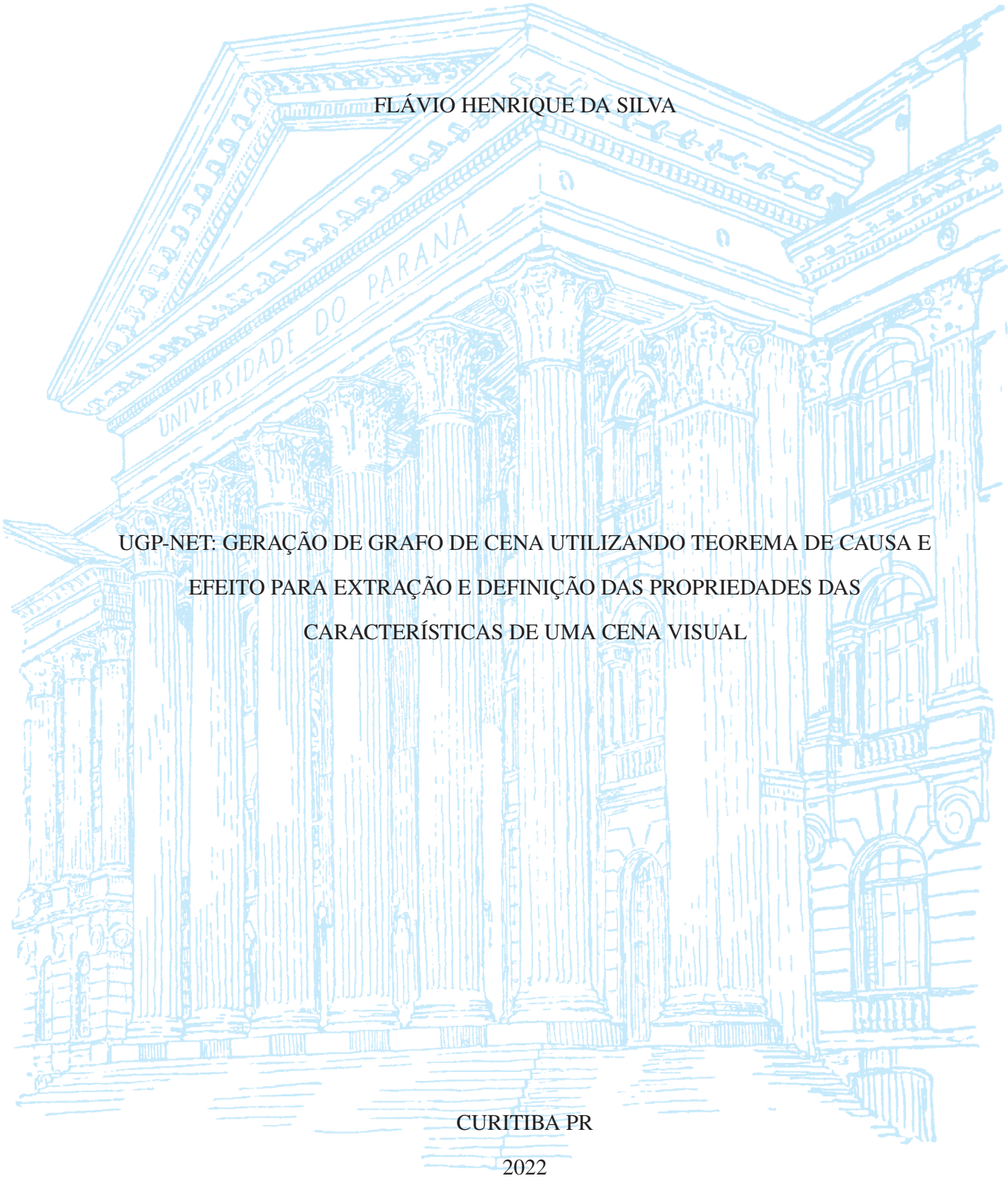
UNIVERSIDADE FEDERAL DO PARANÁ

FLÁVIO HENRIQUE DA SILVA



UGP-NET: GERAÇÃO DE GRAFO DE CENA UTILIZANDO TEOREMA DE CAUSA E EFEITO PARA EXTRAÇÃO E DEFINIÇÃO DAS PROPRIEDADES DAS CARACTERÍSTICAS DE UMA CENA VISUAL

CURITIBA PR

2022

FLÁVIO HENRIQUE DA SILVA

UGP-NET: GERAÇÃO DE GRAFO DE CENA UTILIZANDO TEOREMA DE CAUSA E EFEITO PARA EXTRAÇÃO E DEFINIÇÃO DAS PROPRIEDADES DAS CARACTERÍSTICAS DE UMA CENA VISUAL

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. André Luiz P. Guedes.

CURITIBA PR

2022

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA CIÊNCIA E TECNOLOGIA

Silva, Flávio Henrique da.
UGP-NET : geração de grafo de cena utilizando Teorema de Causa e Efeito para extração e definição das propriedades das características de uma cena visual. / Flávio Henrique da Silva. – Curitiba, 2022.
1 recurso on-line : PDF.

Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.
Orientador: Prof. Dr. André Luiz P. Guedes.

1. Informática. 2. Redes neurais (Computadores). 3. Imagem. 4. Machine learning. I. Guedes, André Luiz P. II. Universidade Federal do Paraná. Programa de Pós-Graduação em Informática. IV. Título.

Bibliotecário: Nilson Carlos Vieira Júnior CRB-9/1797

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **FLÁVIO HENRIQUE DA SILVA** intitulada: **UGP-NET: GERAÇÃO DE GRAFO DE CENA UTILIZANDO TEOREMA DE CAUSA E EFEITO PARA EXTRAÇÃO E DEFINIÇÃO DAS PROPRIEDADES DAS CARACTERÍSTICAS DE UMA CENA VISUAL**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 16 de Novembro de 2022.

Assinatura Eletrônica
16/11/2022 14:23:52.0
ANDRÉ LUIZ PIRES GUEDES
Presidente da Banca Examinadora

Assinatura Eletrônica
18/11/2022 09:40:40.0
MARCO ANTÔNIO RODRIGUES FERNANDES
Avaliador Externo (UNIVERSIDADE ESTADUAL PAULISTA - UNESP)

Assinatura Eletrônica
16/11/2022 13:59:51.0
EDUARDO JAQUES SPINOSA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Dedico essa dissertação a meu irmão Fernando, pelo apoio incondicional e o constante incentivo e força para continuar e terminar o mestrado.

Dedico essa dissertação ao meu orientador Prof. Dr André Luiz Pires Guedes, por acreditar no meu potencial e me apoiar na ideia e criação do projeto, pelo incentivo, amizade e excelente orientação.

Dedico também aos meus pais Selma e Luiz por me apoiarem em vários aspectos nessa jornada árdua e marcante da minha vida.

AGRADECIMENTOS

Meus agradecimentos são direcionados àqueles e àquelas que, de alguma forma, contribuíram com a execução dessa dissertação.

Agradeço primeiramente a Deus por me dar força intelectual e emocional em todas as etapas do mestrado.

Agradeço ao meu orientador Prof. Dr André Luiz Pires Guedes, por acreditar no meu potencial desde o início do mestrado, pela paciência e confiança na ideia do projeto, pelo empenho prático com que me orientou. Agradeço-o profundamente nessa fase da vida.

Agradeço a meu irmão gêmeo Fernando Eduardo da Silva, pelo apoio emocional comigo no período do mestrado, por ser um irmão forte, determinado e guerreiro na vida.

Agradeço também a minha família e notadamente aos meus pais, Selma e Luiz, por nunca terem medido esforços para me proporcionar ensino de qualidade, saúde e acolhimento.

RESUMO

A utilização de grafos está presente em vários domínios do conhecimento e em diversas aplicações, desde a análise social, na fabricação de novos medicamentos e até mesmo em visão computacional em abordagens que buscam a descrição ou representação de uma cena visual. Este trabalho tem o objetivo de investigar e apresentar o método UGP-Net (*Unbiased Graph Property Network for Scene Graph Generation*) que visa tratar as limitações descritas no processo de geração de grafo de cena (SGG), especificamente na etapa de refinamento das características de cada nó, que representa imagens de diferentes níveis semânticos e que possa ser validada em conjuntos de imagens públicos e distintos tais como: *Visual Genome*, *OpenImages* e *Visual Relationship Detection*. Além disso, o trabalho realiza um comparativo referente a eficiência e a performance da abordagem UGP-Net e os métodos considerados estado da arte baseados em aprendizagem profunda que empregam o uso de redes neurais para grafos (GNN) e de redes neurais convolucionais de grafos (GCN) para execução do *pipeline* do processo SGG que consiste em localizar e detectar vários padrões de objetos, atributos e relacionamentos.

Palavras-chave: Geração de Grafo de Cena, Redes Neurais para Grafos, Redes Neurais Convolucionais de Grafos, Classificação de Imagem.

ABSTRACT

The use of graphs in present in various fields of knowledge and in several applications, from social analysis, in the manufacture of new drugs and even in computer vision in approaches that seek the description or representation of a visual scene. This work aims to investigate and present the UGP-Net (Unbiased Graph Property Network for Scene Graph Generation) method, which aims to deal with the limitations described in the scene graph generation process (SGG), specifically in the stage of refinement of the characteristics of each node, which represents images of different semantic levels and that can be validated in sets of public and distinct images such as: Visual Genome, OpenImages and Visual Relationship Detection. In addition, the work compares the efficiency and performance of the UGP-Net approach and state-of-the-art methods based on deep learning that employ the use of graph neural networks (GNN) and graph convolutional neural networks (GCN) for running the SGG process pipeline that consists of finding and detecting various patterns of objects, attributes and relationships.

Keywords: Scene Graph Generation, Graph Neural Networks, Graph Convolutional Networks, Image Classification

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

CNN	Redes Neurais Convolucionais (Convolutional Neural Networks)
GNN	Redes neurais para Grafos (Graph Neural Networks)
GCN	Redes Convolucionais de Grafo (Graph Convolutional Networks)
GRU	Gated Recorrent Unit
IC	Inferência Causal
IoU	Intersecção sobre a União
LSTM	Long Short-Term Memory
MSDN	Multi-Level Scene Description Network
PredCls	Predicates Classification
SCM	Structural Causal Models
SGDet	Scene Graph Detection
SG	Grafo de Cena (Scene Graph)
SGG	Geração de Grafo de Cena (Scene Graph Generation)
SGCls	Scene Graph Classification
RPN	Rede de Proposta de Regiões (Region Proposal Network)
R-CNN	Region-based Convolutional Network
ROI	Region of interest
RNN	Recurrent Neural Networks

SUMÁRIO

1 INTRODUÇÃO

Compreender uma cena visual (Johnson et al., 2015; Lazebnik et al., 2006; Zhou et al., 2014) está além de reconhecer e classificar objetos de maneira isolada, simultânea e rápida, por exemplo: carro, casa, árvore, pessoa, entre outras classes. Em uma cena visual (imagem) existem outras informações que são extremamente importantes no âmbito da área de Visão Computacional, tais como: os relacionamentos entre os pares de objetos formam um rico conjunto de informações semânticas e contextuais em relação a imagem analisada, seja em ambientes controlados ou sem restrições (*in the wild*) e as características das entidades (objetos) que auxiliam no processo de definição dos relacionamentos entre os objetos detectados na imagem.

No entanto, mesmo com o crescente avanço das abordagens e técnicas de aprendizado profundo, inferir relações contextuais complexas e representações gráficas estruturais a partir de dados visuais heterogêneos continua sendo um dos desafios da área de Visão Computacional, que busca constantemente por novos métodos que possam analisar, representar de forma robusta e com maior acuracidade uma respectiva cena visual, simultaneamente identificar o conjunto de objetos e descrever semanticamente o relacionamento entre os pares que compõem a imagem.

Contudo, o processo de compreensão e mapeamento semântico tem como objetivo capturar as informações estruturais em uma imagem, incluindo os objetos, os relacionamentos entre pares de objetos e a região que consiste no ambiente que todas as informações se encontram, por exemplo, parque, casa, shopping, entre outros. Este processo consiste em receber como entrada uma imagem (2D) e tem como saída a classificação dos objetos, atributos e relacionamentos por meio de métodos de detecção, resultando assim, em uma representação estrutural semântica definida como um grafo de cena (Figura 1.1)

Na literatura a geração de grafo de cena foi definida pela primeira vez como uma forma de representação de uma imagem que pode expressar claramente os objetos, atributos e relacionamentos entre os objetos da cena (Johnson et al., 2015), tal processo tem em seu *pipeline* inúmeras etapas em diferentes níveis semânticos. A Figura 1.2 ilustra a estrutura de um grafo de cena completo, que pode representar estruturalmente a semântica detalhada de um conjunto de cenas, onde as representações robustas codificam as imagens e vídeos em seus elementos semânticos abstratos sem qualquer restrição sobre os tipos de atributos e de relações.

Apesar do grande avanço nos últimos anos na pesquisa de detecção de objetos (Ren et al., 2015; He et al., 2015; Liu et al., 2016; Redmon et al., 2016) usando técnicas de aprendizagem profunda (Levi e Hassner, 2015; Ranjan et al., 2017a; Qi et al., 2019a; Tang et al., 2020) especificamente em relação a redes neurais convolucionais (CNN, do inglês *Convolucional Neural Networks*) (Krizhevsky et al., 2012) observou-se que elas não são eficientes no processo de inferir relações contextuais e semânticas complexas a partir de dados visuais de forma estrutural, pois os dados presentes em uma imagem não podem ser representados em um espaço euclidiano, ou seja, são informações que não podem ser representadas através de sequências não esparsas de valores de uma ou duas dimensões (vetores e matrizes), pois apresentam estrutura irregular. Diante as limitações apresentadas pelas CNNs se identificou que a melhor forma de representar as informações semânticas é por meio de grafos que são estruturas de dados compostas por dois subconjuntos denominados de vértices e arestas.

Um grafo G é um conjunto $V(G)$ finito e não vazio de vértices e um conjunto $E(G)$ de arestas, que são subconjuntos de dois elementos de $V(G)$. Dado $\{u, v\} \in V(G)$, se $\{u, v\} \in E(G)$, dizemos que existe uma aresta entre u e v. No contexto de geração de grafo de cena,

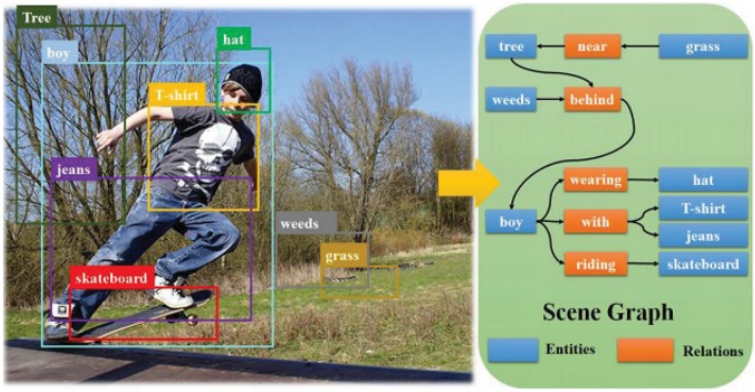


Figura 1.1: Exemplo de grafo de cena: (a) as entidades da imagem foram todas detectadas, por exemplo: menino, árvore, skate, grama e cabeça e (b) associação das entidades por meio dos seus respectivos relacionamentos (Ex: *boy – riding – skatevoard*; *weeds – behind – boy* e *boy – wearning – hat*). Fonte: Qi et. al (2019).

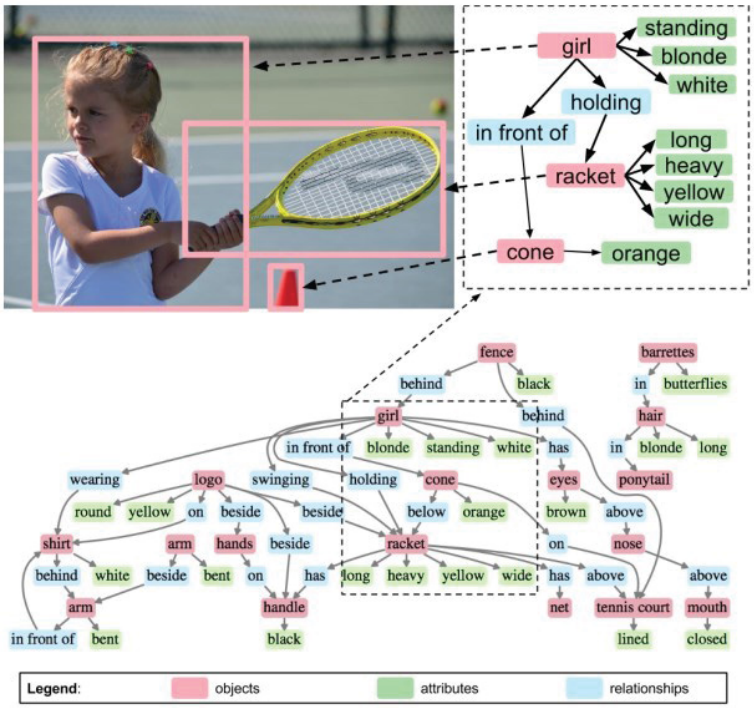


Figura 1.2: Exemplo de grafo de cena (parte inferior) e o grounding (parte superior). O grafo de cena codifica semanticamente os objetos ("girl"), atributos, ("girl is blonde") e relacionamentos ("girl holding racket"). O grounding associa cada objeto do grafo de cena a uma região da imagem. Fonte: Xu et al. (2020).

os vértices correspondem as entidades detectadas que na literatura também são definidas como sujeitos ou objetos e as arestas representam os relacionamentos entre estas entidades (Figura1.3).

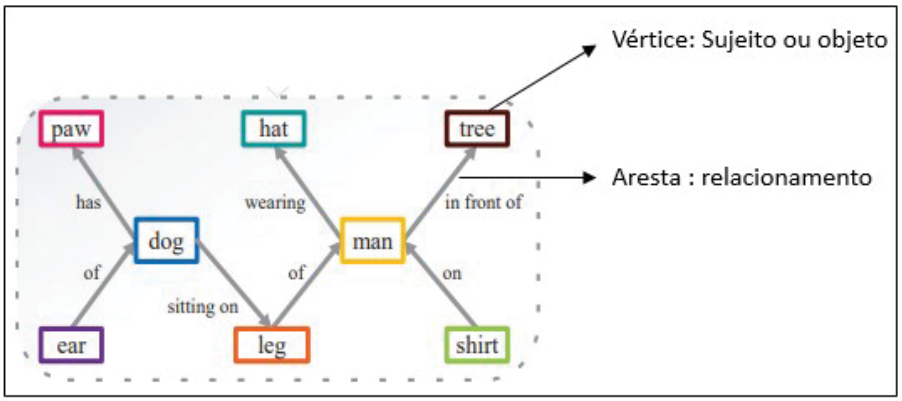


Figura 1.3: Representação de vértices e arestas de um de grafo de cena. Fonte: Adaptado de Lin et al. (2020).

1.1 DESCRIÇÃO DO PROBLEMA

Ao longo dos anos a comunidade de visão computacional tem alcançado grandes resultados em relação ao desenvolvimento de novos métodos de aprendizagem profunda que empregam o uso de CNNs (Krizhevsky et al., 2012). Entretanto, sabe-se que existem diversos problemas que envolvem estruturas irregulares dos quais as CNNs não conseguem descrevê-los, tais como: a identificação, representação estruturada dos padrões de propagação de "boatos/rumores"em redes sociais (Bian et al., 2020), recomendação e combinação de medicamentos a partir da análise estrutural de moléculas (Shang et al., 2019) e principalmente na classificação e interpretação semântica de uma cena visual, considerando todas as entidades envolvidas no contexto, por exemplo objetos, atributos, legendas e relacionamentos.

