Object Segmentation from Online Natural Language Supervision

Georgios Tziafas
Department of Artificial Intelligence
University of Groningen
Groningen, The Netherlands
s3913171

Rohit Malhotra
Department of Artificial Intelligence
University of Groningen
Groningen, The Netherlands
s3801128

Lennart Faber
Department of Artificial Intelligence
University of Groningen
Groningen, The Netherlands
s2500253

***Abstract*—In this paper we propose a software architecture aiming to assist in online robotic supervision for Human-Robot Interaction (HRI) applications. The concerned task is visual grounding, where the agent segments 3D information for an object within its view that is indicated verbally by a human supervisor. At the core of our system we implement a deep multi-modal neural model that learns to locate entities referenced in a natural language caption inside an image frame. Unlike most grounding methods that tackle the challenge in a two-stepped process, we employ an end-to-end zero-shot model from literature that can provide predictions in unseen data with high computational efficiency. Empirical use of the implemented system in online interaction scenarios suggests satisfactory behaviour both in terms of accuracy and performance, further showcasing some interesting generalisation properties over the natural language input.**

***Index Terms*—Zero-Shot Grounding, Multi-Modal Learning, Human-Robot Interaction, Cognitive Robotics**

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 goal object grasping: "Give me the mug") the query phrases that correspond to objects to be segmented (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 (e.g. "Give me the *red mug*" or "Give me the *mug that is next to the laptop*" in the case of multiple mug objects within view). This paper describes a possible bridge between these two modules that tackles the problem by using an end-to-end multi-modal deep learning system.

The task at hand is described as *visual grounding* (VG). In this task, given an image and a natural language caption of the scene that is depicted, the system tracks the locations of all different objects (queries) that comprise the caption, such as in Figure 1. Several research has been done to address this task, both separately and as a sub-process of *visual question-answering* (VQA), where the input phrase is a question and the system must produce groundings of the phrase's queries in order to perform visual reasoning upon them that answers the question. Section II provides a brief overview of such research directions.

Fig. 1: An example of a visual grounding system.

To be practical in an online HRI setting, the VG system has to be able to infer groundings at high speed. Also, when a large variety of objects and query variants for describing them is desired, the system should be able to provide predictions for objects - query variants that is has never encountered during training. These considerations guide our choice of deep learning model, which is described in Section III, as well as the architecture of the implemented system, which is presented in Section IV. Results for the training of our network and the behaviour of the system are demonstrated in Section V, followed by a brief discussion in Section VI.

II. Related work

The task of question and answering (Q&A) is well explored in the field of NLP. This inspired the task of Visual Question and Answering (VQA). The aim here is to answer the specific

questions about an image. The answer can be a word, a scene description, a binary answer, a count, etc. For achieving the task of VQA, visual grounding is a most important aspect of it. Existing VQA models have an image and a question encoder, then these two modalities are merged together and fed to an answer decoder [12]. Thus, the specific task of *visual grounding* is similar to VQA in terms of encoding and merging two modalities, but differs in terms of decoding the answer.

[1] is a joint embedding based model. It uses CNN for feature extraction and LSTM for text processing. Then a joint embedding is obtained by point-wise multiplication. In [7], the authors use the concept of *stacked attention* introduced in [24]. The model learns the important features of image based on attention calculated using the intermediate question features.

The task of visual grounding is considered a challenge because of its high dimensional nature. Not every possible object or situation that could be encountered by a system can be learned in advance and architectures aiming to solve this problem should therefore be capable of dealing with situations in which predictions have to be made concerning unseen data.

[6] addresses this issue by applying a technique described as *semantic self-supervision*. The authors mention that the goal of unsupervised learning is to learn some energy function that assigns lower energy values to data points similar to a training set while assigning higher values to others. Self-supervised learning makes use of a proxy task to mimic this process: its goal is to learn the function that pulls down the energy at the data manifold.

