Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

VISUAL RELATIONSHIP DETECTION

Relatore:
Chiar.mo Prof.
ANDREA ASPERTI

Presentata da:
MATTIA MARANZANA

Sessione 3, Dicembre
Anno Accademico 2021-2022

Indice

1 Introduzione

L'obiettivo di questo elaborato consiste nel presentare lo stato attuale della Visual Relationship Detection (VRD). I vari argomenti saranno presentati seguendo la struttura di [1]. Si partirà dalle basi su cui si fonda la VRD per poi proseguire con un confronto delle varie forme in cui può essere rappresentata una relazione. Verrà posta particolare attenzione ai vari approcci sviluppatosi per risolvere il problema. Successivamente verranno mostrate le sue applicazioni e i principali dataset utilizzati. Infine verranno analizzati i risultati.

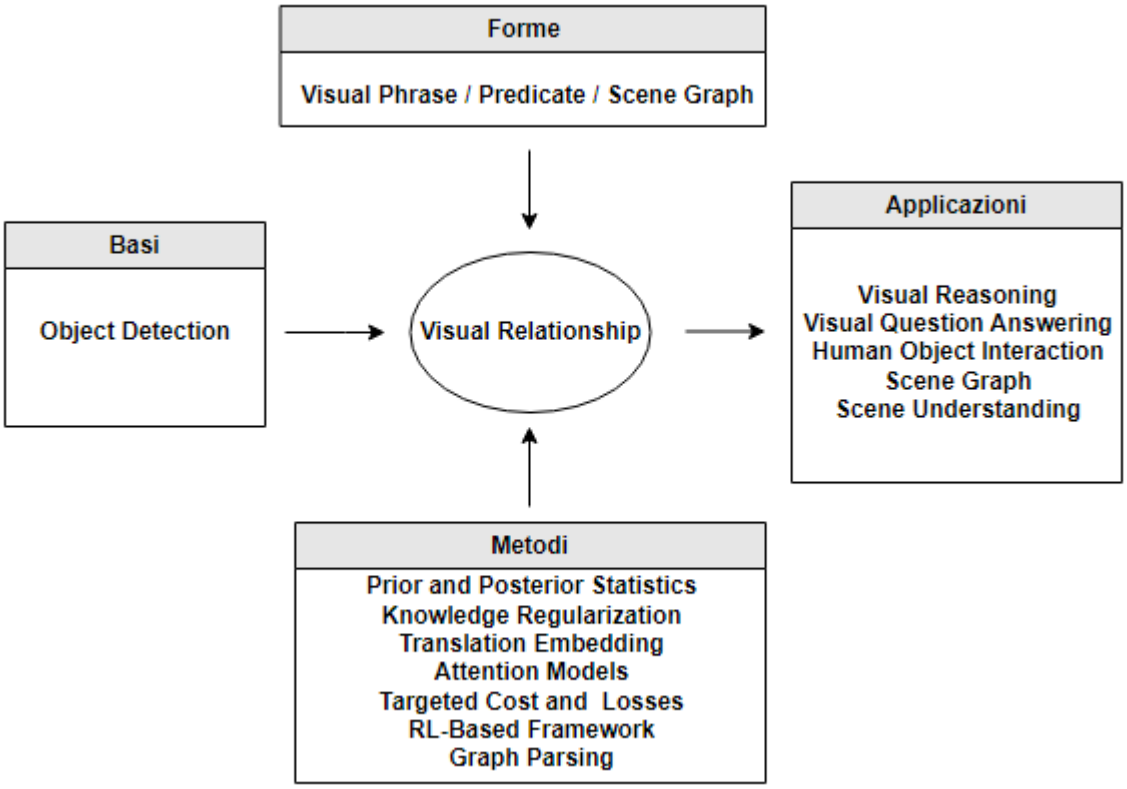

Figura 1: Elementi principali della VRD [1]

2 Fondamenta

La VRD si occupa di rilevare interazioni tra due oggetti in un'immagine. Se creassimo una gerarchia relativa alle attività della computer vision, la VRD si posizionerebbe sopra all'object detection e recognition e sotto a image retrieval, visual reasoning e image caption.
In questa sezione verranno trattate le basi su cui si fonda la VRD e i primi lavori in cui sono state sfruttate le relazioni.

2.1 Object Detection

L'obiettivo dell'object detection consiste nel classificare quali elementi siano presenti in un'immagine e individuare dove essi si trovino [2]. Per indicare la posizione di un oggetto vengono utilizzate le bounding box. Queste rappresentano un rettangolo al cui interno si trova l'oggetto. Esistono due notazioni principali. La prima utilizza il vertice in alto a sinistra e il vertice in basso a destra del rettangolo mentre la seconda utilizza il centro, l'altezza e la larghezza.
L'object detection ha beneficiato particolarmente dallo sviluppo del Deep Learning [3], in particolare con l'utilizzo delle reti neurali convoluzionali (CNN). Questo tipo di rete si è rilevato incredibilmente performante nel rilevare pattern nell'immagini [4]. Una CNN prende in input un' immagine rappresentata tramite una matrice e svolge le seguenti 4 operazioni:

1. Convolution: questa operazione serve a estrarre le caratteristiche dall'immagine. Viene fatta scorrere una matrice di dimensione molto minore rispetto all'immagine chiamata kernel, eseguendo delle moltiplicazioni elemento per elemento tra il kernel e le varie sottomatrici dell'immagine di dimensione pari al kernel. Il risultato di ogni moltiplicazioni compone una cella della matrice restituita come output. La matrice di output viene chiamata Feature Map. Il kernel svolge la funzione di filtro e inizializzare il kernel con valori diversi permette di estrarre caratteristiche diverse.

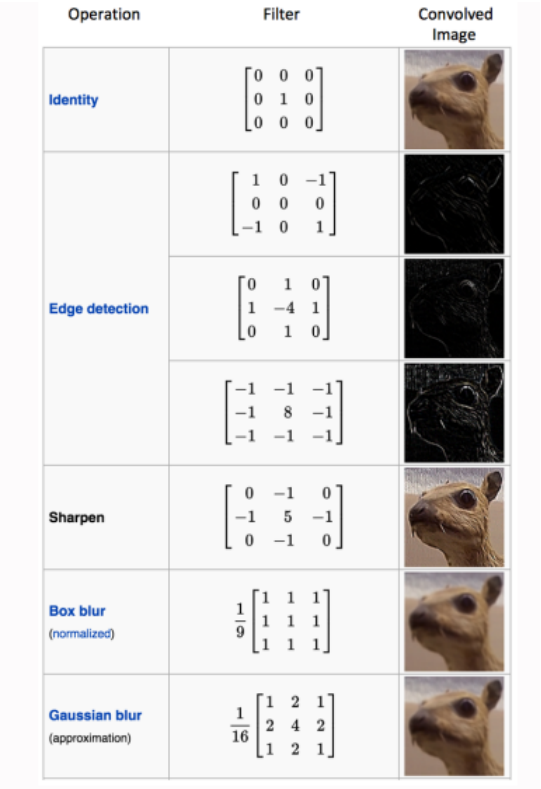

Figura 2: Vari kernel applicati alla stessa immagine [5]

2. Non Linearity (ReLU): viene utilizzata per introdurre non linearità nei dati e può essere definita $ReLU(x) = max(0, x)$. Altre funzioni di attivazioni possono essere usate tuttavia la ReLU risulta più facile da implementare [5].

3. Pooling: serve per diminuire la dimensione della feature map mantenendo le informazioni principali. Questa operazione riduce la complessità del modello.

4. Classification (Fully Connected Layer): fully connected layer indica che ogni neurone nel livello x è collegato a ogni neuroni al livello $x - 1$ e $x + 1$. Quest'ultima operazione restituisce la previsione del modello.

I rilevatori per la object detection possono dividersi in due gruppi [6]:

1. One-stage detector: le operazioni di ricerca e di classificazione degli oggetti vengono svolte in contemporanea.

2. Two-stage detector: vengono prima proposte delle possibili posizioni per gli oggetti cercati chiamate region of interest (RoI). Successivamente avviene la classificazione e vengono migliorate le posizioni. I two-stage detector sono solitamente più precisi ma con un maggiore costo computazionale.

Per il primo gruppo uno degli esempi più importanti è YOLO (You Only Look Once) [7]. Come una persona con un solo sguardo individua gli oggetti, dove si trovano e come interagiscono, YOLO svolge entrambi i compiti della object detection in una sola passata. I vantaggi principali sono la velocità e la possibilità di concentrarsi sull'intera immagine. Fatica invece a localizzare oggetti di piccola dimensione. L'immagine in input viene divisa in una griglia $S \times S$. Se il centro di un'immagine si trova in una cella questa avrà il compito di riconoscere tale oggetto. Le previsioni finali vengono filtrate utilizzando l'intersection over union (IoU).
Per i two stage detector una rete importante anche per il suo utilizzo in alcuni modelli che verranno mostrati in seguito è RCNN [8]. RCNN è composto da 3 moduli:

1. Region proposals: dall'immagine in input vengono proposte circa 200 regioni in cui cercare gli oggetti.

2. Feature extraction: tramite un CNN vengono estratte le caratteristiche per ogni regione.

3. Regions classification: gli oggetti nelle regioni vengono classificate. Per quest'ultima operazione è importante selezionare un valore soglia che non scarti i falsi negativi tramite l'IoU. Prendendo come esempio un classificatore binario per identificare le macchina in un'immagine, alcune regioni potrebbero contenere solo una parte della macchina. Il valore soglia selezionato dopo vari tentativi è 0.3.

Fast R-CNN [9] è un one stage detector proposto per migliorare le prestazioni della RCNN. Oltre a un'immagine prende in input anche un insieme di RoI. Le caratteristiche di ogni RoI vengono trasformate in feature map di dimensione minore. Queste vengono passate a un fully connected layer che produce 2 output: le classi degli oggetti e le relative bounding box.

