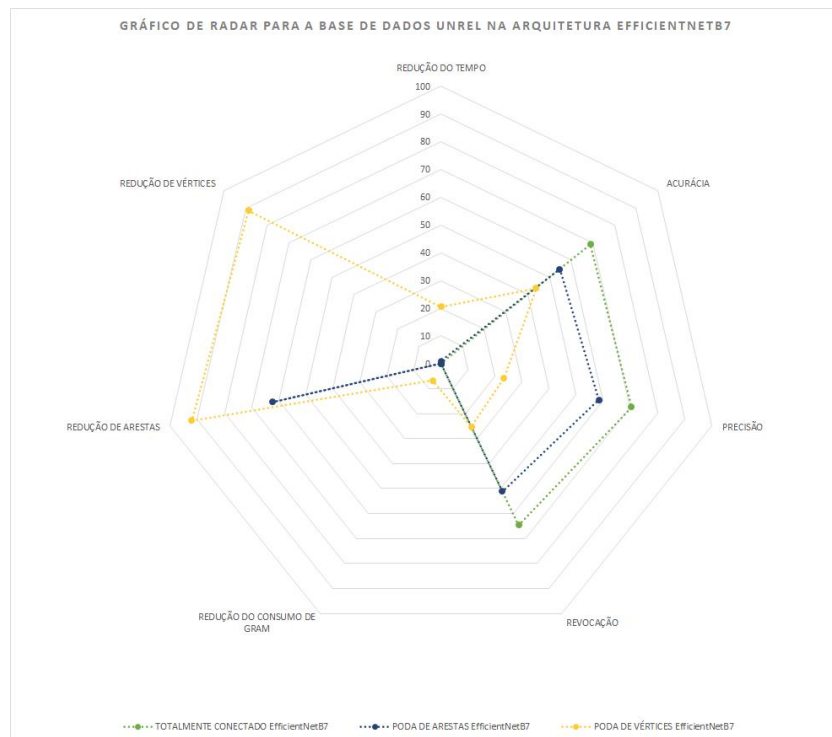


Figura 36: Gráfico de radar dos resultados dos experimentos na arquitetura EfficientNetB7 para a base de dados Unrel

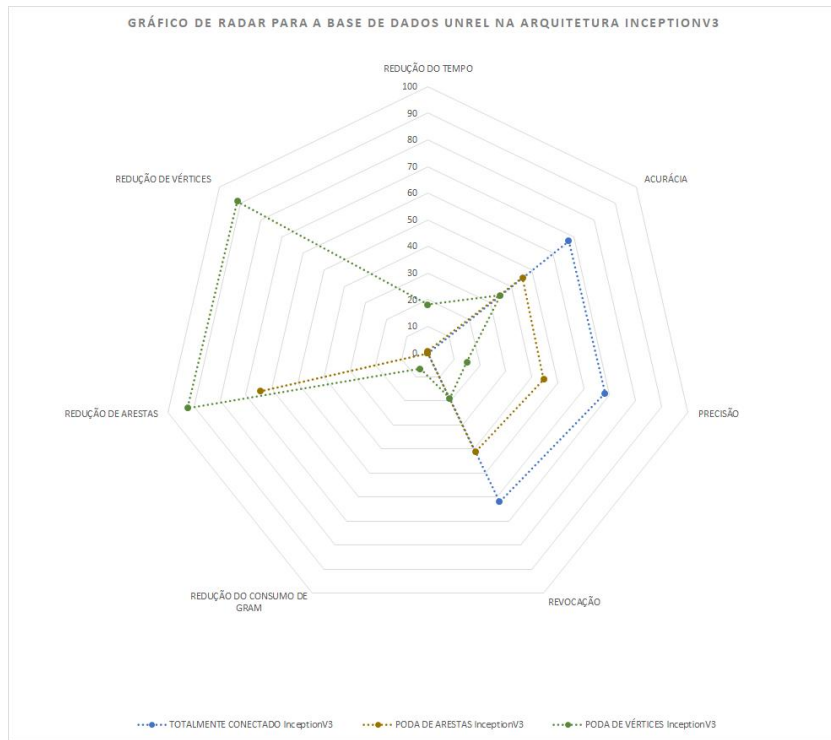


Figura 37: Gráfico de radar dos resultados dos experimentos na arquitetura InceptionV3 para a base de dados Unrel