The proxy task in this case is concept-learning. Concept-learning aims to assist in achieving visual grounding by localising the entity that has to be found in the image. The entity to be found in this case is the main concept in the phrase that was provided as input. The main concept in phrases was extracted using a POS tagger that identifies nouns and randomly selects one if multiple are found. A much more difficult task is to learn the representations of concepts in an unsupervised manner. Javed et al. were able to deal with this problem by introducing concept batches. These batches consist of images that are put together because all of them contain some form of the same concept. For example: a batch could consist of images of a man, a horse, and a cat, with the shared concept being their legs. By training the system on images with such diversity, it could become capable of identifying concepts in many of their forms.

The system was trained using the features extracted by a VGG16 based network. The output of the system consisted of images with an overlay of a heat map indicating in what region the concept described by the phrase was to be found. The architecture proved to be relatively successful when compared to previous efforts, with an average accuracy of 50.10% on the Flickr30k data set calculated using the pointing game metric.

III. Visual Grounding Learning

In this section we present the visual grounding architecture that is developed, the multi-modal datasets utilised for training and the technical details of our implementation.

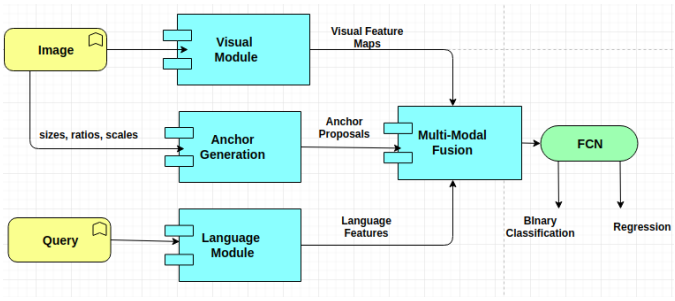


Fig. 2: A schematic of the Zero-Shot Grounding (ZSG) architecture. The model consists of visual and language feature extractors, an anchor generator for producing proposals, multi-modal fusion for creating inter-modal representations and a fully-convolutional layer (FCN) for final predictions.

A. Zero-Shot Grounding Model

The deep learning model that is chosen for our implementation is the *Zero-Shot Grounding* network proposed by [20] and presented schematically in Figure 2. Similar to single-shot architectures employed in object localization, such as [13], 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, as it is commonly applied in models like that of Section II. For each image-query input pair the model generates a set of bounding box proposals $B = \{b_1, b_2, ..., b_N\}$ and at it's output predicts the best candidate box b_i, as well as 4 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 so that the box is tightly attached to the object matching the input query phrase. 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 departing from the limitations posed by other pre-trained visual grounding systems and granting it suitable for integration to real-time applications due to its single stage design and its computational efficiency, especially during inference.

The model consists of a language module that encodes the query phrase into a continuous vector space, a visual module that extracts multiple image feature maps, an anchor generator for proposing multiple scale bounding boxes, a multi-modal fusion scheme for injecting all features into a single representation and a linear layer that predicts the most likely box proposal, as well as the array of regression parameters for its fixed coordinates.

a) ***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 [21] 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.

*b) **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.

*c) **Anchor Generation**:* The first step is to form a grid. For each cell of the grid, anchors of different shapes are proposed. Convention for the grid is (-1,-1) to (1,1), similar to what it is for the bounding boxes, first comes y and then x. -1 is the top-left and 1 is the bottom-right. 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 ratio (r) the aspect is $[s\sqrt{r}, \frac{s}{\sqrt{r}}]$. Then these aspects are sized according to the size of each cell of the grid. Center of each anchor box is the center of the grid cell.

*d) **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 $\mathbf{H_T} \in \mathbb{R}^{d_v \times d_w}$, $\mathbf{H_T} = \begin{pmatrix} \vec{h_t} & \vec{h_t} & ... & \vec{h_t} \end{pmatrix}^T$. 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 by:

$$\mathbf{M_j}(x, y) = \mathbf{V_j}(x, y) \; ; \; \mathbf{H_T}(x, y) \; ; \; \frac{c_x}{W} \; ; \; \frac{c_y}{H}, \quad j = 1, ..., K$$

with ; denoting the concatenation operation. c_x, c_y the centre locations of the normalised feature maps at each location (x, y) and W, H the initial size of the input image. 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.