2.2 Primi Lavori

Scopo iniziale della VRD era quello di migliorare le prestazioni della object detection. Di seguito verranno mostrati alcuni dei primi approcci.
L'obiettivo in [10] consiste nella creazione di un modello capace di individuare vari elementi all'interno di un'immagine. Per meglio selezionare i possibili candidati vengono utilizzate la Non-maximum Suppression e la mutua esclusione. La prima elimina tutte le bounding box che non rispettano certi criteri mentre la seconda indica il vincolo che due oggetti non possano occupare lo stesso spazio all'interno dell'immagine. Per facilitare la ricerca delle varie classi all'interno dell'immagine viene utilizzata la contextual cuening ovvero possiamo ottenere degli indizi sulla posizione di altri oggetti trovato il primo in base al contesto. Se ad esempio venisse individuata una bicicletta, la probabilità che una persona si trovi sopra aumentano. Per meglio identificare la disposizione tra due oggetti viene utilizzato il seguente schema:

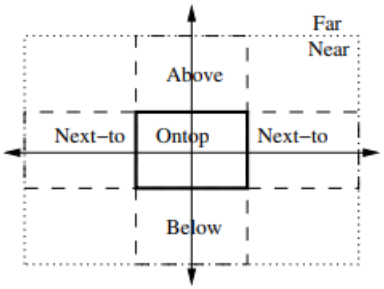

Figura 3: Relazione spaziale in [10]

Nel caso i due oggetti siano vicini vengono identificate 4 ulteriori relazioni(sopra, in alto, in basso, vicino).
L'utilizzo delle relazioni viene usato in [11] per risolvere parzialmente l'ambiguità nell'ottica di realizzare un visual classifier più preciso. L'approccio più semplice al problema consiste nel identificare un elemento di una classe usando un' etichetta corrispondente al valore reale per allenare il modello, tramite vari esempi, per poi poterlo testare con immagini prive di questa etichetta e verificare quanto le sue previsioni si avvicinino alla realtà. Nel caso in cui si vogliano classificare più elementi in un'immagine servirebbero esempi contenenti solamente le singole classi tuttavia questo non è sempre possibile come

nel caso "macchina" e "strada". Per ridurre l'ambiguità vengono sfruttate le relazioni spaziali e delle comparazioni quali più grande, più luminoso.
Come ultimo esempio, le relazioni vengono sfruttate in [12] per rilevare le interazioni tra uomo e oggetto all'interno di alcuni sport per capire quale attività sia presente nell'immagine. Anche in questo caso le relazioni sono spaziali e in ogni immagine è presente un singolo oggetto (racchetta da tennis, mazza da golf) con cui la persona interagisce.
In questi primi lavori le relazioni, per lo più spaziali, sono state sfruttate per ottenere miglioramenti in vari ambiti quali object detection, visual classifier e human object interactions.

3 Visual Phrase e Scene Graph

In questa sezione verranno presentati la Visual Phrase e lo Scene Graph. In entrambi è possibile trovare vari collegamenti con la VRD.

3.1 Visual Phrase

Visual Phrase rappresenta una versione precedente della VRD utilizzata per la rilevazione di relazioni nella forma (soggetto-predicato-oggetto) come (persona-cavalca-cavallo). Mentre la VRD cerca di identificare il soggetto, l'oggetto e la relazione che li lega, la visual phrase cerca di identificare la regione dell'immagine in cui si svolge l'azione.

Figura 4: visual phrase (sinistra) e visual relationship detection (destra) a confronto [13]

In [14]-[15] la visual phrase viene definita come una via di mezzo tra il rilevamento di oggetti e le scene. Una scene può essere definita come "a view of a real-world environment that contains multiples surfaces and objects, organized in a meaningful way" [16]. Prendendo in considerazione l'esempio precedente un approccio potrebbe essere quello di rilevare i componenti della relazione (persona, cavallo) e poi descrivere la relazione. Questa scelta viene motivata dal numero minore di campioni disponibili contenente tutti gli elementi della relazione contro il numero di campioni degli elementi presi singolarmente. Tuttavia la forma di elementi diversi facenti parte di una relazione può cambiare notevolmente. Al contrario di quanto si potrebbe pensare, creare un modello che riconosca una specifica relazione può richiedere un quantitativo di campioni minore rispetto ai campioni necessari per allenare il modello a riconoscere i componenti della

relazioni. Questo avviene perché in una relazione le pose che possono assumere i suoi componenti sono un sottoinsieme delle pose che questi possono assumere presi singolarmente. Inoltre, la presenza di una relazione può celare parti degli oggetti interessati. In (persona-cavalca-cavallo) parti della persona possono essere nascoste dal cavallo.

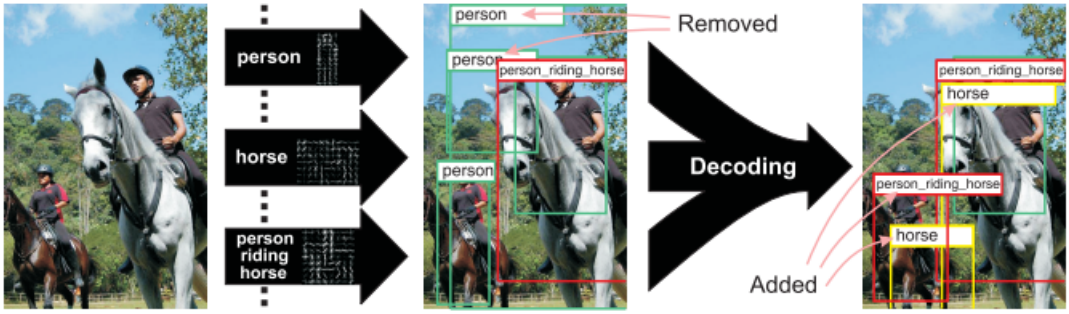

Figura 5: Differenza tra object detection (centro) e visual phrase detector (destra) [14]

La figura 5 confronta i due approcci appena descritti. Nella seconda immagine si è cercato di rilevare le persone e i cavalli per poi cercare la relazione. La terza immagine invece mostra come la visual phrase abbia rilevato tutti gli elementi dell'immagine ed entrambe le relazioni presenti.

3.2 Scene Graph

Dalla definizione di scena, è stata ideata una struttura dati per descrivere il contenuto della scena chiamata scene graph [17] definito come $G = (O, E)$ in cui $O = \{o_1, ..., 0_2\}$ è l'insieme degli oggetti, $E \subseteq O \times R \times O$ è l'insieme degli archi nella forma (Oggetto-Relazione-Oggetto). Ogni oggetto o_i è formato da una coppia (c_i, A_i) dove $c_i \in C$ è la classe dell'i-esimo elemento e $A_i \in A$ è l'insieme dei suoi attributi.
La scene graph verrà approfondita nelle successive sezione.

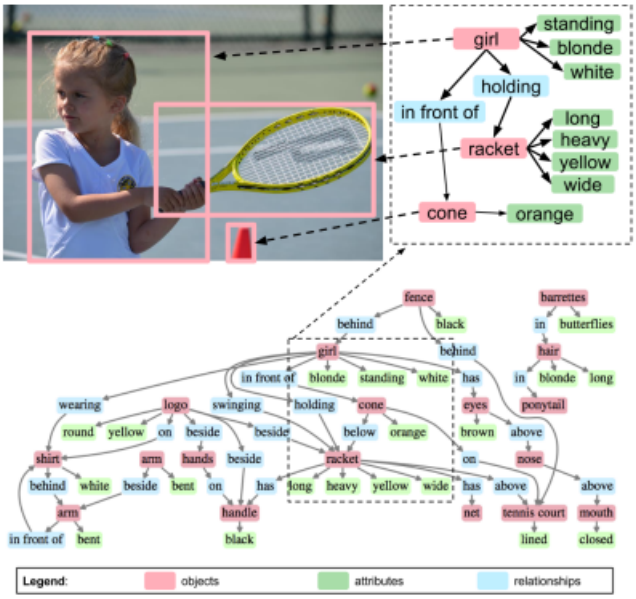

Figura 6: Esempio di scene graph in [17]

Lo scene graph è uno strumento molto potente per rappresentare un'immagine perché rende possibile concentrasi sia sulle singole regioni dell'immagine tramite sottografi che sull'immagine nella sua interezza.

4 Tecniche

In questa sezione verranno mostrate varie tecniche sviluppatesi per ottenere migliori risultati nella rilevazione di relazioni rispetto ai primi approcci precedentemente mostrati. Tutte partono dalla object detection, sfruttando lo stato dell'arte per ottenere i migliori rilevamenti possibili e dalle reti neurali convoluzionali (CNN) per esplorare le coppie di oggetti.

4.1 Prior and Posterior Statistics

Come si evince in [18] una delle principali problematiche della VRD è la possibilità di incontrare relazioni poco o per nulla frequenti nei dati utilizzati per allenare il modello.

Figura 7: Esempi di relazioni tra persona e bicicletta [18]

Dati N oggetti e K relazioni il numero di relazioni è $O(N^2K)$. Per rilevare tutte le relazioni ci servirebbe un ugual numero di rilevatori. Il modello in [18] utilizza $O(N + K)$ rilevatori ovvero uno per ogni classe di oggetti e uno per ogni tipo di relazione. Inoltre alcune relazioni sono semanticamente collegate quali (persona-cavalca-cavallo) e (persona-cavalca-elefante). Oltre a condividere l'azione, in entrambe l'oggetto della relazione è un animale. Queste informazioni possono essere sfruttate per cercare di risolvere il problema delle relazioni poco frequenti.

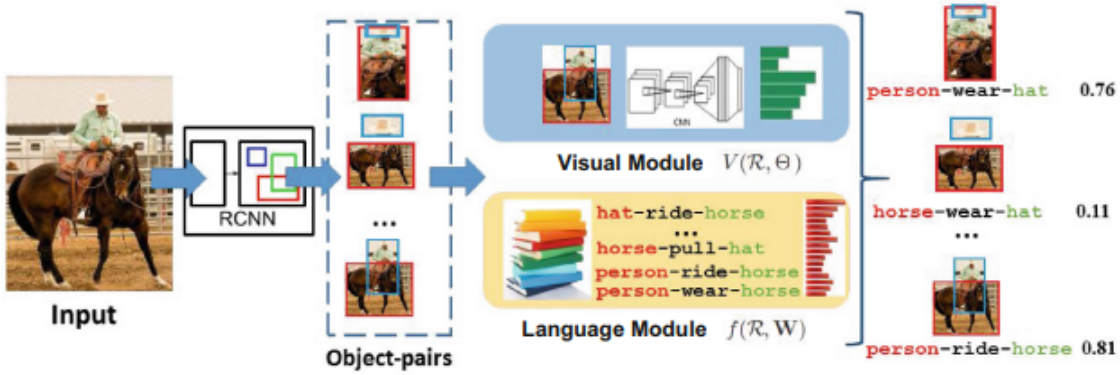

Figura 8: Panoramica modello [18]

Il modello in figura 8 prende come input un'immagine e tramite una Region Based Convolutional Neural Networks (RCNN) vengono create le possibili relazioni. Queste vengono poi valutate tramite i due moduli e i rispettivi parametri (Θ e W) per classificare le relazioni trovate.

Due CNN nel Visual Module sono stata allenate per classificare tutti gli N oggetti (100) la prima e tutte le K relazioni (70) la seconda utilizzando l'unione delle due bounding box facenti parte della relazione. La funzione V viene definita come:

$$V(R_{\langle i,k,j\rangle}, \Theta | \langle O_1, O_2 \rangle) = P_i(O_1)(z_k^T CNN(O_1, O_2) + s_k)P_j(O_2)$$

$R_{\langle i,k,j\rangle}$ rappresenta la relazione effettiva k tra gli oggetti di classe i e j. O_1 e O_2 rappresentano le bounding box di i e j mentre $P_i(O_1)$ e $P_j(O_2)$ indicano la probabilità di categorizzare O_1 e O_2 come elementi di classe i e j. z_k^T e s_k sono parametri per convertire caratteristiche rilevate dalla CNN in probabilità riguardanti la relazione k. Infine $CNN(O_1, O_2)$ rappresenta la relazione trovata dalla CNN dall'unione delle 2 bounding box.