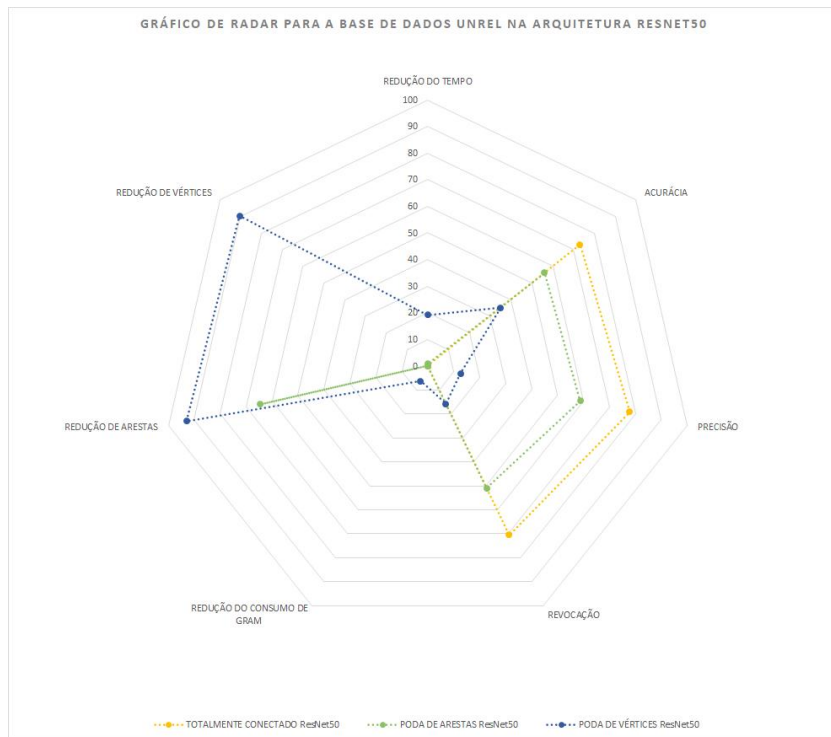


Figura 38: Gráfico de radar dos resultados dos experimentos na arquitetura ResNet50 para a base de dados Unrel

