Few-Shot Visual Grounding for Natural Human-Robot Interaction

Tziafas, Giorgos; Mohades Kasaei, Seyed

Published in:
IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)

DOI:
10.1109/ICARSC52212.2021.9429801

Publication date:
2021

Few-Shot Visual Grounding for Natural Human-Robot Interaction

Giorgos Tziafas
Department of Artificial Intelligence
University of Groningen
g.tziafas@student.rug.nl

Hamidreza Kasaei
Department of Artificial Intelligence
University of Groningen
hamidreza.kasaei@rug.nl

***Abstract*—Natural Human-Robot Interaction (HRI) is one of the key components for service robots to be able to work in human-centric environments. In such dynamic environments, the robot needs to understand the intention of the user to accomplish a task successfully. Towards addressing this point, we propose a software architecture that segments a target object from a crowded scene, indicated verbally by a human user. At the core of our system, we employ a multi-modal deep neural network for visual grounding. Unlike most grounding methods that tackle the challenge using pre-trained object detectors via a two-stepped process, we develop a single stage zero-shot model that is able to provide predictions in unseen data. We evaluate the performance of the proposed model on real RGB-D data collected from public scene datasets. Experimental results showed that the proposed model performs well in terms of accuracy and speed, while showcasing robustness to variation in the natural language input.**

I. Introduction

Humans have the cognitive capacity to process multi-modal data (e.g. vision, language) and make cross-references between parts of the two modalities in real-time effortlessly. They are also very capable of identifying such cross-references in degenerate cases where one modality suffers from noise. However, this is not the case in robotics, where most commonly the visual perception and HRI modules are treated separately. In this regime, when verbal input is given to the robot (e.g. in the context of grasping: "Grasp the *mug*") the referring phrases that correspond to objects to be segmented (aka *grounded*, e.g. "*Mug*") must be predefined explicitly and hard-coded in the agents behaviour. As a result, the agent is unable to comprehend variants of the predefined object category from the verbal input as it is often the case in real-world scenarios, where objects might be referred by their visual attributes or spatial relation to another object (e.g. "Grasp the *red mug*" or "Grasp the *mug* that is *next to* the *laptop*" in the case of multiple mug objects within view). A more *natural* HRI setting would require a bridging between the two modules, allowing the model to interpret ambiguous verbal references and generalize over unseen object categories. This paper proposes such a bridging that tackles the problem by employing an end-to-end multi-modal deep learning model, able to make predictions for image-phrase pairs that were never seen in the training data.

The task at hand is described in literature as *single-query visual grounding* or *referring expression comprehension*. In this task, given an input image and a natural language phrase referring to a specific entity, the model has to localize the entity inside the frame (see Fig. 1). Several research has been done to address this task, both solely and as a proxy task for *visual question-answering* (VQA) and *visual reasoning* downstream applications. Section II provides a brief overview of such research directions. In this work, we aim to design a grounding agent that can be utilized in a practical HRI scenario, such as grasping household objects based on verbal commands given by a human supervisor. To be efficient for this application, our agent needs to be able to infer groundings at high speed. Also, when a broad variation of object categories, as well as robustness to the phrasing of the referring expressions describing them, is desired, the system should be able to provide predictions for object - phrase instances that it has never encountered during training. These considerations guide our choice of deep learning model and the architecture of the implemented system, both of which are described in Section III.

Fig. 1: Examples of grounding predictions made by our model for real (top) and simulated (bottom) scenes along with the corresponding input phrase queries.

We explore the robustness of the proposed approach to variations of the natural language input by probing it in test time with extensive captions of multiple levels of reference, namely: a) multiple instances of the same entity (plural) or b) a specific instance of an entity appeared multiple times in the scene by referring to its visual attribute (color) or some other property (e.g. gender in the case of humans). Additionally, as

978-1-6654-3198-9/21/$31.00 ©2021 IEEE

most benchmarks for visual grounding evaluate on domain-agnostic datasets, in this work we opt to explore the grounding models performance on an HRI-oriented data domain, by evaluating its zero/few-shot performance in two tiny-scale datasets of image-query pairs respecting the assumed HRI scenario mentioned above. The image samples are collected from subsets of two RGB-D datasets, namely: *RGBD-Scenes* [26], used most popularly for 3D vision learning and the *Objects Cluttered Indoors Dataset (OCID)* [1], used for manipulation planning. Details about the evaluation data and some results are presented in section IV. Conclusion and some interesting potential future research directions are given in section V.