Il Language Module permette, tramite la funzione di proiezione f, di rappresentare come due oggetti interagiscano tra loro. f viene definita come:

$$f(R_{\langle i,k,j\rangle}, W) = w_k^T[word2vec(t_i), word2vec(t_j)] + b_k$$

dove W rappresenta l'insieme $\{\{w_1, b_1\}, ..., \{w_k, b_k\}\}$ in cui ogni elemento indica una delle possibili relazioni, t_i rappresenta la i-esima classe di oggetti e $word2vec()$ proietta

i vari oggetti trasformati in vettori,in uno spazio in cui questi si trovano vicini fra loro se semanticamente simili. L'output di f mostra la probabilità della relazione.
La probabilità a priori è stata applicata per fare in modo che una relazione più frequente nei dati in allenamento abbia una probabilità a priori più alta nella fase di testing.
Il modello finale viene poi testato utilizzando i parametri ottenuti dai due moduli Θ e W e le bounding box $\langle O_1, O_2 \rangle$ ottenute tramite la RCNN per ottenere la relazione R che massimizza:

$$R^{\star} = arg \max_{R} V(R, \Theta | \langle O_1, O_2 \rangle) f(R, W)$$

Un esempio di statistica a posteriori può essere trovato in [19]. Gli oggetti rilevati vengono accoppiati come possibili candidati per una relazione. Poiché il numero di coppie $n(n-1)$ comporterebbe un costo computazionale troppo elevato vengono applicati dei filtri basati sulle posizioni e sulle classi dei due oggetti. Solitamente oggetti troppo lontani così come alcune classi non formano relazione significative. Le varie coppie rimanenti verranno utilizzate come input di Joint Recognition module per ottenere una tripla nella forma (soggetto-relazione-oggetto) come output. Questo modulo utilizza informazioni della bounding box contenente entrambi gli oggetti per ottenere le relazioni statistiche tra soggetto, relazione e oggetto. Queste vengono utilizzate per ottenere le classi più probabili per ognuno dei 3 elementi della tripla. La probabilità a posteriori della relazione è la seguente:

$$q_r^{'} = \sigma(W_r x_r + W_{rs} q_s + W_{ro} q_o)$$

in cui σ rappresenta la funzione di attivazione softmax, W indica i parametri del modello, x_r rappresenta le informazioni della bounding box contenente entrambi gli oggetti e q_r, q_s e q_o rappresentano le probabilità attuali di relazione, soggetto e oggetto. Si possono quindi ottenere le probabilità a posteriori come:

$$q_s^{'} = \sigma(W_a x_s + W_{sr} q_r + W_{so} q_o)$$

$$q_r^{'} = \sigma(W_r x_r + W_{rs} q_s + W_{ro} q_o)$$

$$q_o^{'} = \sigma(W_a x_o + W_{os} q_s + W_{or} q_r)$$

4.2 Knowledge Regularization

Con regolarizzazione si intende un insieme di tecniche utilizzate per ridurre la complessità di un modello per limitare il problema dell'overfitting. Overfitting avviene quando un modello si comporta bene con i dati utilizzati per la fase di allenamento ma fallisce nel generalizzare ottenendo risultati negativi nella fase di testing.
Un possibile metodo per ottenere conoscenze linguistiche consiste nel calcolare la probabilità condizionata $P(pred|subj, obj)$. La mancanza di molte relazioni nella fase di allenamento ridurrebbe l'utilità della conoscenza linguistica. Viene proposto in [20] di utilizzare una fonte esterna per ottenere più informazioni linguistiche sulle relazioni non presenti. Il modello sfrutta la knowledge distillation tra due reti seguendo un approccio maestro allievo. L'idea consiste nel utilizzare sia una rappresentazione fisica (un'immagine) che una rappresentazione linguistica(testo) per regolarizzare il processo di apprendimento e ottenere un modello migliore nella previsione e nella generalizzazione.

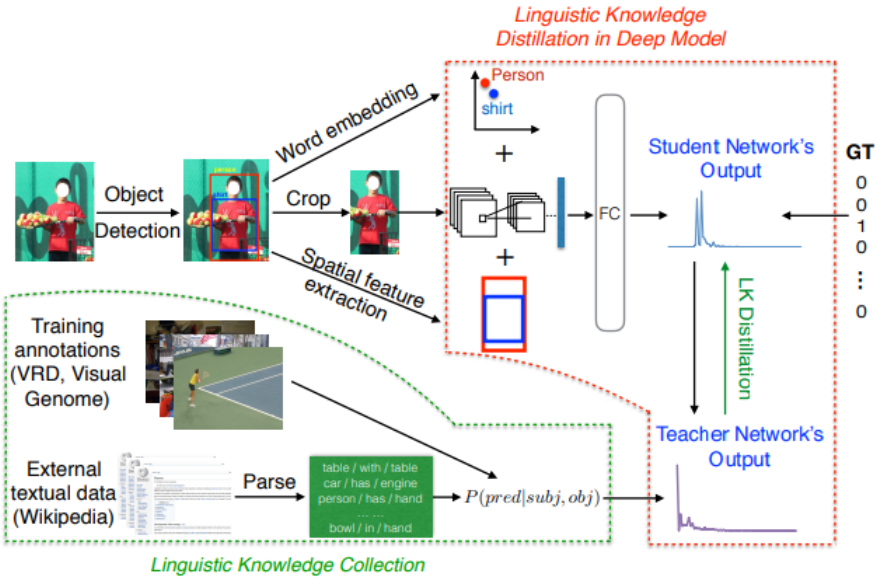

Figura 9: Modello Maestro-Allievo [20]

Partendo da un'immagine, questa viene divisa in 3 parti: l'immagine ritagliata in cui sono presenti il soggetto e l'oggetto, le loro classi, rappresentate tramite vettori, e le rispettive bounding box nell'immagine ritagliata. Questa tripla viene concatenata e

passata insieme al valore reale (GT) alla rete studente S-Net. Alla rete maestro T-Net viene data in input la probabilità condizionata $P(pred|subj, obj)$ ottenuta da informazioni testuali interne quali annotazioni dei dati usati per allenare il modello e dati esterni presi da Wikipedia. In particolare sono state estratte 4 miliardi di parole e 450 milioni di frasi.
La funzione obiettivo premia soluzioni probabili e punisce soluzioni improbabili fornite da S-Net riguardanti la coppia soggetto-oggetto. Prendendo la coppia piatto-tavolo e la probabilità condizionata $P(pred|piatto, tavolo)$ T-Net premierà predicati probabili come "sopra" e penalizzerà "in". S-Net viene quindi allenata sia utilizzando le etichette corrispondenti alle vere relazioni nell'immagine (GT) sia tramite T-Net. Quest'ultima essendo costruita sopra S-Net migliorerà le proprie previsioni grazie al miglioramento della rete studente.
Anche con l'aggiunta di una fonte esterna, numerose relazioni non hanno una rappresentazione per allenare il modello. Una soluzione proposta in [21] utilizza la vicinanza semantica dei predicati. Due predicati vengono definiti simili semanticamente se compaiono nello stesso contesto, definito in questo caso dalle coppie soggetto-oggetto. Il modello in questione premia predicati simili al valore di verità proposti ottenendo prestazioni migliori.

4.3 Translation Embedding

Due relazione contenenti lo stesso predicato possono apparire molto diverse tra loro rendendo ancora più difficile allenare un modello. L'idea in [22] prevede di rappresentare le caratteristiche degli oggetti e delle relazioni in vettori di dimensionalità minore tramite la tecnica dell'embedding vista con word2vec per semplificare il numero di computazioni da eseguire rendendo il modello più veloce da allenare.

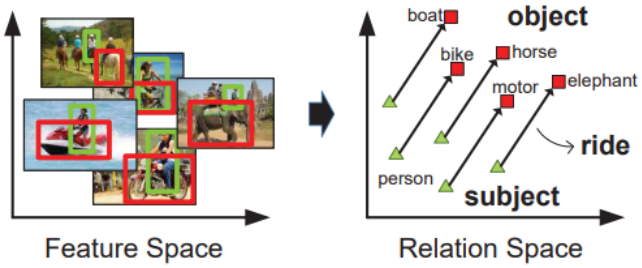

Figura 10: Esempio di translation embedding [22]

In quello che viene definito "Relation Space" in Figura 10, la relazione espressa dalla tripla (subject-ride-object) può essere espressa come una traslazione del vettore "ride" senza doversi preoccupare di quanto possano variare le varie coppie di soggetto e oggetto. Dall'immagini vengono estratte 3 caratteristiche relative alle classi, la posizione e alcune informazioni visive per il soggetto e l'oggetto.
I vettori ridotti vengono indicati con s, p e o dove $s + p \approx o$ rappresenta una traslazione valida, altrimenti vale $s+p \neq o$. Definendo x_s,$x_o \in R^M$ come le caratteristiche di soggetto e oggetto e $t_p \in R^r (r \ll M)$ indica il vettore per traslare la relazione. Vengono infine utilizzate due matrici per proiettare le caratteristiche da "Feature Space" a "Relation Space" definite rispettivamente $W_s, W_0 \in R^{r \times M}$. Si ottiene quindi:

$$W_s x_s + t_p \approx W_o x_o$$

4.4 Attention Models

"Attention is a complex cognitive function that is indispensable for human beings. One important property of perception is that humans do not tend to process whole information in its entirety at once. Instead, humans tend to selectively concentrate on a part of the information when and where it is needed, but ignore other perceivable information at the same time" [23].
In [24] viene sfruttato il contesto in cui si svolge una relazione assumendo che venga rilevato da un rilevatore. Due approcci, come visto in altri esempi, sono di costruire un classificatore per ogni relazione cercata oppure utilizzare la tripla (soggetto-predicato-oggetto) come una classe e costruire un classificatore per ogni combinazione possibile.

Entrambi gli approcci non sono privi di problemi. Il primo non sfrutta pienamente le informazioni contestuali mentre il secondo soffre di scalabilità. [24] cerca di posizionarsi al centro dei due approcci. Viene costruito un classificatore per ogni relazione, come nel primo, e si utilizzano parametri capaci di adattarsi al contesto in cui si sta svolgendo la relazione, come nel secondo.