*e) **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 fully-convolutional layer (FCN) 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. Datasets

Two bi-modal datasets are utilised for the training and evaluation of the ZSG network. Samples from these datasets contain image-caption pairs, with each image potentially appearing multiple times inside the dataset and each caption being a unique reference to some entity depicted in the image. In contrary to the proposed system of Section I, each caption contains only one entity query.

*a) **Flickr30k Entities**:* The *Flickr30k Entities* dataset [17] contains $30k$ annotated images associated with 5 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 data samples concerning the specific image. The annotations are bounding boxes containing also the category labels of the grounded entity.

*b) **ReferIt (RefClef)**:* The *RefClef* strain of the *ReferIt* dataset [8], itself being a subset of the *Imageclef* dataset [3], contains $20k$ images with a total of $85k$ single-query caption sentences.

For both datasets, we use the same training-validation-testing splits as in [5].

C. Implementation

For the implementation of the *ZSG* network, we have utilised the pre-trained *GloVe* word embeddings for English [15] for encoding our vocabulary into vector representations in our language module and the *RetinaNet* [11] network with *ResNet-50* [4] for our visual feature map extractor module. The image samples are resized to $300x300$ and the *GloVe* vector size as well as the bi-LSTM's hidden size is set to 300, so that both feature vectors can be appended to fit the dimensionality of the multi-modal feature maps ($d_m = d_w = d_v = 300$). The input query phrases are padded to a maximum length of 50 words per sentence. A total of 9 candidate anchor proposals of different sizes is generated, as suggested in [11].

*a) **Training**:* For formulating the error signal in the anchor matching process of the training, we utilise the *Intersection over Union* (IoU) metric as in [11], calculated as the total overlapping area between the proposed box and the ground truth box. Following the original implementation, we use an *IoU* threshold of 0.5, meaning that only proposals that fit the ground truth box's area over 50% are considered candidate. This can be formulated as:

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

$$G \doteq \{b_i \mid g_{b_i} = 1\}$$

with $B = \{b_i, \; i = 1, ..., 9\}$ denoting the set of all proposals and gt the ground truth box. The set G is the collection of all candidate proposals ($g_{b_1} = 1$).

For the binary classification output of the model (foreground that possibly contains the query vs. background with low *IoU* scores), we use the *focal loss* L_F, as described in [11], with default parameters $\alpha = 0.25, \; \gamma = 2$. The loss for the prediction score of the i-th proposal p_{b_i}, denoted L_p, can be then formulated as:

$$L_p = \frac{1}{|G|} \sum_{i=1}^{|B|} L_F(p_{b_i}, g_{b_i})$$

For regressing the parameters r_{b_i} of a tighter matching bounding box we employ the encoding scheme suggested by [19] with *smooth-L1 loss* L_S, formulated as:

$$L_r = \frac{1}{|G|} \sum_{i=1}^{|B|} g_{b_i} \cdot L_S(r_{b_i}, gt_{b_i}).$$

with gt_{b_i} denoting the top-left and bottom-down coordinates of the annotated ground truth box. The final loss that we use for backpropagation is $L = L_p + \lambda \cdot L_r$, with hyper-parameter $\lambda = 1$ for our implementation.

The end-to-end model was trained for a total of 12 hours for each dataset (around $7 - 8$ epochs depending on the training dataset) where the overall loss had already saturated and the model had reached it's best accuracy results. We utilised the Adam optimizer [9] with a learning rate of 10^{-4} and a weight decay of 10^{-4} for regularisation. The model was trained on the *Peregrine HPC cluster* on a *NVidia Tesla v100* GPU for parallel processing of batches of 128 image-query pairs.

b) ***Inference****:* During inference time our model produces a forward pass of a single sample sized batch. The proposed box with the highest prediction score is chosen and the regression parameters are used to update the coordinates of the box with respect to the image frame. Given the zero-shot nature of the proposed network, the input data spaces (pixels and text) are directly encoded into feature representations without the need of pre-processing steps, making this model suitable for providing predictions for any image-query input pair, even for unseen entity categories during training time.