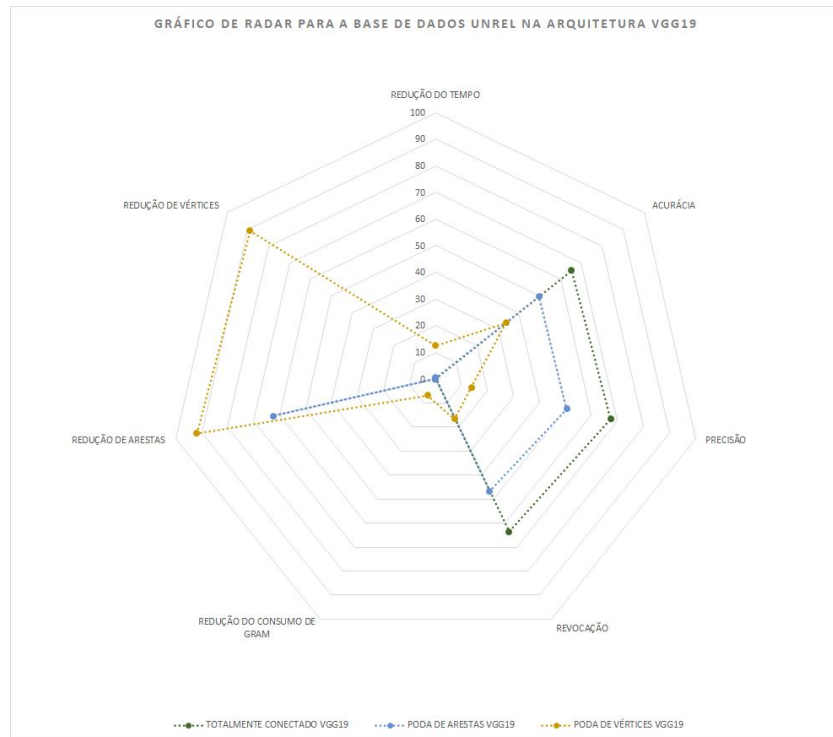


Figura 39: Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados Unrel

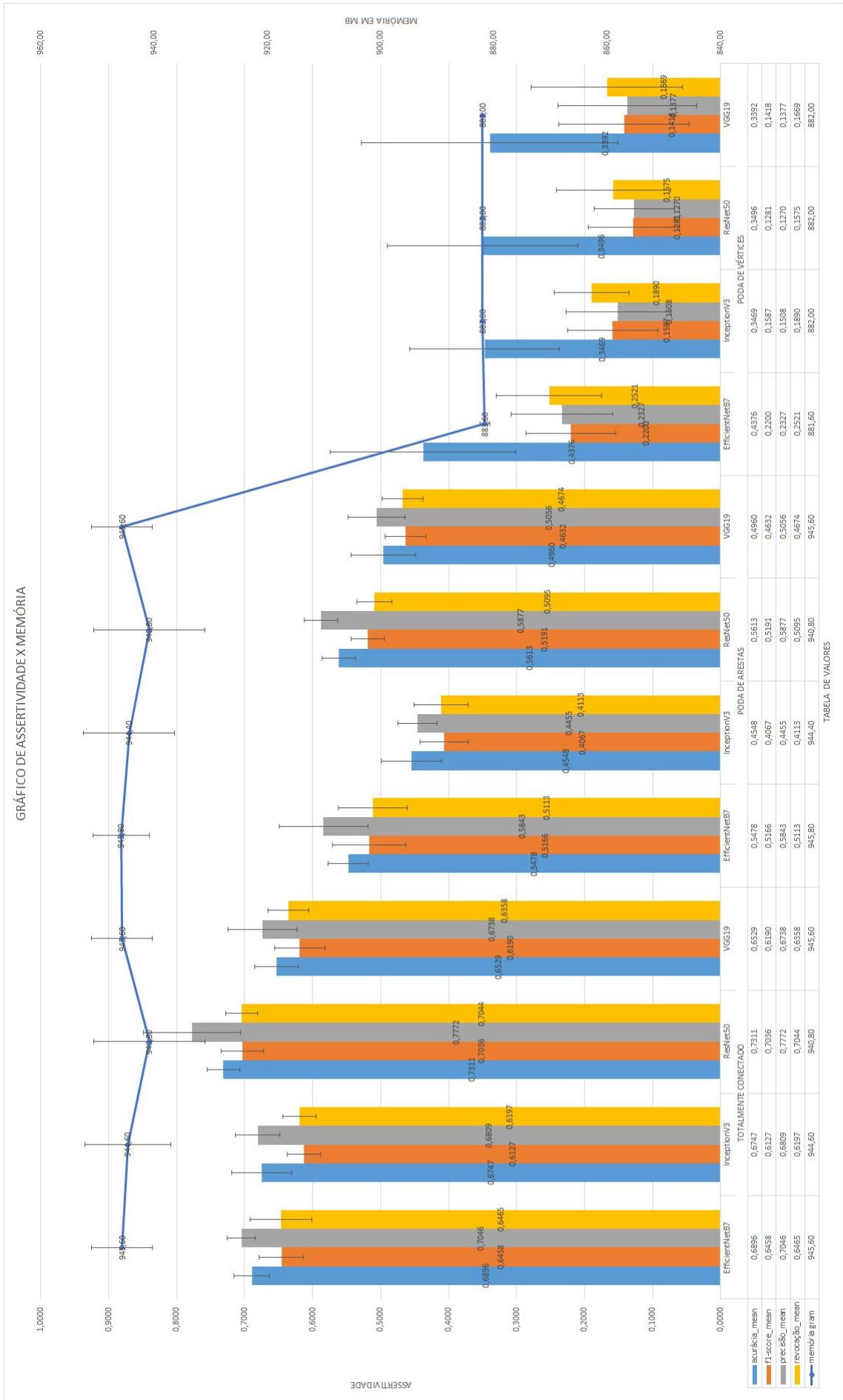


Figura 40: Visualização geral dos resultados dos experimentos para a base de dados Unrel

3.3.3 VRD

A base de imagens VRD apresenta a maior cardinalidade entre as outras. Dessa forma, a redução de custos na mesma torna-se mais aparente em termos computacionais. A redução de memória gram para esta base foi de até 8,92% para poda por arestas, seguida por 17,79% e tendo a menor redução de acurácia em 0,83%. Outrossim, para a poda por vértices, a redução de memória gram foi de até 20,53%, com 60,24% de redução em tempo e apenas 2,41% de redução em acurácia.

É interessante observar por meio das Figuras 41 a 44 que tanto a poda de arestas, quanto a poda de vértices conseguiram manter alta semelhança entre a acurácia do grafo totalmente conectado, bem como em relação às outras métricas como F1-score, precisão e revocação.

No contexto da referida base de imagens, a poda por vértices apresentou os melhores resultados, diferentemente das outras bases que apresentavam uma cardinalidade menor, o que pode ser um indicativo de que em bases de maior cardinalidade, os métodos de poda propostos neste trabalho apresentam uma vantagem superior.

A Figura 45 ilustra a visualização geral em gráfico de barras e os respectivos desvios para todas as métricas consideradas, bem como todas as arquiteturas RNCs e políticas de poda. A partir da mesma, observa-se claramente o empate estatístico existente entre o grafo completo e as duas abordagens de poda aplicadas, em relação às métricas explicitadas (e.g. acurácia, f-measure, etc).

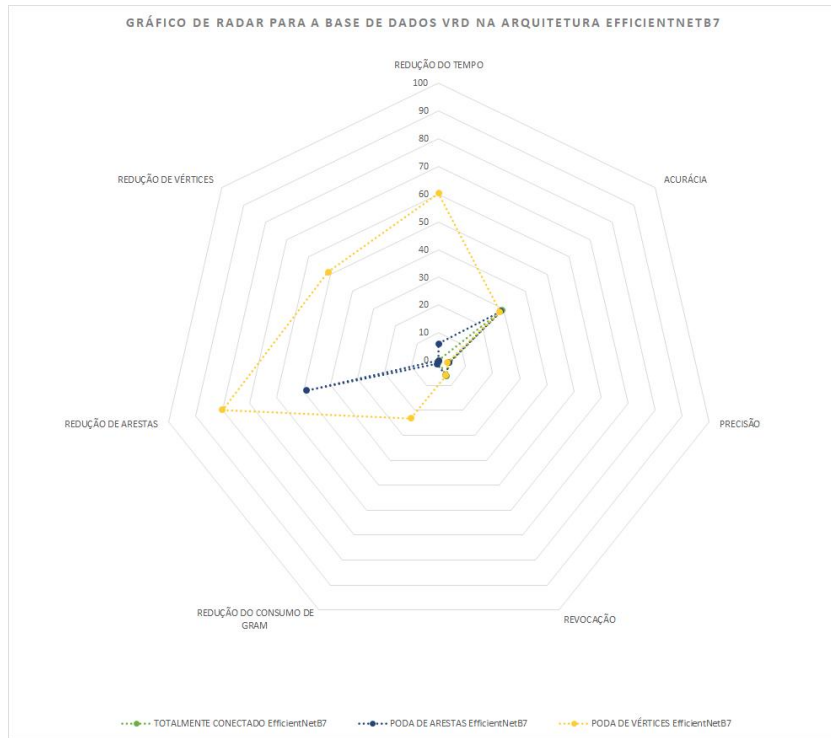


Figura 41: Gráfico de radar dos resultados dos experimentos na arquitetura EfficientNetB7 para a base de dados VRD