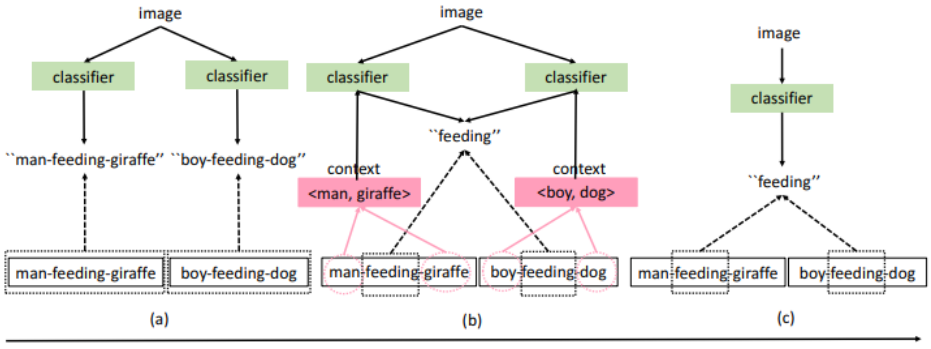

Figura 11: La figura (b) mostra la tecnica ideata in [24]

Il contesto viene codificato tramite word2vec in uno spazio semantico per ricavare un risultato nella classificazione della relazione.
L'attenzione può essere spostata tra vari elementi in un'immagine. Nel lavoro [25] viene realizzato un modello per cercare di disambiguare tra elementi della stessa classe, quali stiano effettivamente partecipando all'azione. Un esempio può essere un calcio di rigore. Nella stessa immagine possono apparire due persone ma solo una di queste sta eseguendo il tiro. Le due problematiche da affrontare sono: la difficoltà nel rilevare oggetti non è costante tra le varie classi e come i predicati possano apparire in forme diverse in base ai due elementi che collegano. Per il primo problema è stato deciso di fornire sempre la posizione dell'altro elemento appartenete alla relazione (se stessimo cercando la persona che calcia la palla, questa verrà evidenziata come input del rilevatore e viceversa). Per il secondo punto si utilizzano i predicati per spostare l'attenzione dal soggetto all'oggetto e viceversa. Le relazioni spaziali sono quelle più semplici da implementare. Se una relazione viene espressa come (X-sopra-Y) e stessimo cercando (X) sapendo la posizione di (Y) basterebbe controllare nella porzione dell'immagine superiore rispetto (Y).
Anche in [26] si cerca di utilizzare il contesto per risolvere il problema della "referring

expression comprehension". Con "referring expression" si intende una frase in linguaggio naturale che fa riferimento a uno specifico oggetto visibile in un'immagine. Viene costruito un grafo diretto dove i nodi sono gli oggetti e gli archi rappresentano le relazioni. Sopra a questo grafo si costruisce un "language-guided graph attention network" (LGRAN). I nodi servono per evidenziare elementi importanti per la relazione mentre gli archi vengono utilizzati per dividere le relazioni in due gruppi. Il primo gruppo identifica relazioni tra elementi della stessa classe mentre il secondo per relazioni tra elementi di classi diverse.

Un approccio tramite grafo in cui i nodi e gli archi vengono utilizzati per rappresentare oggetti e relazioni non tiene in considerazione la probabilità che alcune relazioni hanno di essere entrambe presenti all'interno di un'immagine non condividendo nessun elemento [27]. La relazione (persona-sopra-bicicletta) ha una maggiore probabilità di accadere nello stesso contesto di (macchina-sopra-strada) che (elefante-sopra-erba). Non potendo modellare questa probabilità con un semplice grafo, viene costruito in [27] un "Hierarchical Graph Attention Network" per rappresentare questa dipendenza tra relazioni in cui i nodi rappresentano le relazioni e un arco collega direttamente le relazioni che condividono il soggetto o l'oggetto.

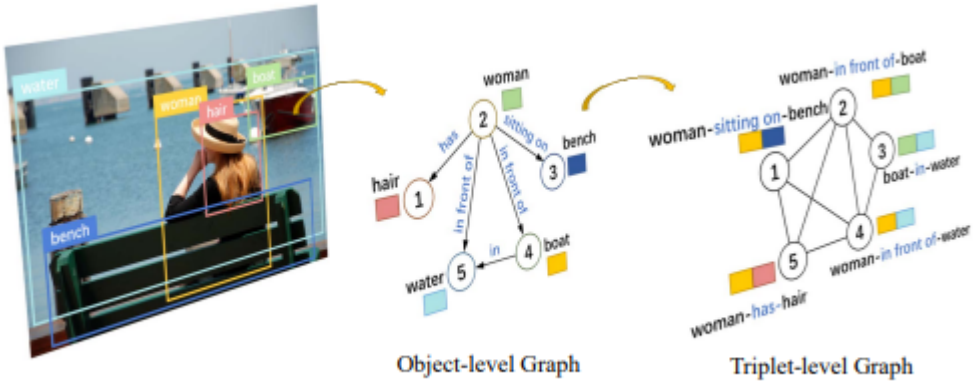

Figura 12: Grafo gerarchico [27]

4.5 Targeted Cost and Losses

La scelta di una o più funzioni di costo permettono al modello di concentrarsi su diversi tipi di relazione. Di seguito verranno descritti alcuni approcci.
Una problematica non ancora affrontata riguarda l'utilizzo di immagini non annotate completamente. I 3 problemi principali sono la mancata rilevazione di tutti gli oggetti nell'immagine, l'assenza di relazioni registrate per coppie di elementi in situazioni in cui esista un' effettiva relazione e l'unidirezionalità con cui vengono annotate le relazioni. In [28] vengono utilizzate informazioni spaziali e testuali per mitigare il problema. Data un'immagine x come input vengono estratte le caratteristiche spaziali relative alla posizione e alla vicinanza ad altri elementi e le caratteristiche linguistiche per affrontare le varie forme in cui una relazione può presentarsi. Queste caratteristiche vengono infine concatenate. Questo procedimento viene indicato con $f(\cdot)$. Partendo da una relazione $r = (s, p, o)$ possiamo definirla come $f(x, s, o)$. Successivamente si definisce una funzione per calcolare la compatibilità tra x e r come:

$$\phi(x, r) = \phi(x, s, p, o) = w_p^T f(x, s, o)$$

dove w_p indica i parametri relativi al p-esimo predicato.
Solitamente le relazioni annotate sono le più significanti. Per evidenziare ciò viene definita una funzione di costo L come:

$$L(x) = \sum_{r \in R} \sum_{r \in R'} [\triangle(r, r^{'}) + \phi(x, r^{'}) - \phi(x, r)]_+$$

dove R e $R^{'}$ rappresentano l'insieme delle relazioni annotate e l'insieme delle relazioni non annotate. $[\cdot]_+ = max(0, \cdot)$ e $\triangle(\cdot, \cdot)$ è una funzione per gestire le relazioni non annotate definita come:

$$\triangle(r, r^{'}) = \triangle(s, p, o, s^{'}, ^{'} p, o^{'}) = 1 + P(p|c_s, c_o) - P(p^{'}|c_s^{'}, c_o^{'})$$

Per ottenere relazioni più affidabili viene creato un modello basato sulla relazione posizionale [29]. Questa relazione si basa su 3 attributi: dimensione, forma e distanza. Definita $A(x)$ l'area di x e r_1, r_2 due regioni viene calcolato il rapporto tra $\frac{A(r_1)}{A(r_2)}$ per

misurare la relazione relativa alla dimensione. Nella relazione (soggetto-sopra-oggetto) solitamente il soggetto è più piccolo dell'oggetto. La forma viene calcolata usando l'altezza e la larghezza delle due regioni. Nella relazione sopra descritta succede spesso che l'altezza del soggetto sia minore rispetto all'oggetto. Infine la distanza viene calcolata usando il rapporto tra l'intersezione delle due regioni e la loro unione. Prendendo ancora una volta in considerazione la relazione precedente accade che la distanza sia maggiore tra i due elementi rispetto alla relazione (soggetto-tiene-oggetto).
In [30] vengono implementate varie funzioni di costo per affrontare 2 problemi: "Entity Instance Confusion" e "Proximal Relationship Ambiguity". Il primo si riferiscono alla difficoltà del modello in presenza di più istanze appartenenti alla stessa classe, in cui solo una è collegata al soggetto/oggetto, di disambiguare l'istanza corretta. Il secondo avviene quando sono presenti varie coppie collegate dallo stesso predicato e il modello fallisce nell'abbinare correttamente soggetto e oggetto.

4.6 RL-Based Framework

L'obiettivo di un modello basato sul reinforcement learning consiste nel trovare la soluzione migliore tra una serie di proposte. In [31] viene costruito un grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ dove $\mathcal{V}$ indica l'insieme dei nodi composto da nomi, attributi e predicati. $\mathcal{E}$ è l'insieme degli archi relativi agli attributi $\mathcal{E}_A \subseteq C \times A$ e ai predicati $\mathcal{E}_P \subseteq C \times P \times C$. Se un arco $(c, a) \in \mathcal{E}_A$ significa che a è attributo del nome c. $\mathcal{G}$ rappresenta l'insieme di tutte le azioni tuttavia solo un sottoinsieme di questo è valido per ogni iterazione. Data un'immagine tramite un rilevatore vengono identificati un insieme $\mathcal{S}$ di possibili elementi. Per ogni $s \in \mathcal{S}$ viene assegnata una classe di appartenenza $s_c \in C$ e una bounding box $B(s)$. Definito $s^{'}$ un oggetto dove $s \neq s^{'}$ il modello in [31] prende come input s, $s^{'}$ e i cammini precedentemente esplorati. Vengono costruiti 3 insiemi $\triangle_a, \triangle_p, \triangle_o$. L'agente cercherà di predire un attributo $g_a \in \triangle_a$ di s, il predicato tra s e $s^{'}$ $g_p \in \triangle_p$ e il prossimo elemento con cui aggiornare $s^{'}$ $g_c \in \triangle_o$. Il passo successivo prenderà in input s, il nuovo $s^{'}$ e l'insieme dei cammini esplorati aggiornato.

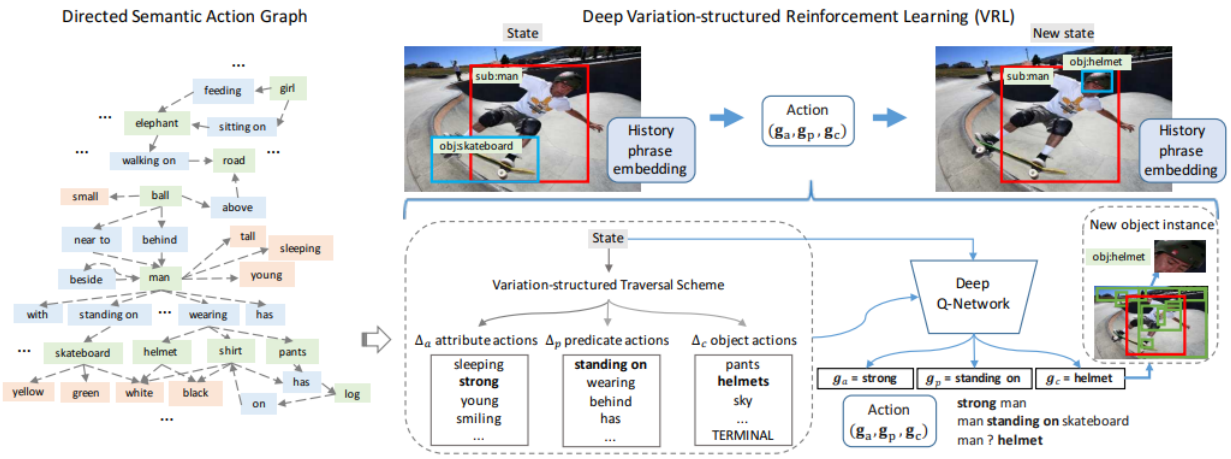

Figura 13: Modello [31]

Deep Q-Netwnork è un modello ideato in [32] per sviluppare un agente capace di giocare a 7 giochi sviluppati per l'Atari prendendo come input un'immagine rappresentante la schermata corrente. DQ-Net viene utilizzata per calcolare i pesi degli attributi θ_a, predicati θ_p e classi θ_c. Supponendo di avere i veri valori dei nodi e degli archi definiti come $\hat{S}, \hat{\mathcal{E}^{\mathcal{A}}}, \hat{\mathcal{E}^{\mathcal{P}}}$ un elemento $s \in S$ e il corrispettivo valore reale $\hat{s} \in \hat{S}$ si dicono sovrapponibili se appartengono alla stessa classe e il rapporto tra l'intersezione e l'unione delle loro bounding box sia almeno pari a 0.5. Indicando con f lo stato vengono definite le seguenti funzioni usate come premio per una scelta corretta del modello:

1. $R_a(f, g_a)$ restituisce +1 se esiste un $\hat{s} \in \hat{S}$ sovrapponibile a s e $(s_c, g_a) \in \hat{\mathcal{E}^{\mathcal{A}}}$. -1 altrimenti.