II. Related work

Visual grounding architectures typically use object detectors such as Faster-RCNN [22] as a pre-processing step to extract bounding box proposals and corresponding object features, thus limiting the categories that the model is able to learn to the ones pre-specified by the pre-trained detector [23]. Refined versions use cross-modal self-attention mechanisms to capture long range dependencies in the two modalities [29]. More recent approaches extract scene graphs from images-text and tackle grounding as a structure prediction task where visual and textual Graph Neural Networks (GNNs) are used to contextualize each modalitys representations and a graph similarity metric is introduced to prune the two graphs appropriately [13].

Another family of state-of-the-art models is that of pre-trained representation learners, such as VisualBERT [8] and ViLBERT [14]. Such models are very parameter-heavy, trained on massive-scale datasets with self-supervised objectives, such as language modeling, masked region modeling, word-region alignment and image-text matching. They often also apply a *multi-task setting* [15], in which supervision from multiple tasks-domains is injected during self-supervised pre-training. Single-task performance is then benefited from attempting to solve all tasks at once. On the downside, the enormous scale of these models grant them potentially inefficient for real-time application.

Several robotics-grounded works attempt to explicitly model natural language input for HRI applications, either as a purely textual or as an end-to-end multi-modal system. In some works, the system maps input natural language instructions into action sequences for task handling [16], [27]. Special care is given so that the model handles variations of natural language input and is able to generalize over unseen environment and task settings. Paul et. al. [17] implemented a probabilistic model that parses natural language input into object/region proposals as well as motion constraints by incorporating notions of abstract spatial concepts. Shridhar et. al. [25] presented an HRI-applied multi-modal system for visual grounding, where the model was able to handle variations of natural language input (for situations of multiple object instances) by interacting with the user through an ambiguity-resolution question, generated by an auto-regressive RNN which is trained jointly with the grounding network. One drawback of their work is that they employed a two-stage approach for providing groundings, intensifying computational requirements in inference time. In this work, we attempt to address this limitation by employing a one-stage zero-shot model.

III. System Architecture

In this section we present the visual grounding architecture that is developed as well as the technical details of the overall architecture of the system.

A. Learning Model

The deep learning model that is chosen as a baseline for our implementation is the *Zero-Shot Grounding (ZSG)* network [23]. Similar to single-shot architectures employed in object localization, such as [12], the proposed model generates bounding box proposals that refer to the input query based entirely on the size of the input image. As a result, an end-to-end trainable image encoder for capturing image representations can replace pre-trained object descriptors. For each image-query input pair, the model generates a set of bounding box proposals $B = \{b_1, b_2, ..., b_N\}$ and outputs the best candidate box b_i, as well as four regression parameters $\{x_1, y_1, x_2, y_2\}$ that correspond to the updated top-left and bottom-right box coordinates in the input image frame. Since this is an end-to-end architecture, the visual features extracted by this model are independent of the trained object intra-class variance. Therefore, the proposed method departs from the limitations posed by other pre-trained visual grounding systems, and is suitable for real-time applications due to its single stage design and computational efficiency, especially during inference.

The model consists of five main components, including: (*1*) a language module that encodes the query phrase into a continuous vector space, (*2*) a visual module that extracts multiple image feature maps, (*3*) an anchor generator for proposing multiple scale bounding boxes, (*4*) a multi-modal fusion scheme for injecting all features into a single representation and a linear layer that predicts the most likely box proposal, (*5*) as well as the array of regression parameters for its fixed coordinates. For coherence purposes, we provide a brief overview of these modules in the following subsections [23].