Diante ao grande desafio de representar dados não euclidianos, os estudos proporcionaram o crescente avanço na abordagem de redes neurais para grafos, habitualmente conhecidas na literatura como *Graph Neural Networks* (GNN) (Johnson et al., 2015), que visam a resolução de problemas em contexto genérico que envolvem o domínio irregular nas estruturas de dados e na realização de tarefas heterogêneas, escabilidade e paralelização de processos.

Dessa forma, com o uso crescente das GNNs em diversas áreas, houve um avanço significativo em relação aos métodos que permitem a representação das características irregulares que compõem uma imagem. Avanço que permitiu a elaboração de novas abordagens de representações semânticas denominada *Scene Graph Generation* - (SGG) (Xu et al., 2017; Yang et al., 2018; Zellers et al., 2018; Li et al., 2017; Qi et al., 2019a; Tang et al., 2020), em sua tradução para o português é definida como processo de geração de grafo de cena, que utiliza subprocessos para a detecção dos objetos existentes em uma imagem, classificação dos atributos, regiões e relacionamentos entre os pares de objetos.

Na Figura 1.4 podemos ver exemplos das etapas do processo extração, detecção e classificação dos tipos de informações que formam uma determinada imagem, por exemplo, a Figura 1.4 (a) ilustra a etapa de detecção de todos objetos que compõem a imagem; em seguida temos o processo de identificação da região (Figura 1.4 (b) responsável por definir o contexto do ambiente que é fundamental na associação dos atributos (Figura 1.4 (c)) consiste na representação de uma ação sendo realizada por um determinado sujeito (mulher), que resultará na classificação de um relacionamento (Figura 1.4 (d)) entre os pares de objetos detectados.

Contudo, o processo de geração de grafo de cena apresenta limitações para representar de forma assertiva o relacionamento entre os pares de objetos em imagens que apresentam um alto nível de complexidade contextual, devido a heterogeneidade de grafos (tamanhos) dificultando

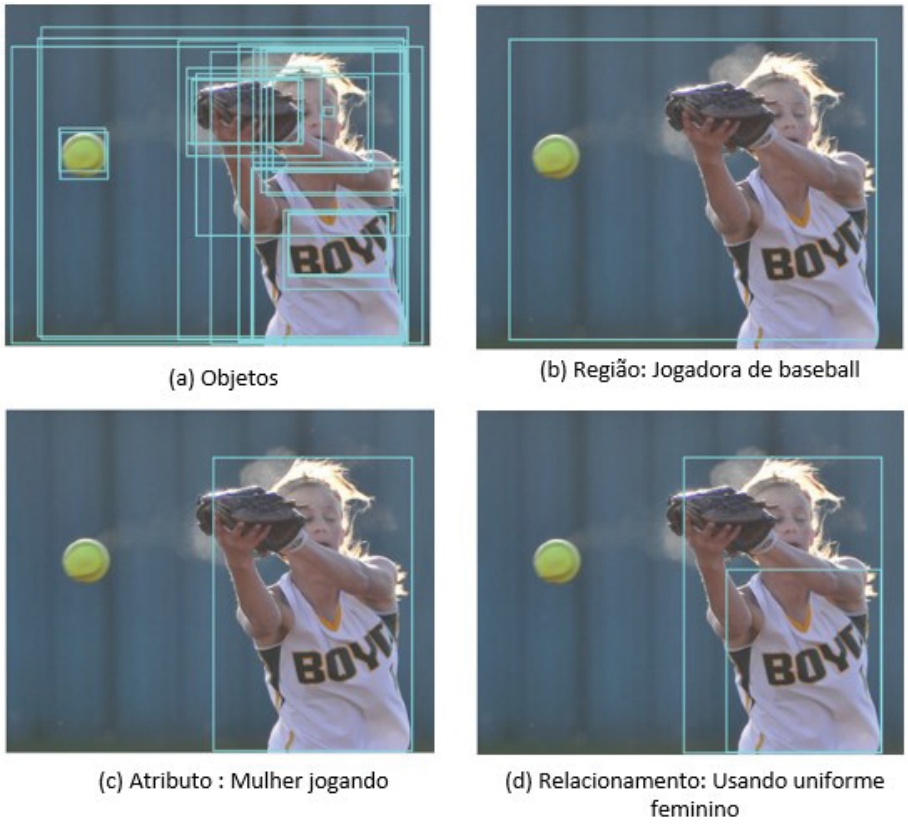


Figura 1.4: Tipos de informações de uma imagem. Fonte: *Dataset Visual Genome* (Krishna et al., 2017).

assim a tarefa de construir de forma dinâmica grafos, conforme o processo de atualização das informações entre os nós.

Diante do exposto, faz-se necessária a busca de alternativas para reduzir as variantes que interferem no processo de geração de grafo de cena em imagem de maior complexidade visual.

1.2 OBJETIVOS E METAS

O objetivo geral deste trabalho é apresentar o método UGP-Net (*Unbiased Graph Property Network for Scene Graph Generation*) que utiliza o teorema de causa e efeito na etapa de refinamento/atualização das características dos nós de um determinado grafo no processo de geração de grafo de cena, que seja capaz de inferir, compreender e classificar relações semânticas complexas de uma determinada cena visual disponíveis em conjuntos de imagens públicos tais como: *Visual Genome* (Krishna et al., 2017), *OpenImages* (Kuznetsova et al., 2020) e *Visual Relationship Detection* (Lu et al., 2016), que são formados por uma diversidade de anotações referentes a classes, atributos e relacionamentos entre pares de objetos que constituem as imagens dos respectivos conjunto de dados.

A abordagem visa empregar o teorema de causa e efeito no contexto de aprendizado profundo, por meio da combinação entre os métodos Causal-TDE (Tang et al., 2020) e GPS-Net (Lin et al., 2020) que tem como entre os principais objetivos a eliminação do víes em grandes conjuntos de dados, que se não for atenuado fará com que o modelo de SGG descreva eventualmente a cena visual incorretamente.

Todavia, os objetivos específicos desse trabalho são: a) realizar uma revisão bibliográfica do estado da arte na área de representações estruturais a partir de dados visuais exclusivamente no processo de geração de grafo de cena por meio de GNNs, que por sua vez, possui uma variedade

de arquiteturas de rede como: *Recurrent Graph Neural Networks*, *Graph Convolutional Networks*, *Graph AutoEncoders* e *Spatial-temporal Graph Neural Networks*; b) efetuar um estudo sobre os métodos de detecção de objetos, atributos e suas limitações; c) apresentar a nova abordagem de refinamento/atualização das características dos nós no processo de geração de grafo de cena que utilize toda a informação contextual e as propriedades presentes em uma imagem.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são descritos os conceitos usados para o desenvolvimento deste trabalho.

2.1 DEFINIÇÕES BÁSICAS

Em um grafo $G = (V, E)$ dizemos que uma aresta e é incidente a um vértice v se $v \in e$. Dois vértices u e v são adjacentes se existe uma aresta entre eles. Neste caso também podemos dizer que u é vizinho de v e vice-versa. Denotamos por $N(v)$ o conjunto de vértices que são vizinhos de v. Dado $S \subseteq V(G)$, definimos $N(S)$ como $\bigcap_{v \in S} N(v)$. O grau de um vértice é o número de arestas incidentes a ele. Se $|V(G)| = n$ existe uma aresta entre cada par de vértices G, dizemos que G é um grafo completo com n vértices, denotado por K_{n}.

Um grafo direcionado $G = (V, E)$ consiste de um conjunto de vértices V não vázio e de um conjunto de arestas direcionadas. Cada aresta está associada a um par ordenado de vértices.

Um caminho direcionado em um grafo direcionado é uma sequência de vértices consecutivos, tal que de cada vértice existe uma aresta para o próximo vértice da sequência, por exemplo: $S \rightarrow T \rightarrow Y \rightarrow Z$.

Dizemos que um ciclo é um caminho que inicia e termina no mesmo vértice. Um grafo que não possui ciclos em sua estrutura é definido como um grafo direcionado acíclico (DAG, do inglês *Directed Acyclic Graph*). Na Figura 2.1 temos um exemplo de grafo direcionado acíclico G no conjunto de vértices $A = \{S, T, W, X, Y, Z\}$ que consiste em um conjunto de pares ordenados de vértices

Um subgrafo de um grafo G é qualquer grafo H tal que $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Um subgrafo H de G é próprio se $V(H) \neq V(G)$.

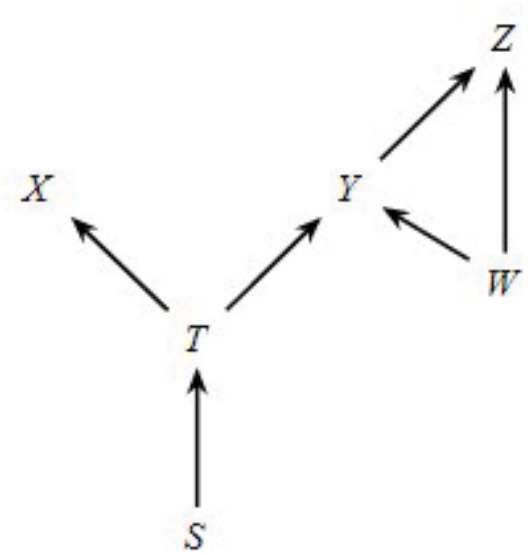


Figura 2.1: Grafo Acíclico Direcionado.

2.1.1 Relacionamento de Grafos

O relacionamento entre os grafos são frequentemente descritos usando a linguagem da geneologia. Na Figura 2.1, a variável X é pai de Y apenas se houver caminho direcionado

de X para Y ($X \rightarrow Y$). Logo, $PA(Y)$ denota o conjunto de todos os pais de Y, por exemplo, $PA(Y) = \{T, W\}$. X é um ancestral de Y e Y é um descendente de X apenas se houver um caminho direcionado de X para Y. Portanto, podemos definir que cada variável é um descendente de si mesma, por exemplo, na Figura 2.1 $DE(T)$ denota o conjunto de todos os descendente de T, onde $DE(T) = \{T, X, Y, Z\}$.

2.1.2 Grafos Causais

Na literatura define-se grafos causais sendo um componente da abordagem intitulada como *Structural Causal Model* (SCM) (Pearl, 2009) que visa inferir quais variáveis devem ser analisadas na resolução de problemas que envolvem grafos, por exemplo, dada uma imagem, determine o grafo de cena da mesma.

2.1.3 Grafos Direcionados Acíclicos (DAGs)

Um grafo direcionado acíclico (DAG) é definido como uma representação gráfica da estrutura de dependência de um vetor aleatório $\mathbf{W} := (W_1, ..., W_d)$ onde cada vértice representa uma variável (Figura 2.2). Em um grafo do tipo DAG, se há uma seta de W_i até W_j, dizemos que W_i é pai de W_j. Denotamos por $\pi(W_i)$ o conjunto de todos os pais do vértice W_i. Para que um DAG represente adequadamente a estrutura de dependência de $\mathbf{W}$, deve se aplicar:

$$p(\mathbf{w}) = \prod_{i=1}^{d} p(w_i|\pi(w_i)) \tag{2.1}$$

Um DAG permite que diversas independências condicionais sejam derivadas facilmente. Por exemplo, em um determinado DAG, temos que W_i é independente de **W'** condicionalmente em $\pi(W_i)$, onde **W'** representa todas as demais variáveis além de $W_i, \pi(W_i)$ e os descendentes de W_i.

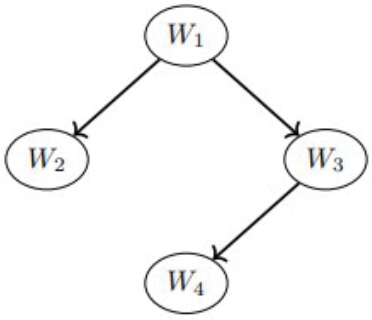


Figura 2.2: Exemplo do DAG.

2.2 INFERÊNCIA CAUSAL

No presente trabalho, a abordagem proposta por meio do método UGP-Net que será descrito no Capítulo 4 tem como base a aplicação do conceito de infererência causal (IC), cujo objetivo é inferir a presença e magnitude de causa e efeito a partir da análise de de informações extraídas de uma determinada imagem, por exemplo, uma variável X é a causa de uma variável Y se Y, de alguma forma, depende de X para definir o seu valor. Outro exemplo de causa e efeito é se X é uma causa de Y se Y ouve X e decide seu valor em resposta ao que ouve.

Segundo Pearl (Pearl, 2009) a inferência causal visa pois descobrir o grafo causal e os mecanismos causais associados, a partir de amostras da distribuição de probabilidade conjunta dos dados observacionais, em outras palavras a inferência causal consiste numa família de métodos que buscam responder o "porquê" das coisas acontecerem. Os métodos comuns como análise de regressão se preocupam em quantificar correlações, como mudanças em X estão associadas em mudanças em X. Já os métodos de (IC) buscam determinar "SE"mudanças em X causam mudanças em Y. Além disso, as abordagens de (IC) buscam responder perguntas do tipo "por quê" Y muda, ou seja, se X está causalmente relacionado com Y, então a mudança de Y pode ser explicada em termos da mudança de X.

Quando nos referimos a teoria dos modelos causais Pearl (Pearl, 2009) classifica a informação causal em termos do tipo de "pergunta"que cada classe é capaz de responder. A classificação forma uma Hierarquia de três níveis, sendo que perguntas do nível i ($i = 1, 2, 3$) só podem ser respondidas se as informações do nível $j \geq i$ estiverem disponíveis. Na Figura 2.3 é possível visualizar os três níveis de modelos causais que estão classificados em: (i) *Association*, (ii) *Intervention* e (iii) *Counterfactuals*.

Level (Symbol)	Typical Activity	Typical Questions	Examples
1. Association $P(y\|x)$	Seeing	What is? How would seeing X change my belief in Y?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $P(y\|do(x), z)$	Doing Intervening	What if? What if I do X?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $P(y_x\|x', y')$	Imagining, Retrospection	Why? Was it X that caused Y? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past 2 years?

Figura 2.3: Hierarquia dos modelos causais.

O método UGP-Net proposto no presente trabalho aplica o modelo causal *Counterfactuals* para a etapa de refinamento das característica que tem como objetivo final a geração do grafo de cena de imagens de baixa, média e alta complexidade semântica.

2.2.1 Modelo Contrafactual

Segundo Pearl (Pearl, 2009) para entender o que é um modelo contrafactual e como podemos ilustrar ele matematicamente, deve-se analisar o seguinte exemplo: suponhamos que, no caminho para casa, nos deparamos com uma bifurcação na estrada e temos que optar por uma das seguintes opções:

- ou seguimos pela autoestrada ($X = 1$), ou
- optamos pela estrada nacional ($X = 0$).

Suponhamos que optamos pela estrada nacional ($X = 0$), e, após chegar a casa, 1 (uma) hora depois, dizemos "Devia ter optado pela autoestrada!". Esta afirmação é muitas vezes usada para dizermos que *se tivesse tomado a outra opção, neste caso a autoestrada, teria chegado mais rapidamente a casa*. Isto é, mentalmente estimamos que o tempo médio que demoraríamos, se tivéssemos optado pela autoestrada, no mesmo dia, nas mesmas circunstâncias e com os mesmos

hábitos de condução, seria menor do que o tempo real que efetivamente demorámos ao ter optado por ir pela estrada nacional.

Portanto, este tipo de situação descrita acima no exemplo é aquilo a que chamamos de **contrafactual**. Ao acontecimento que não é verdadeiro, ou não é possível realizar sob as mesmas condições, definimos como **condição hipotética**. Quando usamos contrafactuais, o objetivo é comparar dois resultados, neste caso os tempos de condução, sob as mesmas condições, com exceção de apenas uma: a *condição hipotética* que, no nosso exemplo, se traduz em "ir pela autoestrada" em oposição a ir pela estrada nacional.

2.2.2 Definição Contrafactual

Definição 2.2.1. *Sejam* ***X*** *e* ***Y*** *dois subconjuntos de variáveis em V. A sentença Confractual "****Y seria y, na situação em que U = u, se X tivesse sido x****"é interpretada como sendo igualdade.*

$$Y_x(u) = y \tag{2.2}$$

onde $Y_x(u)$ é a resposta potencial definida anteriormente.
Consideremos o seguinte *Structural Causal Models* (SCM):

$$\begin{aligned} X &= U_X \\ Y &= X^2 + U_Y \\ Z &= 2Y + X + U_Z \end{aligned} \tag{2.3}$$

onde $U_X, U_Y, U_Z \sim U(-5, -4, ..., 4, 5)$ são uniformemente distribuídos nos inteiros entre -5 e 5. Agora assumimos que observamos $(X, Y, Z) = (1, 2, 4)$. Então $Z_x = x(U_X, U_Y, U_Z$ coloca uma massa em $U_X, U_Y, U_Z = (1, 1, -1$ pois o valor de todas as variáveis exógenas pode ser obtido de forma única através das observações. Portanto o contrafactual $Z_Z = 2$, que representa "o valor Z, na situação em que $U_X, U_Y, U_Z = (1, 1, -1)$, se X tivesse tomado o valor 2"é igual a 11.

Para ilustrar a definição 2.2.1 de confractatuais temos o modelo retratado na Figura 2.4. A variável X representa o de tempo de estudo individual de um aluno após a escola, H representa a quantidade de trabalhos escolares que o aluno realiza em casa e Y representa a nota final obtida pelo aluno no exame. O valor de cada variável é dado pelo número de desvio padrão acima da média, isto é, o modelo está inicializado para que todas as variáveis tenham média 0 e variância 1.

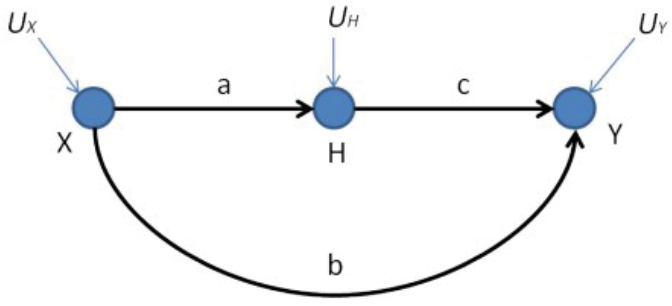


Figura 2.4: Modelo que relaciona o tempo de estudo fora das aulas com a nota final no exame.

O SCM associado é dado pelo modelo linear:

$$
\begin{aligned}
X &\leftarrow U_x \\
H &\leftarrow aX + U_H \\
Y &\leftarrow bX + cH + U_Y
\end{aligned}
\tag{2.4}
$$

Assume-se que todas as variáveis U são independentes e que os coeficientes do modelo foram estimados a partir de dados recolhidos da população: $a = 0.5$, $b = 0.7$ e $c = 0.4$.

Suponhamos que foram medidos para um estudante chamado Nuno, os seguintes valores: $X = 0.5, H = 1$ e $Y = 1.5$. Suponhamos que pretendemos responder à questão: *"Qual seria a nota Y que o Nuno obteria no exame final se ele tivesse duplicado o seu tempo de estudo H, fora das aulas?"*.

Em um SCM linear, como é o caso, o valor de cada variável é determinada pelos coeficientes e pelas variáveis exógenas U. Como sabemos os valores dos coeficientes e das variáveis X, H e Y, podemos determinar o valor das variáveis exógenas associadas ao Nuno. Como estas variáveis exógenas dependem apenas do valor que a natureza lhes atribui, elas são invariantes mesmo que tomemos uma ação hipotética, como é o caso de duplicar a quantidade de trabalhos fora das aulas. Portanto neste caso, as variáveis U (Figura 2.4):

$$
\begin{aligned}
U_X &= 0.5 \\
U_H &= 1 - 0.5x0.5 = 0.75 \\
U_Y &= 1.5 + 0.7x0.5 - 0.4x1 = 0.75
\end{aligned}
\tag{2.5}
$$

Em seguida, simulamos a ação de duplicar a quantidade de trabalhos que o Nuno realiza substituindo a equação de H pela constante $H = 2$ isto é, intervindo $\mathbf{do}(H = 2)$ o que se reflete na omissão das setas que incidem em H obtendo o modelo contrafractual modificado (Figura 2.5).