2. $R_p(f, g_p)$ restituisce +1 se esistono $\hat{s}, \hat{s'} \in \hat{S}$ sovrapponibile a s e s' e $(s_c, g_p, s_c') \in \hat{\mathcal{E}^{\mathcal{P}}}$. -1 altrimenti.

3. $R_c(f, g_c)$ restituisce +5 se il successivo elemento $\bar{s} \in S$ appartiene alla classe $g_c \in C$ e sovrapponibile a $\hat{s} \in S$. -1 altrimenti.

Nei test si utilizza l'azione migliore in $\triangle_a, \triangle_p, \triangle_c$. L'agente continua finché non esaurisce gli elementi non esplorati nell'immagine oppure quando raggiunge il numero massimo di passi.

4.7 Graph Parsing

Anche le Scene Graph descritte nella sezione 3 sono state utilizzate per risolvere il problema della VRD.
In [33] viene proposto un modello per risolvere in contemporanea object detection, VRD e region captioning.

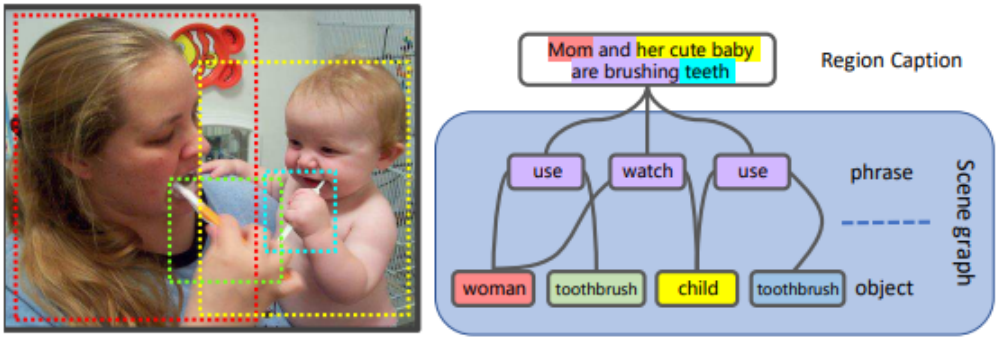

Figura 14: I 3 problemi affrontati in [33]

L'idea alla base consiste nello sfruttare le informazioni ottenute in una delle 3 attività per meglio risolvere le altre due. Un esempio di come le attività siano collegate può essere trovata nella frase (donna-guarda-bambino). Questa assicura l'esistenza di donna e bambino nell'immagine. La struttura del modello può essere riassunta in 4 punti:

1. Region Proposal: vengono generati tre insiemi rappresentanti oggetti, relazioni e didascalia dell'immagine fornita in input.

2. Feature Specialization: i tre insiemi vengono suddivisi per poter ottenere le caratteristiche relative ai 3 problemi che si vuole affrontare.

3. Dynamic Graph Construction: per unire le 3 attività viene generato dinamicamente un grafo. Gli oggetti vengono collegati alle relazioni tramite due archi diretti. Le relazioni vengono collegate alle didascalie tramite un arco senza direzione se la didascalia proposta copre una porzione sufficiente (0.7 come valore soglia) della relazione proposta.

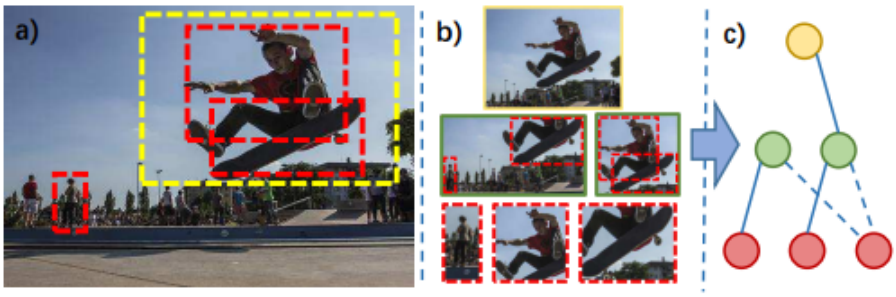

Figura 15: a) rappresenta l'immagine in input. b) mostre le proposte per la didascalia (alto), per le relazioni (centro) e per gli oggetti (basso). c) è il grafico generato dalle connessioni tra le proposte [33]

4. Scene Graph Generation: tramite una matrice viene rappresentato lo scene graph dove (i, i) indica l'i-esimo elemento sulla diagonale, (i, j) con $i \neq j$ la relazione tra l'i-esimo e il j-esimo elemento. Ogni elemento i può rappresentare un'istanza di una classe o (sfondo). Per le relazioni i valori possibili sono un'istanza delle relazioni predefinite oppure (irrilevante). Il grafo viene quindi costruito nel seguente modo: se i e j non sono classificati come (sfondo) e la relazione (i, j) non viene classificata come (irrilevante) viene creato un legame tra i due nodi rappresentanti gli oggetti i e j.

Una possibile ottimizzazione nell'utilizzo degli scene graph riguarda l'utilizzo di motivi: sotto-strutture ricorrenti negli scene graph [34]. In particolare, conoscendo uno dei due elementi aumenta la probabilità della previsione della relazione e dell'altro elemento. Conoscere la relazione invece fornisce poche informazioni aggiuntive.

5 Applicazioni

Di seguito verranno mostrate alcune applicazioni delle VRD.

5.1 Visual Reasoning

"The ability to reason about the relations between entities and their properties is central to generally intelligent behavior" [35]
La visual reasoning (VR) si occupa di dedurre informazioni implicite all'interno di un'immagine come contare le occorrenze di un certo oggetto o indicare quale elemento possegga una certa proprietà.
In [35] viene proposto un modello relazionale. Il modello può essere descritto come:

$$RN(O) = f_\phi(\sum_{i,j} g_\theta(o_i, o_j))$$

dove $O = \{o_1, o_2, ..., o_n\}$, o_i è l'i-esimo oggetto, ϕ e θ sono i parametri di f e g. g_θ viene utilizzata per dedurre se i due oggetti sono collegati tra loro. Il suo output è una relazione.
Superficialmente partendo da un'immagine x e una domanda q sull'immagine come input viene restituita una risposta $a \in A$ da un insieme fissato A di risposte possibili. Un possibile approccio consiste nel utilizzare 2 moduli: un generatori di programmi, $z = \pi(q)$, e un motore per eseguirli $a = \phi(x, z)$ [36]. I programmi hanno una sintassi definita e una semantica per specificare il comportamento del programma. Un programma z viene rappresentato tramite un albero sintattico dove ogni nodo rappresenta una funzione f appartenente a un insieme di funzioni predefinite F. Ognuno di questi nodi ha un numero di figli pari all'arietà della funzione $n_f \in \{1, 2\}$. Il generatore di programmi trasforma, tramite Long short-term memory (LSTM), la domanda intesa come una sequenza di parole in un programma ovvero una sequenza di funzioni. Il motore d'esecuzione crea un modello dove ogni modulo corrisponde a una funzione del programma dato come input. Il dataset utilizzato in questi esempi è Compositional Language and Elementary Visual Reasoning diagnostics dataset (CLEVR) [37]. All'interno del dataset vengono utilizzate 3 forme (cubi, sfere, cilindri) suddivisi per dimensione (piccola, grande) e materiale

(metallo, gomma). Le forme possono assumere 8 colori diversi. Le relazioni spaziali tra gli oggetti sono sinistra, destra, davanti e dietro.

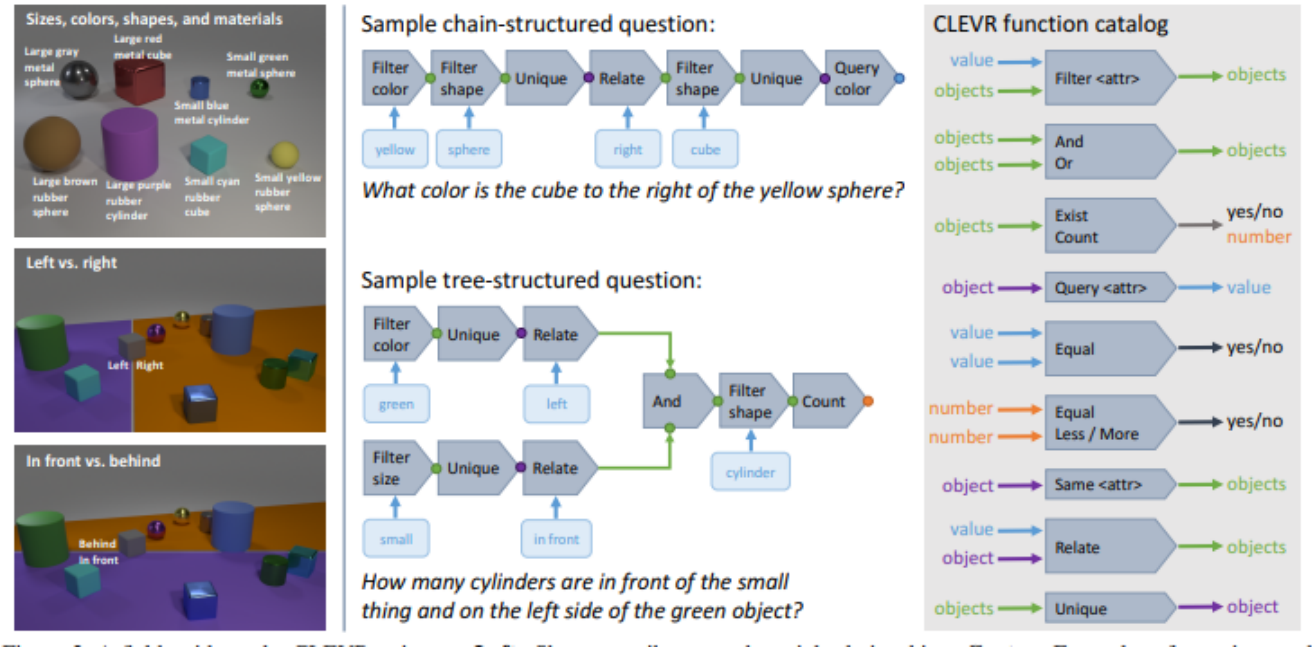

Figura 16: Sinistra: attributi e relazioni degli oggetti utilizzati. Centro: esempi di domande e programmi associati. Destra: funzioni utilizzate per comporre i programmi [37]

5.2 Visual Question Answering

Visual Question Answering (VQA) tratta un problema di classificazione formulato come:

$$\hat{a} = arg\max_{a \in \Omega} p(a|Q, I : \Theta)$$

dove $\hat{a}$ è la risposta più probabile, Q, I, Θ vengono utilizzate per rappresentare una domanda, l'immagine di riferimento e i parametri del modello [38].
Utilizzando il VQA Dataset, il quale contiene 614,163 domande, 7,984,119 risposte riguardanti 204,721 immagini prese dal dataset di Microsoft COCO [39] in [40] viene proposto un modello per rispondere a varie domande utilizzando da 1 a 3 parole. Le domande vengono raggruppate in base alle prime 4 parole della domanda.