IV. System Architecture

In this section we describe the design of the implemented software architecture, shown in Figure 3. All code is written in *Python* using the *PyTorch* deep learning framework and integrated in the *ROS* robotic framework [18] as a *ROS* package. Viewed from within the technical details of the *ROS* framework, our system receives a stream of RGBD topics produced real-time by the driver of a depth sensor as well as a natural language caption topic given by the human supervisor in the terminal's console and produces an RGB image topic with the predicted bounding box drawn into the frame, as well as renders a *PointCloud* topic containing 3D information of the segmented object.

A. Online Signal Buffering

Real-time sensory data are acquired through the *OpenNI ROS* driver for depth sensors [14]. After appropriately calibrating the sensor, the driver is capable of applying some pre-processing steps for rectifying image data as well as registering the depth frame into the RGB frame so that pixels between both frames have a one-to-one alignment. Our image buffering node buffers at 30 fps the rectified versions of the RGB and the registered depth frame into the system. A modular version of the buffer is also implemented, for buffering images from external sources, e.g. local paths.

For the natural language caption input, a simple buffering node was implemented. This node simply waits for a caption input from the terminals console and makes sure that is being published as a time-stamped message for alignment with corresponding frames by the model inference pipeline.

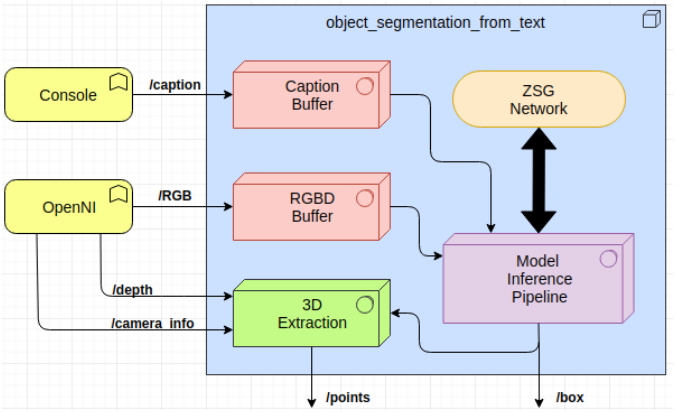


Fig. 3: A schematic of the implemented *ROS* system

B. Model Inference Pipeline

This nodes functions as the communication side between the real-time data and the implemented ZSG network. At its spawn this node loads all the necessary utilities for word embeddings, tensor processing and inference, as well as loads our best trained instance of the ZSG network.

When image-caption data are buffered, appropriate pre-processing is performed on the input RGB frame in order enhance it's contrast, by applying *contrast limited adaptive histogram equalization* (CLAHE) [25] in the YUV colourspace. The resulting image is converted into an 3D tensor-like image $\mathbf{I} \in \mathbb{R}^{3\times300\times300}$. Likewise, the input caption containing l words is parsed, embedded and converted to a 2D tensor-like sequence $\mathbf{W} \in \mathbb{R}^{l\times300}$. A single batch of these tensors is passed as input to the network and a set of prediction score and regression parameters is inferred after a single forward pass.

After inference, some post-processing steps for drawing the predicted box and segmenting it's content foreground for 3D extraction are applied and the resulting frames are published. The whole data preparation - inference - post-processing cycle is executed in 5 Hz, verifying the proposed capability of the selected zero-shot model for usage in real-time robotic applications.

C. 3D Extraction

For the extraction of the *PointCloud*, the *depth_image_proc ROS* package [22] is utilised. Given the */camera_info* topic containing the sensor's intrinsic parameters, as well as a synchronised pair of the segmented RGB object image with it's corresponding registered depth image, this package reconstructs each pixel in the image pair to a 3D space with colour information aligned. Examples of such segmentation can be viewed in Figure 7.