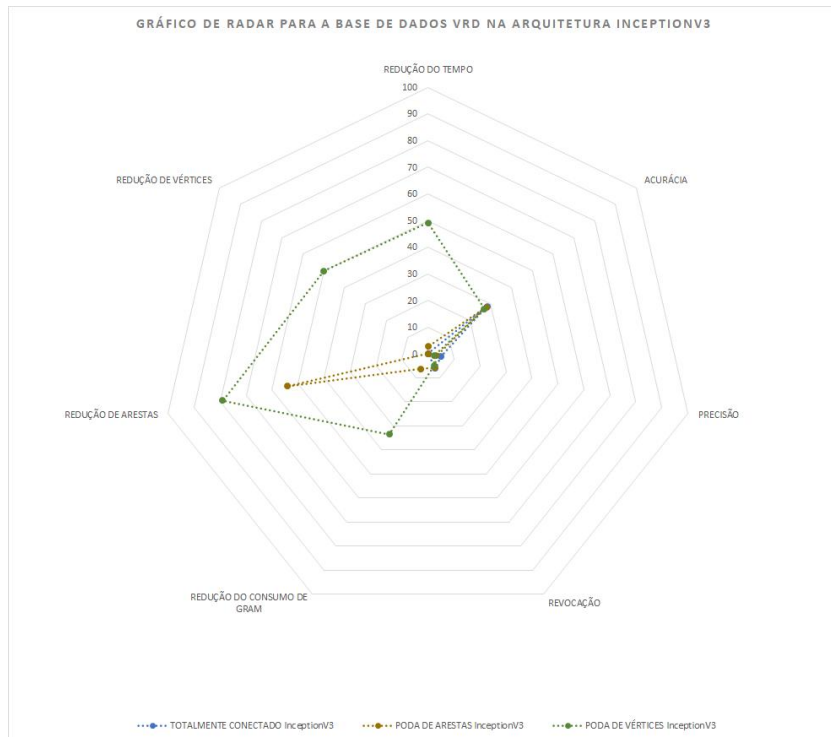


Figura 42: Gráfico de radar dos resultados dos experimentos na arquitetura InceptionV3 para a base de dados VRD

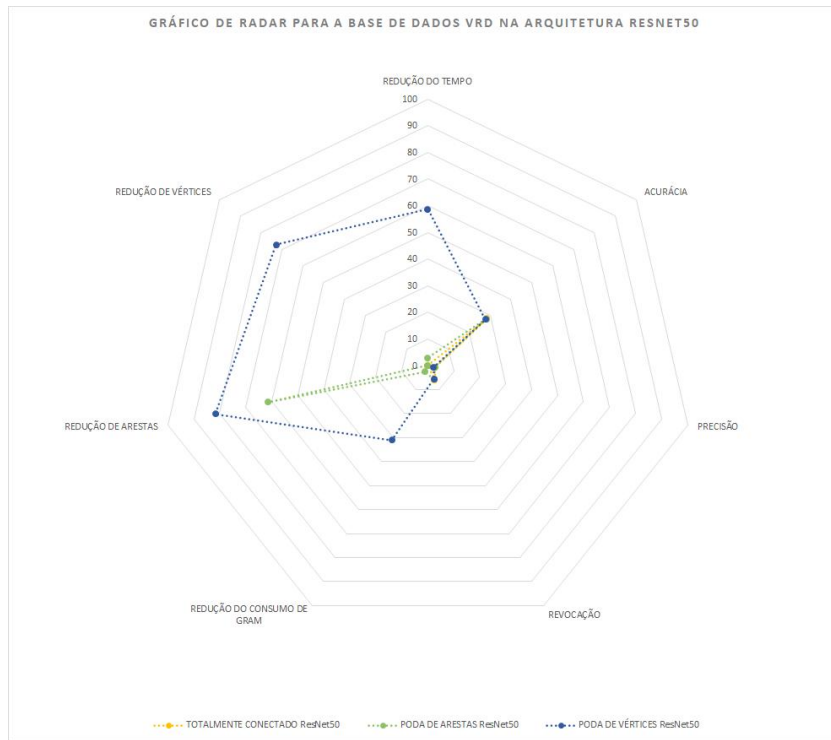


Figura 43: Gráfico de radar dos resultados dos experimentos na arquitetura ResNet50 para a base de dados VRD