1) Language module: This module consists of an embedding layer followed by a recurrent neural encoder for encoding the input query phrase. The embedding layer is responsible for mapping each word W_i in our vocabulary to a dense vector $\vec{w_i} \in \mathbb{R}^{d_w}$. The encoder is a uni-layered bi-directional LSTM architecture [24] that processes sequentially the entire input word vector sequence $\{\vec{w_i}\}, i = 1, ..., T$, in both directions (start to end and vice-versa) and at each step outputs a hidden state vector $\vec{h_i} \in \mathbb{R}^{2d_w}$ that is informed by the context of the entire phrase in both directions at this point. The encoding that we use for representing the entire phrase is the hidden state vector $\vec{h_T}$ produced in the last time step.

Alternatively, we implement phrase contextualization using a single *Transformer Encoder* layer [28] and aggregate the hidden representations $\mathbf{H} \in \mathbb{R}^{T \times d_w}$ for the entire sequence with average pooling.

2) Visual module: This module consists of a deep *Convolutional Neural Network* (CNN) that learns how to represent the input 2D image into a dense feature map. This is a standard CNN architecture used for object recognition without the linear layers that are used for cross-entropy classification. In our version, K feature maps $\mathbf{V_j} \in \mathbb{R}^{d_v \times d_v}, j = 1, ..., K$, are extracted at different resolutions. The model used for encoding is a ResNet-50 [4], augmented into the *Feature Pyramid Network* (FPN) architecture [10] for extracting multi-scale hierarchical feature maps. The feature maps are normalised across the channel dimension.

3) Anchor generation: The first step is to form a grid. For each cell of the grid, anchors of different shapes are proposed. The convention of the grid is (-1,-1) to (1,1), similar to the one defined for bounding boxes. Anchors are defined in terms of ratios and scales. The ratio is the ratio of the height of an anchor box to its width. The scale is used to calculate the unit of the height and width of each anchor box. Given a scale (s) and a ratio (r), the aspect is $[s\sqrt{r}, \frac{s}{\sqrt{r}}]$. These aspects are then resized according to the size of each cell of the grid. The center of each anchor box is considered the centre of the grid cell.

4) Multi-modal fusion: The language feature vector extracted by the neural encoder is expanded to fit the dimensions of the K extracted visual feature maps and it is concatenated along the channel dimension of each feature map, so that $\mathbf{H_T} \in \mathbb{R}^{d_v \times d_w}$, $\mathbf{H_T} = \left(\vec{h_t} \quad \vec{h_t} \quad ... \quad \vec{h_t}\right)^T$. The generated anchor box centres are also appended at each cell of the feature maps. The resulting multi-modal feature representations $\mathbf{M_j} \in \mathbb{R}^{d_m \times d_m}$ are then given for $j = 1, ..., K$ by:

$$\mathbf{M_j}(x, y) = \mathbf{V_j}(x, y) \; ; \; \mathbf{H_T}(x, y) \; ; \; \frac{c_x}{W} \; ; \; \frac{c_y}{H}$$

where ; operator denotes the concatenation, c_x, and c_y represent the center locations of the normalised feature maps at each (x, y) location. The initial size of the input image is represented by W, and H. The scaling operation is performed in order to aid location-based grounding, when input query phrases contain location information, providing functionality for making spatial references between recognised objects in the input image frame.

5) Anchor matching: Each generated anchor proposal of different size is matched to every cell of the produced multi-modal feature maps. For each box, a linear layer maps the multi-modal features into a 5-dimensional vector containing the prediction score (confidence) and the regression box parameters that update the coordinates to bound the referenced object tightly.

B. Online Object Segmentation

In this section, we describe the design of the implemented software architecture, shown in Fig. 2. Our system receives a stream of RGB-D topics produced real-time from a depth sensor along with a natural language caption given by a human user and produces an RGB image with the predicted bounding box drawn into the frame, as well as a *PointCloud* containing 3D information of the segmented object. The images on the RGB-D buffer are refreshed with each new pair and only when a caption is given the last entry is grabbed and used as input to the network.