Method	Flickr30k	RefClef
QRC* [2]	60.21%	44.10%
CITE* [16]	61.89%	34.13%
QRG* [2]	60.1%	-
ZSG [20]	63.39%	58.64%
ZSG (us)	62.73%	50.41%

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

V. Experimental Design & Results

In this section we present our results for both the accuracy of the implemented deep learning model as well as for the overall performance of our system in online human-robot interaction scenarios.

A. Visual Grounding Learning

As we have trained our own end-to-end version of the network in order to integrate it with the model inference pipeline module of section IV for online usage, we evaluate our model's performance on the testing split of each of the two utilised datasets. As proposed in [2], the metric utilised for evaluating the learning of the zero-shot model is *IoU*, measuring the overlapping area percentage between the top-1 predicted bounding box and the annotated ground truth box of each image – query pair.

Following [20], whenever the *IoU* is above 50% the generated box proposal is regarded as accurate. In this manner, the total accuracy of the end-to-end model in our dataset test splits is measured as the total average of correctly predicted boxes for each bi-modal data pair. Our results are summarised in Table I along with other state-of-the-art techniques from literature.

We note that our version of the model succeeds in achieving similar state-of-the-art results as those reported in [20] in the *Flickr30k Entities* dataset. Limitations in training time (12 hours) in combination with the utilised batch size for greater parallelization result in lower accuracy in the very text-rich *RefClef* dataset.

B. Online Object Segmentation Experiments

In order to evaluate our system's performance, we conduct multiple experiments, aiming to test not only the predictive accuracy but also the generalisation potential of our model in variants of the standard input query phrase, as well as the overall online performance of the implemented system.

As real-time sensory data streamed from depth sensors attached in agents operating in natural environments often differ from the "idealised", digitally pre-processed images of the utilised training data, two manually collected tiny-sized datasets are used for evaluation. These datasets do not contain ground truth annotations and therefore the reported results are recorded qualitatively by the authors during online interactive sessions with the implemented *ROS* system. During such sessions, the supervisors provide captions for either static images or recorded RGBD streams in real-time and evaluate the system's predicted segmentations as either accurate or inaccurate. All depicted objects are given as different queries to the system and each prediction is evaluated. A selection of interactions is recorded and edited in a video clip that is uploaded and available for inspection.

Dataset	No.Im.	avg.QPI	No.Sam.	Accc.
Random	40	4.6	169	59.53%
SUN-RGBD	14	2.9	44	50.15%

TABLE II: Total number of image samples. average number of queries per image, total number of samples (query-phrases) and the average accuracy of our system in two tiny-sized manually collected datasets.

The first dataset contains 14 RGBD image-pairs taken from the SUN-RGBD dataset [23], containing almost $10.5k$ RGB-depth image pairs. After extracting the best candidate box, a *PointCloud* is also extracted utilising the registered depth frame. The goal of this experiment is to witness the effectiveness of the system in raw signal data. The second one is a selection of random samples collected from the web, that due to their content present particular interest for the evaluation of our algorithm's generalisation capacity. Figure 5 demonstrates some examples of such experiments. .

As expected, we observe that indeed when the data are real-time signals (RGBD dataset) streamed from a depth sensor they suffer from potentially high amount of noise (reflections, bad illumination etc.) resulting in the visual representations constructed by our image encoder to also suffer from noise. Moreover, we observe that due to the multi-modal nature our model is indeed capable of unravelling ambiguities of queries in the phrase (multiple object appearances) with the analogous variance in the supervisor's natural language input. Specifically, there are examples where the model responds to use of plural, use of definitive pronouns as well as visual cues (colour), as it is manifested in Figure 6.

VI. Conclusion

In this section we discuss our experimental results and provide some empirical insight about the behaviour of the implemented system, as well as some suggestions for future directions.