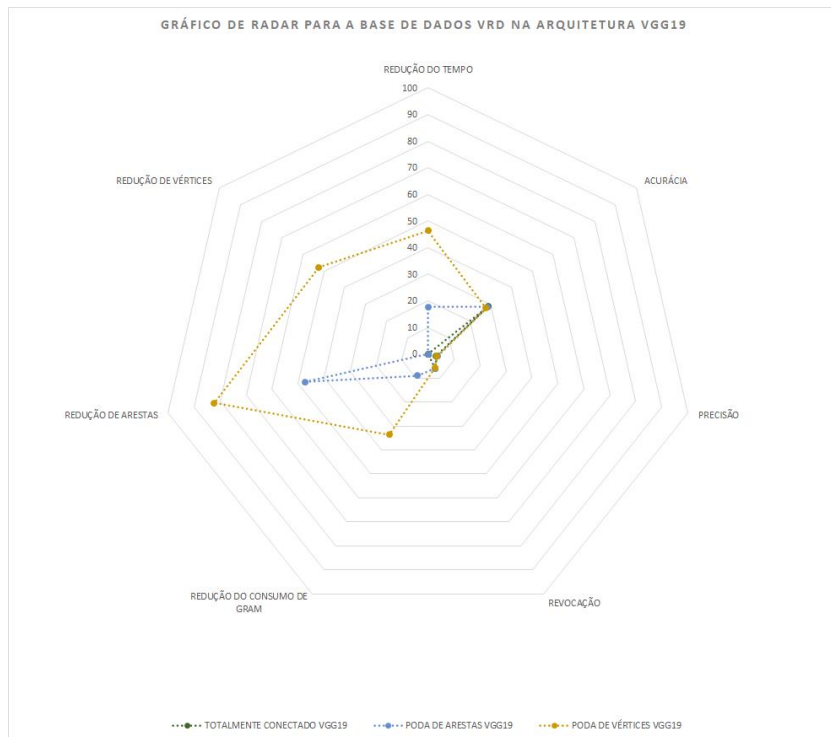


Figura 44: Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados VRD

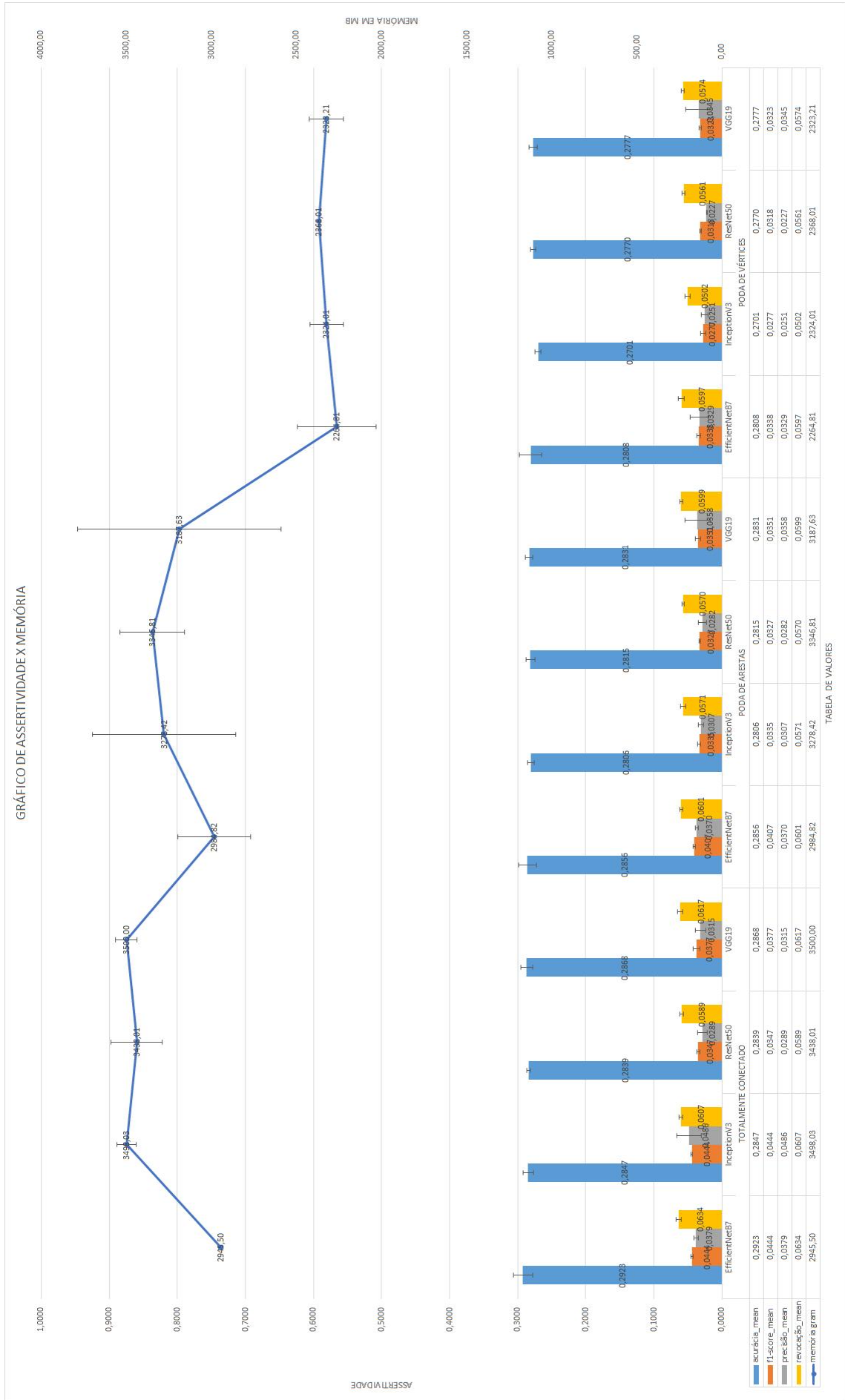


Figura 45: Gráfico de radar dos resultados dos experimentos na arquitetura VGG19 para a base de dados VRD

4 CONSIDERAÇÕES FINAIS

O presente trabalho observou o estado da arte tendo como principal comparativo, grafos cujas arestas estão totalmente conectadas. Foi possível identificar que apesar do declínio de acurácia derivado do processo de poda, conforme apresentado na comparação do modelo totalmente conectado contra o podado, houve uma redução considerável do consumo de memória da placa gráfica durante o treinamento. Outrossim, o algoritmo possui a operação *offline*, ou seja, ele é executado apenas no pré-processamento, não sendo necessário processar a cada época de treinamento.