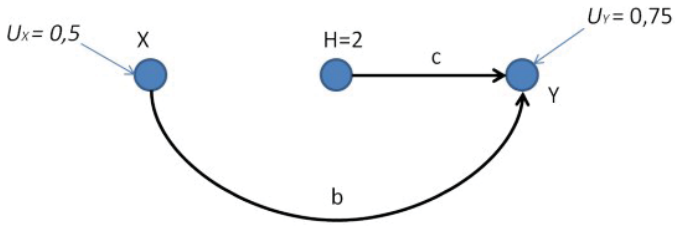


Figura 2.5: Resposta a um contrafactual relativo à nota obtida pelo estudante pressupondo que se aumentou $H = 2$.

Calculando a resposta potencial à ação $\mathbf{do}(H = 2)$ no modelo contrafractual, usando os valores das variáveis exógenas, anteriormente calculados, e obtemos:

$$
Y_{H=2}(U_X = 0.5, U_H = 0.75, U_Y = 0.75) = 0.70.5 + 0.42 + 0.75 = 1.90 \tag{2.6}
$$

Concluo-se que, se o Nuno tivesse duplicado a quantidade de trabalhos a serem realizados em casa, o resultado no exame teria sido1.9 em vez de ser 1.5. Isto significa que, que haveria um aumento de 1.9 desvios padrão acima da média, em vez dos atuais 1.5 desvios padrão.

2.3 REDES NEURAIS CONVOLUCIONAIS

Ao longo dos últimos anos, houve um crescimento significativo no número de trabalhos que incluem em sua abordagem modelos de aprendizado de máquina (*Machine Learning* - ML).

Dentre os modelos, destaca-se o uso de Redes Neurais Convolucionais, no inglês conhecidas como *Convolutional Neural Networks* que empregam o conceito de auto-aprendizagem que permite a criação de extratores de características. Segundo Lecun et al.,2015 as CNNs usam dados brutos para aprender representações, evitando assim, a dependência de extratores de características manuais.

Desse modo, as redes neurais convolucionais consistem em camadas convolucionais que utilizam o algoritmo de *backpropagation* no processo de aprendizagem dos parâmetros/filtros em cada camada. As CNNs têm sido caracterizadas por três propriedades básicas, conexões locais, o compartilhamento de peso e o *pooling* local, camada responsável por simplificar a informação da camada anterior. As duas primeiras propriedades permitem que o modelo aprenda os padrões visuais locais de maior importância com menos parâmetros ajustáveis em um modelo totalmente conectado, e a terceira prepara a rede para conter invariância à translação. Além disso, as redes neurais convolucionais foram desenvolvidas para processar dados em forma de múltiplas matrizes, e são tipicamente treinadas seguindo o modelo *end-to-end* e de forma supervisionada.

A Figura 2.6 exemplifica os estágios de uma CNN, a extração de característica e o processo de classificação. A camada de características é responsável pela extração das características (*features*) que tem a função de aprender características genéricas até um determinado ponto, e que apresentam relevância para a tarefa de destino, que visa a classificação de um conjunto de características. As camadas de extração são capazes de extrair características consideradas invariantes a rotação de formas bidimensionais por meio de uma arquitetura composta de camadas convolucionais que executam a função de filtros, ativações e agregações.

No segundo estágio, as camadas de classificação utilizam características locais e armazenadas em vetores de características para efetuar a classificação da entrada da rede (imagem). O objetivo do procedimento de aprendizado é encontrar conjuntos de matrizes que extraiam as características discriminativas relevantes para serem usadas na classificação de imagens. Dessa forma, a CNN é usada para extrair uma nova representação do conjunto de dados de destino, semelhante ao uso de um extrator de características para uma determinada entrada, o qual obtém uma representação vetorial para cada amostra.

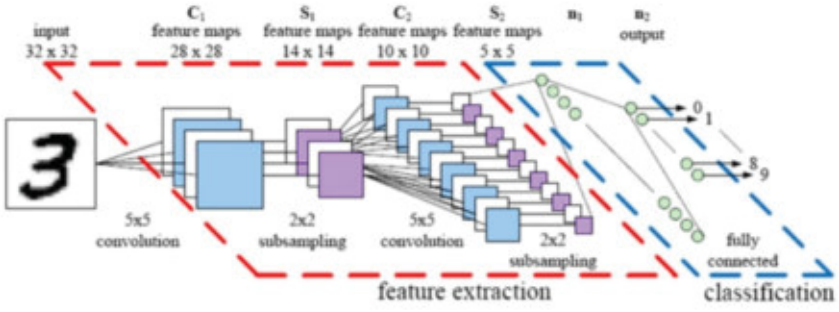


Figura 2.6: Representação dos estágios de uma Rede Neural Convolucional Simplificada. Fonte: Peemen et al. (2011).

2.4 REDES NEURAIS PARA GRAFOS

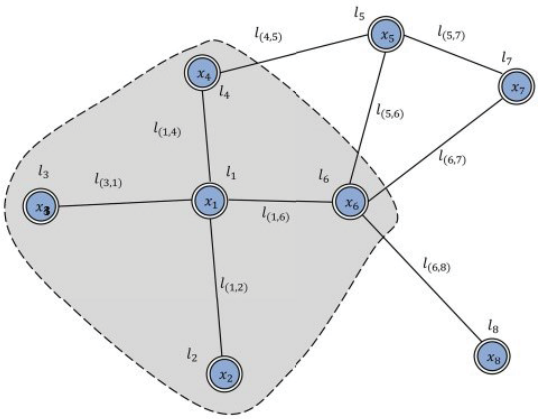
Redes neurais para grafos (*Graph Neural Networks* - GNNs) são arquiteturas de redes neurais profundas que recebem como entrada um conjunto de informações estruturadas no formato de grafo. Este tipo de redes neurais são utilizadas para representação de dados estruturais e irregulares, classificação ou geração de novos grafos, reconhecimento de padrões em conjunto de dados não euclidianos, agrupamento por semelhança de estruturas que podem ser representadas

por meio de grafos, entre outras tarefas. As aplicações são diversas, desde a composição e classificação de estruturas moleculares, até o processo de reconhecimento de padrões em redes sociais e principalmente na identificação e compreensão do contexto semântico dos objetos que compõem uma imagem.

Além disso, as GNNs são classificadas em quatro categorias: (1) *Recurrent Graph Neural Networks* (Li et al., 2016; Zhang et al., 2018; Zellers et al., 2018); (2) *Graph Convolutional Networks* (Bruna et al., 2013; Zhuang e Ma, 2018; Zitnik et al., 2018; Gao et al., 2018); (3) *Graph AutoEncoders* (Tian et al., 2014; Yang et al., 2019) e (4) *Spatial-temporal Graph Neural Networks* (Seo et al., 2018; Jain et al., 2016).

O conceito de redes neurais para grafos foi proposto pela primeira vez por Gori et al.,2005 e Scarselli et al.,2008, onde propuseram um modelo de GNN que consiste em uma generalização das redes neurais recursivas (RNNs), porém com a capacidade de resolver problemas que utilizam grafos como representação ou solução. As *Graph Neural Networks* formam um processo iterativo que propaga as características dos nós até o seu estado de equilíbrio (estado que indica que um determinado nó chegou ao número máximo de iteração no processo de atualização de suas características). O processo de atualização ocorre através das arestas, que são responsáveis em transmitir as características entre os nós que formam o grafo.

Segundo Scarselli et al.,2008 um nó é naturalmente definido pelas suas características e pelos nós relacionados no grafo. As GNNs tem a função de aprender um estado de representação (*embeddings*), que seja capaz de codificar a informação da vizinhança para cada nó. O estado de *embeddings* é utilizado para gerar a saída o_n, que consiste na classificação do rótulo do nó previsto. A Figura 2.7 ilustra um exemplo de grafo não direcionado utilizado no primeiro modelo de GNN proposto por Scarselli et al.,2008, na literatura conhecido como "modelo vanila"ou modelo inicial, onde o modelo define que os nós em um grafo representam os objetos ou conceitos (classes) e as arestas as suas relações. Cada conceito é definido pelo conjunto de características e relações. Assim, podemos dizer que $x_n \in \mathbb{R}^8$ a cada nó n que se baseia na informação contida na vizinhança de n.



$$x_1 = f(l_1, (\underbrace{l_{(1,2)}, l_{(3,1)}, l_{(1,4)}, l_{(6,1)}}_{l_{co[l]}}), (\underbrace{l_2, l_3, l_4, l_6}_{l_{ne[n]}}), (\underbrace{x_2, x_3, x_4, x_6}_{x_{ne[l]}}))$$

Figura 2.7: Modelo vanila da GNN - A variável x_1 depende da informação na vizinhança do nó 1 (região em cinza). Fonte: Scarselli et al.,2008.

Para o processo de atualização do estado de cada nó do grafo de entrada da GNN, aplica-se a função paramétrica f chamada função de transição local, que expressa a dependência

de um nó n em relação a sua vizinhança e definimos g como função paramétrica de saída local que descreve a geração do *output* de um grafo. Assim, definimos x_n e o_n da seguinte forma:

$$x_{\mathrm{n}} = f(l_{\mathrm{n}}, l_{\mathrm{co}[n]}, l_{ne[n]}, x_{ne[n]}) \tag{2.7}$$

$$o_n = g(x_n, l_n) \tag{2.8}$$

onde $l_{\mathrm{n}}, l_{\mathrm{co}[n]}, l_{ne[n]}, x_{ne[n]}$ são respectivamente o rótulo de n, os rótulos das arestas, os rótulos dos nós da vizinhança de n e os estados.

Na região destacada na Figura 2.7 temos o exemplo do conceito agregação de vizinhança aplicado nas GNNs discutido na seção 2.5, onde o nó l_1 é definido como nó referência, x_{l_1} é a variável de entrada das características de l_1 e $l_{\mathrm{co}[l]}$ são as arestas. Dizemos que $x_{ne[l]}$ são os nós vizinhos que irão compartilhar o vetor de características com l_1 para que seu estado seja atualizado.

Definimos x, o, l e l_{N} como sendo as matrizes de armazenamento de x_{n} referente a todos os estados, saídas, características gerais e características dos nós. Neste caso, as Equações 2.7 e 2.8 podem ser reescritas de uma forma mais simplificada:

$$x = F(x, l) \tag{2.9}$$

$$o = G(x, l_{\mathrm{N}}) \tag{2.10}$$

onde F é a função de transição global e G a função de saída global que corresponde as versões empilhadas de $|N|$ instâncias de f e g, respectivamente. A equação 2.9 demonstra um processo iterativo utilizado na etapa de atualização de estado de x em um determinado intervalo de tempo, onde um novo estado consiste na somatória dos estados obtidos na equação 2.7 mais o estado atual de x.

A Figura 2.8 mostra de forma simplificada o objetivo principal de uma GNN, onde a entrada consiste em um grafo direcionado e cada nó apresenta um vetor de características que será compartilhado com os nós vizinhos, conforme a direção das arestas.

Após a extração e refinamento das características de cada nó toma-se como saída a classificação dos nós que formam o grafo, que por sua vez, pode representar qualquer classe de objeto seja ela um número, uma molécula de um medicamento, uma pessoa, um carro, entre outras. O processo de classificação de cada nó, ocorre por meio do método conhecido como passagem de mensagem (*Message Passing*) que permite o envio das características por meio das aresta, abordagem que será detalhada na seção 2.6.

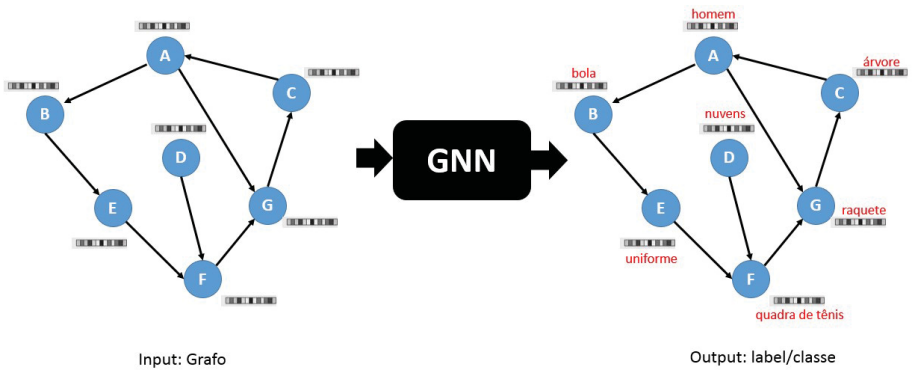


Figura 2.8: Exemplo de uma GNN. Fonte: Autor

Embora os resultados experimentais apresentados ao longo dos anos na literatura tenham evidenciado que a arquitetura da GNN é robusta na modelagem de dados estruturais, ainda há várias limitações do modelo inicial proposto, tais como:

- Em primeiro lugar, os modelos originais de GNN são ineficientes na atualização dos estados ocultos dos nós de forma iterativa para um único nó referência, denominado como estado de equilibrio do processo de passagem de características. O modelo necessita de **T** etapas de cálculo para se aproximar do ponto de equilíbrio.

- Em segundo, as GNNs utilizam os mesmos parâmetros na iteração, enquanto redes neurais mais populares (redes neurais convolucionais) usam parâmetros diferentes em camadas diferentes, permitindo uma extração de caracaterísticas de maneira hierárquica.

- Terceiro, existem características informativas nas arestas que não são possíveis de serem modeladas de forma eficaz no modelo vanila da GNN. Por exemplo, as arestas de grafos de conhecimento que apresentam tipos de relações e propagação de mensagens conforme o tipo de arestas, dificultando assim, o processo de aprendizagem do estado oculto das arestas.

Entretanto, para se obter uma atribuição precisa de valores que venham representar com maior acuracidade as características contidas em um determinado grafo no estágio de treinamento da rede é preciso aplicar a abordagem de Aprendizado Profundo ou *Deep Learning* (LeCun et al., 2015), juntamente com o método de Agregação de Vizinhança mais conhecido na literatura como Passagem de Mensagem (*Message Passing*).

2.5 APRENDIZADO PROFUNDO

Nos últimos anos inúmeras técnicas de aprendizagem profunda vêm sendo aplicadas em diferentes casos, tais como, identificação de objetos (Redmon et al., 2016), análise facial (Ranjan et al., 2017b), geração de grafo de cena (Tian et al., 2014; Yang et al., 2019), entre outros. Como consequência, a aprendizagem profunda (*Deep Learning*) vêm proporcionando grandes avanços na resolução de problemas que a Inteligência Artificial combinada com a Visão Computacinal não alcançaram por muitos anos.

Ao contrário dos métodos tradicionais que pré-definem descritores de características para representar dados de um determinado problema, a aprendizagem profunda busca descobrir uma estrutura intrínseca em grandes conjuntos de dados. Essa técnica utiliza algoritmos de representação da aprendizagem que possuem múltiplos níveis de representações, obtidos pela composição de modelos simples, os quais transformam a representação de um nível em outro nível mais alto (nível abstrato). Dessa forma, é possível representar os dados em múltiplos níveis de abstração, visto que os modelos são capazes de extrair características automaticamente.

2.6 AGREGAÇÃO DE VIZINHANÇA

O processo de aprendizagem das GNNs utiliza o conceito de agregação de vizinhança (*Neighbourhood Aggregation*), também conhecido como passagem de mensagem (*Message Passing*) método que consiste no envio de mensagens entre vértices vizinhos para um vértice referência por meio das arestas que os interligam. Entretanto, caso as arestas não apresentem direção, as mensagens podem ser enviadas para dois sentidos distintos como se existissem duas arestas direcionais em direções opostas.

Nesta etapa aplica-se as redes neurais *feed-forward* que podem utilizar as características da própria aresta se necessário. Nesse caso, diferentes tipos de arestas podem ser atribuídas a diferentes redes neurais *feed-forward* a fim de realizar um processamento diferente para cada tipo de aresta. Portanto, o principal objetivo do método de agregação de vizinhança é agregar todas as entradas (características) de todos os nós vizinhos do nó referência, e em seguida, calcular o valor do nó atual que corresponde ao novo vetor de características do respectivo nó, conforme ilustrado na Figura 2.9.

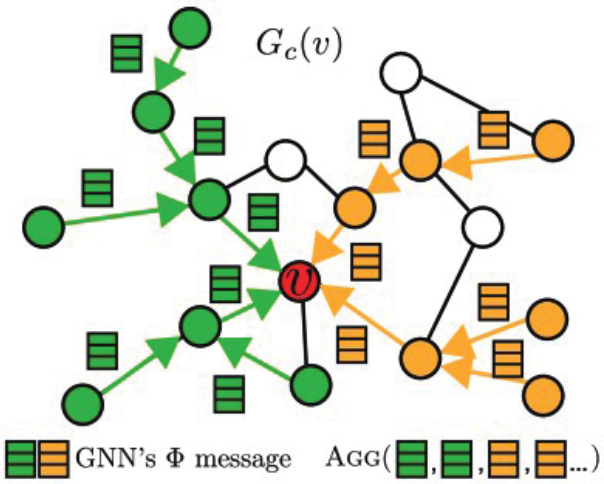


Figura 2.9: Processo de Agregação de Vizinhança da GNN $G_c(v)$. Fonte: Ying et al. (2019).

Na Figura 2.9 observamos o processo de agregação das características do nó v que compõem o grafo da GNN $G_c(v)$, onde destaca-se que determinadas arestas que formam caminhos considerados mais importantes (nós verdes) dos quais possuem características mais relevantes para o cálculo final do nó v, enquanto outras não (nós laranjas). Durante o processo de passagem de mensagens, cada nó é considerado como uma unidade recorrente e cada nó recebe as entradas dos nós vizinhos, conforme representado na Figura 2.9 que mostra a direção de cada aresta. Contudo, para calcular o nó, primeiro é necessário somar (agregar - AGG) todos os valores do nó vizinho e transferir o resultado para a função de cálculo $G(v)$.

A Figura 2.10 também ilustra o processo de agregação de vizinhança considerando apenas 3 nós, onde temos um nó referência (triângulo vermelho) que incorpora as características (envelope branco) dos seus respectivos vizinhos (triângulos de cor laranja) por meio das redes *feed-forward* (quadrado violeta), ou seja, após o envio das mensagens pelas arestas a representação é então atualizada na unidade recorrente, onde o vetor que representa o vértice em questão é atualizado de acordo com a função recorrente definida pela rede neural recorrente (Cho et al., 2014b).

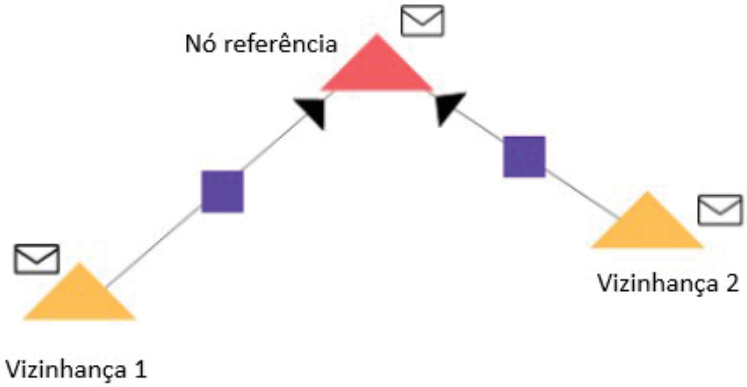


Figura 2.10: Representação do processo de Agregação de Vizinhança. Fonte: Adaptado de Xu et al.,2019.

O processo descrito acima é realizado para todos os vértices em paralelo. Com isso, a cada iteração a representação de um dado vértice sofre influência de uma vizinhança cada vez maior até que seja atingido o limite de vizinhança ou até que o nó sofra influência de todo o grafo.