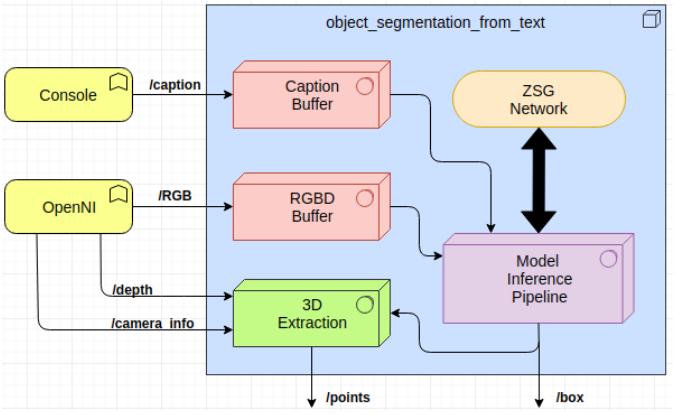


Fig. 2: A schematic of the implemented *ROS* system. The inference pipeline is implemented through an action client that given an input RGB frame and a caption sends a request to our network and outputs the coordinates of the most probable bounding box. A 3D Extraction node extracts a *PointCloud* by combining the RGB box with the corresponding depth box from the registered depth frame.

For real-time visual grounding, we have implemented an action server that functions as the communication side between the sensor data and the implemented ZSG network. At its spawn, the server loads all the necessary utilities (word embeddings etc.) and loads a pre-trained instance of the ZSG network. When a caption is buffered, a client request for a grounding inference is published. The action server appropriately pre-processes the input raw RGB frame in order to enhance it's contrast, by applying *Contrast Limited Adaptive Histogram Equalisation* (CLAHE) [30] in the *YUV* colourspace. The query phrase is embedded and the resulting image-query pair is passed as input to the network for a single forward pass. After inference, some post-processing steps for drawing the predicted box and segmenting the foreground for 3D extraction are applied and the resulting frames are published. The server responses are executed in $>$ 30Hz, even without utilizing GPU-enabled acceleration like during training (tested in an *Intel Xeon E5-2680 v3 @ 2.50GHz* CPU).

IV. Experimental Results

A. Training

We trained our model using two benchmark visual grounding datasets. The first training dataset was *Flickr30k Entities* [20], comprised of 30 000 annotated images associated with five sentences, each having multiple entity queries referring

to the image with an average of 3.6 queries per sentence. As a result, each sentence has been split to provide more grounding samples for the respective image. The annotations are bounding boxes containing the category labels of the grounded entity. The second dataset was the *RefClef* strain of the *ReferIt* dataset [6], itself being a subset of *Imageclef* [3], enumerating approximately 20 000 images with a total of $85k$ single-query caption sentences. For both datasets, we used the same training-validation-test splits as in [5].

We considered two options for embedding input language semantics into word vectors for English, namely: (*i*) pre-trained *GloVe* embeddings [18] and (*ii*) the last hidden layers representations of a pre-trained BERT-BASE [9], followed by a trainable linear layer down-scaling the vectors to desired dimensionality. For extracting visual features we utilized a *RetinaNet* [11] network with a *ResNet-50* [4] backbone. The image samples were resized to 300×300 and the word embeddings size as well as the hidden size of the bi-LSTM contextualizing the input phrase was set to 300, so that both feature vectors could be appended to fit the dimensionality of the multi-modal feature maps. The input query phrases were padded to a maximum length of 50 words per phrase. Following [23], a total of 9 candidate anchor proposals of different sizes is generated.