A. Remarks

The novelty of this method is due to the fusion of state-of-the-art object detection model with language model. This multi-modal learning resulted in capturing semantic relations among different nouns and the intra-class variations due to different attributes of objects. Usually, the features extracted by object detection models fail to capture intra-class variations. For e.g, as shown in Figure 4, the model is able to correctly identify the moped given the query *"silver moped"*. In this case, there was no moped in training examples but there was a car. This shows the model was able to capture the semantic relation that the new word moped is related to the car (as

Fig. 4: Zero-shot grounding as a result of multi-modal learning. A moped is detected correctly by the model. In this case, training samples contained no example of moped.

both are automobiles). Plus it also captured that the asked automobile object is different from the car automobile object present in the image.

The main focus of this paper is not into progress of visual grounding modelling but rather the implementation of a system that bridges the agents perceptive and communicative modules for online interactive use in robotic frameworks; the agent focuses on some part of the scene indicated by a human supervisor. The segmented 3D information of the object can serve as a perceptive utility that can be later processed by any navigation, planning and manipulation modules of the agent, for instance as the goal object that a robotic hand must grasp out of all objects upon a table.

Unlike the proposed system in the introduction, our system functions for only one query per caption. However, it seems to be able to capture some interesting variations of the standard one-to-one object-query correspondence in the input image-caption pair, such as plural, pronouns and visual cues (colour, shape) as it is demonstrated in Figures 5 and 6.

B. Future Work

One interesting future direction in order to make our system more HRI oriented would be the implementation of a speech recognition module for real-time speech to text translation as the input of the indicated captions. This eliminates the need for a physical input terminal in the robot's base, granting it more HRI friendly, as the human supervisor can give verbal indications to the body of the robot equipped with audio sensory hardware.

Finally, research of alternative ways to train a single-stage model or how to adapt it to funciton for multiple query entities per sentence, would also be an interesting direction for further augmenting the multi-modal comprehension of our model.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[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. A. Javed, S. Saxena, and V. Gandhi. Learning unsupervised visual grounding through semantic self-supervision. *arXiv preprint arXiv:1803.06506*, 2018.

[7] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017.

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

[9] 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.

[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. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Oct 2017.

[12] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun. Inverse visual question answering: A new benchmark and vqa diagnosis tool. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[13] 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.

[14] O. organization. *OpenNI User Guide*. OpenNI organization, November 2010.

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

[16] 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.

[17] 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.

[18] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[19] 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.

[20] 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.

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

[22] J. Shade, S. Gortler, L.-w. He, and R. Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, 1998. ACM.

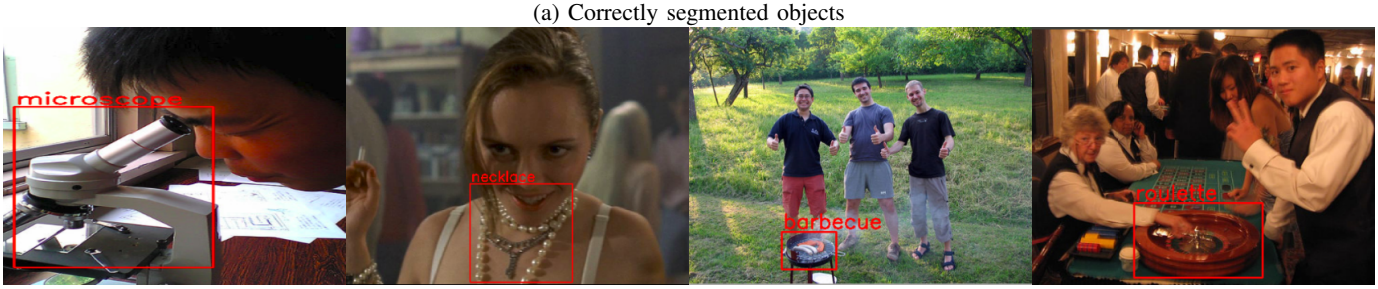
[23] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, June 2015.