Os resultados evidenciam um cenário de custo versus benefício. O custo de reduzir métricas de assertividade versus o benefício de economizar recursos em infraestrutura. O melhor resultado apresenta uma redução de 1,53% em acurácia, enquanto o tempo foi reduzido em 60,24%, e o consumo de memória em 23,11%. Também foi identificado que bases de imagens com volumes maiores tenderam a apresentar menores impactos em assertividade quando comparadas com as de menor tamanho, o que pode ser interessante em cenários em que há a geração de muitos dados.

É possível concluir que o método de poda é promissor em alguns cenários, podendo ter uma aplicabilidade interessante em situações que perdas de assertividade são menores do que custo de infraestrutura. Ainda pode existir a possibilidade da poda performar de forma diferente em bases de imagens ainda não exploradas na pesquisa, especialmente aquelas em que a diversidade de imagens se torna extremamente relevante, dado o objeto de estudo das podas. Um outro ponto, é que o trabalho contribui não só para redução de custos, mas também para redução da emissão de carbono.

4.1 TRABALHOS FUTUROS

A pesquisa explorou uma frente utilizando a informação visual contextual para podar conexões e subimagens que fossem menos relevantes para o treinamento. Possíveis abordagens podem ser feitas a partir do trabalho, como uso de matriz de fatoração para encontrar arestas e vértices relevantes e poda por posição de elementos. Novas métricas de distância, bases de imagens e arquiteturas também podem ser exploradas nos futuros trabalhos.

4.1.1 ALGORITMOS DE MATRIZ DE FATORAÇÃO PARA ENCONTRAR ARESTAS E VÉRTICES RELEVANTES

A matriz de fatoração foi uma abordagem utilizada para algoritmos de recomendação, conforme em Steck (2015). O fator de similaridade entre os vértices, levantado neste trabalho, pode representar uma matriz de recomendação entre diferentes características. Esse algoritmo pode ser um ponto interessante para possíveis recomendações de quais arestas e vértices devem ou não ser podados.

4.1.2 PODA POR GRAU DE DIVERSIDADE

O trabalho explorou a poda por dissimilaridade de objetos através da distância euclidiana. Uma abordagem de poda por itens que não apresentam semelhança pode ser utilizada como um trabalho futuro, validando se nas RNGs, o fator de diversidade de dados pode ser um possível ofensor do declínio de assertividade.

4.1.3 PODA POR POSIÇÃO DE ELEMENTOS

A posição dos retângulos envolventes em uma imagem apresenta uma informação que pode ser interessante para a poda, que tem a ver com a área de representação. Quanto mais ao centro da imagem o objeto estiver, isso pode representar um aspecto de foco, podendo remeter a uma característica de relevância. Uma possível abordagem é aplicar a poda considerando restrições com relação à posição dos objetos, tratando objetos muito centrais como relevantes.

4.2 PUBLICAÇÕES

Campos W.S., Souza L.G., Saito P.T.M., Bugatti P.H. (2021) Contextual Image Classification Through Fine-Tuned Graph Neural Networks. In: Rutkowski L., Scherer R., Korytkowski M., Pedrycz W., Tadeusiewicz R., Zurada J.M. (eds) Artificial Intelligence and Soft Computing. ICAISC 2021. Lecture Notes in Computer Science, vol 12855. Springer, Cham. https://doi.org/10.1007/978-3-030-87897-9_2.

REFERÊNCIAS

ALPAYDIN, E. **Introduction to Machine Learning**. 3. ed. Cambridge, MA: MIT Press, 2014. (Adaptive Computation and Machine Learning). ISBN 978-0-262-02818-9.

ANDONIE, R. Hyperparameter optimization in learning systems. **Journal of Membrane Computing**, Springer Science and Business Media LLC, v. 1, n. 4, p. 279–291, out. 2019.

ARBIB, M. A. **The Handbook of Brain Theory and Neural Networks**. 2nd. ed. Cambridge, MA, USA: MIT Press, 2002. ISBN 0262011972.

AUER, S.; BIZER, C.; KOBILAROV, G.; LEHMANN, J.; CYGANIAK, R.; IVES, Z. Dbpedia: A nucleus for a web of open data. In: **The semantic web**. [S.l.]: Springer, 2007. p. 722–735.

BALCILAR, M.; RENTON, G.; HEROUX, P.; GAUZERE, B.; ADAM, S.; HONEINE, P. **Bridging the Gap Between Spectral and Spatial Domains in Graph Neural Networks**. 2020.

BLALOCK, D.; ORTIZ, J. J. G.; FRANKLE, J.; GUTTAG, J. **What is the State of Neural Network Pruning?** 2020.

BONETTO, R.; LATZKO, V. Machine learning. In: **Computing in Communication Networks**. Elsevier, 2020. p. 135–167. Disponível em: <<https://doi.org/10.1016/b978-0-12-820488-7.00021-9>>.