For formulating the supervision signal as well as quantifying the evaluation performance, we utilised the *Intersection over Union* (IoU) metric, calculated as the total overlapping area between the proposed box and the ground truth box. Following the original implementation, we used an *IoU* threshold of 0.5, meaning that only proposals that fit the ground truth box's area over 50% are considered candidate:

$$g_{b_i} \doteq 1 \cdot [IoU(b_i, gt) \geq 0.5] \; + \; 0 \cdot [IoU(b_i, gt) < 0.5],$$
$$G \doteq \{b_i \mid g_{b_i} = 1\}$$

where $B = \{b_i,\ i = 1, ..., 9\}$ represents the set of all proposals and gt states the ground truth box. The set G is the collection of all candidate proposals ($g_{b_1} = 1$). The training loss is then calculated as the sum of the focal loss L_F described in [11] (applied for deciding the prediction score of the i-th proposal p_{b_i} for its binary classification as foreground that possibly contains the query vs. background), with parameters $\alpha = 0.25,\ \gamma = 2$ and the smooth-L1 loss L_S [21] (used for regressing the parameters r_{b_i} of a tighter matching bounding box for the most probable proposal):

$$L = \frac{1}{|G|} \sum_{i=1}^{|B|} L_F(p_{b_i}, g_{b_i}) + \frac{1}{|G|} \sum_{i=1}^{|B|} g_{b_i} \cdot L_S(r_{b_i}, gt)$$

The reported accuracies are measured as the total average of correctly classified samples with an $IoU > 0.5$. The end-to-end model was trained for a total of 12 hours for each dataset (around $7 - 8$ epochs), where the validation loss had already saturated. We utilised the Adam optimizer [7] with a learning rate of 10^{-4} and a weight decay of 10^{-4} for regularisation. The model was trained on a *NVidia Tesla v100* GPU node for parallel processing of batches of 128 image-query pairs. All code is written in *Python* using the *PyTorch* deep learning framework. Training results compared with several baselines are reported in Table I.

Method	Flickr30k	RefClef
QRC⋆ [2]	60.21%	44.10%
CITE⋆ [19]	61.89%	34.13%
QRG⋆ [2]	60.1%	-
ZSG [23]	63.39%	**58.64%**
VilBERT [15]	**64.61%**	-
ZSG (Glove-LSTM)	62.73%	53.44%
ZSG (Glove-TRM)	61.74%	50.41%
ZSG (BERT-LSTM)	**63.09%**	**54.21%**
ZSG (BERT-TRM)	62.13%	52.12%

TABLE I: Best top-1 accuracies @IoU = 0.5 of the implemented model variants after 12 hours of training in both datasets. Results compared to original implementation and other state-of-the-art methods. Methods with ⋆ further fine-tune their network on the entities of the Flickr30k dataset.

Even though the pre-trained BERT language module provides marginally better embeddings for visual grounding, its massive scale sets a huge computational bottleneck to the system. Therefore, we implement our agent based on the pre-trained Glove model.

B. Evaluation

In order to evaluate the performance of the proposed system, we conducted multiple experiments, aiming to test not only the predictive accuracy but also the generalisation potential of the proposed model in variants of the standard input query phrases. In this vein, we performed online interactive sessions with our implemented agent, during which humans provide captions for arbitrary images and evaluate the system's predicted segmentation qualitatively. Different depicted objects are queried in free linguistic fashion, also including ambiguous query phrases (plural, referring to a visual cue etc.) that challenge our system in regards to the natural HRI element that we strive for. Figure 3 demonstrates some examples of such experiments and a live recorded demo of a session is available online at https://youtu.be/kgQgaghf71o

Real-time sensory data streamed from a depth sensor, often suffer from various forms of noise compared to the digitally pre-processed images of the training data. To investigate the performance of the proposed system in sensor data, we collected two small-scale datasets of pre-recorded RGB-D images. The scenes included in our samples respect the assumed HRI scenario, i.e., they always comprise of a collection of objects in arbitrary spatial arrangements on top of a planar surface in clear front view of the RGB-D sensor.