Após a execução do processo de agregação de vizinhança, obtemos um novo conjunto de representações (*embeddings*) para cada unidade recorrente, onde ao realizar o cálculo de agregação das características obtém-se o vetor H que representa o estado final de um determinado nó.

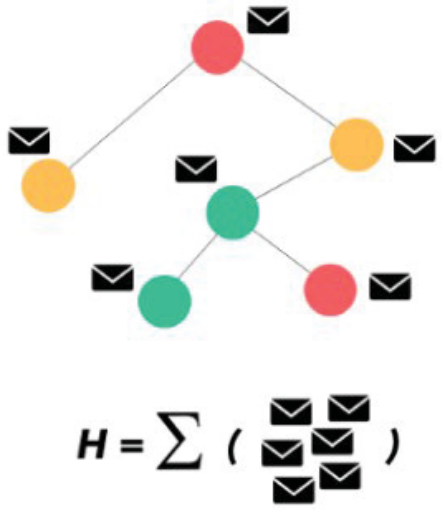


Figura 2.11: Representação do vetor H, após a somatoria dos vetores de características dos nós atualizados após n repetições do processo de passagem de mensagem. Fonte: Adaptado de Xu et al. (2017).

2.7 REDES NEURAIS CONVOLUCIONAIS DE GRAFOS

Redes Convolucionais de Grafos ou *Graph Convolutional Networks* (GCN) (Kipf e Welling, 2016) são consideradas redes variantes das GNNs. As GCNs realizam basicamente as mesmas operações das CNNs, ou seja, refere-se a multiplicação dos neurônios de entrada com um conjunto de pesos que são comumente definidos como filtros ou *kernels*. Os filtros atuam como janela deslizante em toda a imagem e permitem que as CNNs aprendam características de células vizinhas. Dentro da mesma camada, o mesmo filtro será usado para toda a imagem, isso é conhecido como compartilhamento de peso.

Apesar das redes neurais convolucionais de grafos realizam operações semelhantes as CNNs, onde o modelo aprende as características inspecionando os nós vizinhos a principal

diferença entre a CNN e a GCN é que as redes neurais convolucionais são especialmente construídas para operar em dados estruturados regulares (euclidianos), enquanto as GCNs consistem em uma versão generalizada das CNNs, onde o número de conexões de nós variam e os nós são desordenados (irregulares em estruturas não euclidianas).

A Figura 2.12 mostra a diferença entre CNN e GCN, onde na imagem da esquerda (convolução em 2D) a vizinhança é definida pelo tamanho do filtro espacial, a imagem da direita representa uma convolução em grafo e destaca que a vizinhança apresenta variação em tamanho e não é ordenada.

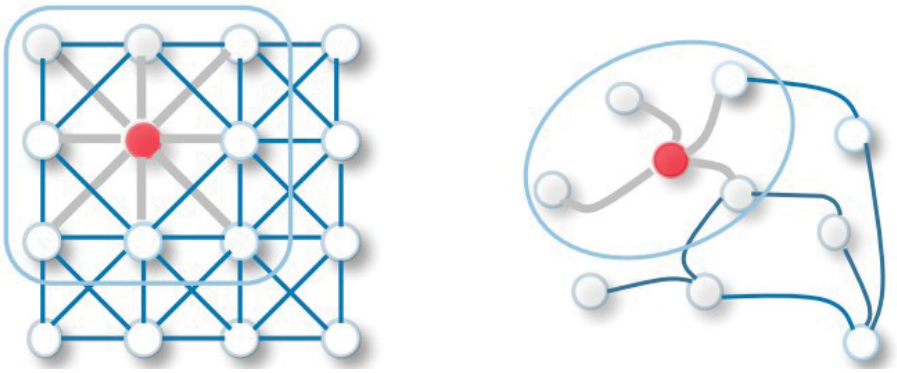


Figura 2.12: Convolução 2D vs Convolução de Grafo. Fonte: Wu et al. (2020).

As GCNs podem ser divididas em duas categorias, espectral e espacial. Na abordagem espectral o grafo possui uma representação no espectro e são aplicadas na classificação dos nós. Segundo Bruna et al.(2013), a convolução é definida no domínio de *Fourier*, calculando os autovetores e autovalores do grafo Laplaciano. Todavia, esse método resulta em cálculos potencialmente intensos e filtros não espacialmente localizados. Estes problemas foram abordados no trabalho de Henalff et al.(2015), onde propuseram o uso de parametrização de filtros espectrais como coeficiente suave o que pode torná-los espacialmente localizados.

A abordagem espacial consiste na realização da operação de convolução diretamente no nó. Entretanto, o maior problema desse método está no fato de definir um operador que consiga executar uma convolução de tamanhos distintos de vizinhança, e que consiga manter a propriedade de compartilhamento de pesos das redes. Duvenand et al.(2015) propuseram um método que visa o aprendizado de uma matriz de peso específica para cada grau de nó. Já Atwood e Toesley (2015) utilizam uma matriz de transição para definir a vizinhança da convolução enquanto aprendem os pesos para cada entrada e o grau de vizinhança.

2.8 GRAFO DE CENA

Um *Scene Graph* (SG), conforme definido por Johnson et al. (2015) trata-se de uma estrutura de dados representada através de um grafo que descreve o conteúdo de uma cena. Um grafo de cena tem como função codificar instâncias de objetos, atributos de objetos e o relacionamento entre os pares objetos.

Dado um conjunto C de classes de objetos, um conjunto $\mathcal{A}$ de tipos de atributos e um conjunto $\mathcal{R}$ de relacionamentos, definimos um grafo de cena G como uma tupla $G = (O, E)$, onde $O = (o_1, ..., o_n)$ é um conjunto de objetos e $E \subseteq O \times \mathcal{R} \times O$ é um conjunto de arestas e cada objeto possui a forma $o_i = (c_i, A_i)$ onde $c_i \in C$ é a classe de objetos e $A_i \in \mathcal{A}$ são os atributos dos objetos.

Portanto, um SG é uma representação estruturada de uma determinada imagem, onde os nós correspondem aos *bounding boxes* dos objetos juntamente com seus respectivos rótulos

definidos por meio do processo de classificação e as arestas correspondem as relações entre os pares de objetos. Em suma, a tarefa de geração de grafo de cena visa basicamente em gerar um grafo visualmente fundamentado de forma semântica, que ilustra as informações contextuais presentes na imagem, ou seja, consiste em distinguir de forma mais assídua um conjunto de dados estrutural.

A Figura 2.13 (a) mostra a representação visual de uma imagem de forma genérica a partir de uma único rótulo. No entanto, com o avanço dos detectores de objetos as imagens passaram a ser representadas como um conjunto de objetos, conforme ilustrado na Figura 2.13 (b). E, por último a Figura 2.14 destaca o avanço na área de visão computacional e das técnicas de aprendizagem profunda, onde uma imagem é representada semânticamente por meio de um grafo de cena.

(a) Imagem como um único rótulo (b) Imagem como um conjunto de objetos

Figura 2.13: Evolução da representação visual estruturada. Fonte:Johnson et al.(2015).

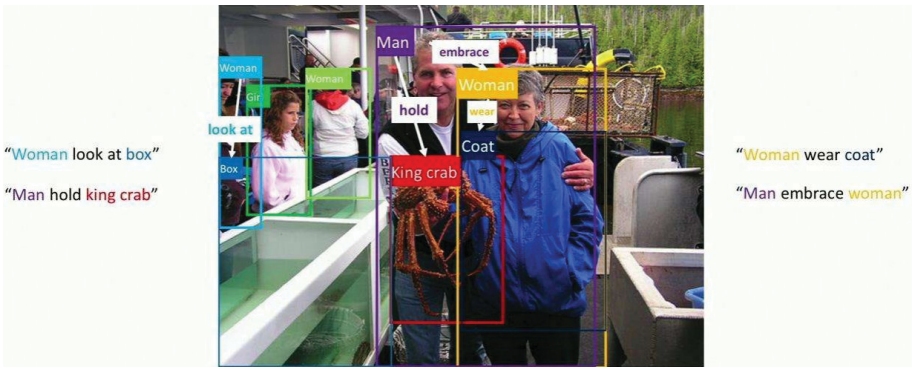


Figura 2.14: Grafo de Cena - Representação semântica. Fonte: Johnson et al.(2015).

Resumidamente, os métodos de geração de grafo de cena visam descrever semanticamente as imagens com maior precisão, ou seja, com um alto nível de detalhamento de tal maneira que possamos associar todas as entidades envolvidas na cena visual e seus relacionamentos.

2.9 PIPELINE GERAL DO PROCESSO DE GERAÇÃO DE GRAFO DE CENA

Com base na literatura revisada sobre a temática de geração de grafo de cena, observa-se que os métodos que serão apresentados e descutidos na seção 3.1 adotam um *pipeline* padrão de etapas que compõem os processos de geração de grafo de cena: (i) detecção dos objetos e extração das características; (ii) construção do grafo inicial; (iii) refinamento das características e (iv) geração do grafo de cena, etapas das quais resultam na representação semântica das

informações de uma imagem em forma de grafo que, por sua vez, ilustra todas as entidades envolvidas, atributos e relacionamentos.

Conforme ilustrado na Figura 4.1, o processo de geração de grafo de cena tem como etapa inicial a entrada de uma imagem, onde esta será a base para análise e extração das características (*features*) que serão representadas por meio de um grafo que descreverá a imagem semanticamente.

A etapa 1 (Figura 4.1 (a)) consiste na detecção de objetos e regiões que compõem a imagem, tal subprocesso visa a utilização de um detector de objetos genérico que possui inúmeras classes em seu modelo que possibilita a classificação de cada entidade detectada na imagem. Atualmente na literatura, temos uma diversidade de detectores tais como: YOLO (Redmon et al., 2016), Faster R-CNN (Ren et al., 2015), SPP-Net (He et al., 2015), SSD (Liu et al., 2016), entre outros. Entretanto, para o processo de geração de cena utilizamos como base o detector Faster R-CNN devido a abordagem RPN que será detalhada ao longo da seção.

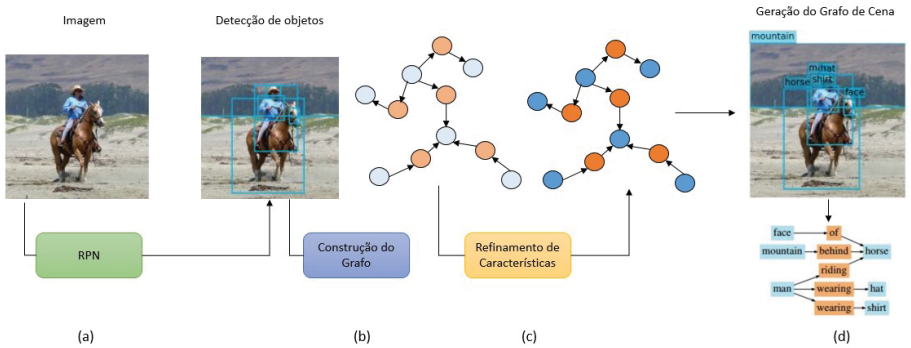


Figura 2.15: Pipeline geral do processo de geração de grafo de cena. (a) Dada uma imagem, o método RPN é utilizado para extração das propostas dos objetos junto com suas características, que são usadas no processo de grafo candidato (b) Após a geração dos primeiros grafos candidatos, o módulo de refinamento (c) é usado para refinar as características, um grafo de cena (d) é inferido de acordo com as característica do nó e da aresta. Fonte: Adaptado de Xu et al.(2017).

No entanto, os trabalhos descritos na seção 3.1 utilizam na fase de detecção de objetos o método proposto por Ren et al. (2015) que trata de uma melhoria da Fast R-CNN (Girshick et al., 2014) que tem como diferencial a adição da rede de Proposta de Região (*Region Proposal Network* - RPN) que permite que a rede compartilhe camadas convolucionais.

O primeiro módulo da Faster R-CNN é composto pela RPN que tem como função identificar a região do objeto de interesse na imagem de entrada e o segundo módulo é composto por uma rede que classifica os objetos que foram propostos anteriormente. A Faster R-CNN inicialmente processa a imagem de entrada usando uma *Convolutional Neural Networks* para extrair as características e obter os mapas de características que serão usados pela RPN. Em seguida, a RPN examina os mapas de características obtidos pela janela deslizante e calcula dois tipos de *score*. O primeiro indica se a janela em análise contém um objeto de interesse de acordo com o último mapa de característica da camada de convolução e o segundo cálculo verifica se o objeto é de interesse ou não.

Na execução do processo de janela deslizante, a detecção de objetos que apresentam formas heterogêneas e tamanhos variados tornam-se difíceis caso usado apenas uma janela de tamanho fixo. Todavia, para que a detecção de objetos de tamanhos variados seja possível, são usadas janelas deslizantes de tamanhos distintos, ou seja, K janelas, onde K corresponde ao número de janelas utilizadas no processo e, que por sua vez, são chamadas de caixas de âncoras (*anchors boxes*) e são geradas em diferentes escalas e proporções. Uma âncora consiste em uma região contida em uma janela deslizante que captura os objetos e prediz as suas coordenadas por

meio do ajuste das dimensões. Na configuração original da Faster R-CNN são geradas 9 caixas de âncoras (K = 9) que consiste em 3 tipos de escalas e 3 tipos de proporções definidas (Figura 2.16).

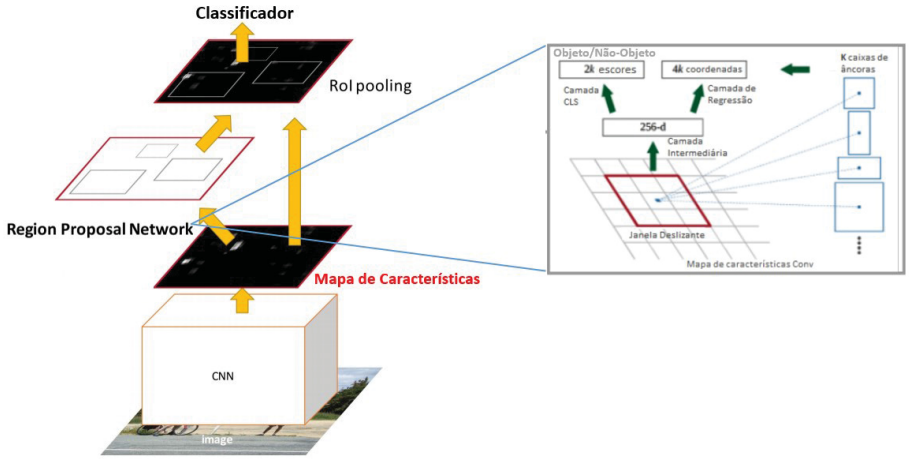


Figura 2.16: Faster R-CNN, à direita rede de proposta de regiões (RPN). Fonte: Ren et al.(2015).

A rede RPN classifica o objeto na âncora e se a intersecção sobre a União (IoU) da região for maior que scrore definido mno início do processo. O IoU corresponde a um limiar pré-definido no qual, um *bounding box* é avaliado como correto se a proporção da área de sobreposição entre o *bounding box* predito e o *bounding box* verdadeiro, e a área abrangida por ambos os *bounding boxes* é maior que o IoU positivo pré-definido (Figura 2.17). Entretanto, um *bounding box* é avaliado como não sendo um objeto se é menor que o IoU negativo parametrizado, neste cenário as regiões estão entre os dois limiares que não são considerados para o treinamento da rede.

O segundo módulo da Faster R-CNN é a própria Fast R-CNN (Girshick et al., 2014) que agrupa as regiões e encaminha para as camadas totalmente conectadas para que a classificação e a regressão possam ser realizadas. O modelo de treinamento proposto na Faster R-CNN é o *End-to-End* que treina a RPN e a Fast R-CNN em conjunto.

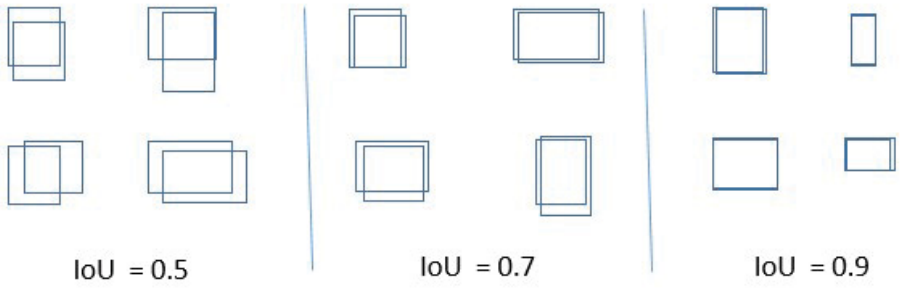


Figura 2.17: Exemplos de limiares para IoU. Fonte: Autor.

A Figura 4.1 (b) do pipeline abrange o processo de construção dos grafos, que consiste em iniciar a geração do relacionamento entre os pares de objetos com base nas características extraídas na etapa anterior. Para essa etapa utiliza uma camada de grafo totalmente conectada onde cada proposta de objeto é considerada como relacionamento a outra proposta de objeto. Uma vez que o grafo iniciado é criado com algumas relações candidatas entre os objetos detectados, inicia-se o módulo de refinamento 4.1 (c) que visa incorporar informações contextuais explícita

ou implicitamente para que o processo de detecção de objetos e de relacionamento se tornem mais dependente do contexto.

Além disso, o processo de refinamento de características tem como objetivo garantir que a regra $< objeto - predicativo - objeto >$ seja cumprida, pois uma vez que as características são refinadas, o grafo final é inferido, ou seja, nessa etapa todos os objetos candidatos que apresentam um *score* maior que IoU positivo e relacionamentos extraídos da etapa anterior são utilizados, e por meio de modelos probabilísticos, temporais, e matemáticos os grafos são refinados mantendo assim, apenas as características semânticas mais relevantes que resultam em um grafo de cena (Figura 4.1 (d)).

2.10 MÉTRICAS

Na construção de modelo de aprendizagem e predição, o desempenho do algoritmo está diretamente relacionado com a sua capacidade de recuperação e de classificação das informações de forma coerente, assim como na abstratação de informações erradas. Tal relação em um método de geração de grafo de cena é representada por meio de uma matriz confusão.

A matriz de confusão apresenta as predições realizadas por um determinado classificador, onde em sua diagonal principal estão contidos os acertos do classificador, enquanto que nas demais posições as confusões (erros) ocorridas (Visa et al., 2011). Em uma matriz confusão representada por 2 classes (0 e 1) (Figura 2.18), é possível classificar os dados em quatro indicadores:

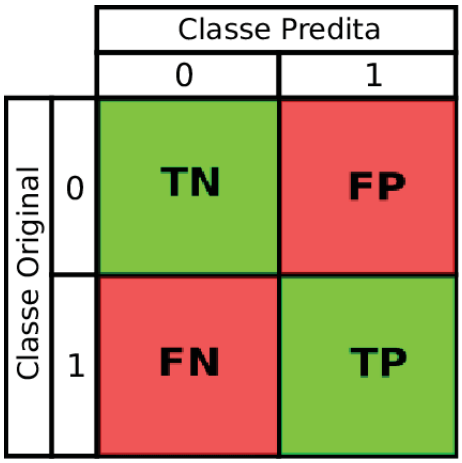


Figura 2.18: Exemplo de matriz confusão. Fonte: Autor.

- *True Positive* (TP): representa a proporção dos casos positivos que foram corretamente classificados.
- *False Negative* (FN): corresponde a proporção de positivos dos casos que foram incorretamente classificados como negativos.
- *False Positive* (FP): corresponde a proporção de casos negativos que foram classificados incorretamente como postivo.
- *True Negative* (TN): representa a proporção de casos negativos que foram classificados corretamente.

A partir da matriz de confusão e seus indicadores, o modelo de grafo de cena pode ser classificado pela sensibilidade/*recall*.