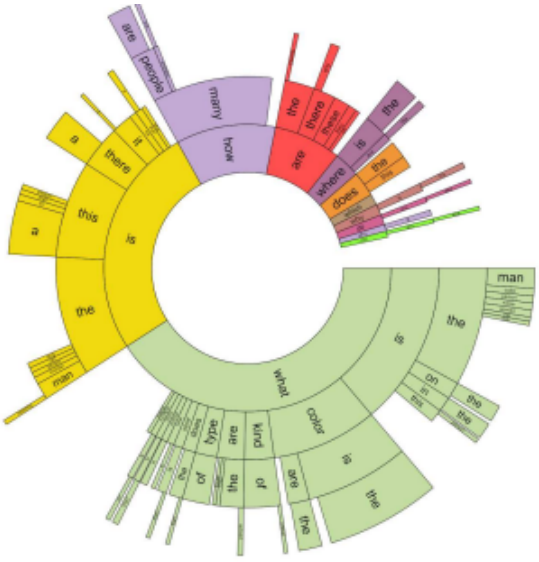

Figura 17: La prima parola della domanda si trova nella sezione più interna e prosegue verso l'esterno. La lunghezza dell'arco è proporzionale al numero di domande aventi quella parola al loro interno [40]

La maggior parte delle risposte è composta da una singola parola. Le distribuzioni delle risposte composte da 1, 2 o 3 parole sono rispettivamente 89.32%, 6.91%, e 2.74%. Sempre in [40] si sono chiesti quale sia l'importanza dell'immagine nel fornire la risposta chiedendo a 3 persone di rispondere a delle domande come "Di che colore è l'idrante".

Input	Tutte	Si\No	Numero	Altro
Domanda	40.81	67.60	25.77	21.22
Domanda + Didascalia	57.47	78.97	39.68	44.41
Domanda + Immagine	83.30	95.77	83.39	72.67

Tabella 1: Risposte corrette in percentuale in [40]

Come si può notare nella colonna "Altro" la sola conoscenza pregressa non basta per poter rispondere correttamente un numero sufficiente di volte e rimarca l'importanza di riuscire a comprendere il contenuto di un'immagine.
L'importanza delle relazioni nel VQA è evidente in [41] come si può evincere dalla *Fig.* 18. Alla domanda "Di quale colore è la maglietta?" le caratteristiche estratte dall'immagine

potrebbero bastare per fornire "rosso" come risposta. Rispondere alla domanda "L'uomo indossa un cappello?" è più complicato, specie se la relazione (Uomo-Indossa-Cappello) non viene rilevata.

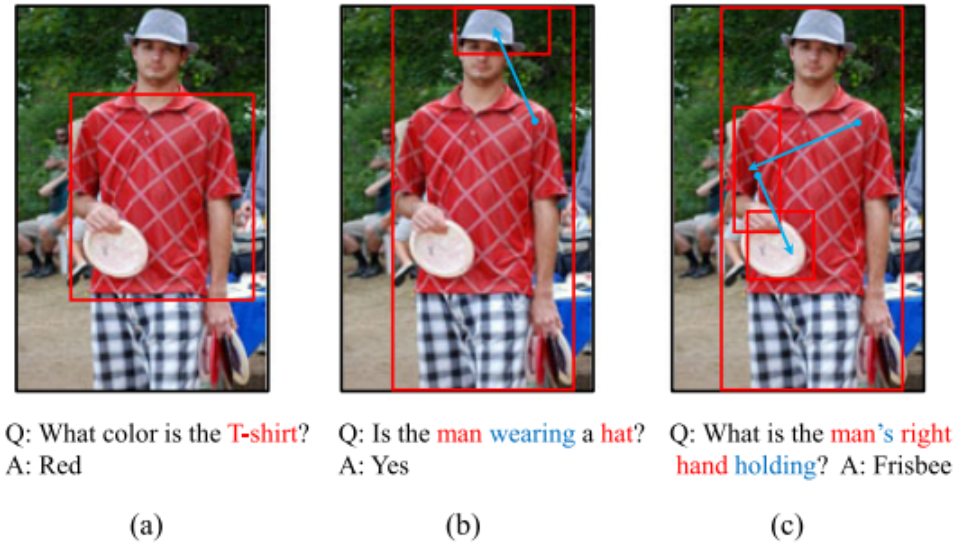

Figura 18: Le bounding box rosse rappresentano gli oggetti facenti parte della domanda mentre le linee blu indicano la relazione [41]

5.3 Human-Object Interactions

Lo scopo della Human-Object Interactions (HOI) riguarda il rilevamento della persona, dell' oggetto e delle interazioni nella forma (persona, predicato, oggetto) concentrandosi sulla persona [42]. Questo perché la posa assunta dalla persona può permettere di dedurre dove si trovi l'oggetto con cui sta interagendo. Nella HOI i predicati rappresentano sempre delle azioni. Il problema di localizzare la persona, l'oggetto e il predicato, viene scomposto in 3 moduli. Si parte da Fast R-CNN [9], la quale produce le bounding box e le rispettive etichette per indicare la classe di appartenenza delle previsioni s_h e s_o. Il secondo modulo assegna una valore relativo alla azione a svolta dalla persona con bounding box h indicata con s_h^a e la previsione della posizione dell'interazione μ_h^a utilizzando la posa della persona. L'ultimo modulo assegna un valore all'interazione a

tra h e o nel seguente modo:

$$S^a_{h,o} = s_h \cdot s_o \cdot s^a_h \cdot g^a_{h,o}$$

$g^a_{h,0}$ è la probabilità che l'oggetto o sia il vero oggetto con cui la persona interagisce nell'immagine.

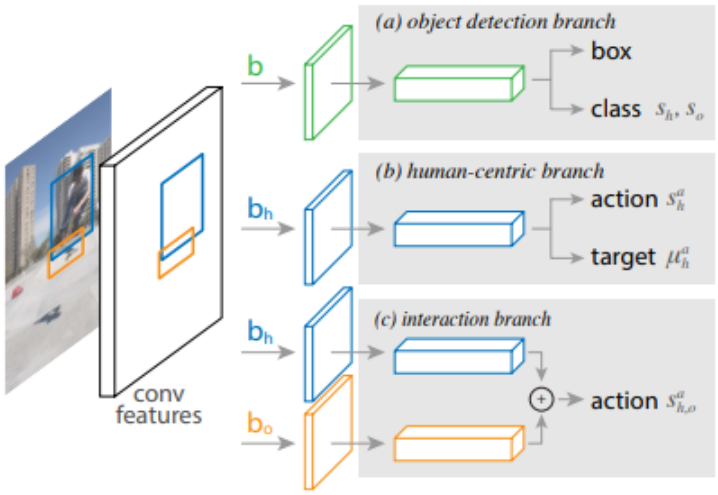

Figura 19: Rappresentazione dei 3 moduli[42]

Così come per la VRD, anche la HOI soffre il problema della mancanza di esempi per alcune interazioni. Un ulteriore approccio sfrutta uno spazio semantico nella forma di un knowledge graph per ottenere informazioni sulla relazione sfruttando i nodi vicini [43]. Il knowledge è un grafo orientato in cui gli archi sono etichettati. Un nodo può rappresentare una qualunque entità, in questo caso una persona o un oggetto e gli archi rappresentano relazioni tra due nodi [44].

5.4 Scene Graphs

Come mostrato nella sezione 3 il contenuto di una scena può essere descritto tramite un scene graph. Una applicazione dello scene graph viene proposta in [45] dove le relazioni vengono sfruttate per ottenere delle proposte migliori nel contesto della modellazione di una stanza 3D. Un artista ha a disposizione una stanza e vari modelli di mobili da poter inserire nella stanza. Capire quali oggetti si adattano meglio alla stanza creata

richiede un metodo per confrontare la scene corrente e scene già presenti nel database. Si parte dalla scena raffigurante la stanza e viene costruito uno scene graph. Le relazioni in questo caso sono spaziali e per ogni istanza viene costruito un arco con una etichetta per indicare il tipo. Per poter confrontare due scene graph servono altri due elementi:

1. Node Kernel: una funzione che prende come input due oggetti e restituisce un numero compreso tra 0 e 1 per indicare quanto essi siano simili, confrontando varie caratteristiche. I nodi dello scene graph oltre a possedere informazioni spaziali quali dimensioni e forma possiedono più classi di appartenenza. Una lampada da scrivania può avere una classe primaria lampada e una classe secondaria scrivania. Le 3 funzioni proposte confrontano l'identità, le classi e la composizione dei 2 oggetti.

2. Edge Kernel: una funzione per confrontare due archi utilizzando una funzione identità applicata agli archi.

Viene quindi costruito un graph kernel con input due grafi, Node Kernel e Edge Kernel e restituisce un valore per indicare quanto i due grafi siano simili.
In [46] viene proposto un modello per tradurre del testo scritto in linguaggio naturale in una scene 3D. Già nell'articolo precedente la capacità del modello di riconoscere come si relazionano gli elementi della scena tra loro era importante, ora è fondamentale. Partendo dal testo viene costruito uno scene graph analogo al precedente. Per favorire le relazioni vengono utilizzate delle informazioni spaziali. Ad esempio, un piatto non può fluttuare nella stanza ma richiede solitamente una struttura su cui appoggiarsi più grande del piatto stesso. Viene quindi sfruttata la probabilità a priori per calcolare:

1. Object Occurrence Priors: $P_{occ}(C_0|C_s) = \frac{count(C_0 \, in \, C_s)}{Count(C_s)}$
2. Support Hierarchy Priors: $P_{support}(C_p|C_c) = \frac{count(C_c \, on \, C_p)}{Count(C_c)}$
3. Support Surface Priors: $P_{surf}(S_n|C_c) = \frac{count(C_c \, on \, surface \, with \, S_n)}{Count(C_c)}$
4. Relative Position Priors: $P_{relpos}(x, y, \theta|(C_{obj}, C_{ref}, C_s, R)$

dove $C_p.C_s$ rappresenta la classe padre e la classe figlio, S_n è il vettore che può assumere un valore per ogni dimensione per un totale di 6. Il punto 4. calcola la posizione di una

istanza di classe C_obj sfruttando un altro oggetto C_ref e il tipo della scena C_s. R indica il legame tra i due oggetti e può assumere due valori: Sibling o ChildParent. Infine x, y, θ vengono utilizzati per indicare il centroide dell'oggetto cercato e l'angolo tra i due oggetti. Un esempio di scena generata parte dalla frase "Una stanza con una sedia e un computer". Il modello riesce a comprendere tramite questi vincoli genera anche una scrivania sulla quale posizionare il computer.