The first dataset is a sub-set of the *RGBD-Scenes* dataset [26], which has a total of 67 RGB-D pairs depicting 22 different typical household object instances from 6 different categories (bowl, cap, cereal box, coffee mug, flashlight, soda can) in 8 different scenes (including multiple desk settings,

(a) Correctly segmented objects

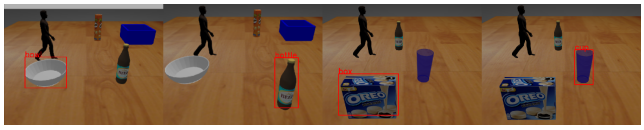
(b) Correctly segmented objects with low number of occurrences in training data

(c) Incorrectly segmented objects

Fig. 3: Several examples of correct and some of incorrect bounding box predictions for image-query pairs. Queries are from left to right: (a) ice cream, hat, guitar, rope, (b) microscope, necklace, barbecue, roulette, (c) cup of coffee, watermelon, beer, bottle.

(a) Correctly segmented objects from RGBD-Scenes

(b) Correctly segmented objects with relative cluttered environments



(c) Correctly segmented objects in synthetic data simulated in Gazebo

Fig. 4: Several examples of bounding box predictions for real-time RGB-D sensory data from real/synthetic scenes. Queries are from left to right: (a) cereal box, paper, bowl, hat, (b) envelope, tortellini, soda can, bowl, (c) bowl, bottle, box, cup.

kitchen, meeting room, small and large table). Bounding box annotations come packed with the dataset. We appropriately sub-sampled the raw training data to include scenes that only contain different object categories or the same objects but viewed from a different angle, resulting in different spatial relationships between them. The second dataset is a curated sub-set of OCID [1], containing a total of 89 different object categories in two scene settings (table and floor) and two different camera placements (front and top view). As this dataset is originally intended for learning motion planning of a robotic actuator in scenes with highly cluttered objects, we filter the raw data to include only samples with none to minimal cluttering. Statistics about the collected RGB-D datasets are described in Table II.

We again qualitatively evaluated and reported accuracy scores representing the zero-shot performance of our system in both domains. We performed preliminary experiments for evaluating the *domain adaptation* potential of our system, where we fine-tuned our model in one RGB-D-scene (e.g. desk) and evaluated in another (e.g. kitchen). By this we aimed to simulate the fine-tuning step that a potential user of our system can employ after gathering minimal samples from the specific object catalogue / environment they wish their agent to operate. Fine-tuning had been implemented identically to the training process with the exception of using a batch size of 32. The results are presented in Table III and examples of real-time inferences are shown in Figure 4.

Dataset	#Img.	avg.QPI	#Samp.	Mult.Inst. (%)
RGBD-Scenes	67	3.13	210	17.97%
OCID	18	2.94	53	22.22%

TABLE II: Total number of image samples. average number of query phrases per image, total number of samples (query-phrase pairs) and percentage of samples that include multiple instances of the same object category.

As expected, we observe a performance drop in the sensory input datasets. In such data, images suffer from potentially high amount of noise (reflections, bad illumination etc.) resulting in noisy visual representations. At the same time, potential cluttering of objects in real-life scenarios also degrades the quality of the proposal generation. These limitations grant our model practically unreliable without any fine-tuning, as we observe by the poor zero-shot performance (below 50%). However, the few-shotting experiments suggest a significant performance boost even with minimal fine-tuning. Moreover, we observe that due to its single-stage nature, our model is capable of unravelling ambiguities of reference in the input phrase. Specifically, there are non-sensory examples where the model responds to use of plural, use of definitive pronouns, as well as visual cues, e.g. colour (see Fig. 5).

V. Conclusion

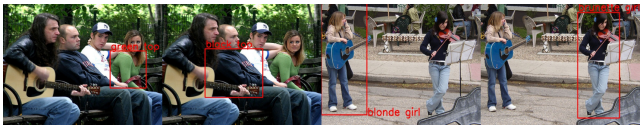
In this work, we employ a state-of-the-art multi-modal deep learning model and develop an HRI agent for real-time visual grounding. The focus of this work is not on improving the

Model	OCID (full)	RGB-D (full)	RGB-D (kitchen)	RGB-D (tables)
ZSG-Flickr30k	32.13%	34.19%	32.50%	34.13%
ZSG-RefClef	31.37%	33.50%	36.35%	34.13%
+desk	-	-	49.49%	50.15%
+desk+meeting	-	-	55.90%	58.20%