O *Recall* relacionada a porcentagem de instâncias classificadass corretamente como positivas (TP), em relação a todos as instâncias realmente positivas (TP + FN). A equação 2.11 denota a relação da métrica *recall*.

$$Recall = \frac{TP}{TP + FN} \tag{2.11}$$

A métrica para avaliar a detecção dos objetos e regiões das imagens é definida como IoU, definida pela Equação 2.12:

$$IoU = \frac{TP}{TP + FP + FN} \tag{2.12}$$

2.11 DATASET DE IMAGENS

Como qualquer algoritmo de detecção de objetos e de relacionamento visual, o processo de geração de grafo de cena requer uma grande quantidade de dados em sua etapa de treinamento.

Portanto, para atender tal necessidade descrita anteriormente a literatura destaca o conjunto de dados *Visual Genome* (Krishna et al., 2017) construído especificamente para a tarefa de SGG. Contudo, há outros conjuntos de dados que são utilizados para tal finalidade, tais como: *OpenImages* (Kuznetsova et al., 2020) e *Visual Relationship Detection* (Lu et al., 2016).

2.11.1 Visual Genome

O conjunto de imagens *Visual Genome* (Krishna et al., 2017) é composto por 108.248 imagens, 75.000 categorias de objetos e 37.000 categorias de predicados. A *Visual Genome* contém 7 componentes principais para cada imagem:

- **Descrição de região:** O conjunto de dados Visual Genome contém em média 42 descrições de região por imagem, visando descrever a imagem de uma maneira mais abrangente.
- **Objetos e *bounding boxes*:** A base de dados possui mais de 17.000 anotações de categorias semânticas com cerca de 21 objetos por imagem e totalmente anotados.
- **Atributos:** Em média são 16 atributos específicos do objeto por imagem que permitem uma classificação e descrição mais assertiva dos objetos. Além disso, são 40.513 atributos únicos e 1.670.180 instâncias de atributos.
- **Relacionamentos:** As imagens do conjunto de dados possuem em média 18 relacionamentos por imagem, que consistem em 40.480 relacionamentos únicos e 1.531.448 instâncias de relacionamento entre pares de objetos.
- **Pares de perguntas e respostas:** São 1.7 milhões de *QA's*, onde cada imagem possui pelo menos uma pergunta de cada um dos 6 tipos que compõem a base: o quê, onde, como, quando, quem e por quê.
- **Grafos de Região:** O conjunto de dados contém um representação em forma de grafo para cada 42 regiões que compóem cada imagem, tendo os objetos, atributos e relacionamentos como nós. Totalizando 3.788,7 regiões, que resultam em 4.297.502 descrições de regiões.

- **Grafos de cena:** São 108.249 grafos de cena

A Figura 2.19 ilustra as categorias de anotações do conjunto de imagens *Visual Genome* em relação a cada imagem.

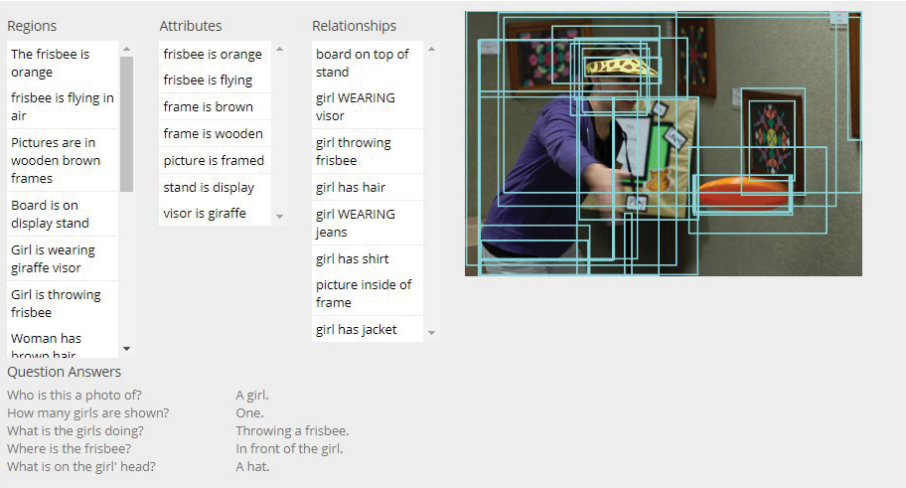


Figura 2.19: Exemplos de anotações no conjunto de imagens Visual Genome. Fonte: Visual Genome Dataset (2017).

2.11.2 OpenImages

O conjunto de dados *OpenImages* (Kuznetsova et al., 2020) é composto por 9.2 milhões de imagens com anotações que permitem a classificação de imagens, detecções de objetos e de relacionamento visual entre pares de objetos. O *OpenImages* oferece 600 classes de objetos, 375.000 anotações de relacionamento compreendendo 57 classes, 15.4 milhões de anotações de *bouding boxes* e 30.1 milhões de rótulos semânticos para 19.8 milhões de contextos.

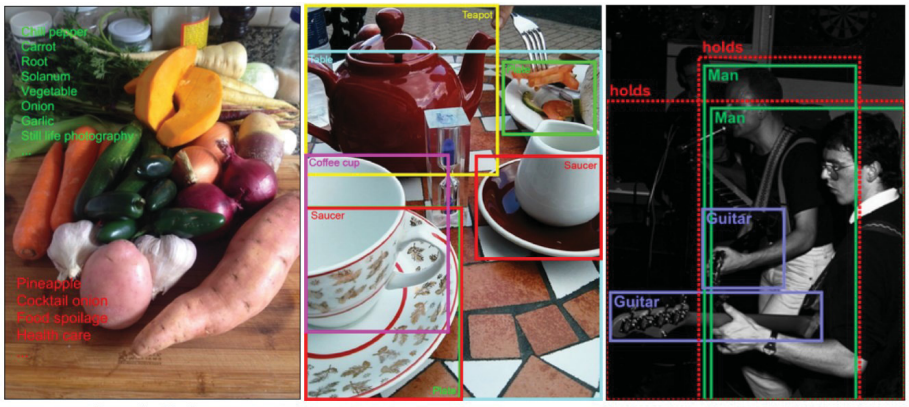


Figura 2.20: Exemplos de anotações no conjunto de imagens OpenImagens. Fonte: (Kuznetsova et al., 2020).

2.11.3 Visual Relationship Detection

O conjunto de imagens *Visual Relationship Detection* (VRD) (Lu et al., 2016) contém 5.000 imagens com 37.993 anotações de relacionamentos. A *Visual Relationship Detection* possui 100 categorias de objetos e 70 categorias de predicados que conectam os objetos, sendo que os predicados são classificados em ação, espaço, preposição, comparativo e verbo.

Figura 2.21: Exemplos de imagens do dataset VRD. Fonte: (Lu et al., 2016).

3 REVISÃO BIBLIOGRÁFICA

Neste capítulo estão descritos os métodos existentes na literatura, que têm como objetivo de resolver os problemas correlacionados a geração de grafo de cena e foram utilizados como base para a elaboração deste trabalho.

3.1 MÉTODOS DE GERAÇÃO DE GRAFO DE CENA

Nos últimos anos a literatura apresentou inúmeros trabalhos que visam encontrar a melhor abordagem de geração de grafo de cena a partir de uma imagem, com o objetivo de exibir de forma coesa um grafo que represente estruturalmente o relacionamento semântico dos objetos que compõem a cena, principalmente em cenas visuais que apresentam um alto nível de complexidade e uma heterogeneidade de informações.

Xu et al. (2017) propôs um novo modelo *end-to-end* para o processo de aprendizagem de geração de grafo de cena baseado em imagem. Dada uma imagem como entrada, seu modelo primeiramente produz um conjunto de propostas de objetos usando uma Rede de Proposta de Região (RPN) e, em seguida, transfere as características extraídas das regiões de objetos para uma formulação de inferência de grafo que refina iterativamente suas características por meio da técnica de passagem de mensagens contextuais ao longo da estrutura topológica do grafo de cena.

A principal contribuição desta abordagem está na introdução de um modelo de ponta a ponta que aperfeiçoa iterativamente a previsão de relacionamentos e objetos por meio da passagem de mensagens baseada em redes neurais recorrentes (RNN) (Li et al., 2016; Zhang et al., 2018) que ao contrário das redes neurais *feed-forward* possuem ciclos entre suas unidades, ou seja, contém conexões (*loops*) com unidades de camadas anteriores ou da mesma camada, que permitem consumir suas próprias saídas após uma determinada entrada. As RNNs permitem que uma informação sequencial seja mantida no estado oculto da rede recorrente, independente do número de etapas.

Entre os trabalhos destaca-se o método proposto por Li et al.(2017) propuseram o método MSDN (*Multi-Level Scene Description Network*) que explora as associações semânticas entre as três tarefas que contemplam a abordagem, tais como: (i) detecção de objetos; (ii) SGG e (ii) geração de legendas para as imagens analisadas, permitindo assim uma melhoria mútua do aprendizado conjunto das respectivas tarefas. Segundo os autores, legendas por região também podem fornecer um contexto útil para geração de grafo de cena, tornando mais precisa a identificação e compreensão dos relacionamentos entre os pares de objetos. Em sua abordagem as características para as três tarefas são altamente correlacionadas e podem atuar como informações complementares entre si.

O método é composto em 4 etapas (Figura 3.1): (1) detecção dos objetos, frases e regiões; (2) construção dinâmica de grafos; (3) refinamento das características e (4) geração de grafo de cena e legenda por região. A etapa 1 é composta por 3 tipos de propostas de detecção, que são definidas em: (i) proposta de região e objeto, que são geradas usando a Rede de Proposta de Região (Ren et al., 2015); (ii) propostas de região de frases, que visa o agrupamento de N propostas de objetos em $N(N-1)$ pares de objetos que conectam as propostas de objetos que possuem arestas direcionadas e (iii) propostas de região de legenda, geradas a partir de outra rede RPN.

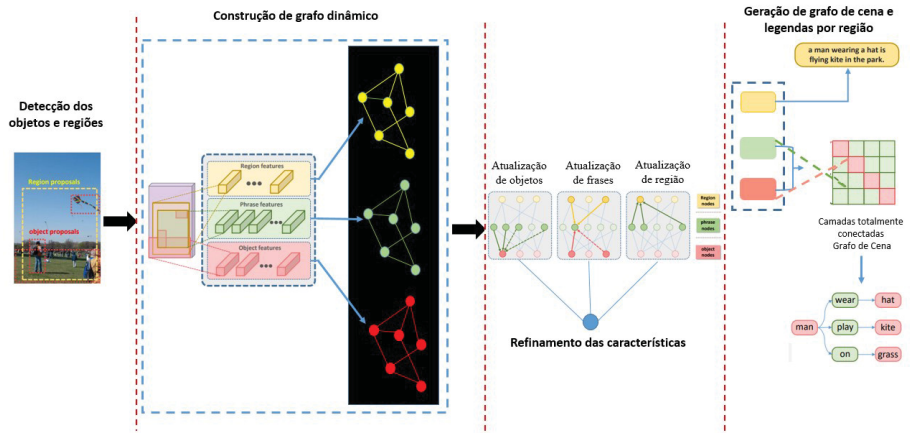


Figura 3.1: Etapas do MSDN. Fonte: Adaptado de Li et al. (Li et al., 2017).

A etapa 2 visa a construção dinâmica dos grafos com base nas características extraídas dos objetos, regiões e frases, ou seja, a rede recebe diferentes tipos de imagens como entrada, assim o grafo de conexão é construído dinamicamente com base nas relações semânticas e espaciais entre os ROIs (*Region of interest*), que correspondem a uma região de interesse proposta a partir da imagem original que nesse contexto fazem referência aos objetos, frases e regiões detectados na imagem. As conexões entre as frases e os objetos são formadas naturalmente durante o processo de geração de propostas de frases, onde cada proposta de frase é conectada a duas propostas/candidatos de objeto como um trio (sujeito-predicado-objeto).

A Figura 3.2 mostra um exemplo prático das etapas 1 e 2 do método MSDN, onde a rede de descrição de cena multinível recebe uma imagem como entrada e, na etapa 1 (Figura 3.2 (1)) todas as propostas de objetos, frases e regiões são detectadas pela rede RPN. No processo de construção do grafo dinâmico (Figura 3.2 (2)) as características dos ROIs referente as propostas são extraídas e o grafo inicial da cenal visual começa a ser construído.

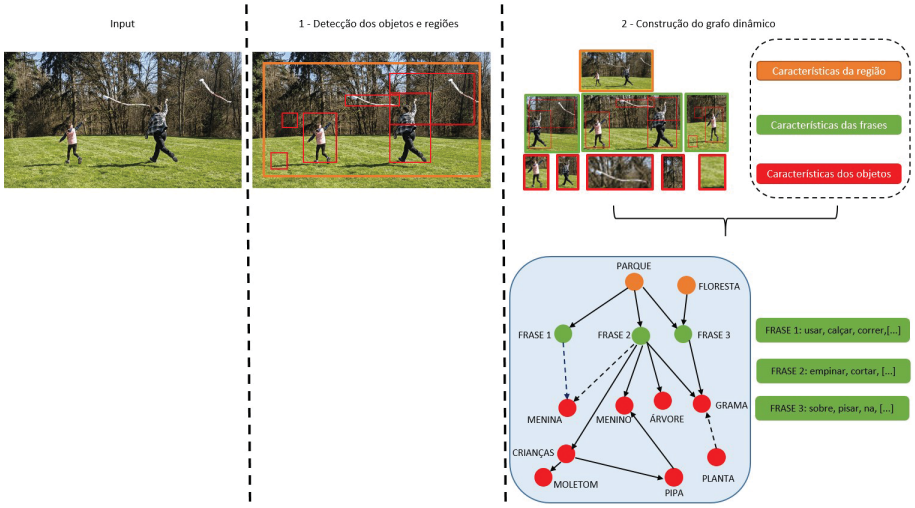


Figura 3.2: Etapas 1 e 2 do método MSDN. Fonte: Autor

Observa-se que os primeiros nós do grafo correspondem a propostas de regiões da imagem (parque e floresta), que será associada aos demais nós que fazem referência as frases (nós verdes) que são responsáveis em conter os predicados que irão compor o relacionamento entre os pares de objetos (nós vermelhos) que são classificados como sujeitos (menina, menino e crianças) e objetos (pipa, moletom, árvore, planta e grama).

O processo de refinamento das características dos nós do grafo, contemplado na etapa 3 é realizado após as primeiras conexões criadas entre os diferentes níveis de nós, as características são transferidas por meio da passagem de mensagens através das arestas do grafo. O procedimento de refinamento visa descartar em cada iteratividade as características que apresentam baixo *score* e mantém apenas as informações consideradas relavantes para a classificação dos nós que representam os objetos, regiões e frases. Nesta etapa do processo, para cada nó de objeto haverá dois tipos de conexões, sujeito-predicado e predicado-sujeito.

A etapa 4 consiste na geração do grafo de cena final, onde as categorias de relacionamentos são previstas diretamente com base nas características e frases. No entanto, também são geradas regiões de legendas que possuem uma ampla variedade de informações relacionadas a cena, sendo assim, necessário treinar um modelo de linguagem baseado em uma rede LSTM (Johnson et al., 2016; Karpathy e Fei-Fei, 2015) para geração das frases que descrevem uma respectiva região. A chave principal do método MSDN está construção dinâmica de grafos a partir da extração das características dos objetos, regiões e legendas.

A Figura 3.3 (3) mostra o processo de refinamento das regiões, frases e objetos em duas etapas de iteratividade, onde na primeira iteração do processo existem duas propostas de regiões que estão interligadas a 3 vetores de frases (nós verdes) que possuem um conjunto de predicados verbais que serão utilizados para interligar os pares de objetos (sujeitos e objetos). Já na segunda iteração nota-se que uma redução no números de nós que representam regiões e objetos, comportamento causado pela atualização das características de cada nó que visa manter apenas os nós que possuem um *score* elevado no vetor de características.

Como resultado final (Figura 3.3 (4)) da abordagem MSDN observa-se a construção do grafo de cena direcionado que contém o trio (sujeito-predicado-objeto), onde as arestas convergem para um determinado nó permitindo assim classificar semanticamente os pares dos objeto que por sua vez, também são representados por meio de legendas.

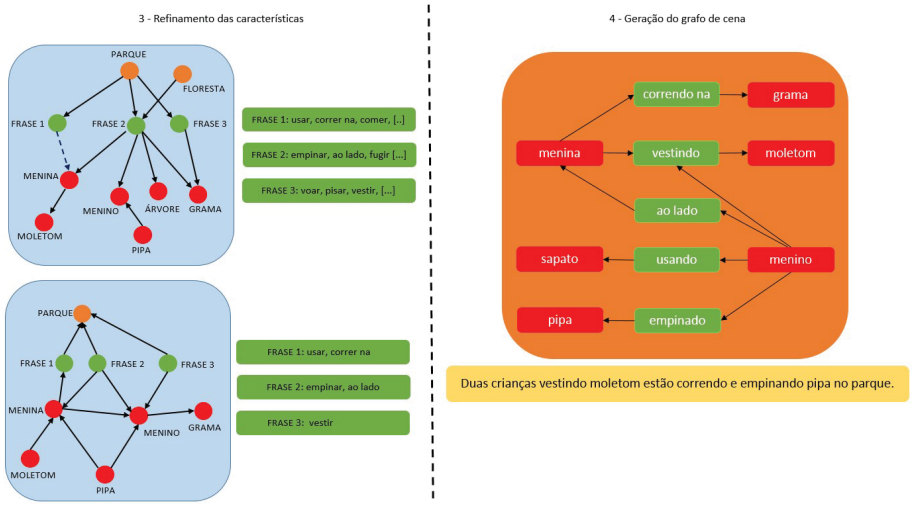


Figura 3.3: Etapas 3 e 4 do método MSDN. Fonte: Autor

Zellers et al.(2018) apresentam em seu trabalho uma nova rede neural denominada *Stacked Motif Network* (MOTIFNET) cujo modelo realiza a transferência de informações contextuais baseado em redes neurais recorrentes (LSTM) que por sua vez, calcula a frequência de co-corrência de pares de objetos e suas subsestruturas. A abordagem tem como característica principal o conceito de frequência *priori* que consiste em utilizar as características do processo de codificação do contexto global de um estágio anterior e empregar o mesmo para prever os

estágios subsequentes. Ou seja, em cada estágio o contexto global é calculado usando do tipo LSTMs (Hochreiter e Schmidhuber, 1997) bidirecionais utilizadas para prever o próximo estágio.

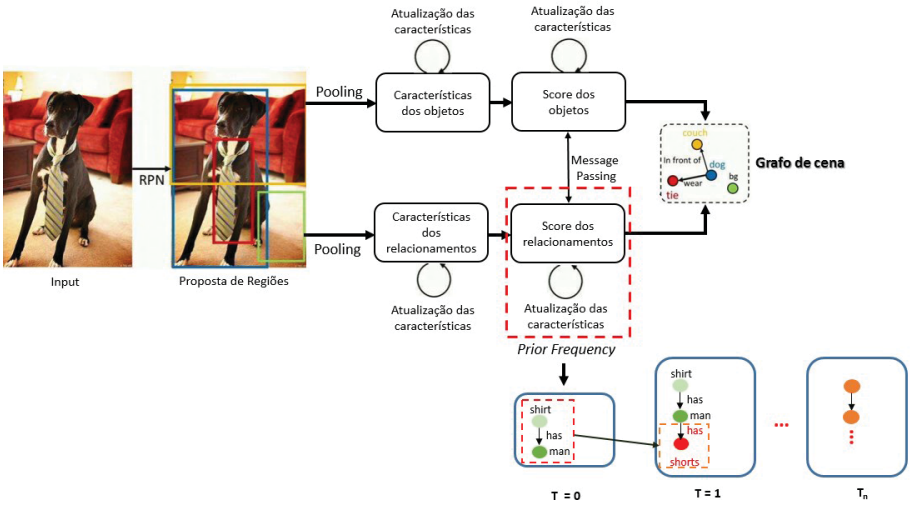


Figura 3.4: Pipeline - Rede neural MOTIFNET. Fonte: Adaptado de Zellers et al. (2018).