[24] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

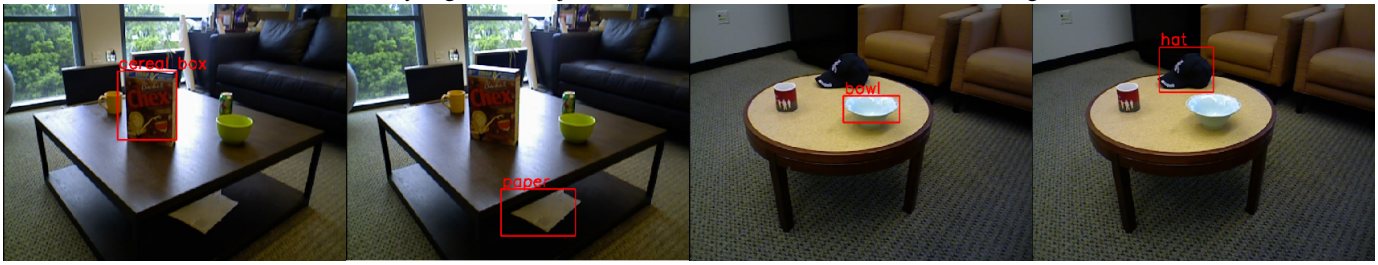
[25] K. Zuiderveld. Graphics gems iv. chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

VII. Appendix

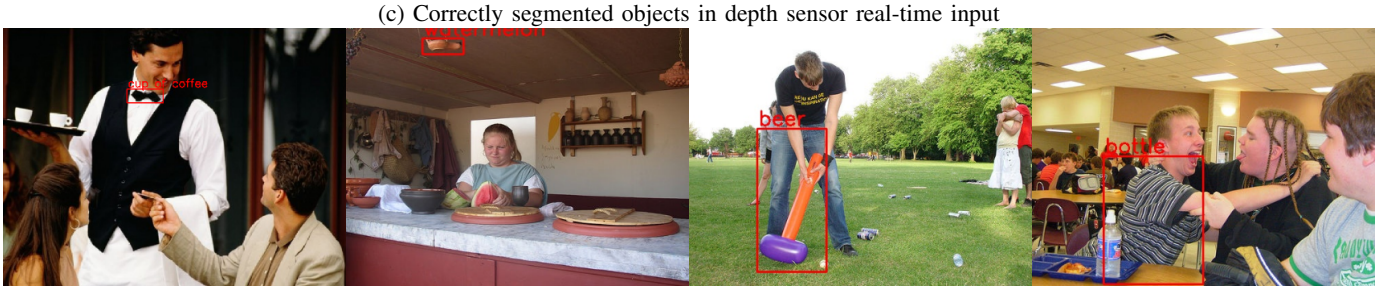
(a) Correctly segmented objects



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



(c) Correctly segmented objects in depth sensor real-time input

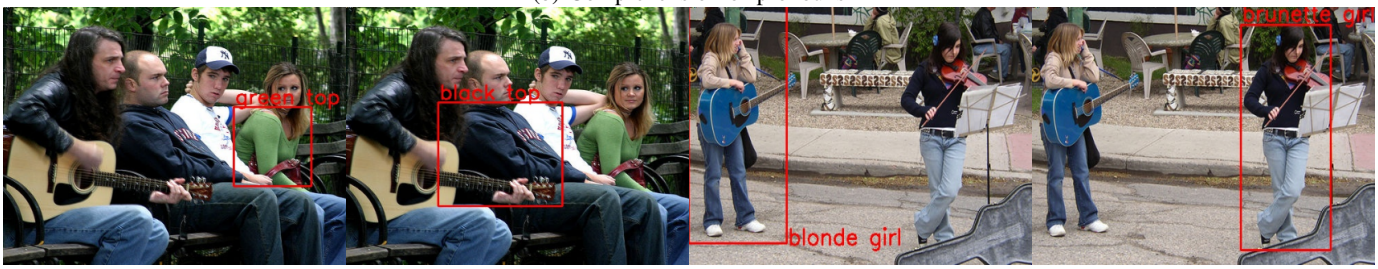


(d) Incorrectly segmented objects

Fig. 5: 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) cereal box, paper, bowl, hat, (d) cup of coffee, watermelon, beer, bottle