TABLE III: Evaluation results for both pre-trained models in HRI-specific data, collected from sub-sets of the *RGBD-scenes* and *OCID* datasets. The + rows denote few-shotting results of the ZSG-Flickr30k model in specific scenes after fine-tuning in similar examples from different scenes

(a) Comprehension of plural

(b) Comprehension of pronouns



(c) Comprehension of visual cues (colour)

Fig. 5: The system is capable of generalising over variants of the natural language input (plural, pronouns, visual cues). Queries are from left to right: (a) sandal(s), human(s) (b) his/the girls face, his/her shoulder (c) green/black top, blonde/brunette girl.

zero-shot model but rather exploiting its single-stage nature, which allows for predictions in unseen data, in order to achieve interaction with humans in dynamic environments (potentially unknown objects) and in natural fashion (robustness to ambiguities of reference). The segmented 3D information of the object can serve as a perceptive utility assisting in navigation, planning, and manipulation actions. We test our methodology by gathering RGB-D data from two robotics-oriented datasets and perform several rounds of experiments to evaluate the systems performance in both zero-shot and few-shot scenarios. Experimental results showed that our approach works well in the fine-tuning scenario, both in terms of accuracy and speed. In the continuation of this work, we would like to investigate the possibility of ameliorating the low quality of visual representations in sensory images by applying *Synthetic2Real Domain Adaptation*, in effect generating synthetic scenes similar to the setup of our HRI scenario and utilizing them as a fine-tuning resource. Another interesting direction is the replacement of the current scene-level visual module with an object-level one, as bounding box extraction can be managed in an RGB-D sensor with the aid of depth data.

REFERENCES

[1] *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*. IEEE, 2019.

[2] K. Chen, R. Kovvuri, and R. Nevatia. Query-guided regression network with context policy for phrase grounding. *CoRR*, abs/1708.01676, 2017.

[3] H. J. Escalante, C. A. Hernández, J. A. Gonzalez, A. López-López, M. Montes, E. F. Morales, L. Enrique Sucar, L. Villaseñor, and M. Grubinger. The segmented and annotated IAPR TC-12 benchmark. *Comput. Vis. Image Underst.*, 114(4):419–428, Apr. 2010.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. *CoRR*, abs/1511.04164, 2015.

[6] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[8] L. H. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019.

[9] L. H. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019.

[10] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[11] T. Lin, P. Goyal, R. Girshick, K. He, and P. DollÃ¡r. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Oct 2017.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[13] Y. Liu, B. Wan, X. Zhu, and X. He. Learning cross-modal context graph for visual grounding. *CoRR*, abs/1911.09042, 2019.

[14] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *CoRR*, abs/1908.02265, 2019.

[15] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee. 12-in-1: Multi-task vision and language representation learning. *CoRR*, abs/1912.02315, 2019.

[16] D. Misra, J. Sung, K. Lee, and A. Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research*, 35:281 – 300, 2016.

[17] R. Paul, J. Arkin, N. Roy, and T. M. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Robotics: Science and Systems*, 2016.

[18] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[19] B. A. Plummer, P. Kordas, M. H. Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik. Conditional image-text embedding networks. *CoRR*, abs/1711.08389, 2017.

[20] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93, 2017.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[22] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[23] A. Sadhu, K. Chen, and R. Nevatia. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4694–4703, 2019.

[24] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, Nov. 1997.

[25] M. Shridhar and D. Hsu. Interactive visual grounding of referring expressions for human-robot interaction. *CoRR*, abs/1806.03831, 2018.

[26] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015.

[27] J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. W. Hart, P. Stone, and R. J. Mooney. Improving grounded natural language understanding through human-robot dialog. *CoRR*, abs/1903.00122, 2019.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[29] L. Ye, M. Rochan, Z. Liu, and Y. Wang. Cross-modal self-attention network for referring image segmentation. *CoRR*, abs/1904.04745, 2019.

[30] K. Zuiderveld. Graphics gems iv. chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.