Yang et al.(2018) em seu trabalho destaca fortemente o conceito de grafo de cena e sua respectiva finalidade, que consiste na representação estruturada e semântica de uma imagem e ressalta o desafio de gerar esse tipo de estrutura a partir de uma cena visual complexa. Diante ao desafio de gerar uma representação estrutural de imagens complexas Yang et al.(2018) propuseram um framework de geração de grafo de cena denominado *Graph R-CNN* (Figura 3.5) que utiliza uma rede candidata de relacionamento e uma rede convolucional de grafo para construir o grafo semântico totalmente conectado gerado no início do processo, que permite a geração de um grafo de cena mais preciso.

O método apresenta 4 (quatro) contribuições, tais como: (1) uma nova rede *Relation Proposal Network* (RePN) que em sua tradução para o português significa "Rede de Proposta de Relacionamentos" que trata de forma eficiente as potenciais relações entre os objetos, sujeitos e predicados em uma imagem; (2) aGCN (*Attentional Graph Convolutional Networks*) cuja função é capturar e integrar as informações contextuais entre os objetos; (3) SGGEN+ uma nova métrica de avaliação da representação estrutural de uma imagem e (4) *Graph R-CNN Framework*, que consiste na apresentação de um novo framework de geração de grafo de cena, que atribui um *score* para cada nó vizinho, de tal forma que no final apenas prevaleça as características mais relevantes.

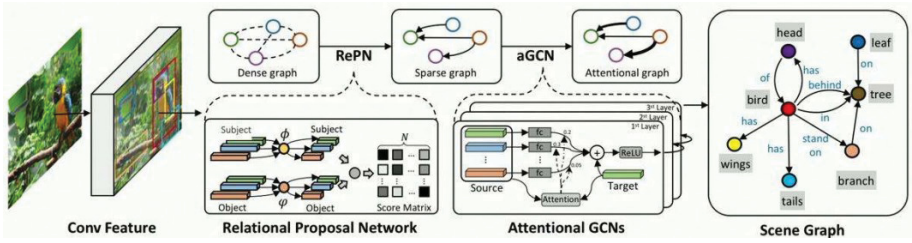


Figura 3.5: *Framework Graph R-CNN*. Dada uma imagem de entrada, o modelo aplica primeiramente o RPN para propor regiões de objetos, em seguida o RePN "estreita"as conexões entre os objetos e as regiões. Em seguida, a rede aGCN é executada para integrar as informações contextuais localizadas nos nós vizinhos no grafo de cena que está em processo de construção. Por último, o grafo de cena é obtido. Fonte: Yang et al. (2018).

O diferencial do método proposto em relação a abordagem apresentada por Li et al.(2017), está em relação o desenvolvimento das redes RePN que juntamente com a camada de *pooling* responsável por simplificar a informação da camada anterior, que permite a geração de uma amostragem aleatória e uniforme de possíveis arestas entre os vértices que possuem maior representatividade em relação aos relacionamentos entre os objetos. Além disso, o framework proposto permite que os *scores* dos objetos e relacionamentos sejam atualizados durante o processo de iteratividade da etapa anterior, por meio da rede aGCN.

Com base no pipeline geral ilustrado pela Figura 4.1, podemos realizar um comparativo entre os métodos de geração de grafo de cena (Li et al., 2017), (Zellers et al., 2018) e (Yang et al., 2018), exemplificados pelas Figuras 3.4, 3.6 e 3.7. Nota-se que ambos os métodos utilizam o método Faster R-CNN (Ren et al., 2015) em sua etapa inicial. Contudo, observa-se que o pipeline proposto por Yang et al.(2018), apresenta um processo de atualização em todo o processo, permitindo que a escolha das características e dos relacionamentos sejam realizados de forma robusta e recorrente ao longo do processo.

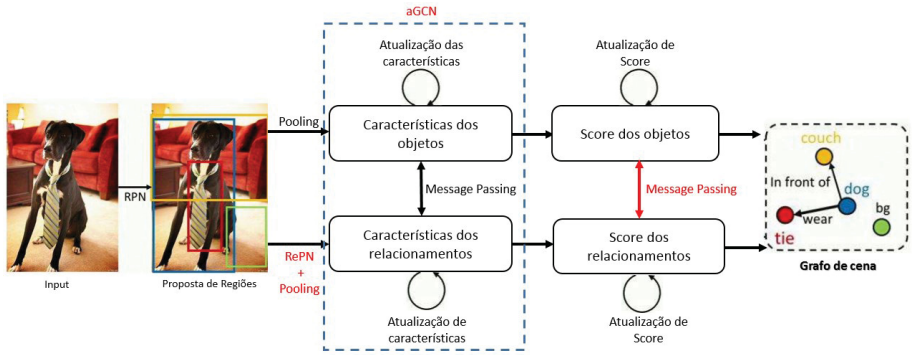


Figura 3.6: Pipeline - *Framework Graph R-CNN*. Fonte: Adaptado de Yang et al.(2018).

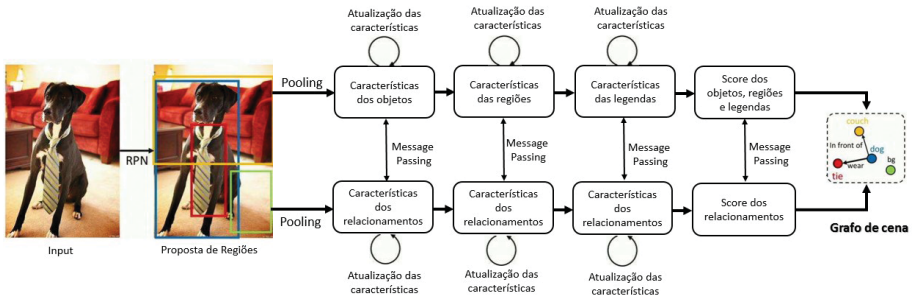


Figura 3.7: Pipeline - MSDN. Fonte: Adaptado de Li et al.(2017).

Qi et al.(2019) propuseram uma nova Rede Relacional Atenta (*Attentive Relational Network*) que traduz a informação visual em uma representação estruturada em grafo, agregando em sua arquitetura dois módulos principais de aprendizagem, sendo um deles que consiste na transformação semântica, que visa incorporar características de relação, entidade e conhecimento linguístico, mapeando simultaneamente palavras e características visuais que pertencem a um espaço comum e, o módulo de auto atenção que analisa a representação conjunta de grafos especificando implicitamente diferentes pesos para diferentes nós vizinhos.

Em uma visão geral o modelo (Figura 3.8) proposto é formado em quatro partes: (1) módulo de detecção de objetos, responsável pela captura das características visuais e a localização dos *bounding box* das entidades juntamente com os *bounding box* da relação de pares. Estas informações são utilizadas pela função *sofmax* para obter a classificação inicial de

cada entidade e relação; (2) módulo de transformação semântica, que produz representações semânticas incorporadas, transformando *embeddings* de rótulos e características visuais em um espaço semântico comum; (3) módulo de auto atenção de grafos, visa a construção de uma matriz de adjacência de entidades baseada na posição espacial de cada nó; (4) módulo de inferência de relação, que é responsável por criar a representação global do grafo e prever os rótulos de entidades e relacionamentos

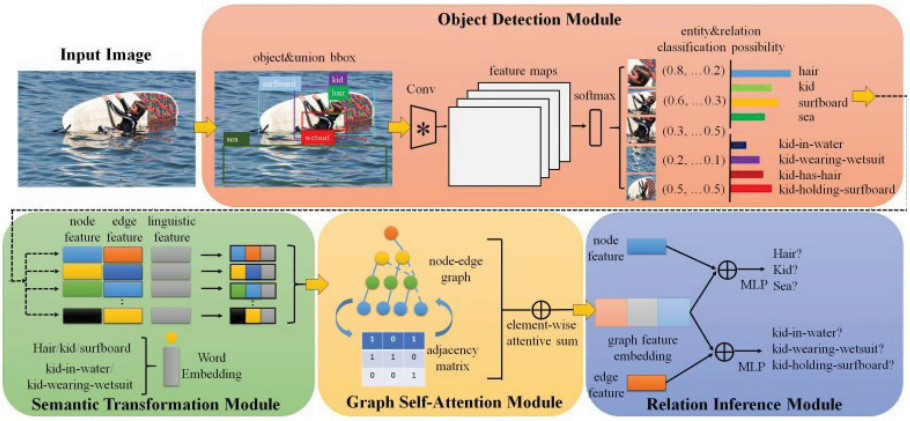


Figura 3.8: Visão geral do método Attentive Relational Network. Fonte: Qi et al.(2019).

Contudo, ao contrário da maioria dos métodos de SGGs que praticamente ignoram totalmente as propriedades dos grafos de cena, por exemplo direção das arestas e prioridade dos nós que compõem um determinado grafo, Li et al.(2020) propuseram uma nova abordagem e rede neural de grafo denominada *Graph Property Sensing Network* (GPS-Net), que realiza a detecção das propriedades dos grafos explorando completamenter as informações de orientação das bordas da imagem em análise, diferenças de prioridade entre os nós e a distribuição de relacionamento em grandes conjuntos de dados.

O método proposto é composto por três módulos responsáveis pelo processo de SGG: (1) o primeiro consiste em nova técnica de passagem de mensagem denominado *Direction-aware Message Passing* - (DMP) baseada na abordagem de decomposição *Tucker* (Ben-Younes et al., 2017), responsável pelo aprimoramento das características do nó com informações contextuais específicas de um determinado objeto; (2) nova função de perda chamada *NPS-loss* que identifica a prioridade de nós diferentes, permitindo que a coerência semântica da cena visual seja mais precisa; (3) módulo de raciocínio adaptativo (*Adaptive Reasoning Module - ARM*) para a classificação do relacionamento entre os pares de objetos. A abordagem foi comparada com outros métodos de geração de grafo de cena existentes na literatura, em três grandes conjuntos de dados de imagens, *Visual Genome* (Krishna et al., 2017), *OpenImages* (Kuznetsova et al., 2020) e *Visual Relationship Detection* (Lu et al., 2016) e conforme resultados apresentado por Li et al. (Lin et al., 2020) a *GPS-Net* superou os métodos anteriores de geração de grafo de cena.

Xu et al. (Xu et al., 2017) propuseram um método que visa gerar grafos de cenas por meio da passagem interativa de mensagens (*Iterative Message Passing*) entre os nós que compõem o grafo, aplicando RNNs (Jain et al., 2016) que consistem em redes neurais desenvolvidas para reconhecer padrões em uma determinada sequência de dados, considerando a sequência temporal das informações, analisando assim os dados com alta taxa de ocorrência e constrói um modelo para prever o próximo dado da sentença, ou seja, prevê o próximo atributo de um respectivo nó em um grafo.

O modelo descrito na Figura 3.9 consiste primeiramente em extrair as características visuais dos nós e arestas de um conjunto de regiões candidatas de objetos, que são utilizadas

como entradas iniciais nas GRUs (*Gated Recorrent Unit*) (Cho et al., 2014a) produzindo assim um conjunto de estados ocultos (a), ou seja, o papel das GRUs é capturar adaptativamente dependências de grandes sequências de dados sem descartar informações de partes anteriores da sequências. Em seguida uma função de agrupamento de mensagens define quais serão passadas para cada nó na próxima iteração (b). Além disso, outro diferencial das GRUs está relacionado aos dois mecanismos presentes, *reset* e *update*, que consistem basicamente na atualização da iteração dos estados ocultos das GRUs (c). Após a última atualização das iterações, os estados finais são usados para prever as categorias dos objetos, definindo assim as classes e seus relacionamentos (d).

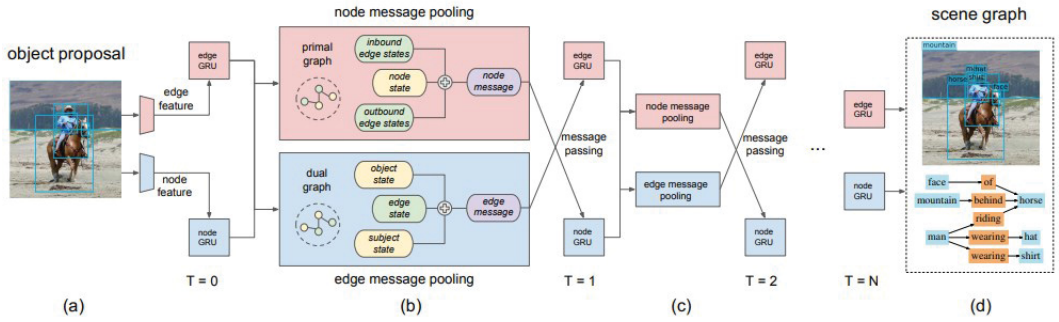


Figura 3.9: Ilustração da arquitetura de GCN do modelo de Iteração por passagem de mensagens. Fonte: (Xu et al., 2017).

A configuração da rede proposta permite que o modelo passe as mensagens entre as unidades GRUs seguindo a topologia do grafo de cena. Um dos diferenciais da abordagem está no módulo da camada de *pool* adpatada para transportar as mensagens entre as arestas e os nós, uma vez que o conjunto de GRUs da borda e GRUs de nó forma um grafo bipartido.

O método VCTREE (Tang et al., 2019) emprega um modelo de árvore de contexto visual com estruturas semelhantes a árvores dinâmicas compostas, usando uma RNN (TreeLSTM) para codificação de textos considerados eficientes para representar relacionamentos visuais hierárquicos e paralelos.

Contudo Tang et al.(2020), propuseram um novo *framework* de geração de cena (SGG) que pode ser aplicado para qualquer modelo de SGG com o objetivo de lidar com o viés presente nos conjuntos de dados disponíveis na literatura. A abordagem baseada no método *Total Direct Effect* (TDE) que utiliza conceitos de inferência causal (VanderWeele, 2015, 2013) que permite estimar previsões de relacionamentos entre pares de objetos por meio de modelos menos tendenciosos. O *framework* consiste em primeiramente em construir um grafo causal ou DAG para SGG que será utilizado na etapa do treinamento tradicional e tendencioso. Em seguida, aplica-se o método de causalidade contrafactual no grafo treinado visando a inferência de efeitos gerado pelo viés incoerente que deve ser removido.

Na Tabela 3.1 é possível visualizar um comparativo geral do estado da arte para geração de grafos de cenas. Por questões cronológicas, os trabalhos estão listados por ano de publicação. O objetivo é ilustrar as semelhanças e as diferenças dos métodos SGGs presentes na literatura.

Em relação aos trabalhos publicados e referenciados no estado da arte, destaca-se a existência de duas abordagens predominantes para o processo SGG, onde a primeira consiste em uma abordagem de estágio único (*Single-Stage*) na qual as classes de objetos e seus relacionamentos são inferidos em conjunto. A segunda abordagem é a de dois estágios (*Two-Stage*) onde a detecção de objetos é realizada separadamente do estágio subsequente de inferência de relacionamentos para os pares de objetos detectados.

Diante da Tabela 3.1 nota-se que a maioria dos métodos estado da arte apresentam um estágio único no processo de geração de grafo de cena. Além disso, pode-se observar em quais tópicos cada método se destacada em relação aos demais.

Mediante ao comparativo apresentado neste capítulo observa-se uma particularidade entre os métodos Motifs e Graph R-CNN, ambos fazem uso do conceito de mecanismo de atenção (Xiao et al., 2015; Zhao et al., 2017) cujo objetivo é tratar diferentes regiões com diferentes níveis de importância visando garantir características que possam ser aplicadas preferencialmente na explorações de regiões-chave para se obter informações mais relevantes para o processo de geração de grafo de cena, tal abordagem tem se mostrado eficaz no aperfeiçoamento do desempenho do SGG.

Tabela 3.1: Comparativo do estado da arte de métodos SGG

Métodos	Object Detection	Approach	Efficient Graph Features Refinement	Efficient Graph Generation	Long-tailed Dataset Distribution	Attention Mechanisms
MSDN (Li et al., 2017)	Faster R-CNN	Single-Stage	X			
Iterative Message Passing (Xu et al., 2017)	Faster R-CNN	Single-Stage	X			
Motifs (Zellers et al., 2018)	Faster R-CNN	Two-Stage	X		X	X
Graph R-CNN (Yang et al., 2018)	Faster R-CNN	Single-Stage	X	X		X
Factorizable net (Li et al., 2018)	Faster R-CNN	Single-Stage		X		
Attentive Relational Networks (Qi et al., 2019b)	Faster R-CNN	Single-Stage	X			
VCTRee (Tang et al., 2019)	Faster R-CNN	Two-Stage	X		X	
Causal-TDE (Tang et al., 2020)	Faster R-CNN	Single-Stage		X	X	
GPS-Net (Lin et al., 2020)	Faster R-CNN	Single-Stage	X			

4 METODOLOGIA

Neste capítulo é detalhada a combinação entre os métodos Causal-TDE (Tang et al., 2020) e GPS-Net (Lin et al., 2020) para o processo de geração de grafo de cena, visando a aplicação da abordagem de inferência causal, juntamente com definição de priorização das características de um determinado nó, por meio da identificação das relações de causa-efeito entre diferentes variáveis.

4.1 VISÃO GERAL

A partir da escolha dos modelos de SGGs como principal metodologia de geração de grafos de cena, o presente trabalho divide-se em duas grandes etapas:

- Avaliar o desempenho da combinação entre os métodos Causal-TDE e GPS-Net, intitulado como Unbiased Graph Property Network for Scene Graph Generation (UGP-Net);
- Gerar grafos de cenas com o uso de metodologia de causa-efeito na identificação de priorização das caracaterísticas de um nó.

Apesar da proposta de combinação dos métodos mencionados anteriormente, o presente trabalho utiliza as etapas definidas no *pipeline* padrão do processo de SGG ilustrado na Figura 4.1, onde as etapas foram detalhadas no Capítulo 2.

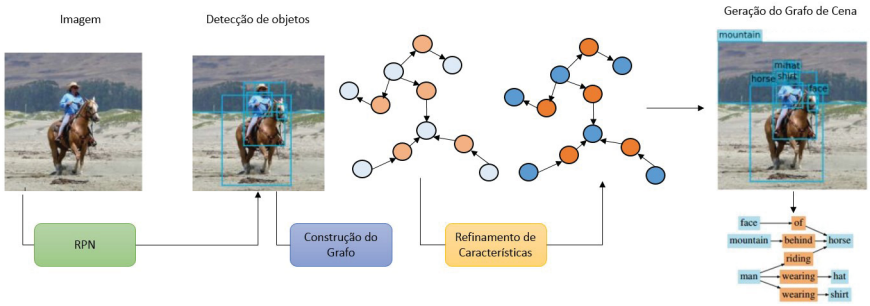


Figura 4.1: Pipeline geral do processo de geração de grafo de cena. Fonte: Adaptado de Xu et al.(2017).

Para etapa de treinamento utilizou-se a abordagem proposta por Tang et al.,2020, pois a principal contribuição e diferencial neste método está na estrutura de análise *Total Direct Effect* (TDE) que permite explorar métodos de treinamento tendenciosos com base na inferência causal. Desta forma, a TDE encontra e elimina vieses indesejáveis aspecto comum em grandes conjuntos de dados, permitindo assim, a extração das relações causais contrafactuais dos dados que compõem o grafo de treinamento.

Contudo, uma vez realizado o treinamento com o método Causal-TDE as dependências entre as varíaveis são aprendidas e para identificar o efeito e a ordem de priorização que uma determinada dependência pode causar em relação a uma outra, optou-se por aplicar o método de detecção de propriedades de grafos (Lin et al., 2020) que explora completamente as informações de orientação de borda, as diferenças de prioridade entre nós e a distribuição de relacionamento entre os objetos detectados em grandes conjuntos de dados durante o processo de geração de grafos de cena.