5.5 Scene Understanding

L'obiettivo della scene understanding consiste nel fornire a un computer la capacità di estrarre le stesse informazioni che una persona riuscirebbe a cogliere da una immagine[47]. Non tutte le scene sono uguali. Per prima cosa bisogna rilevare ogni elemento della scena. Se ad esempio una persona fosse presente nell'immagine, il solo rilevarla non sarebbe sufficiente per poter indicare che il modello ha capito cosa avviene nella scena. L'espressione, la posa, i vestiti sono tutte informazioni con cui noi umani possiamo dedurre cosa stia succedendo alla persona nell'immagine. Le relazioni sono un altro punto fondamentale e alcune di queste assumono importanza solo se confrontate con il resto della scena [48]. Un rilevatore potrebbe catturare tra i vari oggetti: una persona, un muro e come relazione la persona che salta oltre il muro. Queste informazioni, per quanto importanti, non bastano a spiegare perché la persona stia saltando oltre il muro. Anche eventuali scritte possono fornire informazioni importanti. In una scena il contenuto testuale può essere diviso in "Point and Shoot" se l'intenzione originaria dell'immagine era concentrasi sul testo e "Incidental Text" se questo si trova per caso nell'immagine. Il font e le dimensioni possono rendere particolarmente difficile l'azione di classificare la scritta.
Un'ulteriore difficoltà nella classificazione delle scene deriva dalla possibilità di avere scene semanticamente simili ma disposte in modo completamente diverso [49]. Il numero di oggetti e classi e le posizioni in cui questi possono apparire rende la classificazione un problema complicato.

6 Dataset

Per allenare un modello sono richiesti numerosi esempi con cui la rete possa apprendere vari schemi all'interno dei dati. Preparare a mano tutte le volte i dati sarebbe troppo oneroso in termini di tempo. Per questo motivo sono stati costruiti vari dataset per valutare le prestazioni dei vari modelli. Questa sezione verterà su una panoramica dei maggiori dataset utilizzati per la VRD.

6.1 Visual Phrases

Dal 2006 si sono svolte varie sfide riguardanti la capacità di riconoscere certi elementi all'interno di varie scene chiamate The Pascal Visual Object Classes (VOC) Challenge. In queste sfide viene fornito un dataset contenente immagini relative a varie classi e delle annotazioni riguardanti le soluzioni per testare i modelli proposti nelle sfide. Il Phrasal Recognition Dataset descritto in [14] utilizza 8 classi prese da Pascal VOC2008 [50]. Partendo dalle 8 classi (persona, bicicletta, macchina, cane, cavallo, bottiglia, divano, sedia) sono state aggiunte 17 visual phrase. Queste possono riferirsi a un'interazione tra 2 oggetti o su un singolo oggetto. Alcuni esempi di visual phrase nel dataset sono: bicicletta vicino alla macchina, persona che salta, persona beve dalla bottiglia. Le immagini sono state recuperate tramite il motore di ricerca Bing e le bounding box per le 8 classi e le 17 visual phrase sono state ottenute a mano su un totale di 2769 immagini. Con il crescere della complessità della visual phrase il numero delle immagini diminuisce.

6.2 Scene Graph

Real-World Scene Graphs Dataset è stato costruito in [17] selezionando manualmente 5,000 immagini prese dai dataset Miscosoft COCO [39] e YFCC100 [51]. Per ogni immagine è stato prodotto a mano il relativo scene graph. Le 5,000 immagini contengono più di 93,000 oggetti, 110,000 attributi e 112,000 relazioni. Questi numero fanno riferimento alle istanze trovate e non alle classi. Gli attributi possono rappresentare uno stato (ombrello è aperto), un materiale (tavolo di legno) o un colore (maglia gialla). Le relazioni possono rappresentare un'azione (persona guida moto) o un legame spaziale (piatto su tavolo).

6.3 Visual Relationship Dataset

Un dataset sulle relazioni deve catturare sia la posizione degli oggetti in un'immagine, come nei dataset precedenti, sia le relazioni che intercorrono tra le coppie di oggetti. Per questo motivo viene proposto in [18] il Visual Relationship Dataset. Esso contiene 5,000 immagini, 100 classi per gli oggetti e 70 predicati diversi. Nella sua totalità sono state rilevate 37,993 relazioni, 6,672 tipi di relazione e 24.25 predicati per classe di oggetto.

6.4 Visual Genome

Il dataset Visual Genome [52] è formato da 7 moduli:

1. region descriptions: vengono generate varie descrizioni a mano per le varie regioni dell'immagine. Ognuna di queste regioni possiede una bounding box con la possibilità di sovrapporsi tra loro nel caso le descrizioni siano diverse tra loro. Ogni immagine possiede circa 50 descrizioni e ogni descrizione è una frase composta da 1 fino a 16 parole.

2. objects: ogni immagine nel dataset contiene mediamente 35 oggetti indicati tramite bounding box. Queste sono solitamente più piccole rispetto a quelle delle regioni.

3. attributes: un oggetto può avere un certo numero di attributi. Un'immagine possiede mediamente 26 attributi i quali possono riguardare il colore, la forma, etc.

4. relationships: le relazioni collegano due oggetti e possono essere di vari tipi (spaziale, comparazione, descrittivo).

5. region graphs: per ogni regione viene costruito un grafo dove i nodi sono gli oggetti, gli attributi e le relazioni. Gli archi collegano gli oggetti ai loro attributi e le relazioni ai due oggetti coinvolti.

6. scene graphs: vengono uniti i vari region graph per formare un solo scene graph fondendo le informazioni.

7. question answer pairs: vengono create domande riguardanti l'immagini nella sua interezza e domande relative a singole regioni.

Da Visual Genome è stato costruito il dataset GQA [53] il quale viene utilizzato principalmente per la Visual Question Answering anche se viene posta particolare attenzione alle relazioni.

Components\Dataset	Visual Phrases	Scene Graph	VRD	Visual Genome
Images	2,769	5,000	5,000	108,077
Objects Categories	8	266	100	33,877
Visual Phrases	17	-	-	-
Relationship Types	13	23,190	6,672	42,374
Relationship Instances	-	109,535	37,993	2,269,617
Attribute Categories	-	145	-	68,111
Predicates per Obj. Category	-	2.3	24.25	-

Tabella 2: Dataset a confronto in [1]

7 Analisi risultati

In [18] è stato proposto come metodo di valutazione il recall @ 100 e il recall @ 50. Un'alternativa ampiamente utilizzata poteva essere la mean average precision (mAP) tuttavia non potendo annotare tutte le relazioni in ogni immagine è stata scartata. Recall @ x calcola il numero di volte in cui una relazione corretta viene proposta dal modello come soluzione nelle x previsioni con confidence score più alto. Il confidence score rappresenta la probabilità, in percentuale, che la relazione venga rilevata correttamente [54]. Con 100 classi di oggetti e 70 tipi di relazione si ottiene un totale di $100 \times 70 \times 100$ relazioni. Una previsione casuale avrà recall @ 100 pari a 0.00014 ($\frac{1}{100 \times 70 \times 100} \times 100$).

Il lavoro svolto dalla Visual relationship detection può essere diviso in 3 attività:

1. Predicate detection: presa un'immagine e un insieme di bounding box come input vengono calcolate delle possibili relazioni tra le coppie di oggetti. Gli oggetti vengono forniti già localizzati per evitare che alcuni oggetti non vengano rilevati correttamente.

2. Phrase detection: presa un'immagine si produce un'etichetta per descrivere la relazione e una bounding box che la contenga con un valore di sovrapposizione di almeno 0.5 rispetto al valore reale.

3. Relationship detection: presa un'immagine vengono calcolati un insieme di relazioni nella forma (soggetto-predicato-oggetto) oltre a localizzare sia il soggetto che l'oggetto con un valore di sovrapposizione di almeno 0.5 rispetto al valore reale.

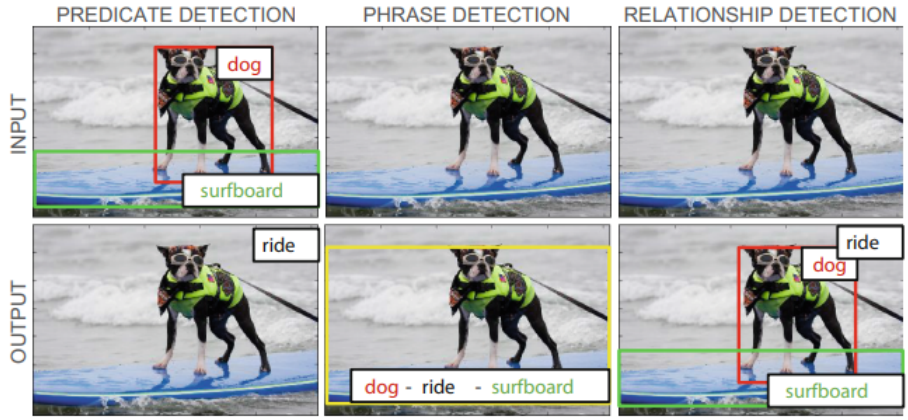

Figura 20: Le tre attività appena descritte: predicate detection (sinistra), phrase detection (centro), relationship detection (destra) [18]

	Predicate Detection		Phrase Detection		Relationship Detection	
Methods	R@50	R@100	R@50	R@100	R@50	R@100
Visual Phrase [14]	-	-	0.54	0.63	-	-
VR-LP [18]	47.87	47.87	16.17	17.03	13.86	14.70
VTransE [22]	44.76	44.76	19.42	22.42	14.07	15.20
VipCNN [13]	-	-	22.78	27.91	17.32	20.01
VRL [31]	-	-	21.37	22.60	18.19	20.79
SD [29]	51.50	51.50	16.94	18.89	14.31	15.77
AP+C+CAT [24]	53.59	53.59	17.60	19.24	15.63	17.39
DLK [20]	54.82	54.82	26.47	29.76	22.68	31.89
MotifNet [34]	65.20	67.10	-	-	-	-
SK [21]	71.02	80.80	-	-	-	-
DR-Net [19]	80.78	81.90	19.93	23.45	17.73	20.88
VRD-DSR [28]	86.01	93.18	-	-	19.03	23.29

Tabella 3: Confronto delle 3 attività utilizzando il dataset VRD [1]

Nella tabella 3 vengo confrontati alcuni degli approcci analizzati in questo elaborato. La predicate detection è l'attività in qui sono stati riscontrati miglioramenti significati-

vi ($47\% - 86\%$). La complessità maggiore della relationship detection risulta evidente nonostante i miglioramenti dovuti anche ai progressi nella object detection.