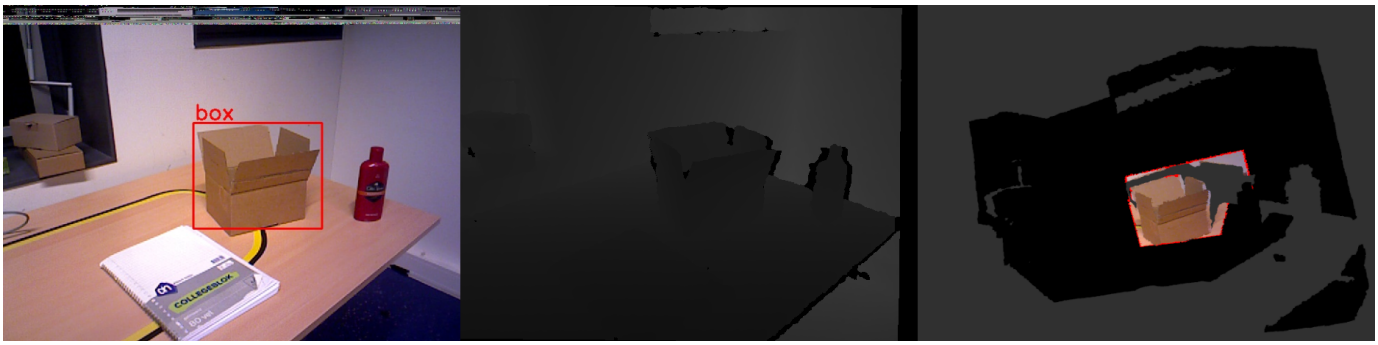
(a) Comprehension of plural

(b) Comprehension of pronouns

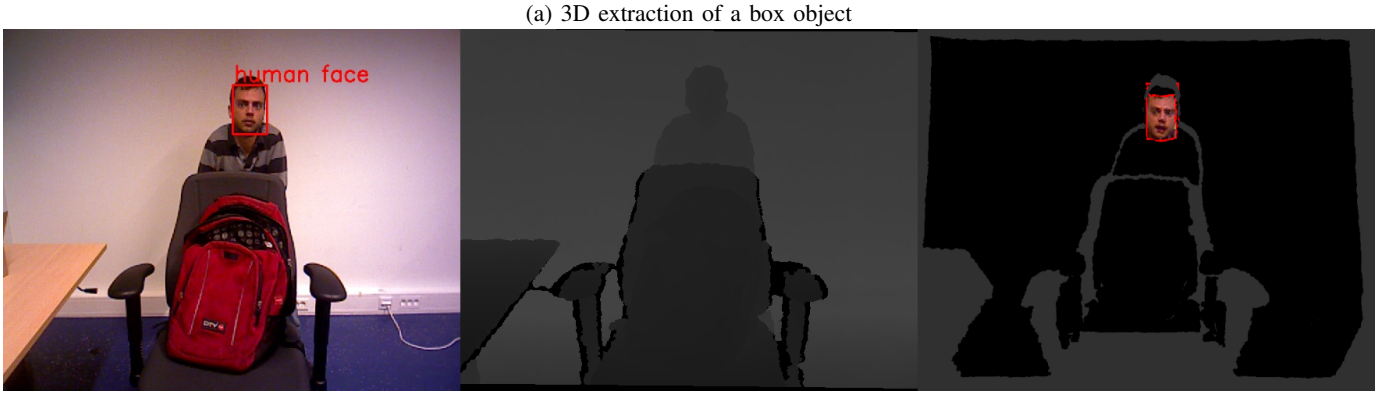


(c) Comprehension of visual cues (colour)

Fig. 6: 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



(a) 3D extraction of a box object



(b) 3D extraction of a human face

Fig. 7: The *PointCloud* extraction of our system. The first two pictures correspond to the RGB-depth data pair from the sensor and the third one is a screenshot of the extracted *PointCloud*. We can see that the system successfully extracts the correct entity, however these images suffer from high amount of noise from the depth sensor (deep black regions are masked by the system in order to keep only the 3D information of the segmented object only).