BRONSTEIN, M. M.; BRUNA, J.; LECUN, Y.; SZLAM, A.; VANDERGHEYNST, P. Geometric Deep Learning: Going beyond Euclidean data. **IEEE Signal Processing Magazine**, IEEE, v. 34, n. 4, p. 18–42, 2017. ISSN 10535888.

BRUNA, J.; ZAREMBA, W.; SZLAM, A.; LECUN, Y. Spectral networks and deep locally connected networks on graphs. **2nd International Conference on Learning Representations**, p. 1–14, 2014.

BURKOV, A. **Machine Learning Engineering**. Andriy Burkov, 2020. ISBN 9781777005467. Disponível em: <<https://books.google.com.br/books?id=BRzZzQEACAAJ>>.

CALAMONERI, T.; FINOCCHI, I.; ITALIANO, G. F. **Algorithms and Complexity**. [S.l.]: Springer, 2006.

Chaudhuri, S.; Ganti, V.; Kaushik, R. A primitive operator for similarity joins in data cleaning. In: **22nd International Conference on Data Engineering (ICDE'06)**. [S.l.: s.n.], 2006. p. 5–5.

CHECHIK, G.; MEILIJSON, I.; RUPPIN, E. Neuronal regulation implements efficient synaptic pruning. **Advances in Neural Information Processing Systems**, v. 11, p. 97–103, 1998.

Chung, H.; Huang, H.; Chen, H. Predicting future participants of information propagation trees. In: **2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)**. [S.l.: s.n.], 2019. p. 321–325.

CO-OPERATION, O. for E.; DEVELOPMENT. **Artificial Intelligence in society**. [S.l.]: OECD Publishing Paris, 2019.

DALIANIS, H. **Clinical text mining: Secondary use of electronic patient records**. [S.l.]: Springer Nature, 2018.

DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. Convolutional neural networks on graphs with fast localized spectral filtering. **CoRR**, abs/1606.09375, 2016.

DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. Convolutional neural networks on graphs with fast localized spectral filtering. arXiv, 2016. Disponível em: <<https://arxiv.org/abs/1606.09375>>.

DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. Convolutional neural networks on graphs with fast localized spectral filtering. **CoRR**, abs/1606.09375, 2016.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255.

DENG, W.; PAN, J.; ZHOU, T.; KONG, D.; FLORES, A.; LIN, G. **DeepLight: Deep Lightweight Feature Interactions for Accelerating CTR Predictions in Ad Serving**. 2021.

FREITAS, L. Q. D. **Medidas de centralidade em grafos**. Tese (Doutorado) — Dissertação de Mestrado apresentada ao Programa de Pós-graduação em . . . , 2010.

Gallicchio, C.; Micheli, A. Graph echo state networks. In: **International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2010. p. 1–8.

GOOGLE Compute Engine – Computation in the cloud. 2012. Disponível em: <<https://cloud.google.com/files/GoogleComputeEngine.pdf>>.

GORI, M.; MONFARDINI, G.; SCARSELLI, F. A new model for learning in graph domains. **Proceedings of the International Joint Conference on Neural Networks**, IEEE, v. 2, p. 729–734, 2005.

GUO, H.; TANG, R.; YE, Y.; LI, Z.; HE, X. **DeepFM: A Factorization-Machine based Neural Network for CTR Prediction**. 2017.

HENAFF, M.; BRUNA, J.; LECUN, Y. Deep convolutional networks on graph-structured data. **CoRR**, abs/1506.05163, 2015.

Hong, L.; Zou, L.; Lian, X.; Yu, P. S. Subgraph matching with set similarity in a large graph database. **IEEE Transactions on Knowledge and Data Engineering**, v. 27, n. 9, p. 2507–2521, 2015.

KARJANTO, N. **Properties of Chebyshev polynomials**. 2020.

KHAN, R. Z.; ALI, M. F. **A Study of Cloud Computing**. International Research Journal of Computer Science (IRJCS), May 2015. Disponível em: <<https://www.irjcs.com/volume-2-issue-05>>.

KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. **5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings**, p. 1–14, 2019.

LANGILLE, J. J.; BROWN, R. E. The synaptic theory of memory: a historical survey and reconciliation of recent opposition. **Frontiers in systems neuroscience**, Frontiers, p. 52, 2018.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. v. 521, n. 7553, p. 436–444, 2015.

LEVIE, R.; MONTI, F.; BRESSON, X.; BRONSTEIN, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. **CoRR**, abs/1705.07664, 2017.

LEY, M. The dblp computer science bibliography: Evolution, research issues, perspectives. In: SPRINGER. **International symposium on string processing and information retrieval**. [S.l.], 2002. p. 1–10.

LI, Q.; HAN, Z.; WU, X. Deeper insights into graph convolutional networks for semi-supervised learning. **CoRR**, abs/1801.07606, 2018.

LIBERTI, L.; LAVOR, C.; MACULAN, N.; MUCHERINO, A. **Euclidean distance geometry and applications**. 2012.

LLOYD, E. K.; BONDY, J. A.; MURTY, U. S. R. Graph Theory with Applications. **The Mathematical Gazette**, v. 62, n. 419, p. 63, 1978. ISSN 00255572.