4.2 MÉTODO PROPOSTO

Conforme mencionado na seção anterior deste capítulo, o método proposto visa a combinação de duas abordagens de geração de grafo de cena, onde ambas irão se complementar durante a execução do processo. A Figura 4.2 mostra a visão geral do método UGP-Net proposto neste trabalho.

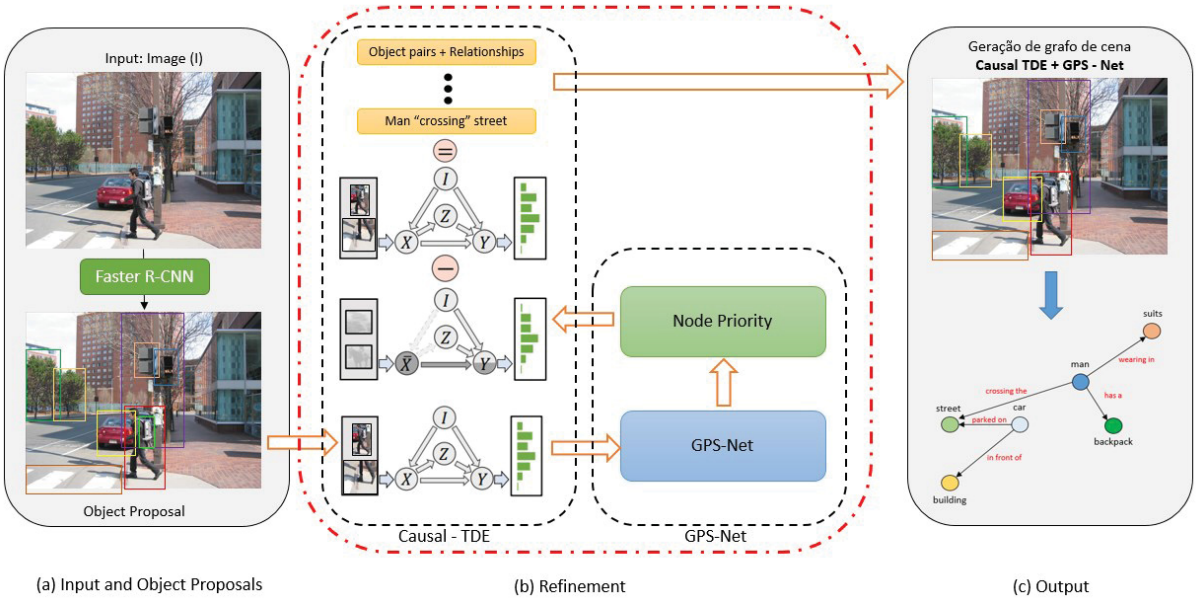


Figura 4.2: Visão geral do método proposto UGP-Net. Fonte: Autor

A abordagem proposta tem como entrada uma imagem ($\mathcal{I}$), onde os objetos são detectados por meio da rede neural Faster R-CNN (Ren et al., 2015). Na etapa (b) que consiste na construção inicial dos grafos e refinamento é criado um grafo causal visando a extração de 3 (três) principais informações tais como: (i) características dos objetos (X), (ii) predição das classes (Z) e (iii) classificação dos predicados entre os pares de objetos (Y), informações obtidas por meio do método Causal-TDE. Após a construção inicial do grafo causal os vetores de informações gerados são utilizados como *input* na rede neural de grafos GPS-Net que foi alterada para receber os vetores com as informações iniciais simulando a técnica de *fine-tuning* e tendo como saída a priorização dos nós.

Em seguida, após a priorização dos nós os vetores de informações são atualizados e devolvidos para a rede neural do método Causal - TDE para que o processo de geração de grafo de cena possa dar andamento. Nesta etapa, são gerados novos grafos causais juntamente com a técnica proposta por (Tang et al., 2020) que visa identificar a causa-efeito que uma determinada variável exerce em relação a outra.

Contudo, o diferencial desta etapa está na eliminação do viés nos conjuntos de dados utilizados nos experimentos, onde o processo de eliminação será um pouco mais refinado, pois levará em consideração a priorização dos nós permitindo assim, uma associação dos relacionamentos com maior probabilidade de assertividade entre os pares dos objetos, ou seja, o refinamento e a priorização dos nós permite que informações de alto nível semântico e com baixa frequência tenha a mesma importância das informações que se repetem com maior recorrência. E, por último temos a geração do grafo de cena, conforme descrito na Figura 4.2(c).

4.3 TREINAMENTO

Na etapa de treinamento utilizou os mesmos conjuntos de dados descrito nos trabalhos mais recentes na literatura (Lin et al., 2020). Dentre os conjuntos de dados está o dataset *Visual Genome* (Krishna et al., 2017), onde foram utilizadas especificamente 150 categorias de objetos e 50 das categorias de relacionamentos de maior recorrência. O conjunto de treinamento é composto por 70% das imagens do *dataset*, das quais 5.000 imagens foram utilizadas como subconjunto de validação. E por último, na etapa de testes foram utilizados os 30% restantes das imagens. Já para o conjunto de imagens *OpenImages* (Kuznetsova et al., 2020) foram utilizadas 53.953 imagens para treinamento e 3.234 imagens na etapa de testes.

4.3.1 Inicialização da UGP-Net

O treinamento do método UGP-Net se inicia primeiramente com a execução da rede neural Faster R-CNN responsável pela etapa de detecção dos objetos. Na entrada da rede neural além da imagem, devem ser adicionados arquivos de extensão $.XML$ referente as informações das imagens, por exemplo, do tamanho da imagem, da classe e das coordenadas iniciais e finais de cada padrão contido na imagem, conforme ilustrado na Figura 4.3.

```
<annotation>
    <folder>VG_100K"</folder>
    <filename>2412112.jpg</filename>
    <source>
        <database>ILDDatabase</database>
    </source>
    <size>
        <width>931</width>
        <height>500</height>
        <depth>2</depth>
    </size>
    <object>
        <name>people</name>
        <bndbox>
            <xmin>147</xmin>
            <ymin>330</ymin>
            <xmax>176</xmax>
            <ymax>347</ymax>
        </bndbox>
    </object>
</annotation>
```

Figura 4.3: Exemplo de arquivo de anotação da Faster R-CNN.

Em paralelo ao *input* das imagens e de suas respectivas anotações, a rede neural de grafos UGP-Net recebe como entrada as anotações referente a região, atributos, predicados e relacionamentos que compõem a imagem, por meio de arquivos de extensão $.JSON$. Na Figura 4.4 pode se observar um arquivo $.JSON$ contento a descrição de todas as regiões de uma determinada imagem.

```
[...
    {
        "image": 2407890,
        "x": 117,
        "y": 79,
        "width": 249,
        "height": 107,
        "phrase": "a cat sitting on a table.",
    },
    {
        "image": 2407890,
        "x": 116,
        "y": 29,
        "width": 239,
        "height": 135,
        "phrase": "a white cat with a tan tail and face markings",
    },
...]
```


Figura 4.4: Exemplo de arquivo de anotação de regiões do método UGP-Net.

Já a Figura 4.5 exemplifica o modelo do arquivo de anotações referente aos atributos, predicados e relacionamentos entre os pares de objetos de uma imagem. Tais informações são de suma importância para a etapa de geração e refinamento dos grafos de cena foco principal deste trabalho.


```
"image": 2407890,
"x": 116,
"y": 29,
"width": 239,
"height": 135,
"phrase": "a white cat with a tan tail and face markings",
"bounding_boxes": [
    {
        "id": 271872,
        "x": 109,
        "y": 37,
        "width": 201,
        "height": 133,
        "boxed_objects": [
            {
                "name": "cat",
                "object_canon": []
            }
        ]
    }
],
"relationships": [
    {
        "predicate": "has",
        "subject": 271882,
        "object": 271883,
        "relationship_canon": [
            {
                "synset_name": "have.v.01",
                "synset_definition": "have or possess, either in a concrete or an abstract sense"
            }
        ]
    }
],
"attributes": [
    {
        "attribute": "white",
        "subject": 271872,
        "attribute_canon": [
            {
                "synset_name": "white.a.01",
                "synset_definition": "being of the achromatic color of maximum lightness; having little or no hue owing to reflection
                 of almost all incident light"
            }
        ]}
```


Figura 4.5: Exemplo de arquivo de anotações de atributos, predicados e relacionamentos.

Pode-se observar que no arquivo de anotações mostrado na Figura 4.5 as informações estão dividas em blocos que são representados pelas *tags*: *relationships*, *predicate* e *attributes*, possibilitanto a identificação das informações que compõem o grafo de cena.

5 EXPERIMENTOS E RESULTADOS

Neste capítulo é mostrados os resultados obtidos dos experimentos com a aplicação do me´todo UGP-Net.

Realizou-se a comparação da abordagem proposta com os métodos mais recentes apresentados pela literatura atual e com maior relevância para a área que estuda o processo de geração de grafo de cena, tais como: Graph R-CNN, Motifs, Iterative Message Passing, VCTREE, Causal-TDE e GPS-Net.

Para o comparativo qualitativo dos experimentos se utilizou a *Recall@K (R@K)* convencional (R@k = 20,50,100), onde o objetivo é calcular a fração de vezes que um determinado relacionamento considerado correto é previsto no "topo"versus as previsões de relacionamentos confiáveis. No entanto, na abordagem apresentada por (Tang et al., 2020) ele destaca que em conjuntos de dados tendenciosos, a métrica *Recall@K (R@K)* apresenta baixo desempenho na classificação de categorias menos frequentes gerando assim o viés, comum em grandes conjuntos de dados.

Portanto, visando o tratamento do viés e com base nos protocolos utilizados no método Causal-TDE (Tang et al., 2020) especificamente no conjunto de dados *Visual Genome* se utilizou uma métrica balanceada denominada *mean Recall@K* (mrR@*K*), onde é calculado o *recall* em cada categoria de predicado de forma independente e, em seguida, calcula a média dos resultados, onde a variável *K* corresponde a quantidade de resultados que atingiram o "*k-top*"de desempenho em cada categoria e que serão utilizados para o cálculo da média. Desta forma, cada categoria contribui igualmente para geração do grafo de cena.

O uso da métrica *mean Recall@K* reduz a influência de alguns predicados comuns, e que impactam na construção semântica dos grafos de cena, por exemplo, "*on*", "*of*", e exerce uma atenção àqueles predicados infrequentes, por exemplo, "*riding*", "*porting*", que são mais valiosos para o alto nível semântico. Portanto, na Tabela 5.1 ilustra o comparativo entre metodos SGGs no conjunto de dados *Visual Genome* (VG).

Além disso, todos os experimentos listados neste capítulo referente a base de dado *Visual Genome* foram avaliados por três tipos de configurações distintas, tais como: (1) *Predicate Classification* (PredCls) responsável por prever os rótulos de relacionamentos, a partir de uma imagem de entrada que contém os *bounding boxes* e as classes dos objetos que compõem ela; (2) *Scene Graph Generation* (SGCls) que prevê os rótulos dos objetos e predicados a partir com base em uma imagem de entrada e (3) *Scene Graph Detection* (SGDet), cujo objetivo é a predição de um grafo de cena.

Na Tabela 5.1 pode ser observado que a abordagem UGP-Net apresentou melhor resultado relativo quando comparado aos métodos SGGs que não realizaram nenhum tipo de combinação (fusão) com o método Causal-TDE (Tang et al., 2020). Contudo, o método Causal-TDE + VCTREE apresentou uma melhor performance de 1.17%, 2.36% e 0.67% a mais para o protocolo PredCls. Já em relação ao método Motifs + Causal - TDE obtivemos uma melhoria relativa de 0.2% para PredCls em mR@20 e mR@100.

Para o protocolo SGCls pode-se observar que o UGP-Net demonstra melhor performace de 0.9% e 0.7% para mR@50 e mR@100 se comparado ao método Motifs + Causal-TDE.

As Figuras 5.1 e 5.2 mostram exemplos do processo de SGGs por meio do método UGP-Net, onde na Figura 5.1 (a) ilustra o exemplo de uma imagem cujo ambiente não é controlado (*in the wild)*, ou seja, nota-se um volume significativo de classes detectadas durante a etapa

de localização de objetos que compõem a imagem e na Figura 5.1 (b) se tem o grafo de cena resultante dos processos de construção e refinamento da abordagem UGP-Net.

Contudo, se observa que no grafo gerado na Figura 5.2 nem todos os objetos detectados foram relacionados entre si ou utilizados no grafo de cena. Tal comportamento reforça a finalidade do processo de refinamento dos vetores de características que apenas mantém as informações de maior relevãncia de associação para o processo de SGG.

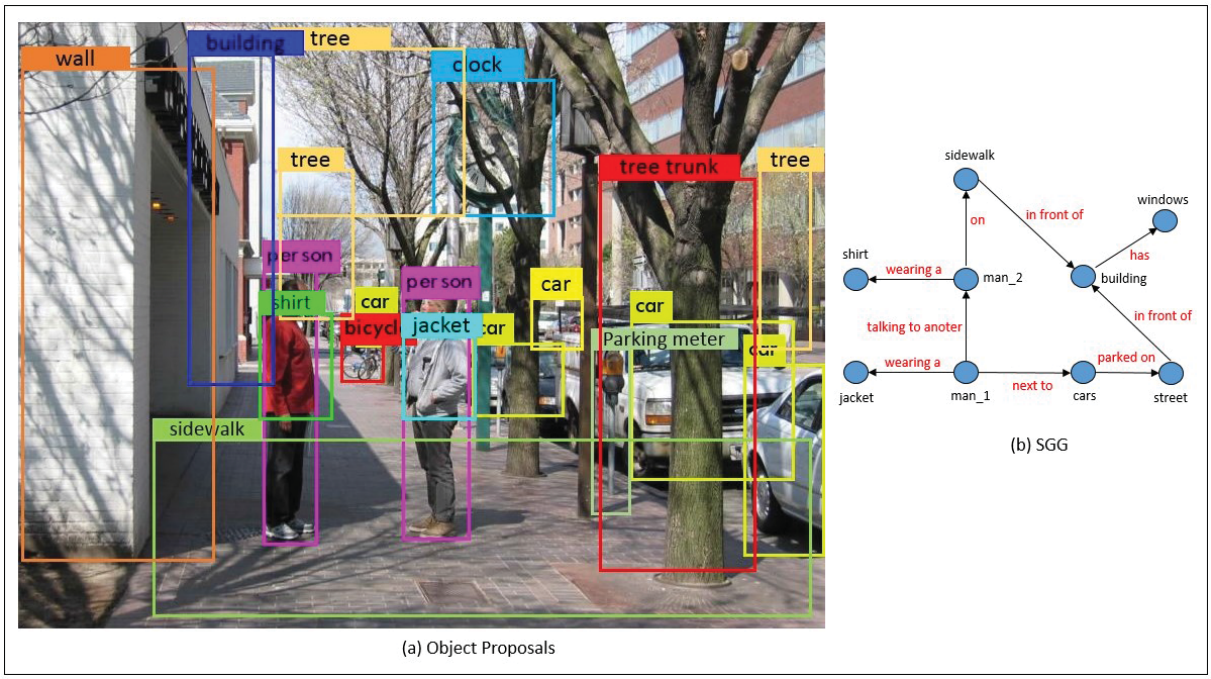


Figura 5.1: Geração de Grafo de Cena em ambientes não controlados.

Já a Figura 5.2 ilustra o processo de geração de grafo de cena em uma imagem com uma quantidade de classes de objeto menor, onde o ambiente que compõem a cena ao ser comparado com a Figura 5.1 é considerado controlado permitindo assim a associação de um ou mais pares de objetos de forma mais assertiva. Neste caso, todos objetos foram utilizados e relacionados no grafo de cena final.

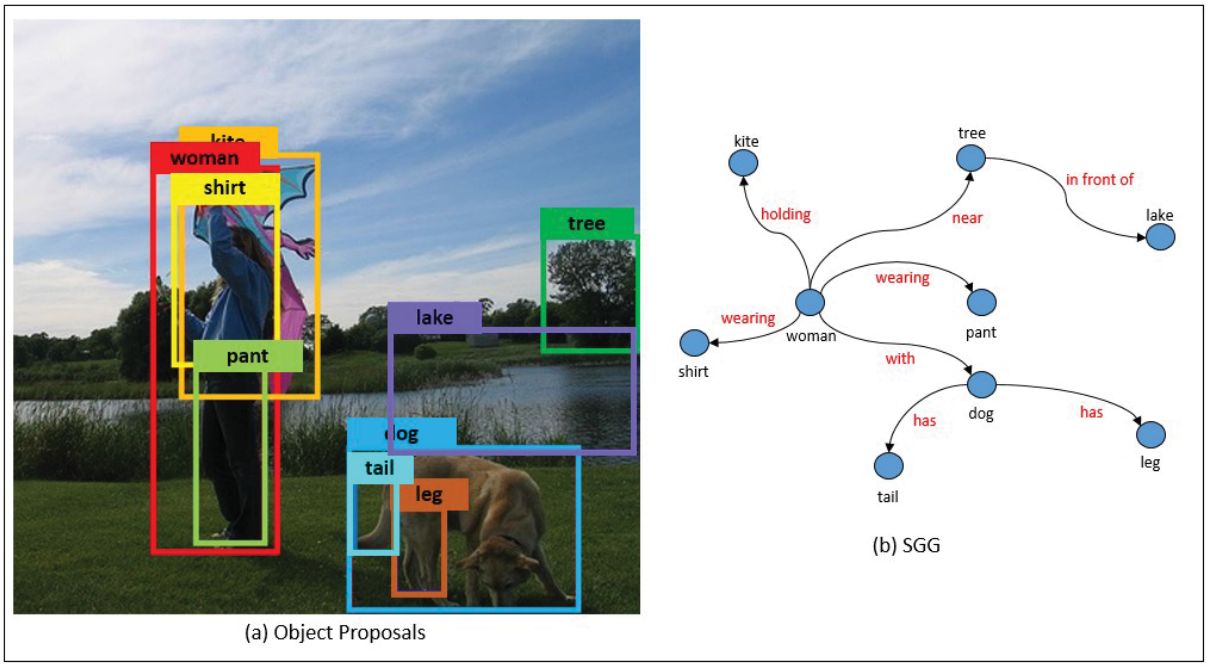


Figura 5.2: Geração de Grafo de Cena em ambientes controlados.

A Figura 5.3 exemplifica de forma detalhada o método UGP-Net. Inicialmente o método adota uma imagem (I) como entrada da rede UGP-Net, a abordagem Faster R-CNN é aplicada para se obter a localização e as características visuais dos objetos existentes na imagem (Figura 5.3(a)). Em seguida, na Figura 5.3(b) aplica-se o método Causal-TDE (Tang et al., 2020) adaptado para gerar inicialmente apenas um grafo causal que será responsável na geração dos vetores de características X, Z e Y que contém as *features* dos objetos, por exemplo, *"grape is red"* ; predição das classes (*grapes, table, kiwi*) e classificação dos predicados entre os pares de objetos (*on, has, in front a*).

Após a geração dos vetores, a terceira etapa do processo consiste em "alimentar"a rede GPS-Net (Lin et al., 2020) que define a prioridade dos nós do grafo gerado e atualiza os vetores X, Z e Y, seguindo a respectiva priorização conforme ilustrado na 5.3(c). Nesta etapa, novos grafos causais são gerados onde predição de causa e efeito entre os pares de objetos são realizadas, até que não seja mais possível atualizar o estado (características) de todos os nós que compõem o grafo final que resultará no grafo de cena com informações semânticas (Figura 5.3(d)).