	Predicate Detection		Phrase Detection		Relationship Detection	
Methods	R@50	R@100	R@50	R@100	R@50	R@100
Visual Phrase [14]	-	-	3.41	4.27	-	-
VTransE [22]	62.63	62.87	9.46	10.45	5.52	6.04
PPR-FCN [55]	64.17	62.86	10.62	11.08	6.02	6.91
VRD-DSR [28]	69.06	74.37	-	-	-	-

Tabella 4: Confronto delle 3 attività utilizzando il VG dataset filtrato [1]

Sono stati fatti alcuni test anche con il VG dataset in cui le relazioni con meno di 5 apparizioni sono state filtrate. Il dataset dopo queste rimozioni contiene 99.658 immagini, 200 classi di oggetti, 100 predicati e 19,237 relazioni diverse. Con dataset di grandi dimensioni la Relationship detection si trova ancora in difficoltà a causa del numero di relazioni che si possono creare. Questo rende difficile per il modello adattarsi a relazioni possibilmente mai incontrate prima.
Sempre in [18] è stato proposto un parametro k per filtrare i predicati basandosi sul loro confidence score. $k = 5$ vuol dire prendere solo i 5 predicati con confidence score più alto e poi calcolare il recall @ x. Il parametro sembra influenzare le prestazioni ma la maggior parte degli articoli non riporta come sono stati calcolati i parametri.
Dai risultati nella relationship detection sono state proposte delle criticità sul metodo di valutazione in [56]. Infatti un modello potrebbe performare meglio perché allenato sulle relazioni più frequenti andando a falsificare le reali prestazioni.

8 Conclusioni

La VRD non è sicuramente un problema chiuso. I vari approcci mostrati cercano di sfruttare varie tecniche e indizi basati su come noi umani percepiamo le relazioni in un'immagine senza riuscire a replicarne il risultato. Tuttavia i miglioramenti ci sono, per quanto lenti se confrontati con altre attività quali l'object detection. Il motivo principale è dovuto al numero, troppo vasto per essere coperto sufficientemente in un dataset, di relazioni e al modo in cui queste modifichino in un certo senso i due oggetti interessati. I vantaggi che un rilevatore sufficientemente performante per la VRD può portare rende il problema sicuramente importante per poter migliorare un considerevole numero di attività che si sviluppano sopra la VRD.

9 Bibliografia

[1] Jun Cheng et al. «Visual Relationship Detection: A Survey». In: *IEEE Transactions on Cybernetics* 52.8 (2022), pp. 8453–8466. DOI: 10.1109/TCYB.2022.3142013.

[2] Zhengxia Zou et al. «Object detection in 20 years: A survey». In: *arXiv preprint arXiv:1905.05055* (2019).

[3] Yann LeCun, Yoshua Bengio e Geoffrey Hinton. «Deep learning». In: *nature* 521.7553 (2015), pp. 436–444.

[4] Keiron O'Shea e Ryan Nash. «An introduction to convolutional neural networks». In: *arXiv preprint arXiv:1511.08458* (2015).

[5] Saad Albawi, Tareq Abed Mohammed e Saad Al-Zawi. «Understanding of a convolutional neural network». In: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, pp. 1–6.

[6] Manuel Carranza-Garcia et al. «On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data». In: *Remote Sensing* 13.1 (2020), p. 89.

[7] Joseph Redmon et al. «You only look once: Unified, real-time object detection». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[8] Ross Girshick et al. «Rich feature hierarchies for accurate object detection and semantic segmentation». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[9] Ross Girshick. «Fast r-cnn». In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[10] Chaitanya Desai, Deva Ramanan e Charless C. Fowlkes. «Discriminative Models for Multi-Class Object Layout». In: *International Journal of Computer Vision* 95.1 (ott. 2011), pp. 1–12. DOI: 10.1007/s11263-011-0439-x. URL: https://doi.org/10.1007/s11263-011-0439-x.

[11] Abhinav Gupta e Larry S. Davis. «Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers». In: *Computer Vision – ECCV 2008*. A cura di David Forsyth, Philip Torr e Andrew Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 16–29. ISBN: 978-3-540-88682-2.

[12] Bangpeng Yao e Li Fei-Fei. «Modeling mutual context of object and human pose in human-object interaction activities». In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 17–24.

[13] Yikang Li et al. «ViP-CNN: Visual Phrase Guided Convolutional Neural Network». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[14] Mohammad Amin Sadeghi e Ali Farhadi. «Recognition using visual phrases». In: *CVPR 2011*. 2011, pp. 1745–1752. DOI: 10.1109/CVPR.2011.5995711.

[15] Ali Farhadi e Mohammad Amin Sadeghi. «Phrasal recognition». In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2013), pp. 2854–2865.

[16] Aude Oliva. *Scene Understanding*. http://people.csail.mit.edu/torralba/courses/6.870/slides/lecture5.pdf.

[17] Justin Johnson et al. «Image retrieval using scene graphs». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3668–3678.

[18] Cewu Lu et al. «Visual relationship detection with language priors». In: *European conference on computer vision*. Springer. 2016, pp. 852–869.

[19] Bo Dai, Yuqi Zhang e Dahua Lin. «Detecting visual relationships with deep relational networks». In: *Proceedings of the IEEE conference on computer vision and Pattern recognition*. 2017, pp. 3076–3086.

[20] Ruichi Yu et al. «Visual relationship detection with internal and external linguistic knowledge distillation». In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1974–1982.

[21] François Plesse et al. «Visual relationship detection based on guided proposals and semantic knowledge distillation». In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2018, pp. 1–6.

[22] Hanwang Zhang et al. «Visual translation embedding network for visual relation detection». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5532–5540.

[23] Zhaoyang Niu, Guoqiang Zhong e Hui Yu. «A review on the attention mechanism of deep learning». In: *Neurocomputing* 452 (2021), pp. 48–62. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.03.091. URL: https://www.sciencedirect.com/science/article/pii/S092523122100477X.

[24] Bohan Zhuang et al. «Towards context-aware interaction recognition for visual relationship detection». In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 589–598.

[25] Ranjay Krishna et al. «Referring relationships». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6867–6876.

[26] Peng Wang et al. «Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1960–1968.

[27] Li Mi e Zhenzhong Chen. «Hierarchical graph attention network for visual relationship detection». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 13886–13895.

[28] Kongming Liang et al. «Visual relationship detection with deep structural ranking». In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[29] Yaohui Zhu, Shuqiang Jiang e Xiangyang Li. «Visual relationship detection with object spatial distribution». In: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2017, pp. 379–384.

[30] Ji Zhang et al. «Graphical contrastive losses for scene graph parsing». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11535–11543.

[31] Xiaodan Liang, Lisa Lee e Eric P Xing. «Deep variation-structured reinforcement learning for visual relationship and attribute detection». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 848–857.

[32] Volodymyr Mnih et al. «Playing atari with deep reinforcement learning». In: *arXiv preprint arXiv:1312.5602* (2013).

[33] Yikang Li et al. «Scene graph generation from objects, phrases and region captions». In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1261–1270.

[34] Rowan Zellers et al. «Neural motifs: Scene graph parsing with global context». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5831–5840.

[35] Adam Santoro et al. «A simple neural network module for relational reasoning». In: *Advances in neural information processing systems* 30 (2017).

[36] Justin Johnson et al. «Inferring and executing programs for visual reasoning». In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2989–2998.

[37] Justin Johnson et al. «Clevr: A diagnostic dataset for compositional language and elementary visual reasoning». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2901–2910.

[38] Pan Lu et al. «R-VQA: learning visual relation facts with semantic attention for visual question answering». In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1880–1889.

[39] Tsung-Yi Lin et al. «Microsoft coco: Common objects in context». In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[40] Stanislaw Antol et al. «Vqa: Visual question answering». In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433.

[41] Liang Peng et al. «Mra-net: Improving vqa via multi-modal relation attention network». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.1 (2020), pp. 318–329.

[42] Georgia Gkioxari et al. «Detecting and recognizing human-object interactions». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8359–8367.

[43] Bingjie Xu et al. «Learning to detect human-object interactions with knowledge». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[44] Vinay Chaudhri et al. «Knowledge Graphs: Introduction, History and, Perspectives». In: *AI Magazine* 43.1 (2022), pp. 17–29.

[45] Matthew Fisher, Manolis Savva e Pat Hanrahan. «Characterizing structural relationships in scenes using graph kernels». In: *ACM SIGGRAPH 2011 papers*. 2011, pp. 1–12.

[46] Angel Chang, Manolis Savva e Christopher D Manning. «Learning spatial knowledge for text to 3D scene generation». In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 2028–2038.

[47] Gregory J Zelinsky. *Understanding scene understanding*. 2013.

[48] Uzair Nadeem et al. «Deep learning for scene understanding». In: *Handbook of deep learning applications*. Springer, 2019, pp. 21–51.

[49] Prajakta Ganesh Pawar e V Devendran. «Scene understanding: A survey to see the world at a single glance». In: *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*. IEEE. 2019, pp. 182–186.

[50] Mark Everingham et al. «The Pascal Visual Object Classes Challenge: A Retrospective». In: *International Journal of Computer Vision* 111.1 (gen. 2015), pp. 98–136. ISSN: 1573-1405. DOI: 10.1007/s11263-014-0733-5. URL: https://doi.org/10.1007/s11263-014-0733-5.

[51] Bart Thomee et al. «YFCC100M: The new data in multimedia research». In: *Communications of the ACM* 59.2 (2016), pp. 64–73.

[52] Ranjay Krishna et al. «Visual genome: Connecting language and vision using crowdsourced dense image annotations». In: *International journal of computer vision* 123.1 (2017), pp. 32–73.

[53] Drew A Hudson e Christopher D Manning. «Gqa: A new dataset for real-world visual reasoning and compositional question answering». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6700–6709.

[54] Sampurna Mandal et al. «Chapter Four - Single shot detection for detecting real-time flying objects for unmanned aerial vehicle». In: *Artificial Intelligence for Future Generation Robotics*. A cura di Rabindra Nath Shaw et al. Elsevier, 2021, pp. 37–53. ISBN: 978-0-323-85498-6. DOI: https://doi.org/10.1016/B978-0-323-85498-6.00005-8. URL: https://www.sciencedirect.com/science/article/pii/B9780323854986000058.

[55] Hanwang Zhang et al. «Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn». In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4233–4241.

[56] Tianshui Chen et al. «Knowledge-embedded routing network for scene graph generation». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6163–6171.