LU, C.; KRISHNA, R.; BERNSTEIN, M.; FEI-FEI, L. Visual relationship detection with language priors. In: **European Conference on Computer Vision**. [S.l.: s.n.], 2016.

MINAR, M. R.; NAHER, J. Recent advances in deep learning: An overview. v. 2006, 2018.

MITCHELL, T. M. **Machine Learning**. New York: McGraw-Hill, 1997. ISBN 978-0-07-042807-2.

PEYRE, J.; LAPTEV, I.; SCHMID, C.; SIVIC, J. Weakly-supervised learning of visual relations. In: **ICCV**. [S.l.: s.n.], 2017.

QUATTONI, A.; TORRALBA, A. Recognizing indoor scenes. In: IEEE. **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.], 2009. p. 413–420.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. [S.l.]: Prentice Hall, 2010.

SACHS, H.; STIEBITZ, M.; WILSON, R. J. An historical note: Euler's königsberg letters. **Journal of Graph Theory**, Wiley Online Library, v. 12, n. 1, p. 133–139, 1988.

SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. **SN Computer Science**, Springer Science and Business Media LLC, v. 2, n. 3, mar. 2021. Disponível em: <<https://doi.org/10.1007/s42979-021-00592-x>>.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. **IEEE Transactions on Neural Networks**, v. 20, n. 1, p. 61–80, 2009.

Sperduti, A.; Starita, A. Supervised neural networks for the classification of structures. **IEEE Transactions on Neural Networks**, v. 8, n. 3, p. 714–735, 1997.

STECK, H. Gaussian ranking by matrix factorization. In: **Proceedings of the 9th ACM Conference on Recommender Systems**. New York, NY, USA: Association for Computing Machinery, 2015. (RecSys '15), p. 115–122. ISBN 9781450336925. Disponível em: <<https://doi.org/10.1145/2792838.2800185>>.

TURING, A. M. Computing machinery and intelligence. **Mind**, Oxford University Press on behalf of the Mind Association, v. 59, n. 236, p. 433–460, 1950. ISSN 00264423. Disponível em: <<http://www.jstor.org/stable/2251299>>.

WANG, H.; RAJ, B. On the Origin of Deep Learning. p. 1–72, 2017.

XU, J.; BAO, Z. Neural networks and graph theory. **Science in China Series F: Information Sciences**, v. 45, n. 1, p. 1–24, 2002. ISSN 1009-2757, 1862-2836.

ZHOU, J.; CUI, G.; ZHANG, Z.; YANG, C.; LIU, Z.; SUN, M. Graph neural networks: A review of methods and applications. **CoRR**, abs/1812.08434, 2018.

APÊNDICE A – DEMAIS CENÁRIOS EXPERIMENTAIS

Essa pesquisa apresentou diversos experimentos decorridos de um extensivo processo de validação. No entanto, outros cenários experimentais foram desenvolvidos, ainda que não representem os valores finais, certamente auxiliaram na tomada de decisão.

Um dos cenários se trata da realização de uma poda de arestas cuja seleção das mesmas se baseou no número gerado do processo de poda apresentado neste trabalho. Para arquitetura InceptionV3, foi obtido o valor de uma acurácia em aproximadamente 60%, contra 69% da poda de arestas e contra 71% do grafo totalmente conectado. Esse resultado auxiliou na compreensão de que o método poderia ser promissor.

Também foi explorado de forma breve o uso da distância de cosseno, que não revelou melhorias quando comparada a distância Euclideana.

Por fim, os primeiros experimentos executados neste trabalho, foram derivados de um notebook cujo processador é um Intel Core i7, de 2 núcleos e 2.5GHz, com 16GB de memória RAM e a placa gráfica GeForce 930M. Posteriormente sendo executado no servidor da universidade UTFPR-CP, em uma máquina cujo processador é um Intel Core i7 com 8 núcleos, 16 threads e 3.40 GHz, 32 GB de memória RAM e a placa gráfica Nvidia GeForce RTX 2080Ti.

APÊNDICE B – DECISÕES DO TRABALHO

Neste trabalho, a decisão de utilizar a poda por informações contextuais partiu da ideia de tornar o método escalável utilizando a estrutura já existente dentro de uma base de imagens. Um dos principais motivadores consistiu em tornar as Redes Neurais de Grafos mais escaláveis para o uso em larga escala.

Outra decisão relacionada aos cenários experimentais, consiste no uso de máquinas em infraestrutura virtual da plataforma Google Cloud. Devido ao momento de pandemia global, o acesso ao campus ficou limitado e, conseqüentemente, aos recursos de *hardware* disponíveis. As bases MIT-67 e Unrel conseguiam serem processadas nas máquinas apresentadas no Apêndice B, contudo, no que tange aos experimentos na VRD, foi extremamente importante executar o processamento em máquinas de poder computacional maior. Graças ao uso da plataforma, também foi possível executar e re-executar experimentos com maior flexibilidade.

Por fim, a decisão de podar vértices após a poda de arestas veio pela observação da topologia do grafo após o procedimento da poda. Sem a remoção das arestas, não seria possível aplicar a técnica utilizada neste trabalho. Neste ponto, uma nova topologia pode despertar uma nova hipótese.

APÊNDICE C – REPOSITÓRIO DA IMPLEMENTAÇÃO

O código base implementado para execução de experimentos nesta pesquisa está no repositório virtual "<https://github.com/camposwalacy/research>".