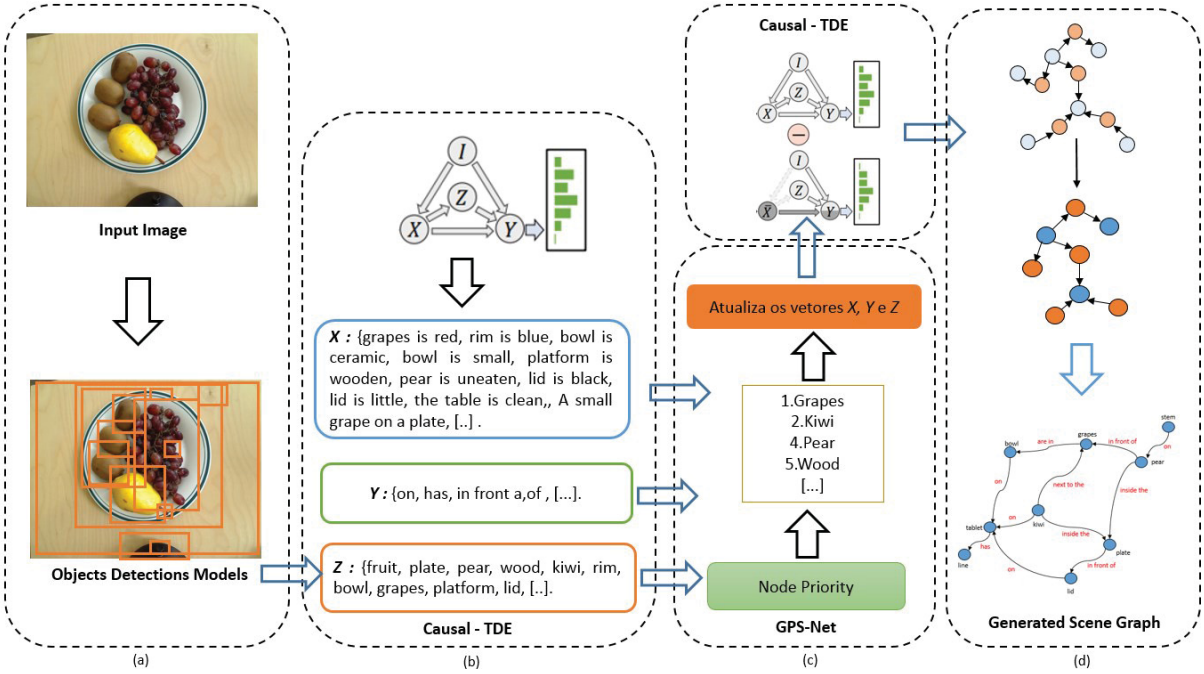


Figura 5.3: Visão detalhada do método UGP-Net.

Tabela 5.1: Comparação de mR@20, mR@50 e mR@100 em % dos três protocolos SGG no conjunto de dados VG.

	Predicate Classification			Scene Graph Classification			Scene Graph Detection		
Método	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
GPS-Net	13.23	20.5	15.4	6.2	7.5	14.0	4.9	16.3	29.0
Motifs	14.17	18.02	19.53	8.18	10.22	10.98	5.66	7.72	9.27
Graph R-CNN	5.3	6.1	18.3	3.8	9.4	10.0	**11.7**	14.1	13.7
VCTREE	14.9	17.9	19.4	8.2	10.1	10.8	5.2	8.9	16.6
Causal - TDE + VCTREE	**19.87**	**26.66**	**29.97**	**13.86**	**18.2**	**20.45**	7.1	**29.69**	**51.6**
Iterative Message Passing	8.85	10.97	11.77	5.4	6.4	6.74	2.2	3.29	4.14
Motifs + Causal - TDE	18.5	25.5	29.1	9.8	13.1	14.9	4.80	25.5	42.0
UGP-Net	**18.7**	**24.3**	**29.3**	**9.5**	**14.0**	**15.6**	**3.82**	**25.0**	**40.32**

Também se avaliou o método proposto no presente trabalho no conjunto de dados *Open Images* respectivamente nas versões V4 e V6. Para a análise comparativa entre os métodos estado da arte utilizou-se a mesma metodologia apresentada no trabalho intitulado *Structured Sparse R-CNN* (SS R-CNN) (Teng e Wang, 2022), os resultados estão ilustrados nas Tabelas 5.2 e 5.3. Dentre as métricas utilizadas para o comparativo temos o *score* que calcula a pontuação de cada método por meio da fórmula *score* = 0.2 x Recall@50 + 0,4 x $wmAP_{rel}$ + 0.4 x $wmAP_{phr}$, onde $wmAP_{rel}$ e $wmAP_{phr}$ correspondem a média poderada dos relacionamentos entre os pares de objetos e das frases detectadas na imagem em análise.

Tabela 5.2: Comparativo entre os métodos de SGG no dataset *Open Imagens* - V4.

Métodos	mR@50	R@50	wmAP$_{rel}$	wmAP$_{rel}$	score$_{wtd}$
ReIDN	70.40	**75.66**	36.13	39.91	45.21
GPS-Net	69.50	74.65	35.02	39.40	44.70
BGNN	72.11	75.46	37.76	41.70	46.87
SS R-CNN	72.62	74.92	43.47	48.17	51.64
SS R-CNN$_{LA}$	**79.23**	74.75	**43.57**	**48.25**	**51.68**
UGP-Net	73.77	74.70	42.11	46.23	50.28

Em comparação a literatura e conforme mostrado na Tabela 5.2 o método UGP-Net obteve melhor performance quando comparado na maioria dos métodos de SGGs em relação a métrica mR@50, apresentando apenas uma performance inferior para a abordagem SS R-CNN$_{LA}$. Entretanto, no comparativo em R@50 o UGP-Net se demonstrou melhor desempenho apenas em relação ao método GPS-Net.

Já em relação a versão 6 do datatset *Open Imagens* o método proposto apresentou melhor desempenho em mR@50 e R@50 quando comparado com a maioria das abordagens conforme mostrado na Tabela 5.3, exceto quando comparado com os métodos SS R-CNN e SS R-CNN$_{LA}$ que apresentaram melhores resultados no contexto geral de avaliação.

Tabela 5.3: Comparativo entre os métodos de SGG no dataset *Open Imagens* - V6.

Métodos	mR@50	R@50	wmAP$_{rel}$	wmAP$_{phr}$	score$_{wtd}$
MOTIFS	32.68	71.63	29.91	31.59	38.03
ReIDN	33.98	73.08	32.16	33.39	40.84
VCTree	33.91	74.08	34.16	33.11	40.21
G-RCNN	34.04	74.51	33.51	34.21	41.84
GPS-Net	35.26	74.81	32.85	33.98	41.69
BGNN	40.45	74.98	33.51	34.15	42.06
SS R-CNN	42.84	**76.66**	**41.47**	**43.64**	**49.38**
SS R-CNN$_{LA}$	**50.73**	75.70	41.14	43.24	48.89
UGP-Net	41.23	75.08	40.16	42.15	47.74

Diante os resultados apresentados no conjunto de dados *Open Imagens* (V4 e V6) pode-se concluir que o método *Structured Sparse R-CNN* apresenta melhor eficácia devido a sua técnica de alinhamento das regiões de interesse (ROIs) referente a localização dos objetos e de seus respectivos relacionamentos, técnica que se repete ao longo do processo de geração de grafo de cena. Além disso, a abordagem utiliza uma arquitetura de rede esparsa e unificada que permite a geração de grafos de cena de forma direta sem que haja a detecção explicita dos objetos para gerar os grafos.

Além disso, o presente trabalho também avaliou o método UGP-Net no conjunto de imagens *Visual Relationship Detection* (VRD) (Lu et al., 2016), onde a métrica de comparação entre as abordagens foi a mesma do trabalho proposto por (Lu et al., 2016) para detecção de relacionamentos, predicados e frases.

A Tabela 5.4 apresenta comparações no conjunto de imagens VRD em relação os métodos de maior relevância na literatura atual referente a área de geração de grafo de cena. Contudo, visando uma comparação equivalente aos protocolos utilizados nas demais abordagens o detector do método UGP-Net foi iniciado com um modelo pré-treinado que utilizou as imagens dos *datasets* apresentados no trabalho ReIDN (Zhang et al., 2019) conhecidos como ImageNet (Deng et al., 2009) e COCO (Lin et al., 2014).

Tabela 5.4: Comparações com o estado da arte no dataset VRD

	Predicado	Relacionamento		Frases	
Métodos	R@50	R@50	R@100	R@50	R@100
VTransE (Zhang et al., 2017)	44.8	19.4	22.4	14.1	15.2
ViP-CNN (Yikang et al., 2017)	-	17.3	20.0	22.8	27.9
VRL (Liang et al., 2017)	-	18.2	20.8	21.4	22.6
GPS-Net (ImageNet) (Lin et al., 2020)	58.7	21.5	24.3	28.9	34.0
GPS-Net (Coco) (Lin et al., 2020)	63.4	27.8	31.7	33.8	39.2
UGP-Net (ImageNet)	**59.2**	**21.8**	**25.0**	**29.75**	**35.22**
UGP-Net (Coco)	**64.5**	**22.3**	**31.7**	**36.2**	**41.0**

Diante dos resultados apresentados na Tabela 5.4 o método UGP-Net apresentou desempenho superior em relação aos demais métodos do estado da arte. Tal resultado está relacionado ao tratamento do viés juntamente com a definição da priorização dos nós que formam um determinado grafo de n graus.

6 CONCLUSÃO

6.1 CONSIDERAÇÕES FINAIS

Neste trabalho fez-se um estudo do problema de geração de grafo de cena, onde foram apresentados os principais métodos mais recentes que buscam solucionar os problemas referente ao viés, heterogeneidade, refinamento robusto e eficaz nos principais conjuntos de imagens.

Os experimentos foram conduzidos de acordo com as metodologias apresentadas na literatura e conforme os conjuntos de dados avaliados. Como previamente descrito, o presente trabalho consiste em propor uma nova abordagem que apresente melhor desempenho no processo de construção e refinamento das informações (atributos, predicados e relacionamentos) durante o processo de SGG.

Conforme descrito no capítulo anterior o método UGP-Net apresentou resultados significativos quando comparado com as demais abordagens presente na literatura consideradas relavantes para a área de conhecimento abordada no presente trabalho. No entanto, no decorrer da realização da fusão entre os métodos Causal-TDE (Tang et al., 2020) e GPS-Net (Lin et al., 2020) se observou que a ordem execução das abordagens impactou diretamente no *output* da rede convolucional de grafo UGP-Net. A primeira tentativa de combinação visou iniciar o processo com o método GPS-Net que subsequentemente alimentaria a rede Causal-TDE, mas se observou que a rede GPS-Net apresentou uma má distribuição dos predicados gerando assim viés nos conjuntos de dados analisados.

Portanto, após observar o comportamento mencionado acima juntamente com uma má detecção da direção das bordas da imagem dada como entrada na rede, optou-se iniciar a combinação primeiramente com o método Causal-TDE visando a construção de um grafo causal "tradicional"que permite a identificação da causa-efeito que um nó tem em relação aos seus respectivos vizinhos. Tal abordagem permitiu que a priorização dos nós de um determinado grafo fosse iniciada a partir de um refinamento menos tedencioso.

A principal contribuição deste trabalho está na combinação entre o métodos e na reestruturação da rede convolucional de grafo GPS-Net para receber como entrada três vetores de informações distintas juntamente com os *bounding boxes* das regiões de interesse e realizar a priorização dos nós, e sem seguida, gerar como saída novos vetores atualizados conforme a priorização definida.

Ainda, como contribuição final do presente trabalho, destaca-se a combinação entre os métodos GPS-Net e Causal - TDE, arquitetura, a qual não era implementada na literatura existente e que apresenta ganhos significativos no processo de geração de grafo de cena.

Em trabalhos futuros, propõe-se o foco na substituição do detector de objetos atual do *pipeline* do processo de SGG para o método Yolo (Bochkovskiy et al., 2020) considerado o estado da arte no processo de detecçãp e classificação de objetos, pois se observou que todos os métodos SGG utilizan a Faster R-CNN, cujo diferencial está no forma de detecção dos objetos que aplicada a técnica de *Region Proposal Networks* que gera conjunto de regiões candidatas em diferentes escalas, comportamente que propocionará um comparativo relevante em relação ao método considerado o estado da arte atualmente.

Além disso, visa se aplicar futuramente o método UGP-Net na geração de grafo de cenas em *Point Cloud* com o objetivo de avaliar seu desempenho e acuracidade em imagens onde sua estrutura é totalmente 3D e com vértices heterogêneos.

REFERÊNCIAS

Ben-Younes, H., Cadene, R., Cord, M. e Thome, N. (2017). Mutan: Multimodal tucker fusion for visual question answering. Em *Proceedings of the IEEE international conference on computer vision*, páginas 2612–2620.

Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y. e Huang, J. (2020). Rumor detection on social media with bi-directional graph convolutional networks. Em *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, páginas 549–556.

Bochkovskiy, A., Wang, C.-Y. e Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Bruna, J., Zaremba, W., Szlam, A. e LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Cho, K., Van Merriënboer, B., Bahdanau, D. e Bengio, Y. (2014a). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. e Bengio, Y. (2014b). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. e Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. Em *2009 IEEE conference on computer vision and pattern recognition*, páginas 248–255. Ieee.

Gao, H., Wang, Z. e Ji, S. (2018). Large-scale learnable graph convolutional networks. Em *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, páginas 1416–1424.

Girshick, R., Donahue, J., Darrell, T. e Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 580–587.

He, K., Zhang, X., Ren, S. e Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916.

Hochreiter, S. e Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jain, A., Zamir, A. R., Savarese, S. e Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. Em *Proceedings of the ieee conference on computer vision and pattern recognition*, páginas 5308–5317.

Johnson, J., Karpathy, A. e Fei-Fei, L. (2016). Densecap: Fully convolutional localization networks for dense captioning. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4565–4574.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M. e Fei-Fei, L. (2015). Image retrieval using scene graphs. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 3668–3678.

Karpathy, A. e Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 3128–3137.

Kipf, T. N. e Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A. et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Krizhevsky, A., Sutskever, I. e Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A. et al. (2020). The open images dataset v4. *International Journal of Computer Vision*, páginas 1–26.

Lazebnik, S., Schmid, C. e Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Em *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, páginas 2169–2178. IEEE.

LeCun, Y., Bengio, Y. e Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Levi, G. e Hassner, T. (2015). Age and gender classification using convolutional neural networks. Em *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, páginas 34–42.

Li, Y., Ouyang, W., Zhou, B., Shi, J., Zhang, C. e Wang, X. (2018). Factorizable net: an efficient subgraph-based framework for scene graph generation. Em *Proceedings of the European Conference on Computer Vision (ECCV)*, páginas 335–351.

Li, Y., Ouyang, W., Zhou, B., Wang, K. e Wang, X. (2017). Scene graph generation from objects, phrases and region captions. Em *Proceedings of the IEEE International Conference on Computer Vision*, páginas 1261–1270.

Li, Y., Tarlow, D., Brockschmidt, M. e Zemel, R. (2016). Gated graph sequence neural networks. *International Conference on Learning Representations*.

Liang, X., Lee, L. e Xing, E. P. (2017). Deep variation-structured reinforcement learning for visual relationship and attribute detection. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 848–857.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. e Zitnick, C. L. (2014). Microsoft coco: Common objects in context. Em *European conference on computer vision*, páginas 740–755. Springer.

Lin, X., Ding, C., Zeng, J. e Tao, D. (2020). Gps-net: Graph property sensing network for scene graph generation. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 3746–3753.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. e Berg, A. C. (2016). Ssd: Single shot multibox detector. Em *European conference on computer vision*, páginas 21–37. Springer.

Lu, C., Krishna, R., Bernstein, M. e Fei-Fei, L. (2016). Visual relationship detection with language priors. Em *European conference on computer vision*, páginas 852–869. Springer.

Pearl, J. (2009). *Causality*. Cambridge university press.

Qi, M., Li, W., Yang, Z., Wang, Y. e Luo, J. (2019a). Attentive relational networks for mapping images to scene graphs. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 3957–3966.

Qi, M., Li, W., Yang, Z., Wang, Y. e Luo, J. (2019b). Attentive relational networks for mapping images to scene graphs. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 3957–3966.

Ranjan, R., Patel, V. M. e Chellappa, R. (2017a). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):121–135.

Ranjan, R., Sankaranarayanan, S., Castillo, C. D. e Chellappa, R. (2017b). An all-in-one convolutional neural network for face analysis. Em *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, páginas 17–24. IEEE.

Redmon, J., Divvala, S., Girshick, R. e Farhadi, A. (2016). You only look once: Unified, real-time object detection. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 779–788.

Ren, S., He, K., Girshick, R. e Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *In NIPS*.

Seo, Y., Defferrard, M., Vandergheynst, P. e Bresson, X. (2018). Structured sequence modeling with graph convolutional recurrent networks. Em *International Conference on Neural Information Processing*, páginas 362–373. Springer.

Shang, J., Xiao, C., Ma, T., Li, H. e Sun, J. (2019). Gamenet: Graph augmented memory networks for recommending medication combination. Em *proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, páginas 1126–1133.

Tang, K., Niu, Y., Huang, J., Shi, J. e Zhang, H. (2020). Unbiased scene graph generation from biased training. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 3716–3725.

Tang, K., Zhang, H., Wu, B., Luo, W. e Liu, W. (2019). Learning to compose dynamic tree structures for visual contexts. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 6619–6628.

Teng, Y. e Wang, L. (2022). Structured sparse r-cnn for direct scene graph generation. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 19437–19446.

Tian, F., Gao, B., Cui, Q., Chen, E. e Liu, T.-Y. (2014). Learning deep representations for graph clustering. Em *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

VanderWeele, T. (2015). *Explanation in causal inference: methods for mediation and interaction*. Oxford University Press.

VanderWeele, T. J. (2013). A three-way decomposition of a total effect into direct, indirect, and interactive effects. *Epidemiology (Cambridge, Mass.)*, 24(2):224.

Visa, S., Ramsay, B., Ralescu, A. L. e Van Der Knaap, E. (2011). Confusion matrix-based feature selection. *MAICS*, 710:120–127.

Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y. e Zhang, Z. (2015). The application of two-level attention models in deep convolutional neural network for fine-grained image classification. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 842–850.

Xu, D., Zhu, Y., Choy, C. B. e Fei-Fei, L. (2017). Scene graph generation by iterative message passing. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 5410–5419.

Yang, J., Lu, J., Lee, S., Batra, D. e Parikh, D. (2018). Graph r-cnn for scene graph generation. Em *Proceedings of the European conference on computer vision (ECCV)*, páginas 670–685.

Yang, X., Tang, K., Zhang, H. e Cai, J. (2019). Auto-encoding scene graphs for image captioning. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 10685–10694.

Yikang, L., Ouyang, W. e Wang, X. (2017). Vip-cnn: A visual phrase reasoning convolutional neural network for visual relationship detection. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 5532–5540.

Zellers, R., Yatskar, M., Thomson, S. e Choi, Y. (2018). Neural motifs: Scene graph parsing with global context. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 5831–5840.

Zhang, H., Kyaw, Z., Chang, S.-F. e Chua, T.-S. (2017). Visual translation embedding network for visual relation detection. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 5532–5540.

Zhang, J., Shih, K. J., Elgammal, A., Tao, A. e Catanzaro, B. (2019). Graphical contrastive losses for scene graph parsing. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 11535–11543.

Zhang, Y., Liu, Q. e Song, L. (2018). Sentence-state lstm for text representation. *arXiv preprint arXiv:1805.02474*.

Zhao, B., Wu, X., Feng, J., Peng, Q. e Yan, S. (2017). Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia*, 19(6):1245–1256.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A. e Oliva, A. (2014). Learning deep features for scene recognition using places database.

Zhuang, C. e Ma, Q. (2018). Dual graph convolutional networks for graph-based semi-supervised classification. Em *Proceedings of the 2018 World Wide Web Conference*, páginas 499–508.

Zitnik, M., Agrawal, M. e Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466.